Chapter 1

Stochastic Control for Cognitive Radar

Alexander Charlish[1], Folker Hoffmann[2]
Kristine Bell[3] and Chris Kreucher[4]

Cognitive radar problems involve the selection of actions based on the uncertain knowledge of a system state that is partially observed through noisy measurements. This process of sequential decision making under uncertainty can be considered as a stochastic optimization problem. This chapter explicitly makes the connection between cognitive radar and stochastic optimization by presenting a framework for describing cognitive radar problems in terms of stochastic optimization, thereby pointing to ways to employ stochastic optimization for designing perception-action cycles in a cognitive radar.

1.1 Introduction

Cognitive radar problems require the selection of actions based on an uncertain perception that is obtained through inexact measurements. There is a broad variety of cognitive radar problems that differ in terms of the relevant perception and the types of actions selected, for example, waveform selection and optimization, measurement scheduling, resource management, detection, tracking and imaging [1]. A single radar may in fact comprise several individual perception-actions cycles, spread over multiple information abstraction levels [2]. Despite their differences, the variety of cognitive radar problems can be described in terms of a set of similar problem components. Consequently, after identifying the problem components, similar methodologies can be applied for designing perception-action cycles for a cognitive radar.

Cognitive radar problems can be classed as types of stochastic optimization problems. Stochastic optimization is a broad term for techniques that perform decision making under uncertainty, which are currently widely deployed in a range of applications including finance, business, logistics and transportation, and science and engineering. Stochastic optimization methods seek a policy that exploits models to map from a perception, which represents all the available information at the current

[1]Fraunhofer FKIE
[2]Fraunhofer FKIE
[3]Metron, Inc.
[4]Centauri

time, into an optimized action. As this policy is essentially a perception-action cycle, the design of perception-action cycles for cognitive radar can benefit from applying algorithmic strategies for finding policies from the stochastic optimization field.

There are many communities focusing on stochastic optimization problems, who have established a wide variety of algorithmic solutions. These stochastic optimization communities have conducted research covering techniques and applications such as decision trees, stochastic search, optimal stopping, optimal control, (partially observable) Markov decision processes (MDPs/POMDPs), approximate dynamic programming, reinforcement learning, model predictive control, stochastic programming, ranking and selection, and multiarmed bandit problems. It has been shown [3] that these problems can be described in a single stochastic optimization framework, and the respective solution methodologies can be grouped into just four classes.

Some of the work in cognitive radar explicitly refers to these stochastic optimization techniques. For example, multiarmed bandits [4], model predictive control [5], and reinforcement learning [6]. However, for many techniques developed in cognitive radar, the connection is less clear. The primary contribution of this chapter is to directly connect the cognitive radar problem with the large body of work done in stochastic optimization. This connection makes the methodologies developed in the stochastic optimization communities directly applicable to the cognitive radar problem, promising to lead to improved methods for designing perception-action cycles in cognitive radar. This chapter extends our previous work in [7].

1.2 Connection to Earlier Work

Current approaches to cognitive radar build on and can be traced back to earlier work which was referred to as sensor management [8]. These earlier efforts, while often applied to radar sensing, were ostensibly agnostic to the sensing modality and as such addressed the broad problem of determining the best way to task a sensor or group of sensors when each sensor may have multiple agilities. This section briefly reviews early work in sensor management to give context and connection to the current state of cognitive radar research.

Sensor management research frequently focused on the use case of tasking sensors to deduce the kinematic state (e.g., position and velocity) and identification of a group of targets as well as the number of targets. Applications of sensor management were often military in nature [9], but also included things such as wireless networking [10] and robot path planning [11].

Like cognitive radar, one of the main issues sensor management research addressed is the many competing objectives an automated decision maker may be tuned to meet, e.g., minimization of track loss, probability of new target detection, minimization of track error/covariance, and identification accuracy. Each of these different objectives taken alone may lead to a different sensor allocation strategy [9, 12]. Sensor management work was interested in mechanisms for capturing the trade-off between these competing objectives to deliver a measurement strategy that effectively addresses all of the objectives.

