Hierarchical Resource Management in Adaptive Airborne Surveillance Radars

Johannes Wintenby

Ericsson Microwave Systems
431 84 Mölndal
Sweden
wintenby@s2.chalmers.se

Vikram Krishnamurthy
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, V6T 1Z4
Canada
vikramk@ece.ubc.ca

Abstract – *In this paper we present a hierarchical resource management algorithm for adaptive airborne surveillance radars. The dynamics of the radar are formulated as a stochastic discrete event system. By abstracting the physical layer sensor performance into a quality of service measure, the resource management problem is formulated as a constrained Markov decision process. A two-level (two-timescale) resource management algorithm is developed based on Lagrangian relaxation. A numerical example is presented on a scenario involving different target densities.*

Keywords: resource management, resource allocation, surveillance radar, stochastic control, sensor control

1 Introduction

The beam agility of an Electronically Scanned Antenna (ESA) radar permits adaptive allocation of transmitted energy in time and space. There is strong motivation in designing scheduling policies that utilizes the agility to achieve an adaptation of an ESA radar to a given set of tasks, such as tracking a set of targets, or searching a sector for new targets. The objective is to enhance the overall system performance, which depends on the quality of the radar output, i.e, on how closely the radar track database matches the actual targets. Due to the stochastic nature of radar detection and target dynamics, scheduling of radar measurements is a stochastic control problem. Furthermore, the resource allocation problem of efficiently conducting several parallel tracking and searching tasks using the radar's antenna is an important part of the scheduling problem, and needs to be considered. Generally, control of ESA radars is studied in the field of sensor management [1]. There are two broad methodologies for formulating radar management problems:

(i) *Heuristic Scheduling based on Rule-based Design*: In this methodology, a scheduling rule is defined based on descriptive, rule based design. Work presented in the literature on heuristic schedulers for radars, which can be regarded to include the overall resource allocation problem, are found in [1], [2]. Some effort has been spent on designing an allocation algorithm for a single target, e.g. [3], and such algorithms suitably forms the base in schedulers. Detailed scheduling of measurement order, given what measurements to make and in what time interval they should be made, is treated in e.g. [4]. Heuristic schedulers operate in real time with limited computational resources. However, relying completely on heuristic schedulers in radar design is often unsatisfactory due to the difficulties of understanding what performance gains that are achievable by updating the design. Furthermore, design of a rule base for system adaptation over both a range of scenarios, and a range of tasks is sophisticated, particularly for cased when the system is downloaded.

(ii) *Optimization based Scheduling*: In the optimization-based approach for radar resource management, a multi-stage cost function over a finite, or infinite, horizon is minimized. Unfortunately, the dimensionality of the resulting formulation typically makes computing the global optimum intractable. One can resort to the *myopic case*, i.e., to optimize an instantaneous cost [5], but this is typically inappropriate. A long term horizon (e.g. a minute) is desirable in resource allocation, for instance due to the following items:

- There is long term dynamics in the coupling between the search scan allocation, and the resulting search performance, mainly because of large surveillance volumes to cover with scarce resources. The coupling also involves the future number of tracked targets, and thus the future track load, giving long term dynamics in the resource demand.
- Platform motion, and course changes in combination with spatially inhomogeneous antenna gain of an ESA, lead to a dynamically changing resource demand of adaptively tracking a set of targets.
- Re-acquisitions of targets that reappear after a blindness period (i.e. Doppler blindness, elevation and vegetation mask), and the tracking of a set of interacting targets with potentially mixing tracks, benefit from planning of measurements over a horizon, and imply dynamics in the resource demand.
- Synchronization of search scans and adaptive track updates may reduce the resource demand. However, the synchronization requires a time horizon in decision-making, stretching at least over the next search scan pass of a target.

Stochastic optimization methods for sensor scheduling using stochastic Dynamic Programming (DP) are presented in [6] and [7]. To solve the full stochastic optimization problem with DP, simplifying assumptions are needed, for example using multi-armed bandit models, or assuming stationarity leading to the employment of Bellman's equation [8]. In the radar resource management problem considered herein, these assumptions are too restrictive, and there is strong motivation to develop alternative sub-optimal formulations.

This paper is a short version of material submitted for publication, also presented in [9, Chapter 5]. We present a hierarchical resource management algorithm for ESA radars that are based on the optimization approach. An important aspect of the method is that searching and tracking are unified in the same framework in a novel manner. Although the primary application is the airborne surveillance radar, the method can be generalized to other applications such as ground and ship based multi function radars, fighter aircraft radars, and multi sensor systems including one or several adaptive sensors. The main ideas of the paper are the following:

(i) Stochastic Discrete Event System formulation and abstraction of physical level aspects of sensor performance into Quality of Service (QoS) Measures. The QoS measures are defined target-wise based on concepts such as track accuracy and track continuity. A single sensor performance measure is defined as an aggregate of the target-wise measures, integrated over a time horizon, see Sec.2.1.

(ii) Approximate, discrete time Markov chain modelling of the dynamics of target wise tracking performance.

(iii) Two-timescale scheduling. What measurements to make, how, and approximately when are decided on batch-wise at a slow, regular timescale with interval Δ_t (typically Δ_t is in the order of a second). This is denoted resource allocation herein. At a fast timescale, the local order of measurements within a batch, and in the joint between two subsequent bathes are arranged. We assume the relevant dynamics of tracking performance is captured by the slow timescale, and that the local arrangement within the batches is of minor significance for the system performance. In the sequel, we assume the regular, slow-timescale in decision-making, while the fast timescale scheduling is disregarded.

