Hierarchical Resource Management in Adaptive Airborne Surveillance Radars

JOHANNES WINTENBY
Chalmers University of Technology
Sweden

VIKRAM KRISHNAMURTHY, Fellow, IEEE
University of British Columbia

In this paper we present hierarchical resource management algorithms for adaptive airborne surveillance radars. 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 presented based on Lagrangian relaxation. A numerical example is given with scenarios involving different target densities.

Manuscript received October 29, 2003; revised April 6, 2004 and September 6, 2005; released for publication November 3, 2005.

IEEE Log No. T-AES/42/2/876418.

Refereeing of this contribution was handled by L. M. Kaplan.

This work was supported by Ericsson Microwave Systems, Sweden, and partially supported by NSERC and the British Columbia Advanced Systems Institute.

Authors' addresses: J. Wintenby, Dept. of Signals and Systems, Chalmers University of Technology, 412 96 Goeteborg, Sweden; V. Krishnamurthy, Dept. of Electrical and Computer Engineering, University of British Columbia, Vancouver, V6T 1Z4, Canada, E-mail: (vikramk@ece.ubc.ca).

I. INTRODUCTION

Electronically scanned antenna (ESA) radars possess an agile beam that permits adaptive allocation of transmitted energy in space and time. There is strong motivation in designing dynamic radar resource management algorithms that exploit this beam agility to facilitate the ESA radar handling a variety of tasks, such as tracking a set of targets and searching a sector for new targets. Radar resource management algorithms aim to enhance the overall radar system performance—this performance is ultimately judged on how closely the radar track database matches actual target trajectories. 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 that needs to be considered. Due to the stochastic nature of radar detection and target dynamics, scheduling of radar measurements is a stochastic control problem. The control of ESA radars is studied in the literature under the field of sensor management [3]. There are the following two broad methodologies in the literature for formulating radar resource management problems.

1) *Heuristic Scheduling based on Rule-Based Design*: Under this methodology, a scheduling rule is defined based on descriptive rule based design, see [3], [14] and also [7], [8], for single target resource allocation algorithms. Detailed scheduling of measurement order, given a set of measurements with specified types and processing time intervals is considered in [13]. Heuristic schedulers are often designed to operate in real time with limited computational requirements. However, since heuristic schedulers are not based on optimizing a cost function, their performance is difficult to predict.

2) *Optimization-Based Scheduling*: In the optimization-based approach for radar resource management, a multi-stage cost function is minimized over a finite or infinite horizon. Globally optimal stochastic optimization methods such as stochastic dynamic programming (DP) can in principle be used to compute the optimal radar resource management policy. Unfortunately, the curse of dimensionality inherent in DP makes their direct application intractable. One can resort to the myopic case, i.e., optimize an instantaneous cost [6], but this is typically inappropriate in airborne surveillance. Indeed, optimizing the radar performance over a long term horizon (e.g. one to several minutes) is desirable in airborne surveillance radar resource allocation due to the following reasons.

1) Due to the large surveillance volumes to be covered with scarce resources, the number of tracked targets and track load depend dynamically on the search scan allocation. In particular, the long-term dynamical behavior of the number of tracked targets

and the system performance as a function of the allocated search resources need to be taken into account. These dynamics also involve future track load and hence the future resource demand.

2) It is preferable to avoid repeated target initiations and drops due to optimistically trying to track resource consuming, low-SNR targets in situations where the system is saturated.

3) Course changes of the ESA radar platform and the spatially inhomogeneous antenna gain of an ESA, lead to dynamically changing resource demand when tracking a set of targets.

4) Reacquisitions of targets that reappear after a blindness period (i.e., Doppler blindness, elevation and vegetation mask for ground targets), and tracking a set of interacting targets with possibly mixing tracks, result in time intervals of increased resource demand. Before such time intervals, it is important for the resource management algorithm to decide which measurements are affordable and to what extent other, parallel tasks of the system will be affected.

5) Synchronization of search scans and adaptive track updates may reduce the resource demand. However, synchronization requires the radar resource manager to consider a time interval stretching at least over the next search scan pass of the target.

To solve the long-term stochastic optimization problem via DP, simplifying assumptions are needed. In the existing literature, several such assumptions are used as we now briefly describe.

1) In [14], [10], and [5] it is assumed that all targets are tracked initially and the scheduling problem concerns only the ordering of track update measurements. In radar resource allocation, this assumption is inappropriate since the search scan is an integral part of the resource management problem. Furthermore, the radar resource manager needs to consider track drops and track reinitiations apart from track accuracy. In the literature, there seems to be little previous work in simultaneous treatment of searching and tracking that explores dynamics in tracking performance and resource demand.

2) In [10] and [5] to facilitate efficient prediction of the tracking dynamics, a regular, discrete timescale is assumed where at each time instant a single measurement occurs. Such an assumption is inappropriate in radar resource management since the measurement times are controlled.

3) In [10] and [9] the multi-dimensional kinematic state was quantized to a Markov chain, and the beam scheduling problem was formulated as a special type of partially observed Markov decision processes (POMDPs) called a multi-armed bandit. However, the multi-armed bandit assumption is quite restrictive for airborne surveillance radar resource management—since it assumes that the tracking dynamics are isolated to one track per time interval. Moreover, for general POMDPs, the number of states becomes prohibitively large, leading to computational intractability. (POMDPs are appropriate for the classification problem where the state is discrete, [4].)

In this paper, we present hierarchical methods for optimization-based resource management of ESA surveillance radars. The purpose is to provide practically feasible methods for optimization-based resource management given a series of simplifications and approximations which do not severely restrict the optimal solution. An important aspect is that both searching and tracking are considered. Moreover, the method provides a tool in radar design for benchmarking heuristic schedulers. Although this paper is primarily directed towards surveillance radars, it can also be applied to other sensor systems, 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 in this paper, and the steps behind arriving at the resource management method, are the following.

1) *Two-Level Two-Timescale Scheduling, and Abstraction of Measurement Operations*: Scheduling of radar measurements naturally decomposes into two different scales. At the slow timescale with regular intervals in the order of seconds, the radar resource manager needs to decide on the batch of measurements to make in the following time interval, i.e., what measurements to make and how to make them. We refer here to this slow-timescale decision process as resource allocation.

Given the decisions on the slow timescale, the local order of measurements within a batch is arranged into a sequence of measurements by a fast-timescale scheduler. We assume the relevant dynamics of tracking performance are captured by the slow timescale, and that optimization of local arrangement of measurements within the batches is of minor significance for the system performance. Therefore, the emphasis here is on the slow-timescale scheduling, and we do not consider joint design of the slow-timescale and the fast-timescale scheduling.

At the slow timescale, measurements are abstracted into measurement operations. A measurement operation is considered to be an algorithm in the radar that generates 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 [7]. Fast feedback measurements that need to be made on a fast timescale are allowed within measurement operations. For example, repeated attempts to update a target are handled by the fast-timescale scheduling. Thus, the fast-timescale scheduler must be able to include the fast feedback measurements in the sequence of measurements, while processing the batch of measurement operations. Thereby, the batch-wise

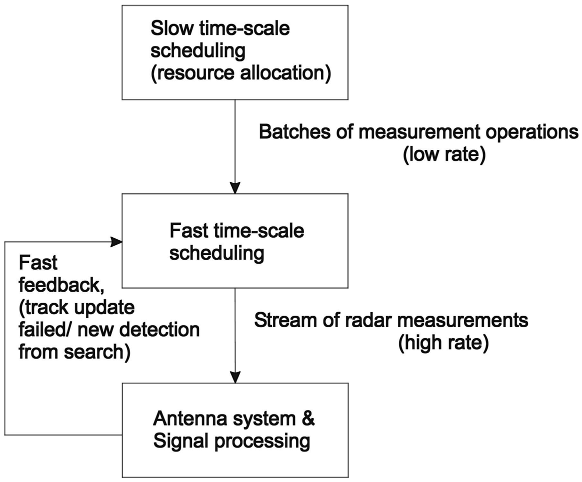


Fig. 1. Two-timescale scheduling.

slow-timescale decision making in resource allocation is facilitated. An overview of the two-timescale approach is shown in Fig. 1. In Section II, ideas behind the two-level two-timescale scheduling approach are presented further.

2) *Formulation of the Slow-Timescale Resource Allocation as a Stochastic Optimization Problem*: The physical level aspects of sensor performance are abstracted into Quality of Service (QoS) measures used in an optimization criterion. The QoS measures are defined target-wise based on concepts such as track accuracy and track continuity, and expressed in terms of tracking utility. Models for predicting the QoS measures, given the decision parameters, are presented in Section III. The models are formulated target-wise based on discrete-time Markov chains, where the sampling period of the chains coincide with the regular, slow timescale.

A single sensor performance measure is defined as an aggregate of the target-wise measures, integrated over a time horizon, see Section IVA. In Section IVC control parameters for the measurement operations on the discrete, slow timescale are described. Finally, resource constraints for the batch-wise planning of measurements are discussed in Section IVB. The resource constraints are made explicit by a series of constraints on the utilized time per time interval (i.e., load), where the time intervals coincide with the intervals on the slow timescale.

3) *Approximate Lagrange Relaxation Formulation and Separation of the Problem into Components*: In Section V, resource constrained optimal resource allocation on the slow timescale is pursued using approximate DP based on Lagrange relaxation [4]. That is, by replacing future stochastic Lagrange multipliers with estimates given average resource constraints, the resource allocation problem is separated into components. Thereby, optimization can be performed component-wise, and the coordination of the components are carried out via the resource constraints. This decomposition into smaller components results in a computationally tractable optimization-based resource allocation which otherwise would suffer from combinatorial explosion. The price is that the uncertainty in the future effects from present decisions will not be fully considered in the predictions during optimization.

4) *Hierarchical Extension and Lagrangian Relaxation Method (LRM)*: To incorporate coordination between track updates and the search scans, a hierarchical extension to the separated problem is proposed in Section VH. The space is divided into a set of sectors where the same search scan sequence is used for each sector. Decisions on track updates are then conditioned hierarchically on the search scan parameters of the sector of the track. Searching and tracking in a sector then 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 Section VI, a numerical example is provided which demonstrates the utilization of the method as a tool in radar design. In the example, we benchmark the performance of LRM against two important heuristic tracking methods: adaptive tracking (AT), and track while scan (TWS), given homogenous scenarios with varying target density. Evaluations were made using a sophisticated testbed which has the capability of realistically modelling an airborne ESA surveillance radar in arbitrary scenarios. The example shows that LRM is useful as an offline reference for radar resource management. Also, the LRM yields insight on how to improve real time utilization of radar resources.

II. OVERVIEW OF PROPOSED TWO-TIMESCALE SCHEDULING ALGORITHM

In this section, we motivate the two-timescale scheduling ESA radar resource management algorithm proposed here. Details are presented in Section V, and in particular, a pseudocode example of optimization-based slow-timescale scheduling is given in Section VG.

The radar resource management problem can be formulated as follows. At time instants when the radar is idle, decide on the next measurement e.g., a single or sequence of coherent processing intervals (CPIs), so as to optimize the future radar performance integrated over a time horizon. The long-term consequences of the decisions are for example modelled by maintaining an optimized plan of future measurements, and after each measurement is completed, new information is received implying that replanning is needed. An optimal (stochastic DP) approach to the problem is practically infeasible due to the large state space (exponential in the number of targets) and the irregular timescale on which decisions need to be made.

In order to cope with the large dimensionality issues and yet develop a computationally feasible algorithm, the following approximations are introduced here.

Slow-Timescale Planning: The initial idea is to maintain a plan of future measurements, and only replan at regular time instants, occurring on a slower rate than the actual measurements. Batches of the plan are then extracted at these instants for further processing. The regular replan instants define a slow timescale, where the length of the time intervals is denoted Δ_t. Herein, the intervals on the slow timescale are called batch intervals, and the planning on the slow timescale is denoted resource allocation.