Information measures, including entropy reduction, Kullback-Leibler divergence (KLD) and mutual information, were a popular way of capturing the utility of sensing actions in foundational sensor management work and as such was explored by a number of researchers. Hintz [13, 14] did early work using the expected change in Shannon entropy when tracking a single target moving in one dimension with Kalman Filters. A related approach used discrimination gain based on a measure of relative entropy, the KLD. Schmaedeke and Kastella [15] used the KLD to determine sensor-to-target taskings. Kastella [16, 17] used KLD to manage a sensor between tracking and identification mode in the multitarget scenario. Mahler [18] used the KLD as a metric for "optimal" multisensor multitarget sensor allocation. Zhao [19] compared several approaches, including simple heuristics and information-based techniques based on entropy and relative entropy.

For multi-stage planning, sensor management was often formulated as a Partially Observable Markov Decision Process (POMDP) [20,21] and researchers worked to develop approximate solution techniques. For example, Krishnamurthy [22, 23] used a multi-arm bandit formulation involving hidden Markov models. In [22], an optimal algorithm was formulated to track multiple targets with an electronically scanned array that has a single steerable beam. Since the optimal approach has prohibitive computational complexity, several suboptimal approximate methods are given and some simple numerical examples involving a small number of targets moving among a small number of discrete states are presented. In [23], the problem was reversed, and a single target was observed by a single sensor from a collection of sensors. Again, approximate methods were formulated due to the intractability of the globally optimal solution.

Bertsekas and Castañon [24] did early work where they formulated heuristics for the solution of a stochastic scheduling problem corresponding to sensor scheduling. They implemented a rollout algorithm based on heuristics to approximate the solution of the stochastic dynamic programming algorithm. Additionally, Castañon [25, 26] investigated the problem of classifying a large number of stationary objects with a multi-mode sensor based on a combination of stochastic dynamic programming and optimization techniques. In [27] Malhotra proposed using reinforcement learning as an approximate approach to dynamic programming.

Chhetri [28] approached the long-term scheduling problem for a single target using particle filters and the unscented transform. The method involves drawing samples from the predicted future distribution and minimizing expected future costs. This requires enumeration of the exponentially growing number of possible sensing actions, a very computationally demanding procedure. This is combined with branch and bound techniques which require some restrictive assumptions on additivity of costs. In a series of works, Zhao [10, 19, 29] investigated sensor management in the setting of a wireless ad hoc network, which involved long term considerations such as power management.

With those connections as background, we now turn our attention to laying out a general framework for describing cognitive radar problems which makes them amenable to modern solution approaches.

1.3 Stochastic Optimization Framework

This section presents a framework, inspired by [3] [1], which enables cognitive radar problems to be described in terms of stochastic optimization problems.

1.3.1 General Problem Components

As described in [3], all the problems addressed by the stochastic optimization communities comprise the problem components described in this subsection. The next subsection shows how these components can be extended for the case when a system state is partially observed through noisy measurements. As the partially observable case is more relevant to cognitive radar problems, it is used as the focus for the remainder of this chapter.

System State - We are interested in the state of a dynamic system, which can be modelled as a random vector X_k for decision step k. A realisation of the random vector at decision step k is denoted $\mathbf{x}_k \in \mathscr{X}$ where $\mathscr{X}$ is the system state space.

Actions and Action Space - We can select an action or action vector at each decision step k, which influences the transition of the system state between time step k and $k+1$. An instantiation of an action for decision step k is denoted $\mathbf{a}_k \in \mathscr{A}$ where $\mathscr{A}$ is the action space.

Exogenous Information - Additional information is revealed at each sequential decision step and can, along with previously revealed information, be used as the basis for the action selection at the current decision step. The information revealed at each time step is modelled as a random vector Z_k and a realisation of this random vector is denoted $\mathbf{z}_k$. For completely observable problems the exogenous information is the system state.

State Transition Function - Between decision steps the system state evolves according to a transition function $\mathbf{x}_{k+1} = f_X(\mathbf{x}_k, \mathbf{a}_k, \mathbf{w}_k)$, where $\mathbf{w}_k$ is a realisation of the state transition noise (alternatively termed process noise). Due to the state transition noise, the transition can be described probabilistically by the transition probability density $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{a}_k)$.

