Performance Improvement of Time-Balance Radar Schedulers Through Decision Policies

ÖMER ÇAYIR
CAĞATAY CANDAN
Middle East Technical University, Ankara, Turkey

The resource management of a phase array system capable of multiple target tracking and surveillance is critical for the realization of its full potential. This paper aims to improve the performance of an existing method, time-balance (TB) scheduling, by establishing an analogy with a well-known stochastic control problem, the machine replacement problem. With the suggested policy, the scheduler can adapt to the operational scenario without a significant sacrifice from the practicality of the TB schedulers. More specifically, the numerical experiments indicate that the schedulers directed with the suggested policy can successfully trade the unnecessary track updates, say of nonmaneuvering targets, with the updates of targets with deteriorating tracks, say of rapidly maneuvering targets, yielding an overall improvement in the tracking performance.

Manuscript received November 14, 2016; revised July 30, 2017; released for publication November 23, 2017. Date of publication January 25, 2018; date of current version August 7, 2018.

DOI. No. 10.1109/TAES.2018.2798418

Refereeing of this contribution was handled by A. Charlish.

Authors' address: Ö. Çayır and Ç. Candan are with the Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara 06800, Turkey, E-mail: (ocayir@metu.edu.tr; ccandan@metu.edu.tr). *(Corresponding author: Ömer Çayır.)*

I. INTRODUCTION

A modern radar system is required to handle a variety of tasks, such as surveillance, multitarget tracking, calibration, guidance, etc. The capabilities of such a system, say a multifunction radar system, come at a significant initial deployment cost mainly due to the installment of possibly thousands of transmit/receive modules. Taking the full advantage of the mentioned capabilities requires an effective radar resource management (RRM). Typically, the multitude of tasks in execution compete for the radar resources, namely time, energy, and computation [1]–[3]. In this paper, we focus on the time allocation problem for such systems.

The allocation of time, among other resources, is generally called *scheduling* in the RRM applications. Scheduling methods can be classified into two classes, adaptive and nonadaptive methods [1]. Nonadaptive scheduling methods, namely heuristic schedulers, are based on a rule-based design. The behavior of schedulers and prioritization (priority assignment) of tasks are predefined by the fixed rules. In contrast, the adaptive scheduling methods dynamically determine the task prioritization and scheduling to optimize overall performance. According to the importance of tasks being scheduled, an efficient task prioritization process is required to rank tasks for the performance improvement of both adaptive and nonadaptive schedulers. Knowledge-based systems using some *a priori* information is suggested to this aim. Knowledge-based systems consist of two subsystems, a knowledge database containing information related to the system environment, and an inference engine making final decisions, taking into account both *a priori* information and existing conditions [4]. In [5], a fuzzy logic-based approach is suggested to rank targets and surveillance sectors for dynamically changing system environments. For tracking tasks, the priorities are assigned according to five different fuzzy variables, such as quality of tracking, hostility, and degree of threat. For surveillance tasks, there are four fuzzy variables including the original priority and number of threatening targets. In [6] and [7], a neural network-based approach is utilized for target ranking with respect to range, radial velocity, membership (friend or foe), acceleration, and object rank (important or not important). Both neural network and fuzzy logic-based approaches provide an adaptive priority assignment, in general. Especially, the learning capability of neural network-based schedulers enable the operator to update the system behavior after the detection of new targets. However, the learning process is far from trivial [2]. The process includes training of several dataset in random order from the same initial starting point [6].

In this paper, we aim to achieve the benefits of adaptive scheduling without a major sacrifice from the low computational load of nonadaptive scheduling methods. More explicitly, the adaptive scheduling schemes in the literature are based on the stochastic control and they are, in general, difficult to implement due to high computational requirements [8]–[10]. To reap the benefits of adaptive scheduling while maintaining a low computational load, we suggest an improvement over a well-known nonadaptive scheduling scheme, namely the time-balance (TB) scheduler, based on a classical stochastic control problem, namely the machine-replacement problem. The suggested improvement, in effect, yields to an automated task prioritization and shown to have good adaptation capabilities to target tracking scenario unfolding to the operator.

The TB method is based on the idea of meeting the "deadlines" of each task with the minimum possible delay. The TB method is robust and achieves the desired task occupancies in the long horizon. In Section II, the general features of TB schedulers are further described. A similar adaptation effort on the TB method is the adaptive task prioritization, as described in [11], with the aim of completing the surveillance task properly even when radar is overloaded with the task of tracking a large number of targets and without adjusting the task update times. With this method, the task prioritization adapts to a predictable task queue and is independent of the radar performance measurements and the track scores.

In this paper, we aim to develop a target selection procedure for the TB method in order to reduce the tracking error, when the radar system is overloaded with the tracking tasks having identical priorities. The proposed method is based on a well-known stochastic control problem known as the machine replacement problem, as given in [12]. Here, our goal is to construct an analogy between the well-known control problem and the target tracking problem, and enable the utilization of the results for this problem in the performance improvement of the TB schedulers. The suggested method and its variants are highly practical and can be immediately applied in the existing systems utilizing the conventional TB schedulers.

II. BACKGROUND: TB METHOD AND MACHINE REPLACEMENT PROBLEM

A. TB Method

The TB metric gives the degree of urgency of each task during the radar operation. A revisit time is assigned to each task and the TB metric is continually updated to reflect the approaching visitation deadline of each task [13], [14]. More specifically, each task is associated with a TB value, t_{TB}. A positive t_{TB} value indicates an overdue task. A negative t_{TB} value indicates a task whose immediate execution would be ahead of the assigned deadline. A zero t_{TB} value indicates a just-on-time task. At any time, a new task can be inserted to the list by assigning a negative t_{TB} value. If a task is scheduled, its t_{TB} is decreased by its task update time (revisit time). Upon execution of any task, the t_{TB} of other tasks which are not scheduled is increased by a fixed amount determined by the designer. Under light-load conditions, the TB method is highly efficient, enabling timely task updates. As the load increases, the TB method suffers a performance loss since this method does not have any capacity to discriminate tasks as urgent and not-so-urgent.

A scheduler algorithm that utilizes the TB method is employed in the multifunction electronically scanned adaptive Radar, [13], [15]. This method allows dividing tasks into subtasks (looks) that can be interleaved to manage radar time efficiently and decrease the delays for the highly prioritized tasks by starting from the highest priority level at each scheduling instant. In [16], the TB scheduler chooses the task which has higher t_{TB} than other tasks as the next task. The scheduler is designed to schedule mainly the tracking tasks and the surveillance task is fragmented by the task fragment time. That is, the surveillance task is not periodically started, but one of its fragments is scheduled whenever all tracking tasks have negative t_{TB} value, i.e., when there is some idle time between the tracking tasks.

The adaptive time-balance (ATB) scheduler is proposed in [11]. Here, the surveillance task is associated with a t_{TB} value so that it is scheduled with respect to task update time to detect new targets. The task update times can be adaptively changed to be a possible solution for the overload conditions or to increase the revisit improvement factor. The ATB scheduler supports user defined priority levels for each task and tasks are scheduled according to these priority levels and t_{TB}'s.

In this paper, we present a further improvement on the TB method. Our goal is to schedule the target tracking tasks according to the track quality. Hence, we would like to adjust the scheduling parameters of the TB method dynamically according to the unfolding tracking scenario.

The target selection problem emerges when there are more than one target requesting the track update. The conventional TB scheduler follows the steps below:

1) Select the targets with the highest priority level.
2) Look for the targets which have the highest t_{TB}.

Typically, if there are multiple overdue tracking tasks at the same priority level, the executed task is selected according to the first-come first-serve principle [17, Ch. 6]. This method aims to minimize the overall lateness in the task execution. Clearly, this is an efficient mechanism for the maneuvering targets requiring a rapid execution depending on the type of maneuver. Our goal is to include the information about track quality in the task selection. To this aim, we construct an analogy between the well-known machine replacement problem and the RRM problem.

B. Machine Replacement Problem

We describe the machine replacement problem with a concrete example. Assume a baker having the main asset of an oven (machine) which can be either in "good" or "bad" state related to its cooking performance. The state of the machine deteriorates due to aging and the products of the machine can be delicious (conforming) or tasteless (defective), depending on the state variable. The true state of the oven is not known, but can be observed by the quality of the products. It is possible to have a bad product in spite of the good state of the machine with a nonzero probability and vice versa. In this problem, it is assumed that the cost of a new machine (replacement cost) and the price for the good and bad products are fixed quantities.

The main question is to determine the time to replace the machine yielding the profit maximization. This type of problem is categorized as partially observable Markov decision process (POMDP). Here, the Markov process is related with an unknown state of the machine that can be only observed in the presence of noise [18].

We establish an analogy between the resource management problem and machine replacement problem as follows: For the target selection problem, a target can be in one of the two states, namely up-to-date and stale. The up-to-date state denotes that the target track is predictable with a high accuracy by the tracker and may not require an immediate track update. Hence, the up-to-date state can be considered the "good" state. The stale state denotes that the target track is not predictable with a good accuracy and this track may require the urgent attention of the scheduler due to its higher probability of target drop. Hence, it is the "bad" state. As in the POMDP problems, we have noisy information on the target states.