Utilization of Measurement Operations, and Two-Timescale Scheduling: The introduction of a slow timescale in replanning is not straightforward because of fast feedback measurements. Fast feedback measurements occur when trying to adaptively confirm search detections, or update a track by using repeated measurement attempts. That is, if an attempt to update a track fails, another attempt is immediately scheduled until the target is detected, or the update attempt failed. Decisions regarding fast feedback measurements cannot be handled at the slow timescale.

To still be able to maintain a slower rate of planning, measurements are abstracted into measurement operations including the fast feedback measurements. A measurement operation is an algorithm which generates the measurements needed to achieve a low-level radar task such as "update track using repeated attempts," or "search a sector with confirmation measurement." The plan is then constructed out of these operations without detailed insight into the actual algorithm defining the operations. A consequence from using measurement operations is that a fast timescale scheduler is needed, which transfers a sequence of measurement operations to a sequence of measurements. The fast timescale scheduler is responsible for fitting the fast feedback measurements into the sequence of measurements, and to look to that the resulting sequence is feasible in terms of ambiguity resolution [13]. We assume a feasible schedule is always possible to achieve by rearranging measurements locally in time.

Fast-timescale scheduling, and slow-timescale resource allocation form together a two-level two-timescale scheduler. In Fig. 2, the two timescales are illustrated. Both the regular replanning instants, and the time interval following a replanning instant, are enumerated with k. A batch of measurement

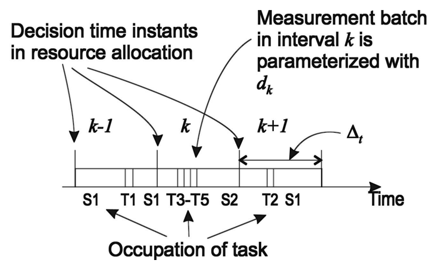


Fig. 2. Batches of search and track update operations are decided on at regular time intervals on slower rate than that of measurements. T1 represents track update operation, and S1 represents sector search.

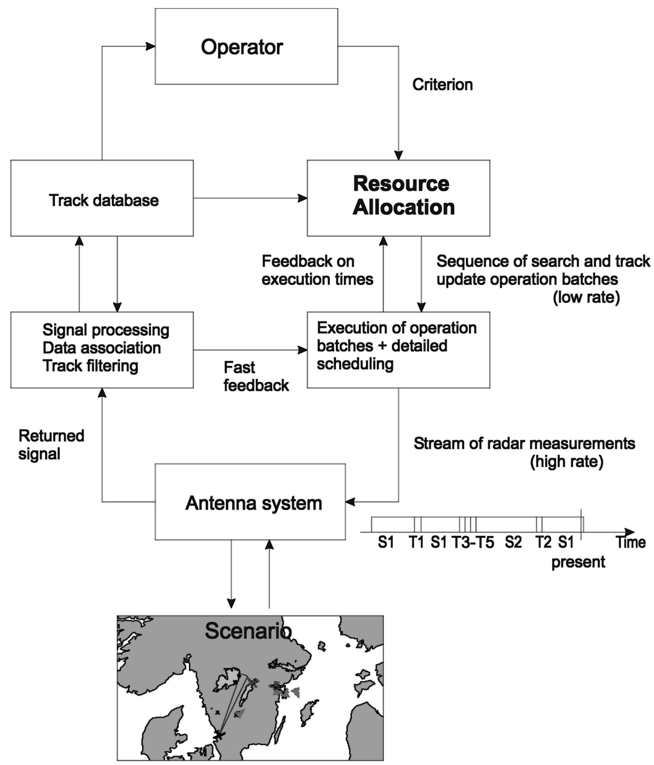


Fig. 3. Overview of two-timescale, online resource allocation in radar system as suggested in this paper. Resource allocation maintains a plan from which batches of measurement operations are extracted at regular time intervals. Measurements in batches are arranged by fast-timescale scheduler also handling fast feedback. Output of fast-timescale scheduler is stream of measurements propagated to antenna system. T1 represents track update operation, and S1 represent search operation in a sector.

operations in interval k is parameterized with d_k. An overview of the approach in a radar system is presented in Fig. 3.

Handling Random Batch Execution Time: Since fast feedback measurements depend on random detection events, the execution time of a batch of measurement operations varies randomly. A new batch of measurements is still generated with a rate of $1/\Delta_t$. Measurements that are not completed at the time when a new batch is generated, are executed before and together with the new batch. As a result, the accumulated time of measurements waiting to be executed may grow with time. To counteract such a growth, a feedback of the execution time from the previous batch is needed. The expected execution time of next batch is then adjusted with the observed variation in time from the previous batch; see Section IVB for a short discussion. If a batch finishes early, the next batch should be ready for processing by the fast-timescale scheduling.

Ignoring Measurement Order in a Batch at the Slow Timescale: Search and tracking performance is fairly insensitive to variations in the time instants of the measurements. That is, the dynamics of tracking performance (e.g., accuracy and track continuity) are slower than the high rate at which measurements are obtained. Consequently, optimizing the detailed order of measurements is not important for system performance. Instead the main issue is how and approximately when to make measurements. We choose Δ_t such that the relevant dynamics of system performance are captured on the slow timescale. Decisions regarding measurement order within slow-timescale batches can then be ignored by the resource allocation algorithm and left to the fast-timescale scheduler. Moreover, optimization-based fast-timescale scheduling or joint optimization of fast-timescale and slow-timescale scheduling is not crucial for the global system performance. Therefore, we do not pursue joint design and the emphasis of this paper is on the slow timescale.

Finally, the sequential nature of the radar (i.e., the radar can only take one action at a given time instant) is relaxed by introducing constraints on the resource utilization of the batches, see Section IVB. In Section V, the resource constraints are included in the slow-timescale resource allocation via Lagrange relaxation, which is a key for achieving a separation of the problem into components. The constraints are then used to coordinate the components. As a consequence, the combinatorial explosion of the resource allocation problem due to the number of targets is alleviated.

We present the formulation of resource allocation on the slow timescale as a stochastic optimization problem in Section IV. First, a model for prediction of tracking performance on the slow timescale is discussed.

III. MODELLING OF TRACKING PERFORMANCE DYNAMICS

In this section, we present a Markov chain model for prediction of tracking performance dynamics on the slow timescale. Such a model is fundamental for the formulation of optimization-based resource allocation, and for evaluation of noninstantaneous sensor performance measures.

The following attributes are important when considering the radar output quality and thus the tracking performance.

1) Detect and maintain tracks of the targets.

2) Keep the same identity of the tracks throughout the surveillance volume, i.e., to avoid track drops leading to reinitiations with new track identities, and to avoid mixing two or more tracks. Track mixes are difficult to predict, and as an approximation we will predict incorrect data associations events instead, although not all association failures lead to track mixes.

3) Sustain the accuracy of the tracks.

A. Markov Chain Formulation

One would like to formulate a model for the prediction of the dynamics of these items from the sequence of measurements d_k. We use target-wise Markov chains to capture the dynamics.

Assume a scenario with targets $\mathcal{T}_i$, $i \in \{1,\ldots,M\}$, where M is the number of targets. In the sequel, i indexes the targets. The target kinematic states are defined as $\xi_i(t) = [r_{x,i}(t), v_{x,i}(t), r_{y,i}(t), v_{y,i}(t)]^T$, where $r_{x,i}(t), r_{y,i}(t)$ are the position parameters of the target, and $v_{x,i}(t), v_{y,i}(t)$ are the velocity parameters. The kinematic states evolve dynamically according to a linear state space model [1],

$$\xi_i(t+T) = F(T)\xi_i(t) + w_i(T) \qquad (1)$$

where $F(T)$ is the state transition matrix of the target state model, and $w_i(t,T)$ is the driving maneuver input modelled as white and Gaussian with,

$$E\{w_i(t,T)\} = 0, \qquad \mathrm{cov}\{w_i(T)\} = Q_i(T). \qquad (2)$$

We assume for simplicity that Kalman filters are used to compute the state estimate $\hat{\xi}_{i,t|s}$ of the track of target i at time t, given the measurements included in the track up to time s. Furthermore, the conditional covariance is $P_{i,t|s} = E\{\tilde{\xi}_{i,t|s}\tilde{\xi}^T_{i,t|s}\}$, where $\tilde{\xi}_{i,t|s} = \xi_i(t) - \hat{\xi}_{i,t|s}$.

For each target, we define a finite valued discrete time state $x_{i,k}$ that measures the tracking performance. The time index k refers to the slow timescale and the tracking performance dynamics are captured on this slow timescale. At each time instant k, the state $x_{i,k}$ is an aggregate of state variables needed when expressing an instantaneous target-wise tracking utility $U_i(x_{i,k})$, for instance:

1) $x_{\mathrm{tracked},i,k} \in \{0,1\}$, indicating if a target is tracked or not in the time interval k.

2) $x_{\mathrm{dropped},i,k} \in \{0,1\}$, indicating if a target has been tracked, but is now dropped.

3) A discrete parameterization of $P_{i,t|t_m}$, the conditional covariance representing the online accuracy, given by the Kalman filter when the target

is tracked. The time t_m represents the time of the last update, $t_m \leq t$. On the discrete timescale, we write $P_{i,k|k_m}$, $k_m \leq k$. Below, an example is presented of a discrete parameterization of $P_{i,k|k_m}$.

4) $x_{\text{reinit},i,k} \in \{0,1\}$, an indicator state of that a track has been reinitiated after a period where the target was dropped. Track reinitiations only occur at time instants when a target is observed by a search scan. The indicator state triggers for one time interval on the slow timescale.

5) $x_{\text{mix},i,k} \in \{0,1\}$, an indictor state for a track mix. We assume track mixes only occur at track updates. The indicator state triggers for one time interval on the slow timescale.

The process $x_{i,k}$, is assumed to evolve according to a finite Markov chain on the slow timescale. However, the modelling of the transitions of the chain involve various probabilities, which depend on the continuous-valued kinematic state $\xi_i(t)$. We assume that the dynamics of $\xi_i(t)$ can be approximated as deterministic when calculating transition probabilities by replacing $\xi_i(t)$ with the estimate $\hat{\xi}_{i,t|t_m}$. Thereby, all stochastic entities involved when expressing tracking performance are treated as discrete, which is highly practical in performance predictions. For instance, this allows formulation of the radar resource allocation problem as a finite-state Markov decision process.

Let $p_{x_{i,k}}$ denote the state probability vector of $x_{i,k}$. Assume that transitions of $x_{i,k}$ are affected by the measurements in batch k, parameterized with d_k. Then, the target-wise dynamic model has the following form,

$$p_{x_{i,k+1}} = P_{tr,1}(d_k,\xi_{i,k})p_{x_{i,k}} \tag{3}$$

where $P_{tr,1}$ is the transition matrix of the Markov chain, and $\xi_{i,k} = \xi_i(k\Delta_t)$. The kinematic state, $\xi_{i,k}$, has been conditioned on explicitly since it affects the detection probabilities and evolves dynamically (although assumed deterministically), and consequently the Markov chain is nonstationary. If there is any case where a deterministic treatment of $\xi_i(t)$ is inappropriate, it is possible to utilize a multiple scenarios approach. That is, a set of multiple future realizations of $\xi_i(t)$ can be assumed, where (3) is evaluated for each realization. The state $x_{i,k}$ is assumed to be fully observed so that when time instant k occurs, the state of the performance model $x_{i,k}$ is known.

The rest of this paper is based on the Markov chain model in (3). We now discuss specific issues regarding the model including an example on how to quantize $P_{i,k|k_m}$, an example of a Markov chain for performance prediction, a discussion on how to calculate transition probabilities, and an estimation of the number of states in the chain. A numerical example is presented where the Markov chain is used for tracking performance prediction.