Reward Function - At each decision step a reward is encountered, which is described by the function $r_x(\mathbf{x}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$. Cost can be handled as a negative reward, and therefore cost and reward are used interchangeably in this chapter. This objective function is described in more detail in the following sections.

These common components allow the breadth of stochastic optimization problems considered by the stochastic optimization communities to be described. Note that although the system state is completely observable, decision making under uncertainty is present due to the stochastic state transitions. A perception-action cycle using these components is illustrated in Figure 1.1.

[1] Although we adopt the framework in [3], we use the terminology and notation that is established in the signal processing community.

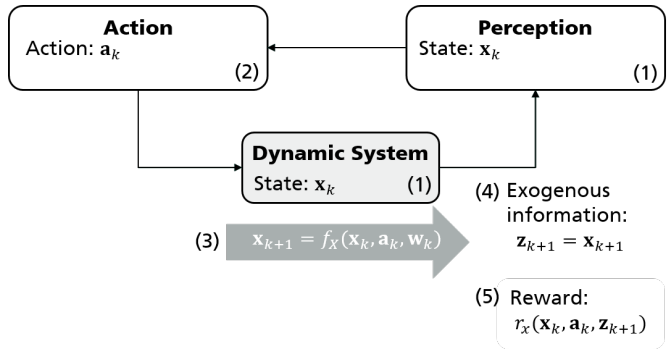


Figure 1.1 General perception-action cycle using stochastic optimization components. The following repetitive steps occur: (1) The system has a state $\mathbf{x}_k$, which is completely observed as the perception, (2) an action $\mathbf{a}_k$ is selected, (3) the system state transitions to $\mathbf{x}_{k+1}$, (4) the system state $\mathbf{x}_{k+1}$ is revealed as exogenous information, (5) a reward is generated.

1.3.2 Partial Observability

A common aspect of cognitive radar problems is that the system state is only *partially observable* through noisy measurements. Therefore, uncertainty is not only present due to stochastic state transitions but also through stochastic measurements. Consequently, we extend and adapt the components described Section 1.3.1 to the more specific partially observable case, which results in a framework closely resembling a POMDP.

Measurements and Measurement Space - The exogenous information described in Section 1.3.1 can now be thought of as a noisy measurement of the system state. Now, the random vector Z_k can be defined more exactly as a measurement with realisation $\mathbf{z}_k \in \mathscr{Z}$ where $\mathscr{Z}$ is the measurement space.

Measurement Likelihood Function - Measurements are related to the system state through the measurement function $\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{a}_{k-1}, \mathbf{v}_k)$ where $\mathbf{v}_k$ is a realisation of the measurement noise. Due to the measurement noise, the measurement process can be described by the measurement likelihood function $\mathscr{L}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{a}_{k-1}) \equiv p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{a}_{k-1})$.

Information State - As the state of the system is not observable, it is necessary to decide on an action based on the information state. The information state is the set of actions and measurements that have occurred prior to the current decision step. The information state for decision step k is denoted $\mathscr{I}_k = (\mathbf{a}_0, \mathbf{z}_1, ..., \mathbf{a}_{k-1}, \mathbf{z}_k)$. This information state grows with each time step, i.e., $\mathscr{I}_k = \mathscr{I}_{k-1} \cup (\mathbf{a}_{k-1}, \mathbf{z}_k)$.

Belief State - As the cardinality of the information state grows with each time step, it is generally undesirable to be used as the perception upon which actions are decided. Instead, decisions can be based on a belief state. The belief state is a set of parameters with fixed cardinality that are an (ideally sufficient) statistic of the

information state. The belief state at decision step k is modelled as a random vector B_k and a realisation of a belief state at decision step k is denoted $\mathbf{b}_k$. For example, under linear Gaussian assumptions a sufficient statistic of the information state is the mean and covariance of the posterior PDF, i.e. $p(\mathbf{x}_k|\mathscr{I}_k) \equiv p(\mathbf{x}_k|\mathbf{b}_k)$. Typical belief states are parameters of a Gaussian, a Gaussian sum, or a set of particles. Although this belief state represents imprecise knowledge on the underlying system state, it is itself completely observable. Consequently, by treating this belief state as the system state in Section 1.3.1, a partially observable problem can be handled like a completely observable problem.