As discussed in the latter parts of this paper, we assign a state to each target and utilize the track quality information as the noisy measurement on the state. The proposed analogy is especially valuable for an overloaded scheduler; but, even for the underloaded case, it can yield some performance improvements. It should be noted that an unnecessary execution of a target update in the up-to-date state could decrease the tracking performance of other targets. Hence, the state-dependent track update selection can also be beneficial in the improvement of overall track quality.

III. PROPOSED MACHINE REPLACEMENT PROBLEM-BASED POLICY

We utilize several results, with some corrections, from the work of Ben-Zvi and Grosfeld-Nir [12]. Here, the binomial observation model for the machine replacement problem refers to the classification of the quality of products as *conforming units* or *defective units* according to observations (measurements), while the production process is in either "good" or "bad" state. The true state of the process is not observable and can only be estimated with some error. Thus, the production process is modeled as a POMDP with some control limits. The POMDPs are known to be usually hard to solve due to prohibitively large size of the state space [19]. In [12], it is proven that the infinite-horizon control limit defined as a function of the probability of obtaining a conforming unit can be calculated by solving a finite set of linear equations.

In the target selection problem, there are many targets and each target is, conceptually, associated with a machine. At each instant of decision-making, a target is selected among a set of overdue targets according to the observed track quality depreciations. To use the machine replacement problem, the cost of the machine renewal, i.e., the track update, should also be specified. Since there can be only one task scheduled at a time, the cost of executing a task should include the cost of not-executing other tasks.

TABLE I
Target Selection Versus Machine Replacement

Problem	Target Selection	Machine Replacement
# of machines	>1	1
States	Up-to-date, Stale	Good, Bad
Observations	good track, bad track	conforming unit, defective unit
Actions	update, not-update	replace, continue
Cost	target dependent	fixed

The state probabilities are obtained with the interacting multiple model (IMM), as described in [20, Ch. 11]. The mode probabilities of the IMM are associated with the state probabilities of the machine replacement problem. We assume that there are two motion models, both of which are constant velocity models having different process noise covariance matrices. The covariance matrix of two models are given as $\mathbf{Q}^2 = 100^2\mathbf{Q}^1$, where $\mathbf{Q}^k$ denotes the covariance matrix for the kth model. The case of higher process noise covariance matrix refers to the case of a target in the stale state. The other case refers to the up-to-date state. We can say that the probability of target being in the up-to-date state is taken as the mode-probability of the model-1.

In addition, the track is considered good; when the trace of the IMM mixed covariance matrix is within the allowed values. Otherwise, it is considered a bad track. Actions are update (UPD), similar to *replace the machine* action comes with a cost $\mathcal{K}$ that will be explained later, and not-update (NUPD). In Table I, the analogy between the target selection and machine replacement problem is summarized. In Section III-A, we provide further details on the analogy given in Table I.

A. Problem Model

It is supposed that there are N_k targets at time k. The target selection problem emerges when there are more than one target concurrently requesting the track update among N_k targets. Then, the scheduler should decide which one of these targets is in more need of a track update than the remaining ones by using the information on the target states. There are N_k distinct Markov chains corresponding to each target and the state transition probabilities of each target are assumed to be independent.

1) *Markov Chain Descriptions:* The target-n obeys a two-state Markov chain with the following description:

a) State is $x_k^n \in \{us, ss\}$, where us denotes the up-to-date state and ss denotes the stale state, with initial probability $P(x_0^n = i) = 0.5$ for $i \in \{us, ss\}$.
b) Observation is $y_k^n \in \{gt, bt\}$, where gt denotes a good track and bt denotes a bad track.
c) Action is $u_k^n \in$ {NUPD, UPD}, where NUPD denotes the *not-update* action and UPD denotes the *update* action,

at time k for $n = 1, 2, \ldots, N_k$.

The state of a target is probabilistically evolving, such that if the target is in up-to-date state at time k, it will remain

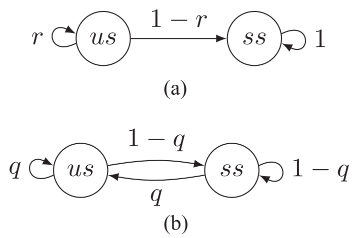


Fig. 1. Markov chains for (a) NUPD (not-update) and (b) UPD (update) actions.

in that state with probability r or it will change its state to the stale state with probability $1 - r$ at time $k + 1$. Once the target enters the stale state, it is assumed to remain in that state until the UPD action is taken, as shown in Fig. 1(a). However, the UPD action may fail. A stale target moves to the up-to-date state with probability q after taking the UPD action, as shown in Fig. 1(b). If the NUPD action is taken, the transition of target states becomes a time-homogeneous Markov chain with the up-to-date state, a transient state, and the stale state, an absorbing state.

The conditional observation probabilities can be expressed as follows:

$$P(y_k^n = gt | x_k^n = us) = \theta_0 \tag{1}$$
$$P(y_k^n = gt | x_k^n = ss) = \theta_1 \tag{2}$$
$$P(y_k^n = bt | x_k^n = us) = 1 - \theta_0 \tag{3}$$
$$P(y_k^n = bt | x_k^n = ss) = 1 - \theta_1. \tag{4}$$

The expressions given above present the probability of having an observation matching the actual state of the target. More specifically, θ_0 denotes the probability of the measurement matching the state of the target, i.e., a measurement indicating a good track quality given that the target is in the up-to-date state.

2) *Cost Function:* Different from the classical machine replacement problem, the cost of updating a specific track (machine renewal) affects the cost of other tracks since there can be only one track that can be updated at an instant and selecting a specific track for the update action leads to the track quality depreciation of other tracks. We propose to use the following cost function $\mathcal{K}_k^n$ taking into account the coupling of track quality scores for individual targets

$$\mathcal{K}_k^n \triangleq \frac{\max \left\{ m_k^{(0,\ell)} x_{\text{requesting},k}^{\ell} \right\}_{\ell=1,\ \ell \neq n}^{N_k}}{m_k^{(1,n)}} \cdot x_{\text{requesting},k}^{n} \tag{5}$$

where $x_{\text{requesting},k}^{n} \in \{0, 1\}$ indicates whether the target-n requests a track update or not. The value of $m_k^{(1,n)}$ denotes the estimated improvement on the IMM mixed covariance of the target-n by taking the UPD action $\left(u_k^n = \text{UPD}\right)$. We assume that once a target is updated, the diagonal elements of the IMM mixed covariance matrix would be reduced due to the measurement update process. The value of $\max \left\{ m_k^{(0,\ell)} \right\}_{\ell=1,\ \ell \neq n}^{N_k}$ denotes the maximum of estimated deterioration on the IMM mixed covariance of all targets that are not updated. This set covers all targets except the target-n. From (5), it can be noted that the cost of updating target-n is related with the cost of not-updating other targets.

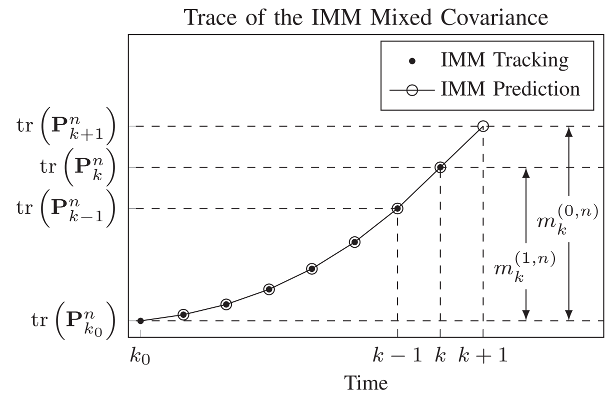


Fig. 2. Description of the parameters used to find the cost function value.

The improvement or deterioration metric, $m_k^{(1,n)}, m_k^{(0,n)}$, can be taken as the trace of the IMM mixed covariance matrix. If the trace decreases at time $k + 1$, an improvement on the track quality occurs; otherwise, the track quality deteriorates.

Fig. 2 is given to visualize the definitions for $m_k^{(0,n)}$ and $m_k^{(1,n)}$, which are written as

$$m_k^{(0,n)} = \text{tr}\left(\mathbf{P}_{k+1}^n\right) - \text{tr}\left(\mathbf{P}_{k_0}^n\right) \tag{6}$$
$$m_k^{(1,n)} = \text{tr}\left(\mathbf{P}_{k}^n\right) - \text{tr}\left(\mathbf{P}_{k_0}^n\right). \tag{7}$$

Here, $\mathbf{P}_{k_0}^n$ is the IMM mixed covariance matrix of the target-n at time k_0 when the target-n has the latest measured track. We assume that if the track is updated at next time $k + 1$, the trace of the IMM mixed covariance matrix would be close to the trace of $\mathbf{P}_{k_0}^n$. Hence, $m_k^{(1,n)}$ and $m_k^{(0,n)}$ depend on $\mathbf{P}_{k_0}^n$.

In Section III-B, we derive the expressions required for the solution of target selection problem with the definition presented in this section.