B. Discrete Parameterization of Accuracy

We present here a discrete parameterization of the Kalman filter covariance to represent a distribution over the covariance at predictions using the Markov chain $x_{i,k}$. All time indices in this subsection are at the slow timescale.

Let k_n denote the discrete time interval when observation n occurs. Let the time since the previous update be $T_n = k_n - k_{n-1}$ (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

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

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 space to the observation space.

Computing $P_{i,k|k_m}$ requires knowledge of the initial filter covariance P_{i,k_0} at time k_0, the mapping $Q_i(T)$, the measurement covariances $R_{i,n}$, the prediction time k, and the time instants of the measurement updates k_n, $n = 1,\ldots,m$. At predictions, only the time instants k_n are stochastic due to uncertainties in when observation instants will occur. The rest of the variables are either known or functions of the sequence k_n and k. Therefore, we include k and, k_n, $n = 1,\ldots,m$ as variables explicitly in the discrete state $x_{i,k}$, while the other variables are treated as implicitly available. Thereby, $P_{i,k|k_m}$ is parameterized by the state $x_i(t)$. Although the sequence k_n grows with time, the memory in the recursion in (4) is typically short, and it is sufficient to include the last few intervals between the observation time instants in the state, e.g. $\{k - k_m, T_m\}$, to get an approximate discrete parameterization of $P_{i,k|k_m}$. To initialize the recursion, a covariance matrix based on the average update rate in the recent past is used.

C. Example of a Markov Chain Model for Performance Prediction

An example of a Markov chain for target-wise tracking performance prediction is presented in Fig. 4. In the chain, there are states included to account for the state components, $x_{\text{tracked},k}$, $x_{\text{dropped},k}$, $x_{\text{reinit},k}$, $x_{\text{mix},k}$. Furthermore, the parameterization of $P_{i,k|k_m}$ suggested above, i.e., $\{k - k_m, T_m\}$, is depicted as a "matrix" valued state. A target is tracked whenever one of the states in the matrix is active. If a tracked target is detected, the Markov chain jumps to one of the leftmost states depending on $k - k_m$ (at updates detections $k - k_m$ becomes T_m). At track updates, an incorrect association is accounted for by activating a

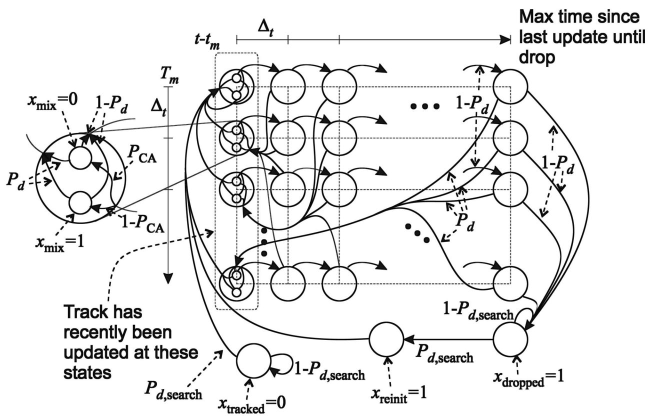


Fig. 4. Example of Markov chain for keeping track of dynamic, discrete performance state, i.e., if target is tracked or not, track reinitiation, track mixes, and $k - k_n, T_n$ given that target is tracked. Transition probabilities are conditioned on when and how measurements are made. On target detection, transition is to one of leftmost states depending on time since previous update. On a detection failure, a step to the right will be taken. States for indication of track mixes with nearby targets are included in leftmost states, which is illustrated in the magnification.

mix state, $x_{\text{mix},k} = 1$, given the predicted probability of correct association P_{CA}. The detection probabilities are conditioned on whether measurements are made in an interval. The search scan has a detection probability $P_{d,\text{search}}$ given an interval where the search scan passes the target, otherwise it is zero. Whenever a target is tracked, both the search scan and the adaptive updates may update the track. Thus, in Fig. 4, P_d depends on if the search scan passes the target, if an adaptive track update is made, or if both events occurs in a time interval.

D. Transition Probabilities

The detection probabilities are determined by the signal-to-noise ratio (SNR), which is calculated via the radar equation [3, ch. 2] for each target. If for example the search scan passes a target in a time interval, the detection probability is determined by $P_{d,\text{search}}(\text{SNR})$. Real time computation of these detection probabilities requires tracking of the expected radar cross section for each known target.

Different expressions apply for various measurement types. Adaptive target updates utilize optimized sequences of track update attempts to achieve a detection probability close to one (though the expected processing time of the sequence increases when the target appears weaker). Since an airborne ESA radar has its antenna fixed to the aircraft hull, the radar has nonhomogenous properties in space. The expressions for SNR include this spatial dependency. We assume detection thresholds are lower for tracked targets compared with those of yet undetected targets. Therefore, P_d given a search scan pass, will be higher for tracked targets than for yet undetected targets, $P_{d,\text{search}}$. This comparative gain is of high importance for the combined performance of searching and tracking of weak targets, since reinitiations are suppressed.

We do not pursue the modelling of detection probabilities and processing times of adaptive updates further here. For details of the modelling of detection probabilities, we refer to [15, ch. 2, 3, 4].

Target-to-track mixes occur as consequences of plot-to-track data association errors in dense parts of the scenario, although only some data association failures lead to track mixes. As a rough estimate of the event, the probability of a plot-to-track association error may be used. Approximate expressions are presented in [12] for scenarios with a homogeneous density of targets, where the number of targets in a volume is Poisson distributed with the density as a parameter. Consider a homogeneous environment of tracked targets in the vicinity of target i (number of targets in a volume is assumed Poisson distributed), and denote the density of tracked targets $\rho_{\text{track},i}$. All neighboring targets are observed simultaneously with $P_d = 1$. The probability of a correct plot-to-track association is then approximated as,

$$P_{\text{CA}} \approx e^{-\pi \rho_{\text{track},i} \sqrt{|S_{i,k|k_m}|}} \qquad (5)$$

where $S_{i,k|k_m} = HP_{i,k|k_m}H^T + R_{i,k}$ is the covariance of the measurement residual given by the Kalman filter at time k.

In [15, ch. 4], prediction of association errors based on open loop assumptions are studied for scenarios given two crossing target trajectories. For approximate predictions in these scenarios, we have simplified forms of calculating the plot-to-track association error events based on the expected future trajectories, and the filter accuracies at the measurement update instants. These filter accuracies are parameterized approximately with the Markov chain above. A DP example for controlling adaptive update instants of two crossing targets is given in Section VF.

The quantization of $P_{i,k|k_m}$ will dominate the number of states in the Markov chain. Assume the maximum time from the last target detection until the target is dropped is 20 s, and $\Delta_t = 1$ s. Thus we need 20 samples of $k - k_m$. Furthermore, assume we sample T_m with 2 s intervals implying 10 samples of T_m. Consequently, the number of states is $20 \times 10 = 200$ plus states needed for accounting for incorrect associations, target drops, reinitiations, etc. Fortunately, the transition matrix of the chain will be very sparse, and therefore we can accept larger chains.

Example of Performance Predictions using the Markov Chain Model: Consider the single target scenario presented in Fig. 5. The target approaches the radar at a speed of 300 m/s, and the platform

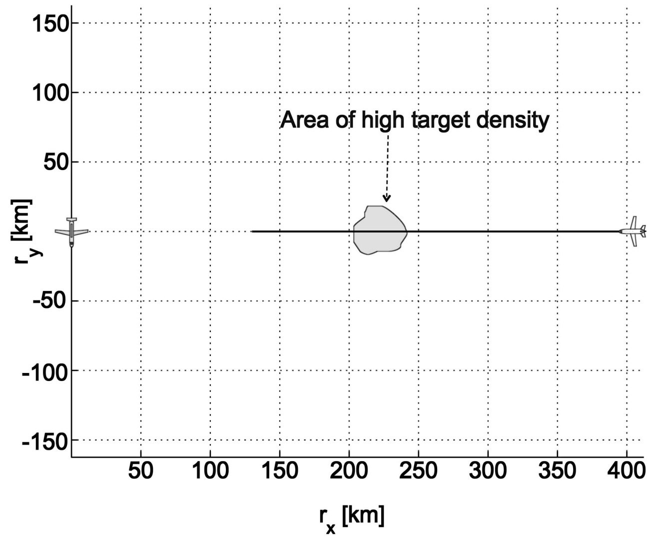


Fig. 5. Incoming target scenario. Target velocity is 300 m/s, and platform is assumed to be stationary at origin. Target passes through an area with increased target density, which is 0.01 targets per km^2 at the maximum.

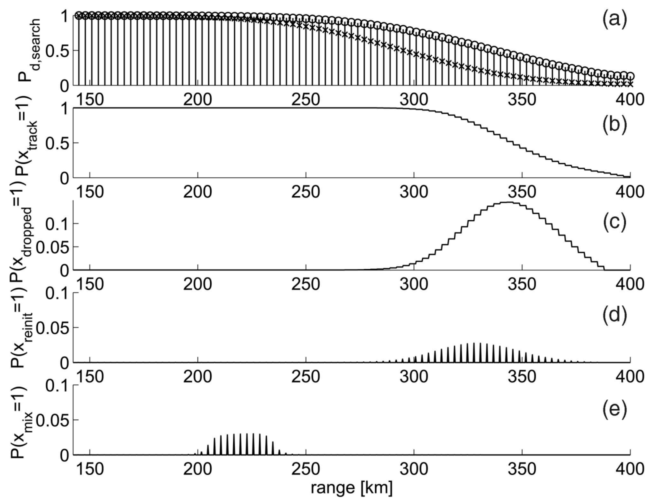


Fig. 6. Results from predictions using Markov chain in scenario. From top to bottom: (a) detection probability of search scan in searching (crosses), and tracking (circles), where detection thresholds are lowered 3dB, (b) probability of tracking target, (c) probability that target is dropped, (d) probability of track reinitiation indication, (e) probability of track mix indication (here, incorrect data association).

is assumed stationary at the origin. Along the path, the target passes through an increased target density area, with target density set to 0.01 targets per km^2. The radar utilizes TWS with a time between search scan passes of 10 s. The Markov chain in Fig. 4 is employed to model the target performance components, where $\Delta_t = 1$ s, and the maximum time until target drops is 40 s. That is, at least one detection every fourth search scan pass is required for track maintenance. In Fig. 6, the predicted results are presented in terms of $P(x_{\text{tracked},k} = 1)$, $P(x_{\text{dropped},k} = 1)$, $P(x_{\text{reinit},k} = 1)$, and $P(x_{\text{mix},k} = 1)$. The relatively high risk of dropping and reinitiating a target indicated by the figure is partly due to the poor initiation criterion, i.e., the track is assumed to be started on the first detection. In reality, an initiation criterion is likely to be used, thus blocking tracking until detections are received more regularly.

IV. RESOURCE ALLOCATION FORMULATION

We now proceed with the formulation of resource allocation on the slow timescale as a stochastic optimization problem, given the tracking performance model from the previous section. This includes formulating objectives and constraints of resource allocation, and a parameterization of measurement control actions.

A. Formulation of Objectives

Based on the tracking performance state, $x_{i,k}$, introduced in Section III, we can now formulate an instantaneous utility measure $U_i(x_{i,k})$. The utility function is specified for each target individually.

Define:

$U_{\text{nom},i}$, a nominal utility measure for tracking a target, also corresponding to a user priority.

$Q_{\text{acc},i}(P_{i,k|k_m})$, a scalar-valued quality measure for accuracy between zero and one.

$C_{\text{reinit},i}$, a cost for a reinitiation of target i.

$C_{\text{mix},i}$, a cost for a track mix of target i.