(iv) Utilization of abstract measurement operations in decision-making on the slow timescale. A measurement operation is considered to be an algorithm in the radar that generate a sequence of measurements needed for achieving a low-level measurement task such as "update track with repeated update attempts", see for instance the track update algorithm designed in [3]. Fast feedback measurements that should be achieved on the fast timescale are allowed within the measurement operations. Thereby, the batch-wise, slow-timescale decision-making in resource allocation is facilitated. An overview of the two-timescale approach is shown in Fig.1.

(v) Formulation of resource constraints and separation of the problem into components. The resource constraints in the antenna are made explicit by a series of constraints on the used time per time interval (i.e. load). These constraints are suitably incorporated in an optimization algorithm using approximate Lagrange relaxation [10]. That is, by replacing future, stochastic Lagrange multipliers with estimates using average resource constraints, the resource allocation problem is separated into components. Such a separation is a condition for a successful optimization based algorithm based, which otherwise suffers from a combinatorial explosion. The price is that the uncertainty in future effects from present decisions will not be fully considered in the predictions. Thereby, we optimally solve a problem which is an approximation of the original, practically unsolvable problem.

(vi) Hierarchical extension and LRM. To incorporate coordination between track updates and the search scans in a sector, a hierarchical extension to the separated problem is proposed. Decisions on track updates are then conditioned hierarchically on the search scan parameters of the sector. The space is divided into a set of sectors, and in each sector, the same search scan sequence is used. Searching and tracking in a sector corresponds to a component of the optimization problem. We refer to the resulting resource allocation method based on the hierarchical extension as the Lagrange Relaxation Method (LRM). The hierarchical method for resource allocation based on the LRM can be viewed as an offline method for benchmarking performance of other resource allocation methods. In Sec.4, a numerical example is provided which demonstrates the utilization of the method as a tool in radar design.

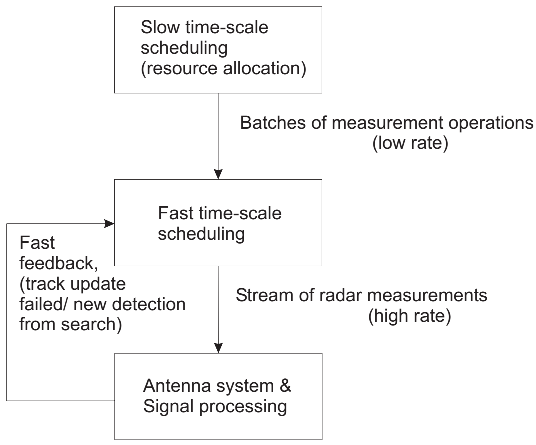

Fig. 1: Two-timescale scheduling.

2 System Model and Resource Allocation Formulation

In this section, we formulate the resource management problem as a finite horizon stochastic optimal control problem. When describing sensor performance, the physical layer of the system is abstracted into QoS measures, which are expressed target-wise in terms of tracking utility, and then aggregated to an overall utility. A Markov chain model is designed for the prediction of the dynamics of the QoS measures.

2.1 Tracking utility

We identify that the following abilities are involved when considering the radar output quality and thus the tracking performance,

- to maintain tracks of the targets,

- to keep the same identity of the tracks throughout the surveillance volume, i.e., to avoid track drops leading to re-initiations with new track identities, and to avoid mixing two or more tracks,
- to sustain the accuracy of the tracks.

Based on these items tracking QoS measures should be defined and aggregated to a single, non-instantaneous sensor performance measure. The approach taken here is to define target-wise, instantaneous tracking utility measures, which are summed over both time and targets to get the aggregated measure. Assume a scenario with targets $\mathcal{T}_i, i \in \{1, \ldots, M\}$, where M is the number of targets. In the sequel, i enumerates the targets. The instantaneous measures are assumed to be functions of a target-wise tracking performance state, $x_i(t)$. The state includes parameters needed to express all relevant aspects of tracking performance, such as covariance, if a target is tracked or not etc. Let $U_i(x_i(t))$ denote an instantaneous tracking utility measure. The overall instantaneous utility of the radar system at time t is defined as, $U(x(t)) = \sum_{i=1}^{M} U_i(x_i(t))$, where $x(t)$ is the aggregated state of all target-wise performance models, i.e., $x(t) = \{x_i(t)\}_{i=1}^{M}$. The utility function is specified for each target individually. In cases where tracking performance calculations involve several targets, e.g., at crossing target situations, we have to condition performance calculations for one target on the tracking performance states of neighboring targets. Define:

- $x_{i,\text{tracked}}(t) \in \{0, 1\}$, a state variable part of $x_i(t)$ indicating if a target is tracked or not.
- $U_{\text{nom},i}$, a nominal utility measure for tracking a target, also corresponding to a user priority.
- $Q_{\text{acc},i}(x_i(t))$, a scalar-valued QoS measure for accuracy between zero and one, where one means good accuracy. Herein, we let $x_i(t)$ include the variances of the target kinematic states in tracking (e.g. position and velocity), and $Q_{\text{acc},i}$ is a nonlinear function of these variances.
- t_n, the time of a measurement update of a track.
- $I_{\text{reinit}}(x_i(t_n)) \in \{0, 1\}$, an indicator of that a track has been re-initiated after a period where the track has been lost at time. The state $x_i(t_n)$ must include the states necessary to make the indication.
- $C_{\text{reinit},i}$, a cost for a re-initiation.
- $I_{\text{mix}}(x_i(t_n), \{x_j(t_n) | j \in \mathcal{C}_i\}) \in \{0, 1\}$, an indictor for track mixes at the track measurement update at time t_n. The set $\mathcal{C}_i$ includes all nearby targets with a mixing risk.
- $C_{\text{mix},i}$, a cost for a track mix.