B. Derivation of Required Expressions

The probability of being in the up-to-date state is

$$\mu_k^n = P(x_k^n = us) \tag{8}$$

for the target-n at time k. Then, the probability of observing a good track is

$$\begin{aligned} \mathcal{P}_{gt}(\mu_k^n) &\triangleq P(y_k^n = gt) \\ &= \sum_{i \in \{us, ss\}} P(y_k^n = gt | x_k^n = i) P(x_k^n = i) \\ &= (\theta_0 - \theta_1)\mu_k^n + \theta_1 \end{aligned} \tag{9}$$

which is considered the function of μ_k^n since θ_0 and θ_1 are the global constants for the problem. Similarly, the probability of observing a bad track is

$$\mathcal{P}_{bt}(\mu_k^n) \triangleq P(y_k^n = bt) = \sum_{i\in\{us,ss\}} P(y_k^n = bt|x_k^n = i)P(x_k^n = i) = 1 - (\theta_0 - \theta_1)\mu_k^n - \theta_1. \tag{10}$$

By applying Bayes' theorem, the posterior probabilities for the up-to-date state are given as

$$P(x_k^n = us|y_k^n = gt) = \frac{P(y_k^n = gt|x_k^n = us)P(x_k^n = us)}{P(y_k^n = gt)} = \frac{\theta_0\mu_k^n}{\mathcal{P}_{gt}(\mu_k^n)} \tag{11}$$

$$P(x_k^n = us|y_k^n = bt) = \frac{P(y_k^n = bt|x_k^n = us)P(x_k^n = us)}{P(y_k^n = bt)} = \frac{(1-\theta_0)\mu_k^n}{\mathcal{P}_{bt}(\mu_k^n)}. \tag{12}$$

With Markov property, it is assumed that x_{k+1}^n is conditionally independent of y_k^n [21], and hence, the conditional probability of the next state is j given that gt is observed and the NUPD action is taken in the current state is i can be written as

$$P(x_{k+1}^n = j|x_k^n = i, y_k^n = gt, u_k^n = \text{NUPD}) = P(x_{k+1}^n = j|x_k^n = i, u_k^n = \text{NUPD}) = P_{ij}(u_k^n) \tag{13}$$

where $i, j \in \{us, ss\}$. By using these expressions and the law of total probability, the conditional probabilities of the next state are obtained. The probability of being in the up-to-date state at next time given that a good track is observed and the NUPD action is taken at current time is expressed as

$$\begin{aligned} P(x_{k+1}^n = us|y_k^n = gt, u_k^n = \text{NUPD}) &= \sum_{i\in\{us,ss\}} P(x_{k+1}^n = us, x_k^n = i| y_k^n = gt, u_k^n = \text{NUPD}) \\ &= \sum_{i\in\{us,ss\}} P(x_{k+1}^n = us|x_k^n = i, y_k^n = gt, u_k^n = \text{NUPD})P(x_k^n = i|y_k^n = gt, u_k^n = \text{NUPD}) \\ &= \sum_{i\in\{us,ss\}, j=us} P_{ij}(u_k^n)P(x_k^n = i|y_k^n = gt) \\ &= r\cdot P(x_k^n = us|y_k^n = gt) + 0\cdot P(x_k^n = ss|y_k^n = gt) \\ &= \frac{r\theta_0\mu_k^n}{\mathcal{P}_{gt}(\mu_k^n)} \end{aligned} \tag{14}$$

and the probability of being in the up-to-date state at next time given that a bad track is observed and the NUPD action is taken at current time is

$$\begin{aligned} P(x_{k+1}^n = us|y_k^n = bt, u_k^n = \text{NUPD}) &= \sum_{i\in\{us,ss\}} P(x_{k+1}^n = us, x_k^n = i| y_k^n = bt, u_k^n = \text{NUPD}) \\ &= \sum_{i\in\{us,ss\}} P(x_{k+1}^n = us|x_k^n = i, y_k^n = bt, u_k^n = \text{NUPD})P(x_k^n = i|y_k^n = bt, u_k^n = \text{NUPD}) \\ &= \sum_{i\in\{us,ss\}, j=us} P_{ij}(u_k^n)P(x_k^n = i|y_k^n = bt) \\ &= r\cdot P(x_k^n = us|y_k^n = bt) + 0\cdot P(x_k^n = ss|y_k^n = bt) \\ &= \frac{r(1-\theta_0)\mu_k^n}{\mathcal{P}_{bt}(\mu_k^n)}. \end{aligned} \tag{15}$$

Since r is a global constant, the conditional probabilities given in (14) and (15) are written as follows:

$$\mathcal{H}_{gt}(\mu_k^n) \triangleq P(x_{k+1}^n = us|y_k^n = gt, u_k^n = \text{NUPD}) = \frac{r\theta_0\mu_k^n}{(\theta_0-\theta_1)\mu_k^n + \theta_1} \tag{16}$$

$$\mathcal{H}_{bt}(\mu_k^n) \triangleq P(x_{k+1}^n = us|y_k^n = bt, u_k^n = \text{NUPD}) = \frac{r(1-\theta_0)\mu_k^n}{1-(\theta_0-\theta_1)\mu_k^n - \theta_1}. \tag{17}$$

According to [19, Lemma 1], both $\mathcal{H}_{gt}(\mu_k^n)$ and $\mathcal{H}_{bt}(\mu_k^n)$ are continuous and strictly increasing functions for $0 < \mu_k^n < 1$, while $\mathcal{H}_{gt}(\mu_k^n)$ is strictly concave and $\mathcal{H}_{bt}(\mu_k^n)$ is strictly convex. Moreover, the inverse functions $\mathcal{H}_{gt}^{-1}(\mu_k^n)$ of $\mathcal{H}_{gt}(\mu_k^n)$ and $\mathcal{H}_{bt}^{-1}(\mu_k^n)$ of $\mathcal{H}_{bt}(\mu_k^n)$ exist, and they are strictly increasing for $0 < \mu_k^n < r$, proofs can be found in [22, Appendix A]. The inverse function $\mathcal{H}_{bt}^{-1}(\mu_k^n)$ is

$$\mathcal{H}_{bt}^{-1}(\mu_k^n) = \frac{(1-\theta_1)\mu_k^n}{(\theta_0-\theta_1)\mu_k^n + (1-\theta_0)r} \tag{18}$$

for $0 < \mu_k^n < r$. The existence of inverse leads to the use of function composition, i.e., $\mathcal{H}_{bt}\left(\mathcal{H}_{bt}^{-1}(\mu_k^n)\right) = \mu_k^n$.

The functions $\mathcal{H}_{gt}(\mu_k^n)$ and $\mathcal{H}_{bt}(\mu_k^n)$ depend on the fixed parameters, r, θ_0, and θ_1 as well. The critical point for choosing the fixed parameters is to ensure the criteria, $r\theta_0 > \theta_1$, so that $\mathcal{H}_{gt}(\mu^*) = \mu^* > 0$ does exist. The value of μ^* is computed as

$$\mu^* = \frac{r\theta_0 - \theta_1}{\theta_0 - \theta_1}. \tag{19}$$

The point μ^* divides the domain of μ_k^n into two subdomains, $\mu^* < \mathcal{H}_{gt}(\mu_k^n) < \mu_k^n$ for $\mu_k^n > \mu^*$ and $\mathcal{H}_{gt}(\mu_k^n) > \mu_k^n$ for $0 < \mu_k^n < \mu^*$. If $r\theta_0 \leqslant \theta_1$, then $0 < \mathcal{H}_{gt}(\mu_k^n) < \mu_k^n$ for $0 < \mu_k^n \leqslant 1$, and hence, $\mathcal{H}_{gt}(\mu^*) = \mu^*$ does not exist, as shown in Fig. 3. In Fig. 3(a), parameters satisfy $r\theta_0 > \theta_1$ and $\mathcal{H}_{gt}(\mu^*) = \mu^*$ exists. If the observation on the state is gt, one knows that the probability of being in the up-to-date state will increase at the next time step, i.e., $\mu_{k+1}^n > \mu_k^n$, for $0 < \mu_k^n < \mu^*$, whereas it will decrease for $\mu^* < \mu_k^n \leqslant 1$,

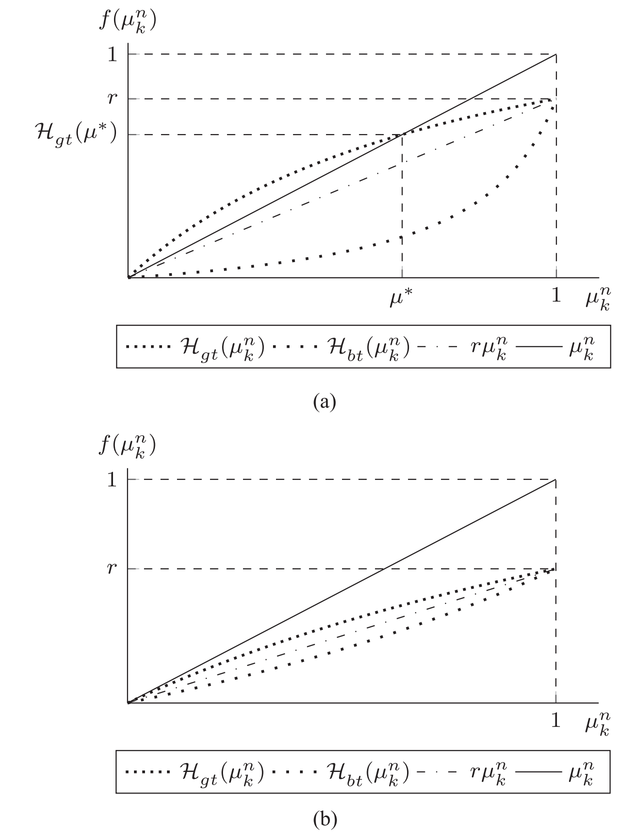


Fig. 3. Functions $\mathcal{H}_{gt}(\mu_k^n)$ and $\mathcal{H}_{bt}(\mu_k^n)$ for (a) $r\theta_0 > \theta_1$, where $r = 0.8$, $\theta_0 = 0.9$, $\theta_1 = 0.4$, and (b) $r\theta_0 < \theta_1$, where $r = 0.6$, $\theta_0 = 0.6$, $\theta_1 = 0.4$.

as shown in Fig. 3(a). If μ^* does not exist, as shown in Fig. 3(b), then the same probability always decreases, i.e., $\mu_{k+1}^n < \mu_k^n$, irrespective of the observation. Thus, the provided observations can be considered to be informative, if μ^* exists, i.e., when $r\theta_0 > \theta_1$. The existence of μ^* is not critical for the implementation of the scheme, but important for its conceptual understanding.