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

$$U_i(x_{i,k}) = x_{\text{tracked},i,k} U_{\text{nom},i} Q_{\text{acc},i}(P_{i,k|k_m}) - C_{\text{reinit},i} x_{\text{reinit},i,k} - C_{\text{mix},i} x_{\text{mix},i,k}. \quad (6)$$

If desired, $U_{\text{nom},i}$, $C_{\text{reinit},i}$, and $C_{\text{mix},i}$ can be made dependent on the kinematic state $\xi_{i,k}$, and thus geographically dependent. The overall instantaneous utility of the radar system at time t is defined as $U(x_k) = \sum_{i=1}^{M} U_i(x_{i,k})$, where x_k is the aggregated state of all target-wise performance models, i.e., $x_k = \{x_{i,k}\}_{i=1}^{M}$.

Example (continued from Section IIID): As an illustration, the expected instantaneous utility measure, $E\{U(x_k)\}$ has been evaluated on the results from the incoming target scenario presented in Fig. 6. The following definitions are used:

$$U_{\text{nom}} = 1$$

$$Q_{\text{acc}}(P_{k|k_m}) = Q_{\text{range}}\left(\sqrt{H_{\text{r}} P_{k|k_m} H_{\text{r}}^T}\right) Q_{\text{cross-range}}\left(\sqrt{H_{\text{c-r}} P_{k|k_m} H_{\text{c-r}}^T}\right)$$

$$Q_{\text{range}}(\sigma) = Q_{\text{cross-range}}(\sigma) = \left\{\begin{array}{ll} 1, & \sigma \le \sigma_0 \\ (\sigma_1 - \sigma)/(\sigma_1 - \sigma_0), & \sigma_0 < \sigma \le \sigma_1 \\ 0, & \sigma > \sigma_1 \end{array}\right\}$$

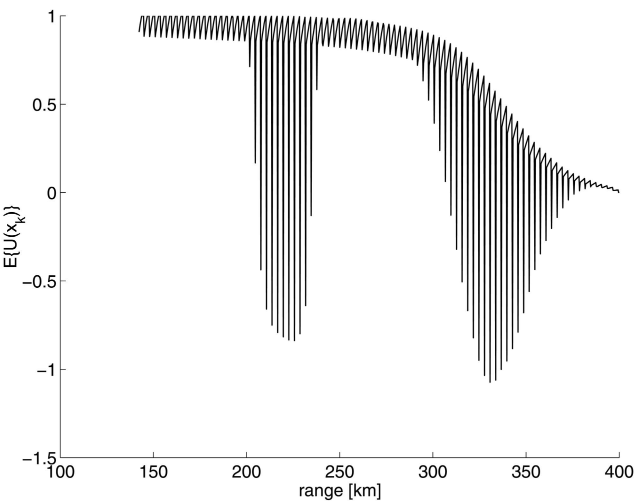


Fig. 7. Example of instantaneous utility measure evaluated on results of incoming target scenario. See Figs. 5 and 6.

where $\sigma_0 = 500$, $\sigma_1 = 5000$, and where H_{r}, $H_{\text{c-r}}$ projects on to the range and the cross-range axes of the target.

$C_{\mathrm{reinit}} = C_{\mathrm{mix}} = 60$, i.e., reinitiations and track mixes are punished with 1 min of accurate tracking.

Computing $P_{k|k_m}$ requires the covariance of the maneuver noise $Q(T)$ (see (2)) and the covariance of the measurement noise R_n. In the numerical examples presented here, the following matrices are used:

$$Q(T) = q\begin{bmatrix} Q_0(T) & \\ & Q_0(T)\end{bmatrix}, \qquad Q_0(T) = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T\end{bmatrix} \tag{7}$$

$$R_{i,n} = \begin{bmatrix} \cos\varphi_{i,n} & -r_{i,n}\sin\varphi_{i,n} \\ \sin\varphi_{i,n} & r_{i,n}\cos\varphi_{i,n}\end{bmatrix}\begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2(\varphi_{i,n})\end{bmatrix} \times \begin{bmatrix} \cos\varphi_{i,n} & -r_{i,n}\sin\varphi_{i,n} \\ \sin\varphi_{i,n} & r_{i,n}\cos\varphi_{i,n}\end{bmatrix}^T \tag{8}$$

where $r_{i,n}$ is the range to a target i at the time of observation n, $\varphi_{i,n}$ is the azimuth to the target in relation to the antenna normal axis, σ_r is the measurement standard deviation in range set to 50 m, and σ_φ is the measurement standard deviation in azimuth set to $\sigma_\varphi(\varphi_{i,n}) = 0.1\pi/(180\cos\varphi_{i,n})$. In the examples herein, $q = 1000$.

In Fig. 7, the resulting expected utility is shown. Note the inverted peaks occurring due to the non-zero probability of track reinitiations and track mixes. The instantaneous utility as defined herein can very well be below zero. The jigsaw pattern is due to the variation in accuracy affecting the utility via $Q_{\mathrm{acc},i}(P_{i,k|k_m})$.

Noninstantaneous Utility: 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_0 at the present time indexed with $k = 0$, the aim is to maximize a noninstantaneous utility,

$$J_0(x_0) = E\left\{\sum_{k=0}^{N-1} U(x_k)\Delta_t \,\middle|\, x_0\right\} \tag{9}$$

with respect to the measurement batch d_0. Here, N denotes the prediction horizon. The expectation is over the future radar measurements, including measurement time instants and measurement errors, and ideally over future target trajectories. In the sequel, the multiplication with Δ_t will be omitted in the notation since it merely acts as a scaling. The objective function in (9) can also be formulated as a discounted, noninstantaneous utility.

For explicitness, the optimization problem is expressed as the recursion,

$$\max_{d_0} U(x_0) + E_{x_1|x_0,d_0}\{J_1^*(x_1)\}, \tag{10}$$

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.,

$$J_k^*(x_k) = \max_{d_k} U(x_k) + E_{x_{k+1}|x_k,d_k}\{J_{k+1}^*(x_{k+1})\}. \tag{11}$$

The decision parameter vectors d_k are here assumed to fulfill the resource constraints on available measurement time. Note that in (11), the modelling of the decision consequences has the form of a recursion with nested maximizations and expectations.

Test Targets and Constant Number of Targets M during Predictions: Targets yet undetected at $k = 0$ may become detected within the prediction horizon, thus entering the evaluation of (9) (this is the benefit of searching). To account for these targets, a set of a priori modelled test targets are utilized to sample the space of trajectories of yet unknown targets. The density of the test targets corresponds to an expected density of targets in the whole or in parts of the scenario, or alternatively, weights can be put on the test targets to achieve the desired density of the model. Online, the prior probabilities that there are still undetected targets along the test target trajectories are computed given the previous search scans launched in the particular areas. This is done for all test targets.

Given the utilization of test targets, the number of targets M in the evaluation of (9) will remain constant during predictions, although the predicted number of tracked targets will change dynamically. This results in a saving of complexity in the formulation compared to if M represented the number of tracked targets, which would imply a dynamically changing M. In Fig. 8, an example is given of test target trajectories incoming from two directions.

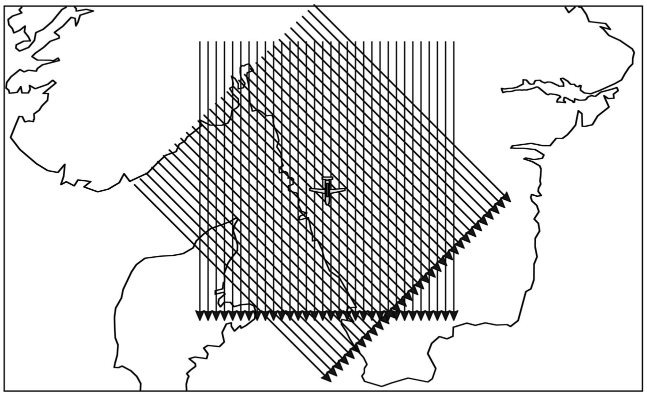


Fig. 8. Illustration of trajectories of test targets used to predict density of tracked targets at $k > 0$, yet undetected at time $k = 0$. Figure shows nonmaneuvering trajectories from two different directions.

B. Resource Constraints

When disregarding order of measurements within a batch in resource allocation, the implicit constraint that the radar only can do one thing at a time is removed. At planning, the constraint must be replaced with constraints on the resource utilization of the batches.

As a consequence of the utilization of measurement operations, the actual execution time of a of a measurement batch will generally be random. Let $u_k(x_k, d_k)$ be the measurement execution time of batch k, and define $l_k(x_k, d_k) = u_k(x_k, d_k)/\Delta_t$ as the load in interval k. Note that the load is also a function of the kinematic state, in particular the target positions in relation to the radar platform. This dependence is assumed implicitly for notational convenience. Desirably, the sequence of resource constraints used in planning are formulated as,

$$l_k(x_k, d_k) \leq_{w.p.1} 1, \qquad \forall \quad k \geq 0. \tag{12}$$

Unfortunately, this formulation is computationally infeasible. Since the standard deviation, σ_{l_k}, is typically a fraction of one, it makes sense to approximate (12) with expectations,

$$\bar{l}_k(x_k, d_k) = E\{l_k(x_k, d_k) \mid x_k, d_k\} \leq_{w.p.1} c_k. \tag{13}$$

Note that the expectation is not over the state at this stage. The sequence c_k is a reaction to observed outcomes of the execution time in previous interval, e.g., in time interval $k-2$. For instance, $c_k = 1 - \delta_{k-2}/\Delta_t$, where δ_{k-2} is the observed deviation from $(k-1)\Delta_t$ in finishing time of the batch in interval $k-2$. A full consideration of the variation of batch processing time requires that the distribution of c_k is carried along in the state x_k. Unfortunately, the computational demand would be large. However, since the deviation δ_{k-2} usually is small, the loss from ignoring the variation at predictions is anticipated to be minor. Therefore, we let $c_k = 1$ for $k > 1$ at planning. At the first time interval, $k = 0$, no approximation is needed. For notational convenience we put $c_k = 1$ for all k subsequently.

C. Parameterization of Measurements on the Slow Timescale

We deal with three kinds of measurements: search scans, adaptive track updates for single targets, and adaptive updates for a group of tracks or crossing targets. 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 modelling, such as integration gain, probability of detection, and the time instant of a scan passing over a certain azimuth location, are computed out of $u_{\mathcal{S}}$.

The decisions regarding a sequence of adaptive target updates are parameterized with a sequence of discrete parameters, $d_{\mathbf{upd},i} = \{d_{\mathbf{upd},i,0}, d_{\mathbf{upd},i,1}, \ldots, d_{\mathbf{upd},i,N-1}\}$, where $d_{\mathbf{upd},i,k} \in$ {"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. A sequence of update attempts is optimized locally with the objective to minimize the expected time to achieve a target detection at a high probability, see e.g. [7]. Discrete decision variables for target updates facilitates DP when controlling track updates. However, the resulting measurement time will be random. Adaptive track updates of closely spaced targets are similar to adaptive updates of one target. A sequence of update attempts are launched until all targets are updated. Each attempt may consist of one or several CPIs of different beam positions, although a gain only occurs if more than one target can be observed per CPI.

The parameter vector d_k is an aggregation of all search scan parameters and track update parameters of time interval k.

V. OPTIMIZATION-BASED RESOURCE ALLOCATION WITH APPROXIMATE DYNAMIC PROGRAMMING

An optimization algorithm for the nested, stochastic control problem formulated in (10) and (11), with constraints (13), typically relies on stochastic DP. 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. In this section, we suggest an approximate relaxation of the resource constraints (13) to separate the problem into components, thereby achieving a simplification. An optimization algorithm is presented, utilizing the separation.