We choose to express the target-wise, instantaneous utility as,

$$\begin{aligned} U_i(x_i(t)) &= x_{i,\text{tracked}}(t) U_{\text{nom},i} Q_{\text{acc},i}(x_i(t)) \\ &- \sum_n C_{\text{reinit},i} I_{\text{reinit}}(x_i(t_n)) \delta(t - t_n) \\ &- \sum_n C_{\text{mix},i} I_{\text{mix}}(x_i(t_n), \{x_j(t_n) | j \in \mathcal{C}_i\}) \delta(t - t_n) \,. \end{aligned} \tag{1}$$

If desired, $U_{\text{nom},i}$, $C_{\text{reinit},i}$, and $C_{\text{mix},i}$ can be made state dependent and thus geographically dependent.

In resource allocation herein, we search for the next batch of measurement operations which maximizes the expected utility of the radar system integrated over a time window. That is, given a global tracking performance state $x(t_0)$, where the present time is denoted as t_0, the aim is to maximize the non-instantaneous utility expressed as,

$$J(x(t_0)) = E\left\{ \int_{t_0}^{t_0+t_h} U(x(t)) dt \Big| x(t_0) \right\}. \tag{2}$$

Here, t_h denotes the prediction horizon. The expectation is over the future radar measurements, including measurement time instants and measurement errors, and ideally over the future target trajectories.

In the application, we assume there are no hard constraints on tracking quality or on how the system measures, e.g., on the track update rates. Instead, these constraints are included softly in $U_i(x_i(t))$.

The non-instantaneous utility is approximated when resorting to the discrete time case. Let $0, \Delta_t, 2\Delta_t, \ldots, N\Delta_t$ be a uniform partition of the interval 0 to t_h. Then we have that,

$$J(x(t_0)) \quad \approx E\left\{ \sum_{k=0}^{N-1} U(x(t_0 + k\Delta_t))\Delta_t \Big| x(t_0) \right\}, \tag{3}$$

where N is the horizon of decision-making in discrete time. In the sequel, Δ_t will be omitted in the multiplication since it acts as a scaling. The objective function in Eq.(3) can also be formulated as a discounted, non-instantaneous utility.

A comment is needed concerning undetected targets which may become detected and tracked during the prediction horizon. At a decision time instant, the position, velocity and radar cross section of these targets are not known, and the evaluation of Eq.(2) with respect to decision on the search program becomes infeasible. Therefore, a set of *test targets* are utilized to sample the space of trajectories of unknown targets. The density of test targets corresponds to an a priori modelled density of targets in the scenario, or alternatively, weights can be put on the test targets to achieve the desired density of the model. Given the utilization of test targets, the number of targets M in the model will remain constant during predictions, although the predicted number of tracked targets will change dynamically.

2.2 Dynamic tracking performance model

In this subsection, we discuss a dynamic state model for prediction of the *target-wise* tracking utility, introduced in the preceding subsection. The state of the model includes parameters needed in describing the instantaneous tracking utility measure. To facilitate the use of finite dimensional dynamic programming (DP) algorithms, the model is formulated as a Markov decision process in Sec.2.2.1.

Accuracy

We assume Kalman filters are used as tracking filters, and that the scalar valued accuracy quality function

$Q_{\text{acc},i}(x_i(t))$ introduced above is a function of the Kalman filter covariance matrix. That is, the filter covariance $P_{i,t|s}$ is part of the state $x_i(t)$ where $P_{i,t|s} = E\{\tilde{\xi}_{i,t|s}\tilde{\xi}_{i,t|s}^T\}$, and $\tilde{\xi}_{i,t|s}$ is the tracking error at time t given the last measurement preceding t occurred at time s.

Let t_n be the time instant when observation n occurs, and let the time since the last update be $T_n = t_n - t_{n-1}$ (herein, the target index i is omitted for the time variables due to convenience in notation). The Kalman filter covariance evolves according to the Riccati equation,

$$\begin{aligned} &P_{i,t_n|t_{n-1}} = F(T_n)P_{i,t_{n-1}|t_{n-1}}F(T_n)^T + Q_i(T_n) \\ &P_{i,t_n|t_n} = (I - K_{i,n}H)P_{i,t_n|t_{n-1}}(I - K_{i,n}H)^T + \\ &K_{i,n}R_{i,n}K_{i,n}^T, \end{aligned} \tag{4}$$

where $Q_i(T_n)$ is the covariance of the covariance input in the target motion model used in a Kalman filter, $R_{i,n}$ is the covariance of the measurement noise, $K_{i,n}$ is the Kalman gain, and H is the linear mapping from the kinematic state of the filter to the observation space.

In order to predict $P_{i,t|s}$, it is enough to know: P_{i,t_0}, the initial filter covariance at time t_0, the mapping $Q_i(T)$, the measurement covariances, $R_{i,n}$, the prediction time t, and the time instants of the measurement updates, t_n, $n = 1, \dots m$, where t_m is the last measurement time instant before t. At predictions, only the time instants t_n, $n = 1, \dots, m$ are stochastic due to uncertainties in when observation instants will occur. The rest of the variables are either known or functions of the sequence t_n and t. Therefore, we include t and the sequence t_n, $n = 1, \dots m$ as dynamic variables explicitly in the state $x_i(t)$, while the other variables are treated as implicitly available. Thus, $P_{i,t|t_m}$ is parameterized by the state $x_i(t)$. Unfortunately, the sequence t_n grows with time. However, the memory in the recursion in Eq.(4) is short, and it is sufficient to include the last few intervals between the observation time instants in the state i.e., $\{t - t_m, T_m, \dots\}$, to get a decent prediction of $P_{i,t|t_m}$. As an initiation of the recursion, a covariance matrix based on the average update rate in the recent past is used.