Next, we discuss the value function and the associated optimal policy for the infinite-horizon problem, where the probability calculations given in this section is required in the computations.

1) *Infinite-Horizon Value Functions:* The solution to the stochastic control problem described is studied only for the infinite-horizon case. There is no fixed period at which the problem restarts with a given set of initial parameters for the target tracking problem. For this problem, each target has a different motion characteristic, and hence, the target-based generalization of initial parameters is not practical. Furthermore, the infinite-horizon case solution has the advantage of requiring less computation. It only requires the computation of a fixed threshold for the asymptotic probability of being in the up-to-date state. Therefore, with a simple threshold policy, whenever the up-to-date state probability is less than the threshold, the optimal action becomes an update action.

The optimal action is determined via *the optimal value function* $V^n(\cdot)$ according to the probability of being in the up-to-date state, μ_k^n

$$V^n(\mu_k^n) = \max\left\{V_{\text{nupd}}^n(\mu_k^n),\, V_{\text{upd}}^n\right\} \tag{20}$$

where $V_{\text{nupd}}^n(\mu_k^n)$ and V_{upd}^n are the infinite-horizon value functions for NUPD and UPD actions, respectively, as follows:

$$\begin{aligned} V_{\text{nupd}}^n(\mu_k^n) &= \mu_k^n + \alpha \sum_{i\in\{gt,bt\}} \mathcal{P}_i(\mu_k^n) V^n\big(\mathcal{H}_i(\mu_k^n)\big) \\ &= \mu_k^n + \alpha\Big[\mathcal{P}_{gt}(\mu_k^n) V^n\big(\mathcal{H}_{gt}(\mu_k^n)\big) \\ &\quad + \mathcal{P}_{bt}(\mu_k^n) V^n\big(\mathcal{H}_{bt}(\mu_k^n)\big)\Big] \end{aligned} \tag{21}$$

$$V_{\text{upd}}^n = -\mathcal{K}_k^n + V_{\text{nupd}}^n(q) \tag{22}$$

where α is a discount factor satisfying $0 < \alpha < 1$.

The value functions (21) and (22) depend explicitly on each other via $V^n(\mu_k^n)$ given in (20). Furthermore $V_{\text{nupd}}^n(\mu_k^n)$ given in (21), is related to the optimal value function $V^n(\cdot)$ through the functions $\mathcal{H}_{gt}(\mu_k^n)$ and $\mathcal{H}_{bt}(\mu_k^n)$, while $V^n(\cdot)$, given in (20), is already related to $V_{\text{nupd}}^n(\cdot)$.

It is not possible to express $V_{\text{nupd}}^n(\mu_k^n)$ without any assumptions on $\mathcal{H}_{gt}(\mu_k^n)$ and $\mathcal{H}_{bt}(\mu_k^n)$. Choosing $\theta_1 = 0$, as stated in [12], is a practical choice which assumes that the probability of observing a good track given that the corresponding target is in the stale state is 0, and it also avoids always-decreasing probability of the up-to-date state by ensuring $r\theta_0 > \theta_1$, i.e., the existence of μ^*. Then, it is only possible that a good track is observed from a target in the up-to-date state. With this assumption, (21) can be simplified as

$$\begin{aligned} V_{\text{nupd}}^n(\mu_k^n) =\; &\mu_k^n + \alpha\big[\theta_0\mu_k^n V^n(r) \\ &+ (1-\theta_0\mu_k^n) V^n\big(\mathcal{H}_{bt}(\mu_k^n)\big)\big]. \end{aligned} \tag{23}$$

Next, an assumption is made about the parameter q, the probability of transition to the up-to-date state upon the update action, as shown Fig. 1(b). To make the model more realistic, perhaps pessimistic, we take $q = r$, $(r \neq 1)$, meaning that update action may fail (due to radar sensing environment). Then, (22) becomes

$$V_{\text{upd}}^n = -\mathcal{K}_k^n + V_{\text{nupd}}^n(r). \tag{24}$$

In order to evaluate the value of $V_{\text{nupd}}^n(r)$ from (23), we need

$$\begin{aligned} V^n(r) &= \max\left\{V_{\text{nupd}}^n(r),\, V_{\text{upd}}^n\right\} \\ &= \max\left\{\mathcal{K}_k^n + V_{\text{upd}}^n,\, V_{\text{upd}}^n\right\}. \end{aligned} \tag{25}$$

Since $\mathcal{K}_k^n$ given in (5) cannot be negative valued, the (25) becomes

$$V^n(r) = \mathcal{K}_k^n + V_{\text{upd}}^n. \tag{26}$$

By using (26), the simplified value function of NUPD action, (23) becomes

$$\begin{aligned} V_{\text{nupd}}^n(\mu_k^n) =\; &\mu_k^n + \alpha\Big[\theta_0\mu_k^n\left(\mathcal{K}_k^n + V_{\text{upd}}^n\right) \\ &+ (1-\theta_0\mu_k^n) V^n\big(\mathcal{H}_{bt}(\mu_k^n)\big)\Big]. \end{aligned} \tag{27}$$

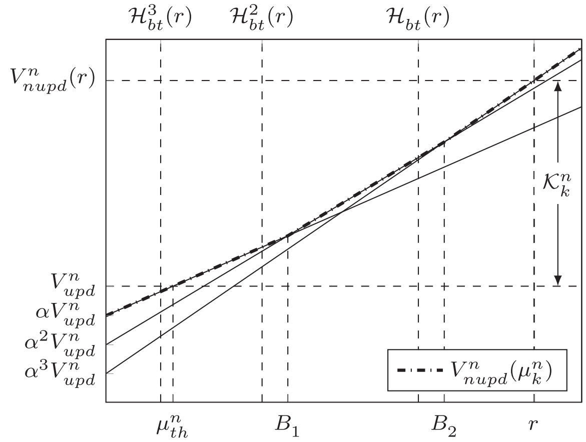


Fig. 4. Value function of NUPD action for $M = 3$.

2) *The Threshold Value for Decision-Making:* The threshold value for the target-n, which is denoted as μ^n_{th}, is the solution of $V^n_{\text{nupd}}(\mu^n_k) = V^n_{\text{upd}}$, where left-hand side and right-hand side are given in (27) and (24), respectively. Then, the decision-making process is given as

$$u^n_k = \begin{cases} \text{NUPD}, & \mu^n_k \geqslant \mu^n_{th} \\ \text{UPD}, & \text{otherwise} \end{cases} \tag{28}$$

such that the action is no-update on the track if the threshold is exceeded.

It is not straightforward to obtain (24) and (27) owing to the presence of $V^n_{\text{nupd}}(r)$ in (24). Fig. 4 is given to visualize the value function of NUPD action obtained from the basic parameters, α, θ_0, r, and $\mathcal{K}^n_k$. The function $V^n_{\text{nupd}}(\mu^n_k)$ is the pointwise maximum of linear functions cutting $\mu^n_k = 0$ axis at $\alpha^3 V^n_{\text{upd}}$, $\alpha^2 V^n_{\text{upd}}$, and αV^n_{upd} for $0 \leqslant \mu^n_k \leqslant 1$. Thus, it is a piecewise linear convex function, see [12, Th. 2]. The intersection points of these linear functions are the breakpoints of $V^n_{\text{nupd}}(\mu^n_k)$ such that $B_1 = \mathcal{H}^{-1}_{bt}(\mu^n_{th})$ and $B_2 = \mathcal{H}^{-1}_{bt}(B_1)$. The number of linear functions, namely linear segments of $V^n_{\text{nupd}}(\mu^n_k)$, M, is determined with μ^n_{th} satisfying $r > \mathcal{H}_{bt}(r) > \mathcal{H}^2_{bt}(r) > \cdots > \mathcal{H}^{M-2}_{bt}(r) > \mathcal{H}^{M-1}_{bt}(r) > \mu^n_{th} > \mathcal{H}^M_{bt}(r)$, see [12, Corollary 1]. To obtain M, the constraint $\mathcal{H}^M_{bt}(r) < \mu^n_{th}$ is utilized, where $\mathcal{H}^M_{bt}(r)$ is found by evaluating (17) recursively. For each M value starting from $M = 1$, μ^n_{th} is computed and compared with $\mathcal{H}^M_{bt}(r)$ until the constraint holds. The expression of μ^n_{th} depending on M and the basic parameters will be provided later.