A. Separation into Subtasks

An approach to large-scale control problems is 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 resource constraints (13) in this case. 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 affect tracking performance of the target, and perhaps of nearby targets, but not of 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 have local effects in the performance modelling of the subtasks. The utility of a subtask $\mathbf{s}$ is then written

$$U_{\mathbf{s}}(x_k) = \sum_{\{i \mid \mathcal{T}_i \text{ belongs to } \mathbf{s}\}} U_i(x_{i,k}). \tag{14}$$

Define $x_{\mathbf{s},k}$ as the aggregated state for the targets sorted to $\mathbf{s}$. The system utility at time k is then $\sum_{\mathbf{s}} U_{\mathbf{s}}(x_{\mathbf{s},k})$. For subtask $\mathbf{s}$, and time interval k, the decision parameters are denoted as $d_{\mathbf{s},k}$. 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 $\mathbf{s}$ are assumed to be locally dependent on $d_{\mathbf{s},k}$,

$$p_{x_{i,k+1}} = P_{tr,1}(d_{\mathbf{s},k}, \xi_{i,k}) p_{x_{i,k}}, \qquad \{i \mid i \text{ belongs to } \mathbf{s}\}. \tag{15}$$

A separation with this property is a division of space into exclusive 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. 9.

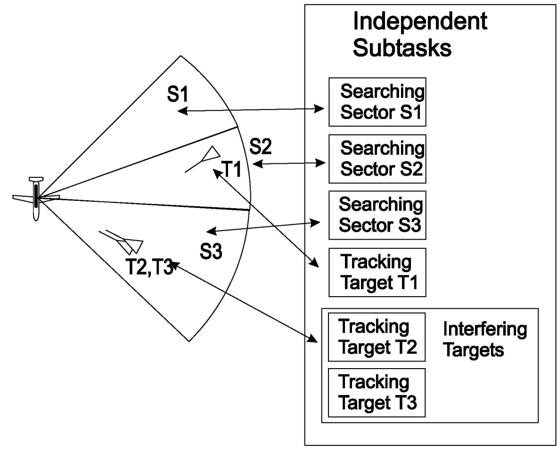


Fig. 9. Formation of independent subtasks. S1, S2, and S3 represent sectors that should be searched, while T1, T2, and T3 are tracked targets.

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

$$\sum_{\mathbf{s}} \bar{l}_{\mathbf{s},k}(d_{\mathbf{s},k}, x_{\mathbf{s},k}) = \sum_{\mathbf{s}} E\{l_{\mathbf{s},k}(d_{\mathbf{s},k}, x_{\mathbf{s},k}) \mid d_{\mathbf{s},k}, x_{\mathbf{s},k}\} \leq 1, \qquad k \in \{0, N-1\}. \tag{16}$$

B. Approximate Lagrange Relaxation

According to (15), decisions on track updates and search scans have local effects on tracking performance for each subtask disregarding one fact: the measurements compete for the same constrained resources. By introducing Lagrange relaxation, the constraints on the resources are included explicitly in the optimization

$$\begin{aligned} L_k(x_k, d_k, \lambda_k) &= U(x_k) + \lambda_k(1 - \bar{l}_k(x_k, d_k)) \\ &\quad + E_{x_{k+1} \mid x_k, d_k}\left\{ \max_{d_{k+1}} L_{k+1}(x_{k+1}, d_{k+1}, \lambda^*_{k+1}(x_{k+1})) \right\}. \end{aligned} \tag{17}$$

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 Lagrange multipliers are regarded as internal variables in an optimal decision mapping $d^*_k = \mu^*_k(x_k)$ which includes the resource constraint at time k, however, we chose to make the Lagrange multipliers explicit. The optimal Lagrangian at time k, $L_k(x_k, d^*_k(x_k), \lambda^*_k(x_k))$ is equal to the optimal value-to-go function $J^*_k(x_k)$. At the last stage N, $L_N(x_N) = U(x_N)$.

One of the aims of introducing the Lagrange multipliers is to separate the optimization problem into subtasks. Using the expressions from (14)–(16) the Lagrangian is reformulated. For clarity, the $k+1$ stage is also expanded,

$$
\begin{aligned}
L_k(x_k,d_k,\lambda_k) &= \sum_{\mathbf{s}} U_{\mathbf{s}}(x_{\mathbf{s},k}) + \lambda_k\left(1-\sum_{\mathbf{s}} \bar{l}_{\mathbf{s},k}(x_{\mathbf{s},k},d_{\mathbf{s},k})\right)\\
&\quad + E_{x_{k+1}|x_k,d_k}\left\{\max_{d_{k+1}}\sum_{\mathbf{s}} U_{\mathbf{s}}(x_{\mathbf{s},k+1}) + \lambda^*_{k+1}(x_{k+1})\right.\\
&\qquad \times\left(1-\sum_{\mathbf{s}} \bar{l}_{\mathbf{s},k+1}(x_{\mathbf{s},k+1},d_{\mathbf{s},k+1})\right)\\
&\qquad \left. + E_{x_{k+2}|x_{k+1},d_{k+1}}\left\{\max_{d_{k+2}} L_{k+2}(x_{k+2},d_{k+2},\lambda^*_{k+2})\right\}\right\}.
\end{aligned} \tag{18}
$$

Rearranging terms gives

$$
\begin{aligned}
L_k(x_k,d_k,\lambda_k) &= \sum_{\mathbf{s}} (U_{\mathbf{s}}(x_{\mathbf{s},k}) - \lambda_k \bar{l}_{\mathbf{s},k}(x_{\mathbf{s},k},d_{\mathbf{s},k})) + \lambda_k + E_{x_{k+1}|x_k,d_k}\\
&\quad \times\left\{\max_{d_{k+1}}\sum_{\mathbf{s}} (U_{\mathbf{s}}(x_{\mathbf{s},k+1}) - \lambda^*_{k+1}(x_{k+1})\bar{l}_{\mathbf{s},k+1}(x_{\mathbf{s},k+1},d_{\mathbf{s},k+1}))\right.\\
&\quad + E_{x_{k+2}|x_{k+1},d_{k+1}}\left\{\max_{d_{k+2}} L_{k+2}(x_{k+2},d_{k+2},\lambda^*_{k+2})\right\}\\
&\quad \left. + \lambda^*_{k+1}(x_{k+1})\right\}.
\end{aligned} \tag{19}
$$

A separation of (19) in terms of subtasks requires that the inner sums can be moved outside both the maximizations and the expectations. In these expressions, $\lambda^*_k(x_k)$ is a function of the global state, and this prevents the separation. However, there will be many subtasks, and we assume that the variation of $\lambda^*_k(x_k)$ is moderate compared with the average $E_{x_k|x_0}\{\lambda^*_k(x_k)\}$. It is then reasonable to replace $\lambda^*_k(x_k)$ for $k > 0$ with estimates. These estimates are denoted $\hat{\lambda}^*_k$, and chosen such that,

$$
E_{x_k|x_0}\{\bar{l}_k(x_k,d^*_k(x_k))\} = 1, \qquad k > 0. \tag{20}
$$

Note that the expectation now includes the state, which is not the case in (13). Since $\hat{\lambda}^*_k$ are conditioned on the present state x_0, and future information input is disregarded, they are regarded as open loop estimates. It will be a part of a global optimization algorithm to search for $\hat{\lambda}^*_k$.

There are low-pass effects which will reduce the fast term variation of $\lambda^*_k(x_k)$, motivating the use of estimates. That is, since a Lagrange multiplier can be interpreted as a marginal price for a resource, a large λ_k compared with λ_{k-1} and λ_{k+1} will reduce the desire to allocate in interval k while moving allocations to the neighboring intervals, thus reducing the price λ_k. Measurements tend to spread out in time, and the pressure on the constraints will be slowly varying. The long term variation of the Lagrange multipliers typically depends on the number of tracked targets which will be stochastic as well. This variation is usually small due to averaging effects. However, the variation of batch execution time, which gives rise to a reaction in the capacity c_k of a following interval (see Section IVB), will induce a variation in λ^*_k due to correlation between c_k and λ^*_k. To account for this variation, an ad-hoc, small variation of λ^*_k which is white and independent of the state is suitable to include in the modelling. Then, $\hat{\lambda}^*_k$ is assumed to be the mean of this variation. At optimization, an expectation should be taken over this variation as well. Since the expectation will be placed outside the maximization, it will help making derivatives exist. This is useful for speeding up optimization.

Using the open loop estimates $\hat{\lambda}^*_k$, a recursive argument is carried out that separates the problem. For notational convenience, define a vector of Lagrange multiplier estimates $\tilde{\lambda} = [\lambda_0, \hat{\lambda}^*_1, \ldots, \hat{\lambda}^*_{N-1}]$, and denote the element corresponding to time k as $\tilde{\lambda}_k$. Assume that the Lagrangian at time $k+1$ can be rewritten as a sum of Lagrange components for each subtask, plus a term depending on the multipliers only,

$$
L_{k+1}(x_{k+1},d_{k+1},\tilde{\lambda}) = \sum_{\mathbf{s}} L_{\mathbf{s},k+1}(x_{\mathbf{s},k+1},d_{\mathbf{s},k+1},\tilde{\lambda}) + \sum_{n=k+1}^{N-1}\tilde{\lambda}_n. \tag{21}
$$

The Lagrangian at time k is then expressed as,

$$
\begin{aligned}
L_k(x_k,d_k,\tilde{\lambda}) &= \sum_{\mathbf{s}} (U_{\mathbf{s}}(x_{\mathbf{s},k}) - \tilde{\lambda}_k \bar{l}_{\mathbf{s},k}(x_{\mathbf{s},k},d_{\mathbf{s},k})) + \tilde{\lambda}_k\\
&\quad + E_{x_{k+1}|x_k,d_k}\left\{\max_{d_{k+1}}\sum_{\mathbf{s}} L_{\mathbf{s},k+1}(x_{\mathbf{s},k+1},d_{\mathbf{s},k+1},\tilde{\lambda})\right\}\\
&\quad + \sum_{n=k+1}^{N-1}\tilde{\lambda}_n.
\end{aligned} \tag{22}
$$

The maximum operation is separable in the subtasks due to the local influence of decision parameters in subtasks, e.g., if $f_{\mathbf{s}}(d_{\mathbf{s}})$ is a set of functions representing local consequences of the decision $d_{\mathbf{s}}$ regarding measurements of subtask **s**, we have that

$$
\max_d \sum_{\mathbf{s}} f_{\mathbf{s}}(d_{\mathbf{s}}) = \sum_{\mathbf{s}} \max_{d_{\mathbf{s}}} f_{\mathbf{s}}(d_{\mathbf{s}}). \tag{23}
$$

Thus, the sum can be moved outside the maximization. Furthermore, the expectation is carried out per subtask due to the independence assumptions regarding target-wise performance. Consequently, given the assumption in (21), the Lagrangian at stage

k is also separable in the subtasks:

$$L_k(x_k,d_k,\tilde{\lambda}_k) = \sum_{\mathbf{s}} \Bigg(U_{\mathbf{s}}(x_{\mathbf{s},k}) - \tilde{\lambda}_k \bar{l}_{\mathbf{s},k}(x_{\mathbf{s},k},d_{\mathbf{s},k}) + E_{x_{\mathbf{s},k+1}|x_{\mathbf{s},k},d_{\mathbf{s},k}} \left\{ \max_{d_{\mathbf{s},k+1}} L_{\mathbf{s},k+1}(x_{\mathbf{s},k+1},d_{d_{\mathbf{s}},k+1},\tilde{\lambda}) \right\} \Bigg) + \sum_{n=k}^{N-1} \tilde{\lambda}_n \triangleq \sum_{\mathbf{s}} L_{\mathbf{s},k}(x_{\mathbf{s},k},d_{\mathbf{s},k},\tilde{\lambda}) + \sum_{n=k}^{N-1} \tilde{\lambda}_n. \quad (24)$$

At the last stage, we have that $L_N(x_N) = \sum_s U_{\mathbf{s}}(x_{\mathbf{s},k}) \triangleq L_{\mathbf{s},N}(x_{\mathbf{s},N})$. Carrying out a recursive argument from time N and proceeding backwards, the Lagrangian at the decision time instant $k = 0$ separates in the subtasks.