To be able to store a distribution over the state $x_i(t)$, we chose to quantize the state, and for efficient calculations, only $\{t - t_m, T_m\}$ is included. The quantization is made such that $t - t_m, T_m \in \{k\Delta_t\}$ in order to suitably match the slow-timescale in resource allocation.

Target is tracked

The variable $x_{i,\text{tracked}}(t)$ is modelled with a Markov chain. The transition probabilities depend on the measurements made on the target, in particular on the detection probabilities P_d. Both a policy for track drops, and a policy for track confirmation measurements are suitably modelled in the Markov chain. In the modelling of P_d, the expected SNR is calculated by the radar equation [1, Chapter 2] with target position, velocity, and expected radar cross section (RCS) inserted (see [9] for modelling details). The target position and velocity are extracted from the future trajectory, which is modelled either deterministically or stochastically.

Re-initiation events

The modelling of re-initiation events is made by augmenting the Markov chain above with a state for remembering that a target has been dropped, denoted as $x_{i,\text{dropped}}(t)$, and a state for indication of the re-initiation event, connected to the indicator $I_{\text{reinit},i}(t_n)$. From the Markov chain we can then extract $P(I_{\text{reinit},i}(t_n) = 1)$.

Track mixes

Target-to-track mixes occur as a consequence of plot-to-track data association errors in dense parts of the scenario, although only some data association failures lead to track mixes. The events are assumed to occur at track update instants. As a rough estimate of the event, the probability of a plot-to-track association error may be used. Approximate expressions are presented in [11] for scenarios with a homogeneous density of targets. In [9, Chapter 4], prediction of association errors are studied for scenarios given two crossing target trajectories.

The calculation of plot-to-track association error events is based on the predicted accuracies at the measurement update instants. Thus, the information needed in the state $x_i(t_n)$ to predict the filter accuracy at time t_n, i.e., the sequence $\{t_n, T_n, T_{n-1}, T_{n-2}, \dots\}$, is likewise needed in the prediction of association error events.

2.2.1 *Discrete Event State model formulation*

We now formulate a dynamic model where the instantaneous tracking performance is a function of the state of this dynamic model. Based on the discussion above, the state should describe the observation process, and for accuracy prediction and track mix predictions, only the recent past of the observation process. The following discrete variables evolve dynamically and stochastically with prediction time, and are explicitly included in the state:

- The quantization of $\{t - t_n, T_n\}$.
- A discrete variable representing if a target is tracked or not, $x_{i,\text{tracked}}(t)$.
- A discrete variable remembering that target has been tracked, but is now dropped, $x_{i,\text{dropped}}(t)$, plus a variable indicating track re-initiation events.

The track information available at the start of a prediction interval is static and is only implicitly regarded as being a part of the state, i.e., P_{i,t_0}, $R_{i,n}$, etc. Detection probabilities at time t depend on the target kinematic state (i.e. distance, azimuth and radial velocity), here denoted as $\xi_i(t)$, and therefore, $\xi_i(t)$ is also regarded as a part of the performance model state $x_i(t)$. However, in the calculation of detection probabilities, $\xi_i(t)$ is assumed to develop deterministically, and in the formalism we will treat $\xi_i(t)$ as separated from $x_i(t)$ for convenience. Thereby, all dynamic, stochastic components in $x_i(t)$ are discrete. A suitable model of the dynamics of $x_i(t)$ is then a stochastic discrete event system. With the state variables listed above, the transitions in the system occur due to two types of events:

- Detection opportunity events. When a target has an opportunity of being detected i.e., when the target is scanned, a detection opportunity event occurs. The state transitions resulting from the event depends on the detection probability, P_d.

- Quantization adjustment events. When time passes, $t - t_n$ increases and the quantization must be adjusted by switching state. The event occurs on the slow time scale.

The dynamics of the system can be modelled as a generalized semi-Markov decision process (GSMP) [12]. However, for computational tractability, we consider a discrete time Markov chain model for the discrete event system. The event times are then exclusively multiples of Δ_t. Thus, detection times are approximated to occur at the discrete time instants on the slow timescale.

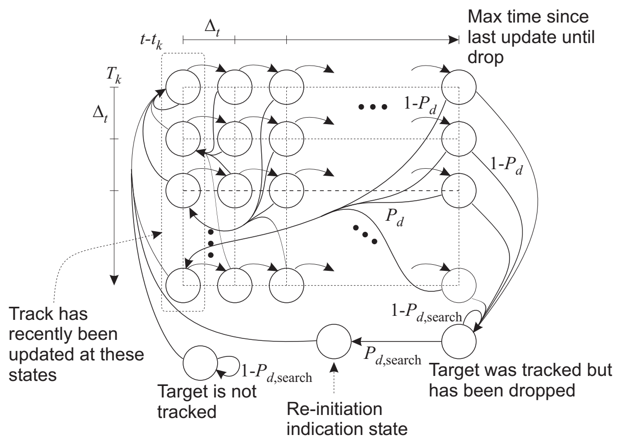

Fig. 2: An example of a Markov chain for keeping track of if a target is tracked or not, track re-initiation, and $t - t_n, T_n \in \{k\Delta_t\}$ given that the target is tracked. The transition probabilities are conditioned on when and how measurements are made. On target detection, the transition is to one of the leftmost states depending on the time since the previous update. On a detection failure, a step to the right will be taken.