The relation between μ^n_{th} and V^n_{upd} can be found from (27) at $\mu^n_k = \mu^n_{th}$ by replacing $V^n_{\text{nupd}}(\mu^n_{th})$ and $V^n\big(\mathcal{H}_{bt}(\mu^n_{th})\big)$ with V^n_{upd}, where the latter is the result of $\mathcal{H}_{bt}(\mu^n_{th}) < \mu^n_{th}$, as shown in Fig. 3(a). Then, μ^n_{th} can be computed as

$$\mu^n_{th} = \frac{1-\alpha}{1+\alpha\theta_0\mathcal{K}^n_k} V^n_{\text{upd}} \tag{29}$$

if V^n_{upd} is known, see [12, Prop. 4]. Thus, the value of V^n_{upd} is the most critical point of the calculations.

The expression of V^n_{upd} depending on M and the basic parameters is found by solving $M+1$ distinct equations, $V^n(r)$, $V^n\big(\mathcal{H}_{bt}(r)\big), \ldots, V^n\big(\mathcal{H}^{M-1}_{bt}(r)\big)$ from (23), owing to $V^n(\mu^n_k) = V^n_{\text{nupd}}(\mu^n_k)$ for $\mu^n_k > \mu^n_{th}$, and $V^n\big(\mathcal{H}^M_{bt}(r)\big) = V^n_{\text{upd}}$. Then, V^n_{upd} can be computed by

$$V^n_{\text{upd}} \triangleq \frac{\mathcal{A}_M(r)(1+\alpha\theta_0\mathcal{K}^n_k) - \mathcal{K}^n_k}{1-\alpha\theta_0\mathcal{A}_M(r) - \mathcal{B}_M(r)} \tag{30}$$

where $\mathcal{A}_M(r)$ is defined as

$$\mathcal{A}_M(r) \triangleq \begin{cases} r, & M = 1 \\ r + \sum_{i=1}^{M-1} \alpha^i \mathcal{H}^i_{bt}(r) & \\ \prod_{j=0}^{i-1}\big(1-\theta_0\mathcal{H}^j_{bt}(r)\big), & M \geqslant 2 \end{cases} \tag{31}$$

and $\mathcal{B}_M(r)$ is defined as

$$\mathcal{B}_M(r) \triangleq \prod_{i=0}^{M-1} \alpha\big(1-\theta_0\mathcal{H}^i_{bt}(r)\big) \tag{32}$$

for $M \geqslant 1$ by supposing that $V^n_{\text{nupd}}(\mu^n_k)$ consists of M segments, see [12, Th. 3 and 4]. Derivations of $\mathcal{A}_M(r)$ and $\mathcal{B}_M(r)$ can be found in [22, Ch. 4].

A careful examination of (30) reveals that V^n_{upd} can be negative valued depending on the $\mathcal{K}^n_k$ value. If $\mathcal{K}^n_k$ is high enough, then V^n_{upd} is negative valued, and hence, μ^n_{th} also becomes negative according to (29). Therefore, the NUPD action immediately becomes the optimal action for the negative threshold values since μ^n_k is always positive. This observation simplifies the computations for the decision-making process since whether μ^n_{th} is negative or not can be possibly inferred from $\mathcal{K}^n_k$. When

$$\mathcal{K}^n_k > r/(1-\alpha r)$$

a degenerate policy emerges and the optimal action is always NUPD [23, Prop. 2], see the proof of Proposition 3 [22, Ch. 4].

Then, the threshold value is determined by

$$\mu^n_{th} = \begin{cases} 0, & \mathcal{K}^n_k > \dfrac{r}{1-\alpha r} \\ \dfrac{1-\alpha}{1+\alpha\theta_0\mathcal{K}^n_k} V^n_{\text{upd}}, & \text{otherwise.} \end{cases} \tag{33}$$

The computation of threshold μ^n_{th} is explicitly described in Table II. Here, line 2 checks whether the parameters cause a degenerate policy or not. Lines 5 to 18 start with $M = 1$ and increase M until $\mathcal{H}^M_{bt}(r) < \mu^n_{th}$ is satisfied, meanwhile, μ^n_{th} is computed by evaluating (17), (31), (32), (30), and (33), respectively.

We present the outputs of the algorithm in Table II for some special cases in Table III. This table can also be used for debugging purposes. Here, we assume that $\alpha = 0.99$ and other basic parameters, θ_0, r and $\mathcal{K}^n_k$, are changed to obtain distinct infinite-horizon value functions. The algorithm outputs, namely the threshold values and the numbers of segments, are given in Table III.

Some comments on the data of Table III can be given as follows:

TABLE II
Algorithm for Computing the Threshold Value

```
function THRESHOLD(α, θ_0, r, K_k^n)
    if K_k^n > r/(1 − αr) then
        μ_th^n = 0
    else
        M = 1
        compute H_bt^M(r) from (17)
        compute A_M(r) from (31)
        compute B_M(r) from (32)
        compute V_upd^n from (30)
        compute μ_th^n from (33)
        while H_bt^M(r) ⩾ μ_th^n do
            M ++
            compute H_bt^M(r) from (17)
            compute A_M(r) from (31)
            compute B_M(r) from (32)
            compute V_upd^n from (30)
            compute μ_th^n from (33)
        end while
    end if
    return μ_th^n
end
```

TABLE III
Comparison of the Threshold Value and Number of Segments for Different $\mathcal{K}_k^n$ Values With $\alpha = 0.99$

		$r = 0.90$			$r = 0.95$		
$\mathcal{K}_k^n$	θ_0	0.75	0.80	0.90	0.75	0.80	0.90
0.1	μ_{th}^n	0.8069	0.8073	0.8082	0.8569	0.8573	0.8582
	M	1	1	1	1	1	1
1.0	μ_{th}^n	0.3984	0.3933	0.3799	0.4667	0.4611	0.4464
	M	2	2	2	2	2	2
2.5	μ_{th}^n	0.1809	0.1755	0.1730	0.2503	0.2429	0.2315
	M	3	3	2	3	3	2

1) The higher r makes μ_{th}^n higher since $\mathcal{A}_M(r)$ increases with r, and V_{nupd}^n also increases. This statement can be deduced from (30).
2) The higher $\mathcal{K}_k^n$ makes μ_{th}^n smaller. That is

$$\mathcal{K}_k^n < r/(1-\alpha r) \wedge \mathcal{K}_k^n \to r/(1-\alpha r) \Rightarrow \mu_{th}^n \to 0.$$

3) M depends on both θ_0 and $\mathcal{K}_k^n$, as illustrated more clearly for $\mathcal{K}_k^n = 2.5$ in Table III.

Up to now, we have discussed how to decide whether a *single target* requires an update action or not based on its estimated track accuracy and calculated threshold given by the suggested algorithm. For the multiple target tracking scenarios, there can be a situation that several targets requiring an update at the same time, i.e., the up-to-date state probability falls below the corresponding threshold level. For such cases, we need to develop a policy for the selection of the most suitable target.

C. Problem Solution: Target Selection Policies

We present three policies for the selection of the track to be updated. The first policy is based on the machine replacement problem and it utilizes the threshold policy for the selection of track. The other policies use IMM outputs, but not the threshold value; hence, they are simpler to implement.

1) *Decision Policy (DecP):* This policy selects the track to be updated among the tracks satisfying the condition $\mu_k^n < \mu_{th}^n$ for $n = \{1, 2, \ldots, N_k\}$. It should be remembered that this condition is analogous to the decision of machine replacement.

The DecP selects the target-i according to

$$i = \underset{n \in \{1,2,\ldots,N_k\}}{\arg\min} \left\{\mu_k^n - \mu_{th}^n\right\} \quad \text{subject to } x_{\text{requesting},k}^n = 1, \quad \mu_k^n < \mu_{th}^n. \tag{34}$$

Ideally, the condition $\mu_k^n < \mu_{th}^n$ should be satisfied only if the track is sufficiently degraded. If none of the tracks is not sufficiently degraded, there is no track update. Hence, under the ideal conditions, the radar resources are not wasted by updating the tracks solely by the lateness value.

It should be remembered that this method has multiple criterion to be satisfied to grant a track update. First, by the requirement of the TB method, t_{TB} value should be nonnegative; hence, the lateness parameter should be either zero or a positive number. Second, the track should be sufficiently degraded, which is a condition checked by $\mu_k^n < \mu_{th}^n$. Third, among all tracks satisfying first two conditions, the track with the highest gap to threshold is selected.

2) *Tracking Error Minimization Policy (MinTE):* This is a greedy policy implementing the update of the track based on the IMM mixed covariance matrices of the targets. The aim is to select the target among the set of targets with nonnegative t_{TB} value and the worst case tracking error. To select the target-i, this method takes into account the mixed covariance matrix but not the mode-probabilities of IMM

$$i = \underset{n \in \{1,2,\ldots,N_k\}}{\arg\max} \left\{\operatorname{tr}\left(\mathbf{P}_k^n\right)\right\} \quad \text{subject to } x_{\text{requesting},k}^n = 1. \tag{35}$$

It should be noted that the target with the highest trace of IMM mixed covariance matrix might not necessarily correspond to a rapidly maneuvering target. This policy does not exert any effort in detecting the maneuvering action of the target.