In [4], a similar approximate DP approach given averaged resource constraints is used with the aim of achieving a separation of a different sensor management problem, namely that of optimizing target classifications.

C. Separated Optimization Solution

A possible optimization algorithm for the problem is to locally optimize each subtask given $\tilde{\lambda}$, and to iteratively search for $\tilde{\lambda}$ such that the expected resource constraints (20) are fulfilled. Let $\tilde{\lambda}^j$ denote the Lagrange multiplier estimate at iteration j. Moreover, denote $E\{\bar{l}^j\} = [\ldots, E_{x_k|x_0}\{\bar{l}_k^j\}, \ldots]$, the vector of predicted, expected load given the optimized decisions at iteration j. Assume that $E\{\bar{l}^j\}$ is continuous and differentiable with respect to $\tilde{\lambda}$. Via a first-order Taylor expansion, the equations in (20) are solved iteratively for $\tilde{\lambda}$ according to,

$$\tilde{\lambda}^{j+1} = \tilde{\lambda}^j + \Delta\tilde{\lambda}^j, \qquad E\{\bar{l}^j\} + \frac{\partial E\{\bar{l}^j\}}{\partial \tilde{\lambda}} \Delta\tilde{\lambda}^j = 1. \quad (25)$$

Thus, at each iteration, $d^*_{\mathbf{s},k}$ are optimized locally for all k, given the present estimate of $\tilde{\lambda}$. The expected load coming from the subtask is computed based on $d^*_{\mathbf{s},k}$, and $E\{\bar{l}^j\}$ is formed by summing over all subtasks. Furthermore, $\partial E\{\bar{l}^j\}/\partial\tilde{\lambda}$ is summed from the partial derivatives of each subtask. Then, (25) is used to generate the new $\tilde{\lambda}$ iterate.

For target tracking subtasks, $L_{\mathbf{s},0}(x_{\mathbf{s},0},d_{\mathbf{s},0},\tilde{\lambda})$ is discontinuous with respect to $\tilde{\lambda}$ due to the maximization over the discrete update variables. Then, a subgradient method is applicable. However, by assuming that future λ_k will vary randomly, where $\hat{\lambda}^*_k$ only represents a nominal value, a faster search in a continuous space is possible.

Next, we outline how to calculate the Lagrangian for three different subtasks. After that, pseudocode for resource allocation based on the approach outlined herein is given.

D. Adaptive Tracking Subtasks

If the subtask is to generate updates for a single target where the state space is fairly small, (24) forms a base for DP. Resource constraint are included via the Lagrange multiplier cost terms. The Markov decision process in (3) based on a chain such as in Fig. 4 can be used straight off.

Consider a subtask of tracking target $\mathcal{T}_i$. Denote the value-to-go function for $\mathcal{T}_i$ at time k as $J_{i,k}(x_{i,k})$. Let $J_{i,k}(x_{i,k})$ be equal to the local Lagrangian of the subtask given the vector $\tilde{\lambda}$ i.e., $J_{i,k}(x_{i,k}) = L_{\mathcal{T}_i,k}(x_{i,k},d_{\mathbf{upd},i,k},\tilde{\lambda})$, ($J_{i,k}(x_{i,k})$ is implicitly a function of $d_{\mathbf{upd},i,k}$ and $\tilde{\lambda}$). Given $\tilde{\lambda}$, the local decision problem is characterized as a Markov decision process. The optimization is achieved with a DP backwards recursion,

$$J^*_{i,k}(x_{i,k}) = \max_{d_{\mathbf{upd},i,k}} \Bigg(U_i(x_{i,k}) - \tilde{\lambda}_k \bar{l}_i(x_{i,k},d_{\mathbf{upd},i,k}) + \sum_{x_{i,k+1}\in X_i} J^*_{i,k+1}(x_{i,k+1}) P(x_{i,k+1} \mid x_{i,k},d_{\mathbf{upd},i,k},\xi_{i,k}) \Bigg). \quad (26)$$

Here, $J_{i,N} = U_i(x_{i,N})$, and $\bar{l}_i(x_{i,k},d_{\mathbf{upd},i,k})$ is the expected load given a scheduled track update, and given the filter covariances predicted by the state in time interval k. Since $d_{\mathbf{upd},i,k}$ is a binary variable, and since $\bar{l}_i(x_{i,k},d_{\mathbf{upd},i,k})$ is zero given no update, the maximum operation is possible to interpret as the following comparison. At each time and state, the marginal difference in future utility from launching an update or not,

$$\sum_{x_{i,k+1}\in X_i} J^*_{i,k+1}(x_{i,k+1}) \times (P(x_{i,k+1} \mid x_{i,k},\mathbf{upd},\xi_{i,k}) - P(x_{i,k+1} \mid x_{i,k},\mathbf{wait},\xi_{i,k})) \quad (27)$$

is compared with the cost of the update, $\tilde{\lambda}_k \bar{l}_i(x_{i,k},\mathbf{upd})$; update a track if the marginal utility gain from making the update is larger than the cost.

When $\hat{\lambda}^*_k$ is modelled as stochastic (in order to improve modelling faults connected to the open loop assumption, e.g., by modelling a small amount of

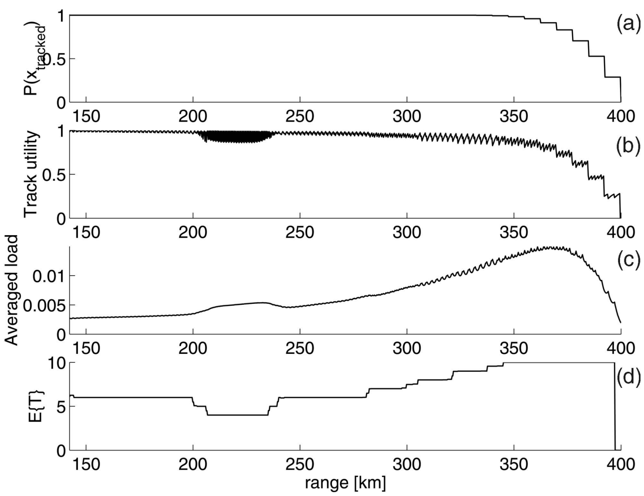


Fig. 10. Results from DP of adaptive track update instants for incoming target in Fig. 5. From top to bottom: (a) probability of tracking target, (b) expected utility, (c) average track load given by AR-filter of true, peaky load curve, (d) expected time between adaptive track updates.

white Gaussian noise in $\hat{\lambda}_k^*$), the optimal value-to-go function $J_{i,k}^*(x_{i,k})$ will become stochastic for $k > 0$, even conditioned on the state. When accomplishing the DP recursion, the probabilities, $P(d_{\mathbf{upd},i,k}^* = \mathbf{upd})$ and $P(d_{\mathbf{upd},i,k}^* = \mathbf{wait})$ are computed instead of $d_{\mathbf{upd},i,k}^*$ for $k > 0$, in combination with the mean and approximate variance of $J_{i,k}^*(x_{i,k})$. A detailed explanation is out of scope of the paper.

Example of DP in Single Target Tracking (continued from Sections IIID and IVA): Consider the incoming target scenario from Fig. 5. Assume a stationary load situation in a radar system, where $\tilde{\lambda}_k$ is equal to a constant for all k. In the example, $\tilde{\lambda}_k$ is chosen from a standard scenario. The typical range of $\tilde{\lambda}_k$ depends on, for instance, the system in question, models for determination of $\bar{l}_i(x_{i,k}, \mathbf{upd})$, target signal strength and density, and the choice of utility measure. Fig. 10 shows results from carrying out the recursion (26) for the incoming target. The models and the utility function are the same as in Section IIID and Section IVA, with the following differences.

1) Tracking is carried out only by adaptive updates, no search scan measurements update the track once it is started.

2) The time between search scan passes is 25 s instead of 10 s. A lower scan rate brings the possibility to increase coherent integration and thereby improve the detection range.

3) The maximum time between adaptive updates is set to 10 s.

E. Search Subtasks and Open Loop Optimization

In search subtasks, the local state space is still too large for DP due to the 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 in (24) are moved outside the expectations. The recursive definition in (24) is thus updated for open loop assumptions to,

$$\begin{aligned} L_{\mathbf{s},k}&(x_{\mathbf{s},k}, d_{\mathbf{s},k}, \tilde{\lambda}) \\ &= U_{\mathbf{s}}(x_{\mathbf{s},k}) - \tilde{\lambda}_k \bar{l}_{\mathbf{s},k}(x_{\mathbf{s},k}, d_{\mathbf{s},k}) \\ &\quad + \max_{d_{\mathbf{s},k+1}} E_{x_{\mathbf{s},k+1}|x_{\mathbf{s},k},d_{\mathbf{s},k}} \{L_{\mathbf{s},k+1}(x_{\mathbf{s},k+1}, d_{\mathbf{s},k+1}, \tilde{\lambda})\}. \end{aligned} \tag{28}$$

The maximizations in open loop are suitably extracted from the recursion,

$$\begin{aligned} &\max_{d_{\mathbf{s},0}} L_{\mathbf{s},0}(x_{\mathbf{s},0}, d_{\mathbf{s},0}, \lambda_0^*) \\ &\quad = \max_{d_{\mathbf{s},0}, d_{\mathbf{s},1},\ldots} (U_{\mathbf{s}}(x_{\mathbf{s},0}) - \lambda_0^* l_{\mathbf{s},0}(x_{\mathbf{s},0}, d_{\mathbf{s},0}) \\ &\qquad + E_{x_{\mathbf{s},1}|x_{\mathbf{s},0},d_{\mathbf{s},0}} \{U_{\mathbf{s}}(x_{\mathbf{s},1}) - \hat{\lambda}_1^* l_{\mathbf{s},1}(x_{\mathbf{s},1}, d_{\mathbf{s},1}) + E_{x_{\mathbf{s},2}|x_{\mathbf{s},1},d_{\mathbf{s},1}}\{\ldots\}\}). \end{aligned} \tag{29}$$

In a similar manner as for $\hat{\lambda}_k^*$, a random variation can be added to $d_{\mathbf{s},k}$ in order to account for the fact that $d_{\mathbf{s},k}$ is random for $k > 0$, although the open loop assumption makes them erroneously look deterministic. Thus, $d_{\mathbf{s},k}$ will be regarded as a nominal decision which will most likely be altered in the future due to the actual feedback. The open loop optimization results in a control structure called open loop feedback control, see [2, ch. 6]. Note that the approach only disregard future information in the predictions, and feedback will indeed be utilized.

The search scan performance depends on the future tracking utility and track load of yet undetected targets. To predict these entities, we use test targets as discussed in Section IVA, and illustrated in Fig. 8. For each test target, we can predict the future utility and track load utilizing Markov chains and DP, as demonstrated above in Fig. 10. This is the approach taken in the demonstration in Section VI, where the chain in Fig. 4 forms the performance model for a set of nonmoving test targets.