An example of a Markov chain based on the discussion above is shown in Fig.2. Let $p_{x_{i,k}}$ be the state probability vector of $x_{i,k} = x_i(k\Delta_t)$. Assume $x_{i,k}$ is a Markov chain which is affected by control actions d_k i.e., $x_{i,k}$ is a Markov decision process. The target-wise model has the following form,

$$p_{x_{i,k+1}} = P_{tr,i}(d_k, \xi_{i,k}) p_{x_{i,k}}, \tag{5}$$

where $P_{tr,i}$ is the transition matrix of the Markov decision process, and $\xi_{i,k} = \xi_i(k\Delta_t)$ is the kinematic state of target i. The target motion state $\xi_{i,k}$ has been conditioned on explicitly since it affects the detection probabilities and evolves dynamically, and consequently, the process is non-stationary. Typically, $P_{tr,i}$ has a sparse structure which should be used in the resource allocation algorithms.

The state is assumed to be fully observed i.e., when time instant k occurs, the state of the performance model $x_{i,k}$ is known.

2.3 Parameterization of measurements

We deal with two kind of measurements: search scans, and adaptive track updates. A search scan is herein parameterized with a vector of parameters representing allocated time per time interval on the slow-timescale. One parameter is required for each interval in which the scan may be allocated. For search scan j, the allocation vector is denoted as $u_j = [u_{j,k_{0,j}}, u_{j,k_{0,j}+1}, \ldots, u_{j,k_{1,j}}]$, where $k_{0,j}$, and $k_{1,j}$ are the first and last intervals in which allocation is allowed. A sequence of search scans associated with a sector, $\mathcal{S}$, is then parameterized with an aggregated vector of allocation vectors,$u_{\mathcal{S}} = [u_1, u_2 \ldots, u_j, \ldots]$. Other parameters relevant in the performance modelling, such as integration gain, probability of detection, and the time instant of a scan passing over a certain azimuth location, are calculated out of $u_{\mathcal{S}}$.

The decisions regarding a sequence of adaptive target updates are parameterized with a sequence of discrete parameters, $d_{\text{upd},i} = \{d_{\text{upd},i,0}, d_{\text{upd},i,1}, \ldots, d_{\text{upd},i,N-1}\}$, where $d_{\text{upd},i,k} \in \{$'update track i at time interval k', 'do not update track i at time interval k' $\}$. An update command triggers a sequence of update attempts, which results either in a successful or a failed detection. The sequence is optimized locally with the objective to minimize the expected time to achieve a target detection with a high probability, see e.g. [3]. Discrete decision variables for target updates facilitates DP when controlling track updates. However, the resulting measurement time will be random so to that the processing time of a planned measurement batch will be random. This randomness should be considered in an online approach, for example by adjusting the available time in the following time interval, [9]. Herein this randomness is disregarded for simplicity.

3 Optimization of resource allocation with approximate stochastic Lagrange relaxation

The resource allocation problem is now to decide on what measurements to do in the next time interval on the slow-timescale, with the aim of maximizing the overall, expected tracking performance predicted over a time horizon. Let $k = 0$ represent a decision time instant, where $k > 0$ is the future. Formally, the problem is here expressed as,

$$\max_{d_0} U(x_0) + E_{x_1|x_0,d_0}\{J_1^*(x_1)\}, \tag{6}$$

where $J_1^*(x_1)$ represents the future utility as a consequence of decisions d_0 made at $k = 0$, and given a sequence of optimal future decision, i.e.,

$$J_k^*(x_k) = \max_{d_k} U(x_k) + E_{x_{k+1}|x_0,d_0}\{J_{k+1}^*(x_{k+1})\}. \tag{7}$$

Thus, the modelling of the decision consequences has the form of a recursion with nested maximizations and expectations. An optimization algorithm for this stochastic control problem typically relies on stochastic Dynamic Programming. Unfortunately, the size of the state space explodes combinatorially with the number of targets in the scenario, and an optimal approach is infeasible. Therefore, approximate solutions are needed.

Problem separation

An approach to large scale control problems is to try to achieve a separation into components, where each component can be optimized locally, and then coordinated globally via the constraint on the control signals, i.e., on the allocated resources in this case. Define the load l_k as the total utilized time per time unit in interval k. The resource constraints are formulated as: $l_k(x_k, d_k) < 1, k \in \{0, N-1\}$. The control problem studied herein has a separable structure in the targets which is suitably explored. A key observation is that measurements have local effects in space on tracking performance on targets. For example, track updates of a target affects tracking performance of the target, and perhaps of nearby targets, but not on targets distant in space. Likewise, search scans in one sector do not affect tracking performance of targets in other sectors. Formally, this reasoning is treated by grouping targets into *subtasks* such that decision parameters has local effects in the performance modelling of the subtasks. The utility of a subtask $\mathtt{s}$ is then written

$$U_{\mathtt{s}}(x_k) = \sum_{\{i|\mathcal{T}_i \text{ belongs to } \mathtt{s}\}} U_i(x_{i,k}). \tag{8}$$

Define $x_{\mathtt{s},k}$ as the aggregated state for the targets sorted to $\mathtt{s}$. The system utility at time k is then $\sum_{\mathtt{s}} U_{\mathtt{s}}(x_{\mathtt{s},k})$. For subtask $\mathtt{s}$, and time interval k, the decision parameters are denoted as $d_{\mathtt{s},k}$. The decision parameters of the total batch of scans in interval k is then an aggregate of the decision parameters for all subtasks. State transitions for targets part of $\mathtt{s}$ are assumed to be locally dependent on $d_{\mathtt{s},k}$,

$$p_{x_{i,k+1}} = P_{tr,i}(d_{\mathtt{s},k}, \xi_{i,k}) p_{x_{i,k}}, \ \{i|i \text{ belongs to } \mathtt{s}\}. \tag{9}$$

One separation with this property is a division of space into sectors defined by, e.g., target density, task definitions, and sensor characteristics. Another possibility is to disregard the search scan support in tracking, i.e., the search is only used to cue adaptive tracking. Then, each tracked, independently acting target is regarded as an independent subtask. Neighboring, interacting targets are merged to form new independent subtasks, and searching a sector for undetected targets is also an independent subtask. This gives a flat separation into subtasks as illustrated in Fig.3.