3) *Pursuing the Most Maneuvering Target Policy (PurMM):* It should be remembered that the probability of up-to-date state is given as μ_k^n. Then, the updated target can be chosen according to the product of the trace of IMM mixed covariance matrix and the stale state probability, $1 - \mu_k^n$

$$i = \underset{n \in \{1,2,\ldots,N_k\}}{\arg\max} \left\{\left(1 - \mu_k^n\right) \operatorname{tr}\left(\mathbf{P}_k^n\right)\right\} \quad \text{subject to } x_{\text{requesting},k}^n = 1. \tag{36}$$

This method aims to give higher priority to the targets with having a high probability of maneuvering, namely probability of the mode with high process noise. Hence, this method is called as the method of PurMM.

IV. NUMERICAL COMPARISONS

The assumed instrumented range for the simulator is 200 km. Within this detection range, the assigned priority changes from 1 to 5 according to the detected target range with a range step of 40 km. For example, the target at the range of 60 km is assigned to the priority level 4, which is the second highest priority level. The measurement noise is $\mathcal{N}(0, \sigma_r^2)$ and $\mathcal{N}(0, \sigma_a^2)$ for range and azimuth, respectively, where $\sigma_r = 80$ m and $\sigma_a = 3$ mrd. Further details on radar simulator can be found in [22]. Due to the nature of conventional TB scheduler, the target selection policies are applied on the targets with the same priority. Hence, the tracking error of targets at only similar ranges are compared by these policies.

A. Case 1: Single Target Tracking Case

To illustrate the improvement brought by the DecP, we compare the performance of TB method utilizing the DecP with the conventional TB method. The conventional TB method does not utilize the track information provided by IMM. Hence, its performance is expected to be inferior to the one utilizing the DecP. Our goal is to contrast the difference between the two.

Fig. 5(a) shows the performance of conventional method on nonmaneuvering [left panel of Fig. 5(a)] and maneuvering (right panel) targets. The tracking performance of the TB method with DecP is given in Fig. 5(b). In this figure, the confidence ellipses are drawn to illustrate the tracking performance. A visual comparison of top and bottom panels of Fig. 5 immediately reveals that the DecP yields a better performance by refraining from, or postponing, unnecessary track updates.

More specifically, for the nonmaneuvering target, the average tracking error, namely the average of the trace of IMM mixed covariance matrices, decreases from 1.93×10^5 m^2, given in the top part of Fig. 6(a), to 1.59×10^5 m^2, given in the top part of Fig. 6(b). For the maneuvering target, the average tracking error decreases from 2.63×10^5 m^2 to 2.39×10^5 m^2. Furthermore, the maximum value of the tracking error is also smaller with the DecP, which is a criterion that can be especially important for rapidly maneuvering targets.

However, it is important to remind that the DecP does not guarantee a better operation at every run, but can present significant improvements in the scenarios where the beginning of target maneuvering can be effectively sensed with the IMM mode-probabilities.

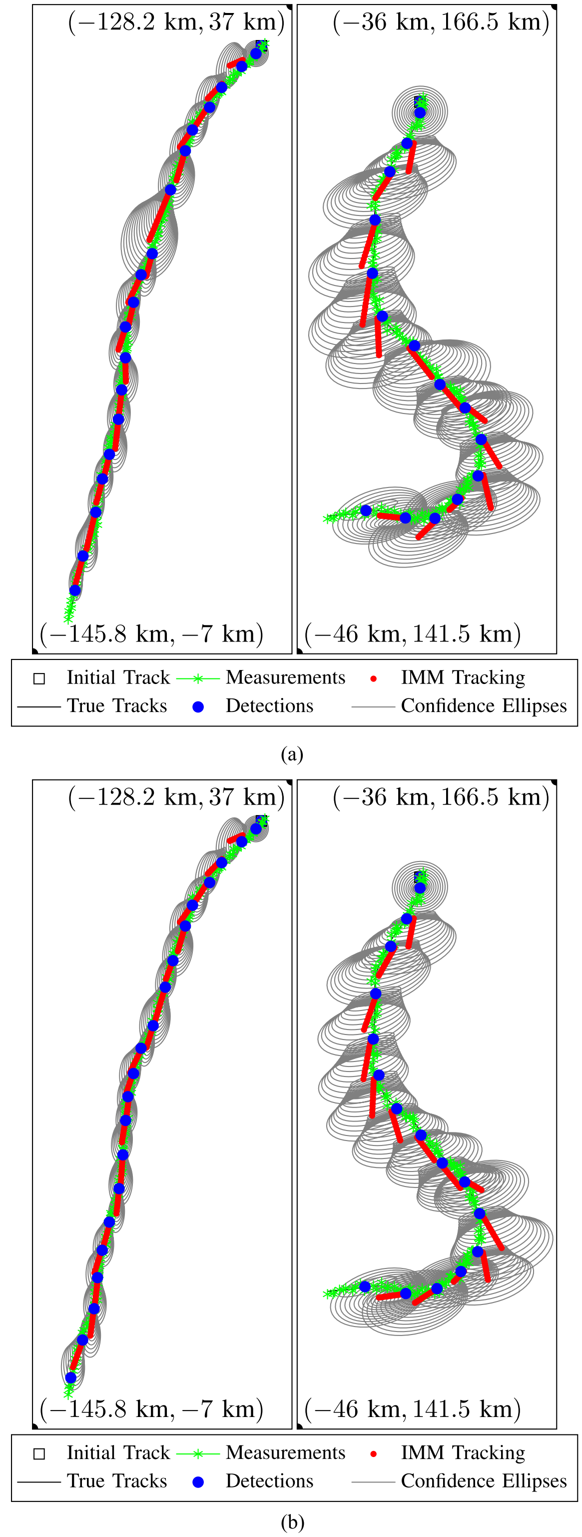


Fig. 5. Tracking of nonmaneuvering (left) and maneuvering (right) targets by using (a) the conventional TB method and (b) the TB method with DecP.

B. Case 2: Multiple Target Tracking Case

The proposed methods are evaluated for the scenario of multiple targets in addition to the surveillance tasks. The target tracks are randomly generated for each scenario of 200 s. Each target has randomly chosen transition probability matrix out of five matrices, while the IMM tracker makes use of fixed transition probabilities for all tracks. There are 100 distinct scenarios for each comparison case. The comparisons are made on the average of the scheduler performance that is measured with the following criteria:

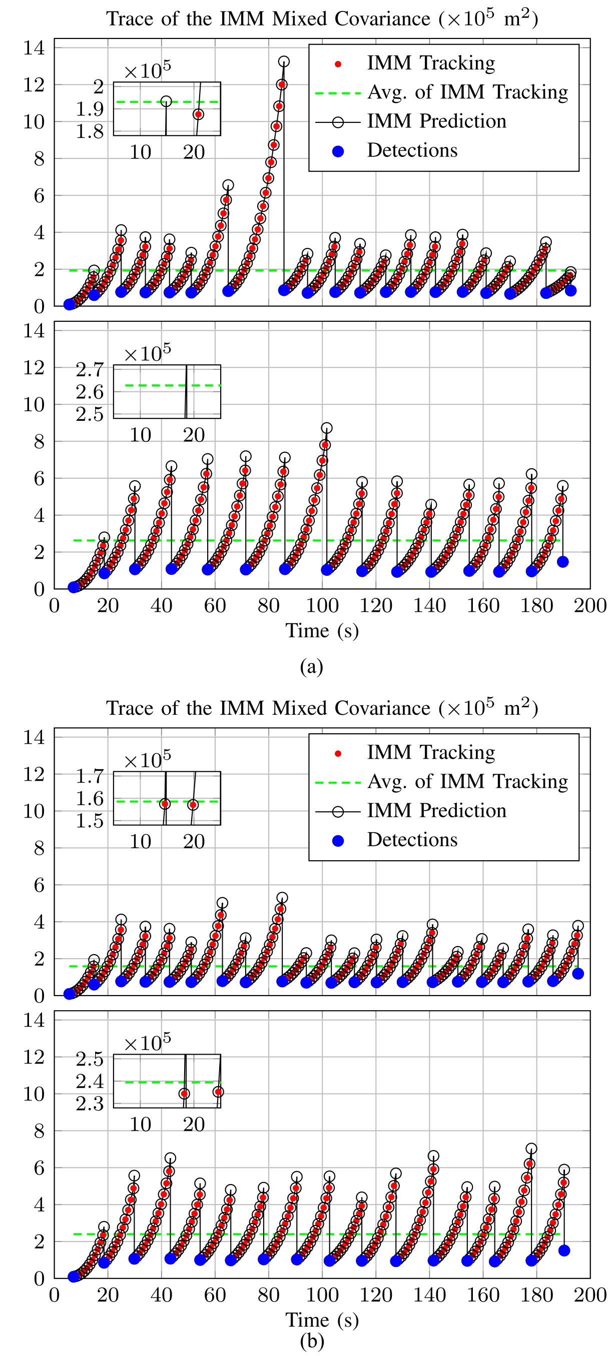


Fig. 6. Trace of IMM mixed covariance matrices for nonmaneuvering (top) and maneuvering (bottom) targets by using (a) the conventional TB and (b) the TB method with DecP.