To the large number of possible test target trajectories and limited computational resources, the state space for each test target may have to be restricted to one or a few states, e.g., to $x_{\text{tracked},k}$. The utility and load for the tracked targets are then estimated by either assuming a fixed, average track update rate for all detected targets, or by optimizing average track update rates based on time averages of the predicted values of $\hat{\lambda}_k^*$. The later results in a hierarchical optimization problem when searching

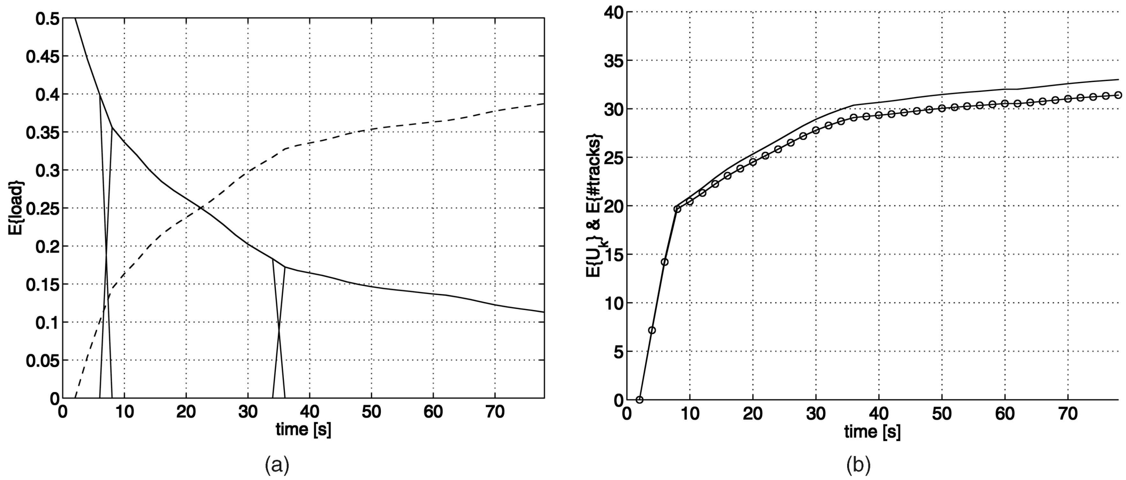


Fig. 11. Results from search example: (a) search load with load allocation of first three search scans (solid), and track load (dashed), (b) expected, instantaneous utility (curve with circles), and number of tracks, (solid without circles).

for $\hat{\lambda}_k^*$. A short example is now given of this approach.

Search Performance Prediction Example: Consider a scenario where test targets enter the scenario along trajectories illustrated in Fig. 8 from eight regularly sampled directions. The test targets trajectories are divided into cells of 7500 m. We assume that all test targets have the same velocity of 250 m/s. The test targets move to the next neighboring cell on the trajectory in a time discrete fashion every 30 s ($250 \cdot 30 = 7500$). The probability that there is an undetected target in a cell is modelled a priori for every cell such that the test targets yield a target density of $\rho_0 = 1.5 \cdot 10^{-4}$ targets per km^2. The probabilities are regarded as state variables which are affected by the history of search activities, and are carried along from one time instant to the other.

A search subtask is assumed in the scenario where a sector from -75 to $+75$ deg needs to be searched, given half the radar resource. No targets are tracked initially. In the example, $\hat{\lambda}_k^*$ is optimized to yield a total search and expected track load of 0.5. Average track update rates are decided for each tracked test target individually by extracting an asymptotic time average of the optimized $\hat{\lambda}_k^*$. The track load, track accuracy and the utility of a test target depend on the resulting update rate and on the location of the target. In this example we have not considered track mixes when deciding on the update rates, nor the coordination of search scans and track updates.

In Figs. 11 and 12, results from open loop optimization with a time horizon of 80 s, and with $\Delta_t = 2$ s are presented. Fig. 11(a), shows the search and track load as a function of time, together with the load allocation of the first three consecutive search scans. (The utilized search scan in this example includes range ambiguity resolution, binary integration, and long coherent integration times, and is therefore seemingly slow.) In Fig. 11(b), the predicted, instantaneous utility and the predicted number of tracked targets are shown as a function of time. In Fig. 12, the resulting coverage after each of the three search scans is illustrated in terms of the expected number of yet undetected targets. A white color implies a low risk of yet undetected targets.

F. Crossing Targets Tracking Subtasks

Consider the scenario with two crossing targets in Fig. 13. We have applied DP to the scenario, aiming at predicting update time instants and future track load. Although the scenario only contains two targets, modelling with a fully observed system is computationally demanding. Therefore, the following simplifications are considered.

1) An adaptive update algorithm is used with $P_d \lesssim 1$, and where both targets are observed simultaneously. This means that the sequences of update instants are captured by a common state, e.g., with tracking performance dynamics modelled by the Markov chain in Fig. 4.
2) An open loop assumption is made regarding the effect of the measurement value outcome on the kinematic state estimate. Note, in crossing target situation, the actual outcome of the state estimate affects the control reactions, which is not the case for single target tracking.
3) The probability of track mix is predicted at each stage and state by calculating two-stage plot-to-track association errors [15, ch. 4].

Predictions are made using the same constant Lagrange multiplier values, and utility function, as in Section VD. The track load model differs to that in Section VD since the targets are coscanned when fruitful.

In Fig. 14, some results from DP in the crossing target scenario are presented. The time between

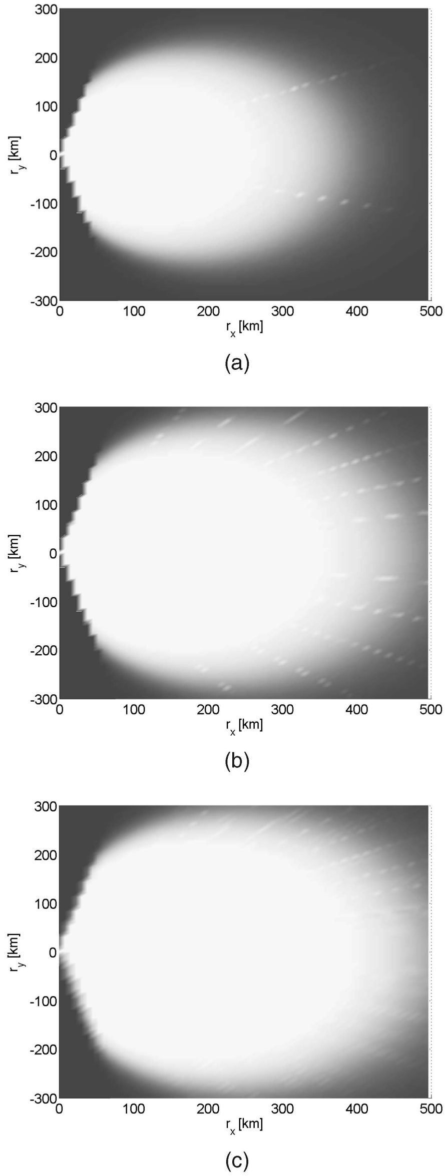


Fig. 12. Expected number of yet undetected targets as function of location after (a) first, (b) second, (c) third scan. White implies there are no undetected targets.

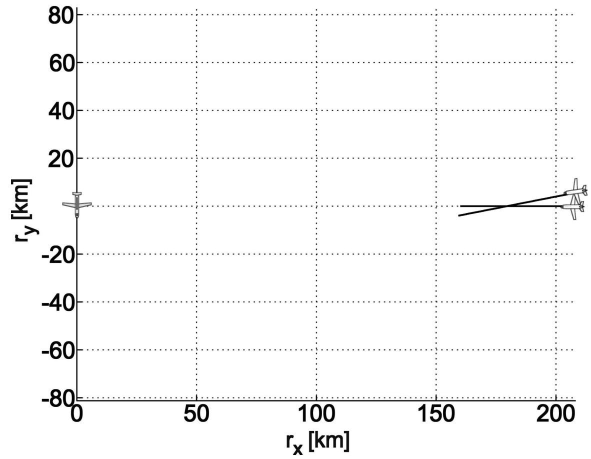


Fig. 13. A scenario of two crossing targets.

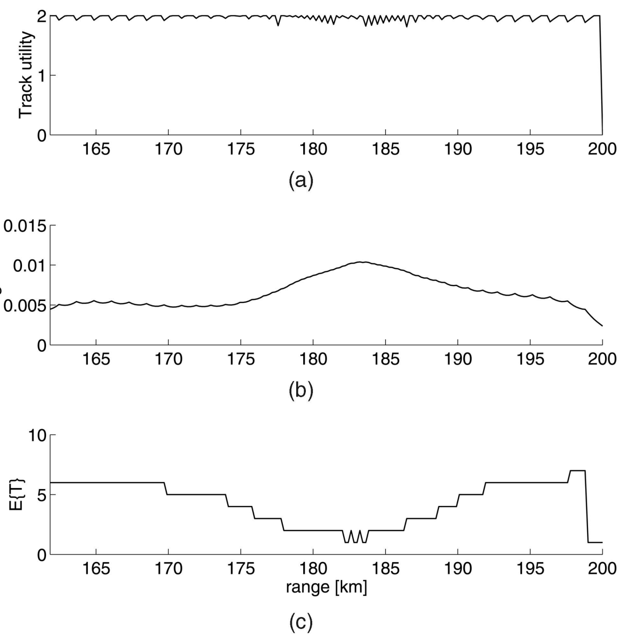


Fig. 14. Results from DP in crossing target scenario in Fig. 13. From top to bottom: (a) expected utility of the two targets, (b) expected combined track load, (c) expected time between updates as generated by DP.

updates, T, is predicted to drop when track mixes are likely, thereby counteracting mixes. The predicted instantaneous utility remains mainly unaffected during the crossing, but the expected track load is nearly doubled.

G. Pseudocode for the Separated Solution to Resource Allocation

Assume that an initial estimate of $\tilde{\lambda}$ exists, then an algorithm for generating a measurement batch is now the following.

1. Form subtasks:
   1.1. Each tracked target forms a subtask.
   1.2. Space is divided into sectors, for which the optimized search scan properties may differ, for instance due to spatially inhomogeneous utility function definitions, or varying spatial antenna properties. A search subtask is formed in each sector.
   1.3. Group tracked targets which may interact during the prediction interval, or which may benefit from simultaneous scanning. Each group forms a subtask.
2. Optimize parameters locally for each subtask, given the estimate of $\tilde{\lambda}_0$, and calculate $E\{\bar{l}^j_{\mathbf{s}}\}$, and $\partial E\{\bar{l}^j_{\mathbf{s}}\}/\partial\tilde{\lambda}$ for each subtask.
   2.1. For each tracked target which has not been grouped, utilize backwards recursive DP to optimize the time instants of updates.

2.2. For each search task, optimize the time allocation of the sequence of search scans of the task.
2.3. For each group of tracked targets, optimize the future measurements needed to maintain the tracks using open loop assumptions, possibly in combination with DP.
3. Sum $E\{\bar{l}_s^j\}$ and $\partial E\{\bar{l}_s^j\}/\partial\tilde{\lambda}$ over the subtasks to form $E\{\bar{l}^j\}$ and $\partial E\{\bar{l}^j\}/\partial\tilde{\lambda}$.
4. Update the Lagrange multiplier vector, $\tilde{\lambda}$ according to (25).
5. If track load is sufficiently close to one for all time intervals, proceed to 6, otherwise continue with 2.
6. To form the next measurement batch, extract the measurements of the first time interval from each subtask.

The algorithm is restarted when next measurement batch should be produced. Initial estimates in optimization are taken from the previous time instant. In online solutions, one should try to reduce the number of parameters and rely on tabulated results from offline optimization. Furthermore, it is not meaningful to spend computational resources on finding the exact optimum. The goal is primarily to coarsely assess the future resource situation.

H. Extension to Coordinate Search Scans and Adaptive Track Updates

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 [16].

The method, including the hierarchical ordering of searching and tracking tasks, has been implemented as a benchmarking reference algorithm. In the numerical example section, Section VI, the method is denoted LRM. Due to computational demand it is infeasible as an online solution. However, as an offline benchmarking method, it proves to be very useful.