The load in an interval coming from the measurements associated with $\mathtt{s}$ is here denoted as $l_{\mathtt{s},k}$. Based on the load of all subtasks, the resource constraints are reformulated as,

$$\sum_{\mathtt{s}} l_{\mathtt{s},k} \leq 1, \ \ k \in \{0, N-1\}. \tag{10}$$

Approximate Lagrange relaxation

According to above, decisions on track updates, and search scans have local effects on tracking performance for each subtask disregarding one fact: the measurements compete of the same constrained resources. By introducing Lagrange relaxation, the constraints on the resources are included explicitly in the optimization,

$$\begin{aligned} L_k(x_k, d_k, \lambda_k) &= U(x_k) + \lambda_k \left(1 - l_k(x_k, d_k)\right) + \\ &+ E_{x_{k+1}|x_k, d_k} \left\{ \max_{d_{k+1}} L_{k+1}(x_{k+1}, d_{k+1}, \lambda^*_{k+1}(x_{k+1})) \right\}. \end{aligned} \tag{11}$$

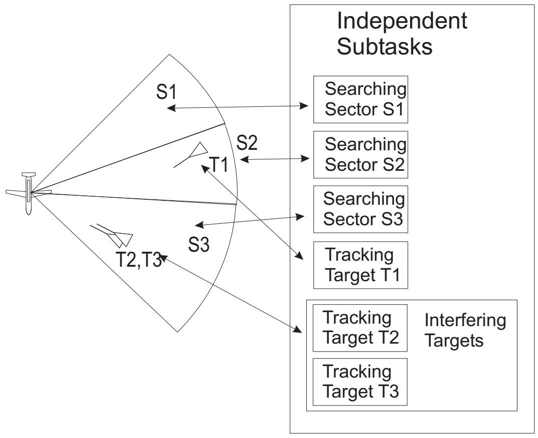

Fig. 3: Formation of independent subtasks. S1, S2 and S3 represent sectors that should be searched, while T1,T2 and T3 are tracked targets.

Here, λ_k is the Lagrange multiplier at time k, and $\lambda^*_k(x_k)$ is the Lagrange multiplier such that the resource constraint is fulfilled with equality at optimum d^*_k. The aim by introducing the Lagrange formulation is to separate the problem into components given an algorithm that globally searches for the Lagrange multipliers. However, the optimal Lagrange multipliers at future stages $\lambda^*_k(x_k), k > 0$, depends on the global state x_k, and therefore, such a separation is not feasible without further simplifications. An operation that achieves a separation is to replace the future, state dependent Lagrange multipliers with estimates $\widehat{\lambda}^*_k$ based on constraints on the *expected* resource utilization [10],

$$E_{x_k|x_0}\{l_k(x_k, d_k)\} = 1, \ k \in \{1, \dots, N-1\}. \tag{12}$$

Due to the expectation, the Lagrange multipliers λ_k are no longer dependent on the realization of the global state x_k. Using Eqs.(8-10), one can show that the total Lagrangian in Eq.(11), with the multiplier estimates inserted, is separable with respect to the subtasks. It is possible to write (the approximation comes from using the estimates),

$$L_k(x_k, d_k, \lambda_k) \approx \sum_{\mathtt{s}} L_{\mathtt{s},k}(x_{\mathtt{s},k}, d_{\mathtt{s},k}, \lambda_k) \ + \lambda_k + \sum_{n=k+1}^{N-1} \widehat{\lambda}^*_n,$$

where the Lagrangian local to subtask $\mathtt{s}$ is defined recursively as

$$\begin{aligned} L_{\mathtt{s},k}(x_{\mathtt{s},k}, d_{\mathtt{s},k}, \lambda_k) &= U_{\mathtt{s}}(x_{\mathtt{s},k}) - \lambda_k l_{\mathtt{s},k}(x_{\mathtt{s},k}, d_{\mathtt{s},k}) \\ &+ E_{x_{\mathtt{s},k+1}|x_{\mathtt{s},k}, d_{\mathtt{s},k}} \left\{ \max_{d_{\mathtt{s},k+1}} L_{\mathtt{s},k+1}(x_{\mathtt{s},k+1}, d_{\mathtt{s},k+1}, \widehat{\lambda}^*_{k+1}) \right\}. \end{aligned} \tag{13}$$

Due to the separation, dual programming can be employed to globally search for the sequence of Lagrange multiplier estimates $\{\widehat{\lambda}^*_k\}_{k=1}^{N-1}$ and λ_0, while the recursion in Eq.(13) is solved backwards to time $k = 0$ for each subtask individually given the Lagrange multipliers. If the subtask is to generate updates for a single target where the state space is fairly small, Eq.(13) forms a base for DP, including the resource constraint via the Lagrange multiplier cost terms. The Markov decision process in Eq.(5) based on a chain such as in Fig.2 can be used straight of [9].