1) *The number of probable drops* is the number of updates that are too late for target tracking. The probable drop occurs when the update interval exceeds the sum of task update time and allowable lateness.
2) *Cost* is the sum of weighted lateness values squares after each scheduling epochs. The priority values are assigned as the weights.
3) *Average of errors* is the average of the trace of IMM mixed covariance matrices of all targets.
4) *Occupancy* is the ratio of utilized radar time to the total available time interval.

TABLE IV
Comparison of the Decision Policies for 15 Targets

(a) The number of frequency bands is 2.				
	Average of statistics after 100 simulations			
	Conv.	DecP	MinTE	PurMM
# of tracking tasks	288.91	**292.11**	289.73	289.20
# of surveillances	**17.52**	17.43	17.37	**17.52**
# of prob. drops	23.76	**23.04**	23.12	23.08
Occupancy (%)	48.43	**48.69**	48.37	48.47
Cost (s^2)	$\mathbf{9.13 \times 10^4}$	1.01×10^5	9.49×10^4	1.11×10^5
Avg. of errors (m^2)	4.70×10^5	$\mathbf{3.69 \times 10^5}$	3.97×10^5	4.64×10^5
	Distributions of standings in avg. of errors			
Best	20	**38**	19	23
Runner-up	24	24	**33**	19
Honorable Mention	25	21	**30**	24
Last	31	17	18	**34**
(b) The number of frequency bands is 7.				
	Average of statistics after 100 simulations			
	Conv.	DecP	MinTE	PurMM
# of tracking tasks	502.54	**504.67**	504.22	503.18
# of surveillances	**19.13**	19.10	18.98	19.09
# of prob. drops	8.38	**8.22**	8.39	8.30
Occupancy (%)	74.17	**74.32**	74.18	74.15
Cost (s^2)	9.49×10^2	8.13×10^2	$\mathbf{7.28 \times 10^2}$	9.58×10^2
Avg. of errors (m^2)	8.95×10^4	$\mathbf{8.77 \times 10^4}$	8.82×10^4	8.89×10^4
	Distributions of standings in avg. of errors			
Best	20	26	**32**	22
Runner-up	24	**30**	17	29
Honorable Mention	16	**35**	26	23
Last	**40**	9	25	26

In Tables IV and V, the suggested methods are compared for the scenarios of 15 and 25 in-track-targets, respectively. Our main goal is to compare the performance statistics resulting from the application of the conventional TB method and TB method augmented with the suggested policies.

In order to illustrate the effect of the loading condition more explicitly, we assume that the radar system can utilize multiple-frequency bands concurrently. An increase in the number of frequency bands reduces the load seen from the resource management side [22]. The number of frequency bands is selected as 2 in Tables IV(a) and V(b), and the number of bands is selected as 7 in Tables IV(b) and V(b).

When the proposed methods, (DecP, MinTE, and PurMM) are compared with the conventional TB method, it can be seen that the DecP is the method which has the smallest number of probable drops and the smallest average of errors for all cases. Yet, the alternatives to the DecP (MinTE and PurMM) are almost equally good for this case.

Tables also illustrate the performance comparison of the methods in a competitive sense. From the bottom part of the tables, it can be noted the conventional method has most frequently provided the poorest tracking error performance

TABLE V
Comparison of the Decision Policies for 25 Targets

(a) The number of frequency bands is 2.

	Average of statistics after 100 simulations			
	Conv.	DecP	MinTE	PurMM
# of tracking tasks	294.93	296.39	296.66	**298.07**
# of surveillances	24.55	24.55	24.40	**24.63**
# of prob. drops	38.40	**37.37**	37.98	37.94
Occupancy (%)	55.70	55.84	55.72	**56.11**
Cost (s^2)	5.21×10^5	5.22×10^5	$\mathbf{4.79 \times 10^5}$	5.28×10^5
Avg. of errors (m^2)	8.73×10^5	$\mathbf{7.44 \times 10^5}$	8.60×10^5	8.14×10^5
	Distributions of standings in avg. of errors			
Best	20	28	18	**34**
Runner-up	26	**31**	21	22
Honorable Mention	25	24	**32**	19
Last	**29**	17	**29**	25

(b) The number of frequency bands is 7.

	Average of statistics after 100 simulations			
	Conv.	DecP	MinTE	PurMM
# of tracking tasks	538.33	**538.74**	537.18	537.53
# of surveillances	25.86	25.94	26.02	**26.10**
# of prob. drops	39.48	**39.33**	40.48	40.10
Occupancy (%)	83.76	83.88	83.79	**83.92**
Cost (s^2)	$\mathbf{1.68 \times 10^5}$	1.89×10^5	1.80×10^5	1.89×10^5
Avg. of errors (m^2)	3.35×10^5	$\mathbf{3.28 \times 10^5}$	3.49×10^5	3.58×10^5
	Distributions of standings in avg. of errors			
Best	22	**35**	21	22
Runner-up	**35**	25	21	19
Honorable Mention	21	28	16	**35**
Last	22	12	**42**	24

TABLE VI
Overall Distributions of Standings in Average of Errors
After 400 Simulations

	Conv.	DecP	MinTE	PurMM
Best	82	**127**	90	101
Runner-up	109	**110**	92	89
Honorable Mention	87	**108**	104	101
Last	**122**	55	114	109

for the duration of complete scenarios, while the DecP has the smallest number of bad performances.

In Table VI, the distributions of standings given in Tables IV and V are combined for a clearer comparison. Table VI indicates that the DecP is the most frequently successful policy among the four. It can be said that the DecP successfully traded the unnecessary track updates of targets having accurately predictable tracks, e.g., nonmaneuvering targets, with the track quality depreciating targets, e.g., maneuvering targets. This conclusion can be further justified by examining the average of errors criteria in the tables where the DecP is the best policy in all cases. Hence, as in the single target case, the DecP, in essence, manages to “detect” the beginning of a maneuver successfully and does not grant unnecessary updates to a track in spite of its potentially large lateness value. Interested readers may examine [22] for more comparisons.

TABLE VII
Comparison With Task Prioritization Methods for 15 Targets

(a) The number of frequency bands is 2

	Average of statistics after 100 simulations			
	Conv.	DecP	Neural N.	Fuzzy L.
# of tracking tasks	288.91	**292.11**	287.21	283.26
# of surveillances	**17.52**	17.43	16.70	16.61
# of prob. drops	23.76	23.04	21.75	**21.34**
Occupancy (%)	48.43	**48.69**	47.46	47.16
Cost (s^2)	$\mathbf{9.13 \times 10^4}$	1.01×10^5	1.29×10^5	1.56×10^5
Avg. of errors (m^2)	4.70×10^5	$\mathbf{3.69 \times 10^5}$	5.13×10^5	6.94×10^5
	Distributions of standings in avg. of errors			
Best	21	**49**	24	6
Runner-up	**37**	25	22	16
Honorable Mention	22	19	**35**	24
Last	20	7	19	**54**

(b) The number of frequency bands is 7

	Average of statistics after 100 simulations			
	Conv.	DecP	Neural N.	Fuzzy L.
# of tracking tasks	502.54	504.67	**509.66**	509.08
# of surveillances	**19.13**	19.10	18.56	18.48
# of prob. drops	8.38	8.22	**7.45**	7.58
Occupancy (%)	74.17	74.32	**74.38**	74.22
Cost (s^2)	9.49×10^2	8.13×10^2	$\mathbf{7.90 \times 10^2}$	4.95×10^3
Avg. of errors (m^2)	8.95×10^4	$\mathbf{8.77 \times 10^4}$	8.88×10^4	9.49×10^4
	Distributions of standings in avg. of errors			
Best	15	**36**	32	17
Runner-up	**31**	28	23	18
Honorable Mention	**32**	27	21	20
Last	22	9	24	**45**

C. Comparisons With Task Prioritization Methods

We compare the conventional TB and DecP with two other task prioritization methods based on neural network [6], [7], and fuzzy logic [5]. The detailed descriptions on these methods, such as the choice of training set for the neural network based scheme and the membership functions for the fuzzy logic based scheme, can be found in the extended version [24] of this paper. Similar to the decision policies, these methods incorporate the tracking error into decision-making. In addition, they use some other inputs such as the radial velocity and the allowable lateness for the task prioritization. Unlike the decision policies, which are applied only when the conventional TB requires selecting one of targets having the same priority level, the task prioritization methods are continually applied.