VI. NUMERICAL EXAMPLE

In this section we illustrate the use of the LRM algorithm as a tool in radar design. The LRM is compared with both a TWS policy, which updates tracks while scanning for new targets, and an ad-hoc AT policy. AT uses the search scans only to cue AT, for which adaptive track updates are scheduled once every 6th second per track. 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. In this example, the three methods utilize a quick search scan which resolves range ambiguities from scan to scan. Targets may be eclipsed by the radar transmission intervals and thus remain undetected at one scan pass, but will likely be detected a following scan. The search scans cover 360 deg before new scans are started. This results in scan times of about 2.5 to 20 s depending on level of coherent integration. The TWS method herein utilizes a 6 s scan giving a fair balance between search and tracking performance in the example. AT utilizes search scans with increasing level of coherent integration to detect smaller targets. The search may proceed until the system becomes saturated by track load.

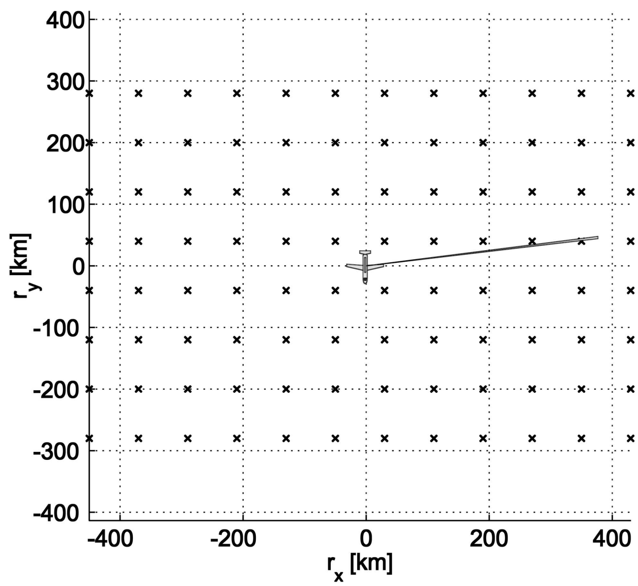


Fig. 15. Scenario of regularly distributed test targets. Density of targets is $\rho_0 = 1.5 \cdot 10^{-4}$ targets per km^2.

The scenario in the demonstration consists of nonmoving test targets placed regularly in space, see Fig. 15, where the target density is $\rho_0 = 1.5 \cdot 10^{-4}$ targets per km^2. No targets are tracked at $k = 0$. Each target is given a weight to achieve target densities $\rho \in \rho_0 \cdot \{1,3\}$ (corresponding to medium and high densities for the modelled radar).

Due to the open loop assumptions of search scan parameters 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.

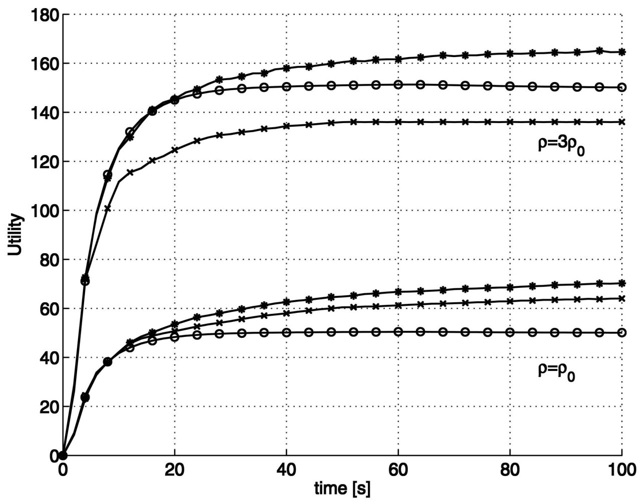


Fig. 16. Startup transients of utility for densities $\rho \in \rho_0 \cdot \{1,3\}$. Upper three curves corresponds to $\rho = 3\rho_0$, and lower three curves to $\rho = \rho_0$. LRM corresponds to curves with stars, TWS to circles, AT to crosses.

The sensor, performance prediction models, and utility function are similar to those used in previous examples with the following additions and exceptions:

$\Delta_t = 2$ s for reduced size of the state space, and $k - k_m$ is sampled up to 30 s.

$C_{\text{reinit}} = 30$, and $C_{\text{mix}} = 30$.

In order to include both the startup phase of the performance transient, and a period where the transient has levelled out, the time horizon is set to 100 s plus a burn in time of the Markov decision processes.

All targets have the same definition of utility function. The Markov chain model of Fig. 4 used in optimization of LRM is also used for evaluations of TWS and AT. The detection probabilities are then computed for each target individually, and for each time instant k conditioned on the following entities: target location in relation to the radar geometry, target radar cross section, sequence of search scans in the sector of the target u_S, scheduled target updates $d_{\mathbf{upd},i,k}$. By inserting the detection probabilities in the Markov chain transition matrix, the probability vector of the target performance state is simulated using (3). Further, the probability of correct association is computed with (5), based on an estimate of the background density of tracked targets. In homogenous density scenarios at the chosen density levels and update rates, track mixes are rare and have little impact on the overall utility.

Results: Fig. 16 shows comparisons of utility transients for LRM, TWS, and AT given the two target densities. LRM optimizes the area under the utility transient given the resource constraints.

At the medium density scenario, $\rho = \rho_0$, AT performs better than TWS. The reason is that while AT increases the coherent integration gain of the search scan to produce more tracks, TWS is forced to maintain a high scan rate to sustains tracking quality. Thus, AT starts more low-SNR targets. Additionally, the tracking quality of AT tracks is better than that of TWS tracks. AT updates each target every 6th second. With TWS on the other hand, tracks may remain undetected for one or a few scan passes due to eclipsing and a stretched out shape of the P_d-curve. LRM copies the behavior of AT at low and medium density scenarios. A difference is the adaptive control of the time between updates, illustrated in Fig. 10(d). This adaptivity gives a slight advantage over AT in that more resources are used for searching, and even more low-SNR targets can be started. The utility gain of these costly targets is not substantial.

At the the high density scenario, $\rho = 3\rho_0$, conditions are reversed. AT saturates from adaptive tracking. TWS maintains substantially more tracks than AT, although still at a lower tracking performance per track. LRM now combines TWS and AT behavior. Half the resource is utilized for TWS, while the rest is utilized in adaptive updates when TWS detection fails, and to fill in with updates in between the search scan passes.

Discussion: TWS distributes the energy equally in all directions, and is therefore likely to operate well for homogenous target densities and homogenous task definitions. In dense scenarios particularly, TWS achieves an efficient coordination of track updates. The performance difference between TWS and LRM in the dense scenario is therefore small. With varying target densities, inhomogeneous task definitions, and inhomogeneous sensor characteristics, the performance difference can be significant in favor of LRM. For instance, LRM has the ability to adapt the resource allocation policy sector-wise, where for each sector the three properties are reasonably homogenous. In a sector with high target density, LRM combines the benefits of TWS and AT, while in a low density sector, LRM acts more like AT. For tracks with high performance requirements, adaptive updates are favored. The essence of the LRM behavior is possible to capture in a heuristic policy.

VII. CONCLUSIONS

In this paper we formulated the radar resource management problem for adaptive airborne surveillance radars as a stochastic optimal control problem. We modelled performance of radar tracking target-wise as Markov decision processes. Since an optimal (stochastic DP) approach is computationally intractable, a novel suboptimal method based on hierarchical time decomposition and Lagrange relaxation is presented. The method was implemented as an offline tool for benchmarking other methods. In a numerical example, the method was compared with two adhoc resource allocation policies. Thereby, the power of the method as an offline reference in radar design has been demonstrated.

The techniques in this paper can be generalized to other resource allocation applications in which dynamic stochastic effects are important, and where similar simplifications apply. In future work it is interesting to study the method in a scenario with two adaptive airborne radars, or with one radar supported by ground sensors. Modelling of tracking performance using Markov chains can be developed further. Moreover, it is worthwhile studying efficient algorithms for solving the optimization problem, particularly for online applications. It is also worthwhile consider a game theoretic version of the above problem where multiple radars are allocated to multiple targets—see [11] for a similar problem.

REFERENCES

[1] Bar-Shalom, Y., and Li, X.
Estimation and Tracking, Principles Techniques and Software.
Norwood, MA: Artech House, 1993.

[2] Bertsekas, D.
Dynamic Programming and Optimal Control (2nd ed.).
Belmont, MA: Athena Scientific, 2000.

[3] Blackman, S., and Popoli, R.
Modern Tracking Systems.
Norwood, MA: Artech House, 1999.

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

[5] Howard, S., Suvorova, S., and Moran, B.
Optimal policy for scheduling of Gauss-Markov systems.
In *Proceedings of the 7th International Conference on Information Fusion*, 2004.

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

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

[8] Koch, W.
On adaptive parameter control for phased-array tracking.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, 1999, 3809.

[9] Krishnamurthy, V.
Decentralized emission management for low probability of intercept sensor platforms in network centric warfare.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 1 (Jan. 2005), 133–152.

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

[11] Maskery, M., Krishnamurthy, V., and O'Regan, C.
Decentralized algorithms for netcentric force protection against anti-ship missiles.
IEEE Transactions on Aerospace and Electronic Systems, submitted.

[12] Mori, S., Chang, K., and Chong, C.
Multitarget-Multisensor Tracking: Applications and Advances, Vol. II.
Norwood, MA: Artech House, 1992, ch. 7.

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

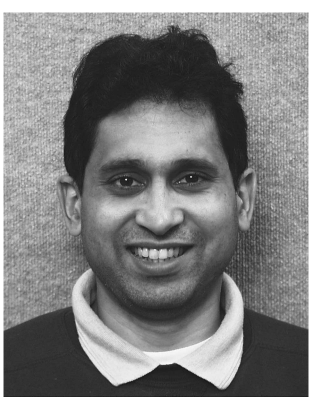
[14] Vannicola, V. C., and Mineo, J. A.
Expert system for sensor resource allocation.
In *Proceedings of the 33rd Midwest Symposium on Circuits and Systems*, 1990.

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

[16] Wintenby, J.
Resource allocation in airborne surveillance radar.
Ph.D. dissertation, Chalmers University of Technology, Sweden, 2003.

Johannes Wintenby was born 1972 in Kolmŕden, Sweden. He received the M.Sc. degree in applied physics and electrical engineering from Linköping University, Sweden, in 1997, and the Ph.D. in signal processing from Chalmers University of Technology, Gothenburg, in 2003.

During 2004 and 2005 he was employed at the Department of Signals and Systems at Chalmers University of Technology. Since 2000, he has also been associated with Ericsson Microwave System, Mölndal, Sweden, working with development of airborne radar systems. His research interests include control and signal processing aspects in radar and sensor systems, in particular sensor resource allocation, scheduling, data association, target tracking, and data fusion.



Vikram Krishnamurthy (S'90—M'91—SM'99—F'05) was born in 1966. He received his bachelor's degree in electrical engineering from the University of Auckland, New Zealand in 1988, and Ph.D. from the Australian National University, Canberra, in 1992.

Since 2002, he has been a professor and Canada research chair at the Department of Electrical Engineering, University of British Columbia, Vancouver, Canada. Prior to this he was a chaired professor at the Department of Electrical and Electronic Engineering, University of Melbourne, Australia. His research interests span several areas including stochastic scheduling and network optimization, biological nanotubes, statistical signal processing, and wireless telecommunications.

Dr. Krishnamurthy is currently an associate editor for *IEEE Transactions on Signal Processing*, *IEEE Transactions Aerospace and Electronic Systems*, and *Systems and Control Letters*. He is also guest editor of a special issue of *IEEE Transactions on NanoBioScience*, on biological nanotubes published in March 2005. He has served on the technical program committee of several conferences in signal processing, telecommunications, and control.