In searching subtasks, the local state space is still too high dimensional for DP, mainly because of a large number of test targets. To deal with this complexity, we use open loop assumptions on the search scan parameters, meaning that the modelling excludes observations of future states. The future decisions are then no longer conditioned on the observed future states, but on the expected future state predicted from the present state. Formally, the maximizations are moved outside the expectations. The recursive definition in Eq.(13) is updated for open loop assumptions to,

$$L_{s,k}(x_{s,k}, d_{s,k}, \lambda_k) = U_s(x_{s,k}) - \lambda_k l_{s,k}(x_{s,k}, d_{s,k}) \quad (14)$$
$$+ \max_{d_{s,k+1}} E_{x_{s,k+1}|x_{s,k},d_{s,k}} \left\{ L_{s,k+1}(x_{s,k+1}, d_{s,k+1}, \widehat{\lambda}^*_{k+1}) \right\} .$$

Synchronization of track updates and search scans is achieved in the framework by ordering subtasks and parameter dependencies hierarchically. Tracking of a target within a sector is then considered as a subtask to the subtask of maintaining a radar image in the sector. The DP optimizations of track updates will be conditioned on the sequence of search scans in the sector, and on the Lagrange multiplier estimates. The optimization of search scan parameters and the generation of Lagrange multiplier estimates are made at a global level using nonlinear programming. For details we refer to [9].

A price to pay with expected resource constraints is that the uncertainty in the future Lagrange multipliers is modelled incorrectly in the predictions. For instance, the future Lagrange multipliers are correlated to the number of tracked targets, and consequently, the global effects in the optimization coming from the uncertainty in the future number of tracked targets will be handled incorrectly. A possible solution is to consider multiple scenarios of test target densities in the optimization.

4 Numerical Example

The method including the hierarchical ordering of searching and tracking tasks has been implemented as a benchmarking reference algorithm. The method is denoted LRM (Lagrange Relaxation Method) herein. In this section we illustrate the use of the LRM algorithm as a tool in radar design. The LRM is compared with both a Track While Scan (TWS) policy, which updates tracks while scanning for new targets, and an ad-hoc adaptive tracking (AT) policy. AT uses the search scans only to cue adaptive tracking, for which track updates are scheduled once every 10 second per track. In the scenario herein, 10 seconds is enough to give satisfactory track quality. More targets result in higher track load, and left over time is used for searching. Track loads larger than one result in that AT has to drop tracks, leading further to reduced overall performance. For both AT and TWS, search scans are assumed to cover 360 degrees before new scans are started.

Due to the open loop assumptions concerning search scans and Lagrange multipliers in optimization, detailed evaluations of LRM require extensive Monte Carlo simulations. A quick but low precision alternative is to do evaluations using the prediction given by the optimization. The evaluations of LRM are then based on the open loop assumptions regarding search scans and Lagrange multipliers. Furthermore, the Markovian discrete event model of Fig.2 is used also for evaluations of TWS and AT. The detection probabilities are then calculated for each target individually, and for each time instant k conditioned on the following entities: target location in relation to the radar orientation, target radar cross section, the sequence of search scans in the sector of the target $u_{\mathcal{S}}$, scheduled target updates, $d_{\text{upd},i,k}$. By inserting these probabilities in the Markovian discrete event model, the probability vector of the target performance state is simulated using Eq.(5).

ESA Radar: The sensor is an airborne ESA radar with the antenna fixed to the aircraft hull. Along the antenna axis, the possibility to focus the energy is poor, and the antenna gain is small. Along the broadside of the antenna, the antenna gain is high. The sensor thus has non-homogenous properties in space. The models are standard for airborne surveillance radar. For example, a Swerling 1 model is assumed for the fluctuation of the target radar cross section. The specifications for the radar equation is that a signal to noise ratio is 1 for a 1m^2 sized target at 500km, given 0.01s integration time. The two-way 6dB-beamwidth is in the order of a degree. The Kalman filter used is a standard cartesian filter in position and velocity based on a Singer model [1, Chapter 4]. Measurement uncertainty standard deviations are 50 meters in range and 0.1 degrees in azimuth. See [9] for details.

Task Definition: The task definition is chosen homogenously in that each target is given the same utility function. A unity reward is given for each second of tracking when the desires on tracking performance are fulfilled i.e., $U_{\text{nom},i} = 1$ in Eq.(1) for all i. Here, we assume that the desired accuracy is a standard deviation less than 500 meters both in range and azimuth. The reward is dropped linearly from one to zero when the standard deviation in either range or cross range is dropped from 500 meters to 5000 meters. Let $\sigma_0 = 500$ and $\sigma_1 = 5000$, and let $\sigma_{r,i,t|s}$, $\sigma_{cr,i,t|s}$ be the filter standard deviations in range and azimuth. In the example, we have used the following definition of the function $Q_{\text{acc},i}(P_{i,t|s})$ in Eq.(1),

$$Q_{\text{acc},i}(P_{i,t|s}) = Q_0\left(\sigma_{r,i,t|s}\right) Q_0\left(\sigma_{cr,i,t|s}\right) , \quad (15)$$
$$Q_0(\sigma) = \begin{cases} 1 , & \sigma \le \sigma_0 \\ (\sigma_1 - \sigma)/(\sigma_1 - \sigma_0) , & \sigma_0 < \sigma \le \sigma_1 \\ 0 , & \sigma > \sigma_1 \end{cases}$$

Track mixes and re-initiations are penalized with costs $C_{\text{reinit},i} = C_{\text{mix},i} = 20$. The cost for a track mix at measurement update n when taking the expectation is then $C_{\text{mix},i}P(I_{\text{mix},i}(t_n))$, where $P(I_{\text{mix},i}(t_n))$ is calculated approximately based on the track density in the scenario.