Belief State Transition Function - It is necessary to define a transition function for belief states, analog to the system state transition function. This transition function is denoted $\mathbf{b}_{k+1} = f_B(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$. As the belief state can be thought of as parameters of the posterior PDF $p(\mathbf{x}_k|\mathbf{b}_k)$, the transition function represents the standard Bayesian prediction and update steps. As a cognitive radar is an observer, it is often the case that the system state transition is not influenced by the selected sensing action. However, the belief state transition certainly will be influenced by the selected action.

Reward Function - A reward function is now defined as a function of the belief state, i.e. $r(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$. This differs from the reward function described in Section 1.3.1, which was a function of the system state. The reward function maps to the reward that is associated with the measurement realisation $\mathbf{z}_{k+1}$ when the belief state was $\mathbf{b}_k$ and action $\mathbf{a}_k$ was taken. The next subsection describes specific forms of this reward function.

A perception-action cycle for the case of a partially observable system state is illustrated in Figure 1.2. In this figure, $\mathbf{a}_k = A^{\pi}(\mathbf{b}_k)$ is the policy function that maps from belief states to actions. This policy function is described in detail in Section 1.5.

For the remainder of this chapter we will assume a partially observable problem. However, a completely observable problem can be recovered by substituting the belief state with the observable system state, considering the likelihood function as a Dirac delta function, and using the state transition function instead of the belief state transition function.

1.4 Objective Functions for Cognitive Radar

The exact form of the reward function $r(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$ is crucial, as it must accurately represent the physical problem to be solved. Specifying reward functions for cognitive radar can be loosely categorized into task, information, or utility (quality-of-service) based approaches. However, the separation between the categories is not always distinct and existing approaches form more of a continuum.

1.4.1 Task Based Reward Functions

Task based reward functions calculate the cost or reward of an action in terms of a measure that is specific to the task being performed. Relevant task based metrics

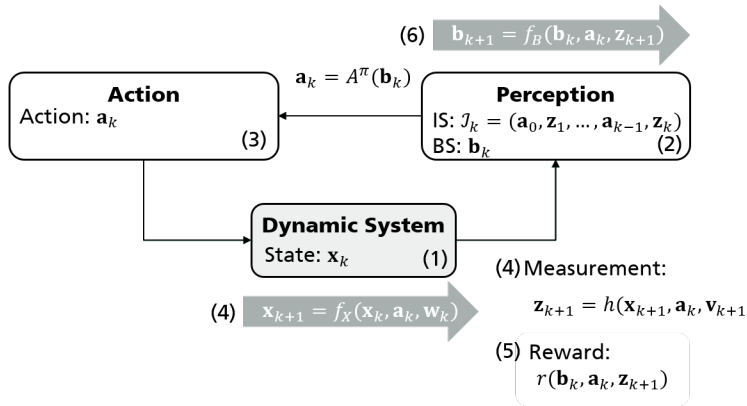


Figure 1.2 Partially observable perception-action cycle using stochastic optimization components. The following iterative steps occur: (1) The system has a state $\mathbf{x}_k$, (2) the perception of the system state is summarized in a belief state $\mathbf{b}_k$ (3) an action $\mathbf{x}_k$ is selected according to the policy function, (4) the system state transitions to $\mathbf{x}_{k+1}$ and a measurement $\mathbf{z}_{k+1}$ is generated, (5) a reward is produced, (6) the belief state transitions to $\mathbf{b}_{k+1}$.

include radar timeline or spectrum usage, probability of target detection, detection range for an undetected target density, tracking root mean square error (RMSE), track sharpness, track purity, track continuity, and probability of correct target classification, to name a few.

Each task-based reward function can be regarded as some function $q(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$ that is combined in some way to produce a scalar function that maps into the quality space $\mathscr{Q}$. It is often the case that a desired task-based metric is difficult to calculate and is replaced by a surrogate metric such as signal-to-interference plus noise ratio (SINR) or an information theoretic metric.