In Tables VII and VIII, the DecP is compared with the task prioritization methods for 15 and 25 targets, respectively. From the viewpoint of minimum average tracking error, the DecP policy remains as the best choice. On the

TABLE VIII
Comparison With Task Prioritization Methods for 25 Targets

(a) The number of frequency bands is 2

	Average of statistics after 100 simulations			
	Conv.	DecP	Neural N.	Fuzzy L.
# of tracking tasks	294.93	**296.39**	290.72	276.92
# of surveillances	**24.55**	**24.55**	23.84	23.62
# of prob. drops	38.40	37.37	**35.03**	35.99
Occupancy (%)	55.70	**55.84**	54.58	53.16
Cost (s^2)	$\mathbf{5.21 \times 10^5}$	5.22×10^5	5.76×10^5	7.10×10^5
Avg. of errors (m^2)	8.73×10^5	$\mathbf{7.44 \times 10^5}$	9.23×10^5	1.36×10^6
	Distributions of standings in avg. of errors			
Best	24	**38**	30	8
Runner-up	**39**	36	16	9
Honorable Mention	25	14	**35**	26
Last	12	12	19	**57**

(b) The number of frequency bands is 7

	Average of statistics after 100 simulations			
	Conv.	DecP	Neural N.	Fuzzy L.
# of tracking tasks	538.33	**538.74**	538.31	519.42
# of surveillances	25.86	**25.94**	25.27	25.31
# of prob. drops	39.48	39.33	**35.29**	36.96
Occupancy (%)	83.76	**83.88**	83.24	81.53
Cost (s^2)	$\mathbf{1.68 \times 10^5}$	1.89×10^5	2.45×10^5	3.70×10^5
Avg. of errors (m^2)	3.35×10^5	$\mathbf{3.28 \times 10^5}$	4.55×10^5	6.15×10^5
	Distributions of standings in avg. of errors			
Best	35	**48**	16	1
Runner-up	**46**	36	15	3
Honorable Mention	13	10	**53**	24
Last	6	6	16	**72**

other hand, the task prioritization methods provides the minimum number of probable drops due to the inclusion of the allowable lateness parameter in the scheduler design.

V. CONCLUSION

In this paper, we adapt the solution methods for the well-known machine replacement problem to the RRM problem. We propose practical performance improvement policies for the TB method. The conventional TB method does not have the capacity to adapt to the unfolding target tracking scenario. To provide some adaptation capability, we present a decision policy, DecP, and two other alternatives.

The results show that DecP-based TB method yields better tracking performance by trading the unnecessary updates of targets having accurately predictable tracks with the targets suffering from track quality degradations, say maneuvering targets. This is achieved, in effect, with the early detection of the track quality degradations via the utilization of information provided by IMM filter in the decision-making. In the numerical comparisons, it has been noted that the suggested DecP-based TB method is the method with the fewest worst case tracking performance. Furthermore, the suggested policy does not only improve the average tracking performance, but can also reduce the target drops.

The suggested policy is also compared with the knowledge-based task prioritization methods based on neural networks and fuzzy logic. The neural network-based scheme shows a competitive performance due to the efficient training process, while the fuzzy logic shows a rather poor performance and requires more computational time due to large number of rules. Thus, the capabilities of knowledge-based methods are limited by training process or inference rules. The suggested decision policy is rather simple and does present a good track quality improvement according to several performance metrics.

ACKNOWLEDGMENT

The authors would like to thank Prof. U. Orguner for his kind support, suggestions, and insightful comments.

REFERENCES

[1] Z. Ding
A survey of radar resource management algorithms
In *Proc. Can. Conf. Elect. Comput. Eng.*, May 2008, pp. 1559–1564.

[2] P. Moo and Z. Ding
Adaptive Radar Resource Management. New York, NY, USA: Academic, 2015.

[3] G. van Keuk and S. S. Blackman
On phased-array radar tracking and parameter control
IEEE Trans. Aerosp. Electron. Syst., vol. 29, no. 1, pp. 186–194, Jan. 1993.

[4] F. Gini and M. Rangaswamy
Knowledge-Based Radar Detection, Tracking, and Classification. Hoboken, NJ, USA: Wiley, 2008.

[5] S. L. C. Miranda, C. J. Baker, K. Woodbridge, and H. D. Griffiths
Fuzzy logic approach for prioritisation of radar tasks and sectors of surveillance in multifunction radar
IET Radar, Sonar Navig., vol. 1, no. 2, pp. 131–141, Apr. 2007.

[6] W. Komorniczak and J. Pietrasiñski
Selected problems of MFR resources management
In *Proc. 3rd Int. Conf. Inf. Fusion*, Jul. 2000, vol. 2, pp. WEC1/3–WEC1/8.

[7] W. Komorniczak, T. Kuczerski, and J. F. Pietrasinski
The priority assignment for detected targets in multifunction radar
J. Telecommun. Inf. Technol., no. 4, pp. 30–32, 2001. [Online]. Available: http://www.nit.eu/archive?Itemid=71&option=com_czasopisma&view=wynikiszukania&jezyk=en&nr_mies=&rok=&autor=komorniczak&tytul=&opis=&slowakl=&limit=50

[8] J. Wintenby and V. Krishnamurthy
Hierarchical resource management in adaptive airborne surveillance radars
IEEE Trans. Aerosp. Electron. Syst., vol. 42, no. 2, pp. 401–420, Apr. 2006.

[9] V. Krishnamurthy and R. J. Evans
Hidden Markov model multiarm bandits: A methodology for beam scheduling in multitarget tracking
IEEE Trans. Signal Process., vol. 49, no. 12, pp. 2893–2908, Dec. 2001.

[10] V. Krishnamurthy and R. J. Evans
Correction to 'Hidden Markov model multiarm bandits: A methodology for beam scheduling in multitarget tracking'
IEEE Trans. Signal Process., vol. 51, no. 6, pp. 1662–1663, Jun. 2003.

[11] R. Reinoso-Rondinel, T.-Y. Yu, and S. Torres
Task prioritization on phased-array radar scheduler for adaptive weather sensing
Proc. 26th Int. Conf. Interactive Inf. Process. Syst. Meteorol., Oceanography, Hydrol., Atlanta, GA, USA, 2010, Paper 14B.6.

[12] T. Ben-Zvi and A. Grosfeld-Nir
Partially observed Markov decision processes with binomial observations
Oper. Res. Lett., vol. 41, no. 2, pp. 201–206, 2013.

[13] W. K. Stafford
Real time control of a multifunction electronically scanned adaptive radar, (MESAR)
In *Proc. IEE Colloquium Real-Time Manage. Adaptive Radar Syst.*, Jun. 1990, pp. 7/1–7/5.

[14] M. Wray
Software architecture for real time control of the radar beam within MESAR
In *Proc. Int. Conf. Radar*, Oct. 1992, pp. 38–41.

[15] J. M. Butler
Tracking and control in multi-function radar
Ph.D. dissertation, Dept. Electron. Elect. Eng., Univ. College London, London, U.K., 1998.

[16] R. Reinoso-Rondinel, T.-Y. Yu, and S. Torres
Multifunction phased-array radar: Time balance scheduler for adaptive weather sensing
J. Atmos. Ocean. Technol., vol. 27, pp. 1854–1867, 2010.

[17] A. Silberschatz, P. B. Galvin, and G. Greg
Operating System Concepts. Hoboken, NJ, USA: Wiley, 2013.

[18] V. Krishnamurthy
Partially Observed Markov Decision Processes. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[19] S. Anily and A. Grosfeld-Nir
An optimal lot-sizing and offline inspection policy in the case of nonrigid demand
Oper. Res., vol. 54, no. 2, pp. 311–323, 2006.

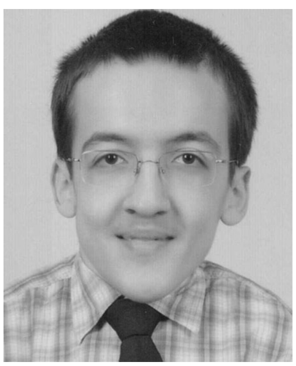
[20] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2001.

[21] P. R. Kumar and P. Varaiya
Stochastic Systems: Estimation, Identification, and Adaptive Control. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.

[22] Ö. Çayır
Radar resource management techniques for multi-function phased array radars
M.S. thesis, , Dept. Elect. Electron. Eng., Middle East Technical University, Ankara, Turkey, Sep. 2014. [Online]. Available: http://etd.lib.metu.edu.tr/upload/12617869/index.pdf

[23] M. Givon and A. Grosfeld-Nir
Using partially observed Markov processes to select optimal termination time of TV shows
Omega, vol. 36, no. 3, pp. 477–485, 2008.

[24] Ö. Çayır and Ç. Candan
Performance improvement of time-balance radar schedulers through decision policies (extended version)
arXiv:1712.00590 [eess.SP], Dec. 2017. [Online]. Available: https://arxiv.org/abs/1712.00590



Ömer Çayır received the B.S. degree (with rank 1) from Hacettepe University, Ankara, Turkey, in 2011, and the M.S. degree from Middle East Technical University, Ankara, in 2014, both in electrical and electronic engineering, and is currently working toward the Ph.D. degree with the Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara.

He is currently a Research Assistant with the Department of Electrical and Electronics Engineering, Middle East Technical University. His research interests include statistical signal processing and its applications in waveform optimization, stochastic control, radar resource management, and software defined radio.

Çağatay Candan received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1996, the M.S. degree from Bilkent University, Ankara, in 1998, and the Ph.D. degrees from Georgia Institute of Technology, Atlanta, GA, USA, in 2004, respectively, all in electrical engineering.

He is currently a Professor with the Department of Electrical and Electronics Engineering, Middle East Technical University. His research interests include statistical signal processing and its applications in array signal processing, radar signal processing and communications.