To include the startup transient in our simulations, $k = 0$ corresponds to system startup and no targets are tracked at $k = 0$. The horizon of the decision-making is chosen as 200 seconds in order to include both the startup phase of the performance transient, and a period where the transient has levelled out. The horizon corresponds to $N = 100$ stages where Δ_t is chosen as 2 seconds.

Scenario: We consider 2 sectors with homogenous target density, where the density differ between the sectors, see Fig.4. The large sector represents a background target density, while the small sector represents all sectors with an increased density. We assume the high density areas can be identified. The background density was set to $\rho_{\mathrm{S1}} = 0.5\rho_0$, where $\rho_0 = 1.5 \cdot 10^{-4}$ per km^2 is the density of test targets distributed homogenously in space, while three different densities were chosen for the small sector, $\rho_{\mathrm{S2}} = \rho_0 \cdot \{2, 3, 5\}$. Fig.5 shows the resulting transients, where the utility is proportional (approximately with a factor between 0.7 to 0.9 herein) to the number of tracked targets. TWS is only considered for $\rho_{\mathrm{S2}} = 5\rho_0$. In the other cases TWS is outperformed. The search scans of both TWS and AT are started favorably at the edge of the high density sector S2, and continued 360 degrees before a new search scan is started.

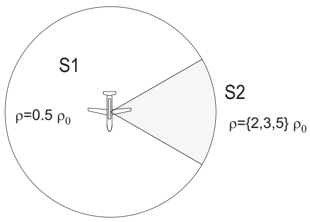

Fig. 4: A scenario with two sectors where the average target density differs between the sectors.

When $\rho_{\mathrm{S2}} = \rho_0 \cdot \{2, 3\}$, the total density is such that AT can maintain all tracks, and the performance difference is minor to LRM, as expected. However, for the final scenario with high total density, $\rho_{\mathrm{S2}} = 5\rho_0$, the difference is significant, also to TWS. LRM improves by adapting measurement behavior in the sectors such that it resembles AT more in S1 and TWS more in S2.

The two-sectors scenario is simplified compared to real scenarios. Therefore, the example only gives an indication on the possible performance gains achievable by improving the AT and TWS heuristics. Real scenarios are considerably more inhomogeneous, leading to that the gain of LRM would be higher, if the sectors with increased target densities are identified.

5 Conclusions

In this paper we formulated resource allocation in adaptive airborne surveillance radars as a stochastic optimal control problem on a regular and slow timescale. We modelled performance of radar tracking target-wise as Markov decision processes. An optimization algorithm based on Lagrange relaxation was employed to achieve an approximate solution to the complicated optimization problem. The algorithm relies on a separation of the problem into components, which are optimized locally. The method was implemented and demonstrated as an offline tool for benchmarking other methods. For that purpose, it has great value in radar design.

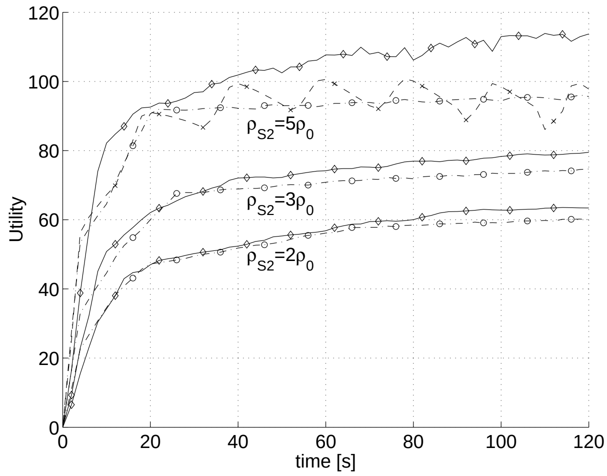

Fig. 5: Startup transients of utility for the scenario in Fig.4. LRM corresponds to solid curves, TWS to dashed, and AT to dash-dotted. For the low density scenarios, TWS is outperformed, and the transient is not shown.

References

[1] S. Blackman and R. Popoli. *Modern Tracking Systems*. Artech House, 1999.

[2] V.C. Vannicola and J.A. Mineo. Expert system for sensor resource allocation. *Proceedings of the 33rd Midwest Symposium on Circuits and Systems*, 1990.

[3] G. Van Keuk and S. Blackman. On phased array radar tracking and parameter control. *IEEE Transactions on Aerospace and Electronic Systems*, 29, 1993.

[4] D. Strömberg and P. Grahn. Scheduling of tasks in phased array radar. *IEEE International Symposium on Phased Array Systems and Technology*, 1996.

[5] K. Kastella. Discrimination gain to optimize detection and classification. *IEEE Transaction on Systems, Man and Cybernetics*, 27(1), 1997.

[6] R. Washburn, M. Schneider, and J. Fox. Stochastic dynamic programming based approaches to sensor resource management. *5th International Conference on Information Fusion*, 2002.

[7] V. Krishnamurthy and R. Evans. Hidden markov model multiarmed bandits: A methodology for beam scheduling in multitarget tracking. *IEEE Transactions on Signal Processing*, 49, 2001.

[8] D. Bertsekas. *Dynamic Programming and Optimal Control, second edition, Volume I*. Athena Scientific, 2000.

[9] J. Wintenby. *Resource Allocation in Airborne Surveillance Radar*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 2003.

[10] D. Castañon. Approximate dynamic programming for sensor management. *Proceedings of the 36th Conference on Decision and Control*, 1997.

[11] S. Mori, K. Chang, and C. Chong. *Multitarget-Multisensor Tracking: Applications and Advances, Volume II*, chapter 7, Performance Analysis of Optimal Data Association with Applications to Multiple Target Tracking. Artech House, 1992.

[12] C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.