1.4.2 Information Theoretic Reward Functions

A second class of reward functions used in cognitive radar and related fields is based on information theory. Broadly speaking, an information theoretic function gauges the relative merit of a sensing action in terms of the information flow it provides. While this does not correspond directly to an operational criteria like track hold probability, information flow does capture actions that ultimately lead to good operational performance. A primary motivation for information-based reward functions is the ability to compare actions which generate different types of knowledge (e.g., knowledge about a target class versus knowledge about target position) using a common measuring stick.

A review of the history of information metrics in this context is provided in [8]. Here, we highlight some of the most commonly used reward functions. The most basic information theoretic cost function is the Posterior Shannon Entropy, given as:

$$\mathscr{H}(X_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1}) = \int p(\mathbf{x}_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1}) \ln p(\mathbf{x}_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1}) d\mathbf{x}_{k+1} \tag{1.1}$$

Note that $p(\mathbf{x}_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1}) \equiv p(\mathbf{x}_{k+1}|\mathbf{b}_{k+1})$ in the case that the belief state is a sufficient statistic of the information state.

A related approach computes the information *gain* between densities rather than just the information contained in the posterior. The most popular approach uses the KLD, which is defined using the prior and posterior densities as:

$$\mathscr{D}\Big(p(\mathbf{x}_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1})||p(\mathbf{x}_{k+1}|\mathbf{b}_k)\Big) = \int p(\mathbf{x}_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1}) \ln \frac{p(\mathbf{x}_{k+1}|\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1})}{p(\mathbf{x}_{k+1}|\mathbf{b}_k)} d\mathbf{x}_{k+1} \tag{1.2}$$

The KLD has several desirable properties [30], including its connection to Mutual Information. There are a number of generalizations of the KLD in the literature, including the Rényi Divergence [31], the Arimoto α-divergences, and the $f-$divergence [32].

A third approach specific to parameter estimation is the Fisher Information Matrix (FIM) and related Bayesian Information Matrix (BIM) [33], which characterize the amount of information that a distribution contains about individual parameters (such as target position or velocity). The inverse of the FIM is the Cramér-Rao Lower Bound (CRLB) and the inverse of the BIM is the Bayesian CRLB, which quantifies the uncertainty in the parameter estimates. The (square root of) the Bayesian CRLB has the property that it is in the units of the parameter being estimated and is a lower bound on the RMSE. Thus it is often used as a surrogate for the RMSE and categorized as a task-based metric.

The Bayesian CRLB approach is actually closely related to the KLD approach, since the BIM is related to a more general version of the KLD [34], and there is an equivalent Bayesian α-CRLB that is derived from the Bayesian version of the Renyi divergence [35] . Thus, these approaches have at their core the same information theoretic quantities, and the distinction is in the separation and weighting of individual tasks in the task-based Bayesian CRLB method versus a global approach in the information-based KLD method. A comparison between the approaches for fully adaptive radar resource allocation is explored in Chapter 10.

1.4.3 Utility and QoS Based Objective Functions

Quality-of-service approaches [2, 36] differ from task or information-based reward functions in that they optimize the user or operator satisfaction that is derived from a

task. A utility function is defined on the task quality space $\hat{u}: \mathscr{Q} \mapsto [0,1]$ that should accurately describe the satisfaction that is derived from the different possible task quality levels. Combining the quality and utility functions results in a function of the required form $u(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1}) \equiv (\hat{u} \circ q)(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$, where $(\hat{u} \circ q)$ is the composite function of $\hat{u}$ following q.

Using utility functions allows a user to specify requirements on task qualities, which are generally tangible to the user. This is very valuable in the context of radar resource management [2] as it enables a radar with limited resources to optimize multiple tasks based on the task quality levels that are required by the mission. Mapping the quality levels of differing radar tasks into the common utility space enables trade-offs between tasks evaluated using differing quality metrics. The global utility across the multiple tasks is typically formed by taking a weighted sum of task utilities. When considering the resource usage, a resource function $g(\mathbf{b}_k, \mathbf{a}_k)$ can be used as a constraint on the permissible actions. This quality-of-service conceptual approach can also be identified in other work [37, 38].

1.5 Multi-Step Objective Function

A general objective is to find a policy that determines a feasible action based on the belief state. The policy is a mapping from belief state to action denoted $\mathbf{a}_k = A^{\pi}(\mathbf{b}_k)$, where π carries information about the type of function and its parameters. As the belief state is a set of parameters describing a perception of the system state, the policy can be thought of as the perception-action cycle for a cognitive radar. The policy is not necessarily an analytical function and may actually represent an optimization problem. This section describes how a multi-step objective function is used to define optimal values and policies that are the basis for the design of perception-action cycles in the following section.

1.5.1 Optimal Values and Policies

The objective of a stochastic optimization problem is to maximize rewards or minimize costs over a time horizon comprising H future decision steps. The expected reward achievable over the current and future decision steps that originate from the current belief state is termed the value of the belief state. Let $V_H^{\pi}(\mathbf{b}_k)$ denote the value of a belief state when following policy A^{π}. It is defined as the expected value of the summed rewards with respect to the set of future measurements $(Z_{k+1}, ..., Z_{k+H})$, conditioned on the belief state $\mathbf{b}_k$:

$$V_H^{\pi}(\mathbf{b}_k) = \mathbb{E}\left[\sum_{t=k}^{k+H} r(B_t^{\pi}, A^{\pi}(B_t^{\pi}), Z_{t+1}) \,|B_k^{\pi} = \mathbf{b}_k\right] \tag{1.3}$$

where the belief state random variables in the summation evolve according to the belief state transition function when following policy π, i.e. $B_{k+1}^{\pi} = f_B(B_k^{\pi}, A^{\pi}(B_k^{\pi}), Z_{k+1})$.

It is common to rewrite (1.3) by splitting it into the expected reward for the current time step and the expected reward for subsequent time steps to give:

$$V_H^{\pi}(\mathbf{b}_k) = R(\mathbf{b}_k, A^{\pi}(\mathbf{b}_k)) + \mathbb{E}\left[V_{H-1}^{\pi}(B_{k+1}^{\pi})|B_k^{\pi} = \mathbf{b}_k\right] \tag{1.4}$$

where the expectation is taken with respect to the future measurement Z_{k+1}. The single step reward, $R(\mathbf{b}_k, A^{\pi}(\mathbf{b}_k))$, is the expected reward with respect to the future measurement Z_{k+1}:

$$R(\mathbf{b}_k, A^{\pi}(\mathbf{b}_k)) = \mathbb{E}\left[r(B_k, A^{\pi}(B_k), Z_{k+1})\,|B_k = \mathbf{b}_k\right] \tag{1.5}$$

Note that the expectation with respect to the remaining future measurements $(Z_{k+2}, \ldots, Z_{k+H})$ in (1.3) is now contained in the future value term $V_{H-1}^{\pi}(B_{k+1}^{\pi})$ in (1.4). Equation (1.4) can be identified as a form of Bellman's equation. The calculation of the value of a belief state when following policy π is illustrated in Figure 1.3.

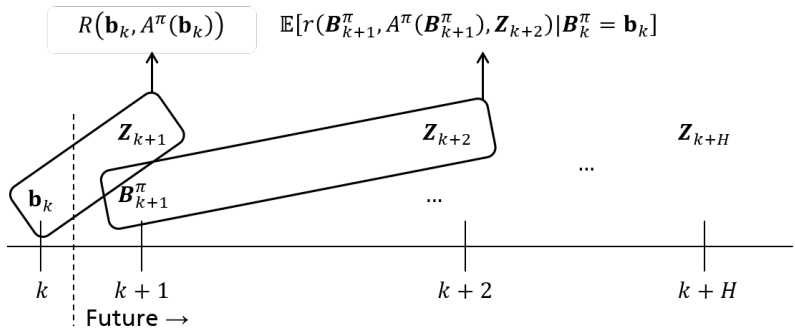


Figure 1.3 Calculation of the value of a belief state when following policy A^{π}. The single step expected reward is calculated using the current belief state realization $\mathbf{b}_k$ and with respect to the future measurement random variable Z_{k+1}. The expected reward from future times steps is calculated with respect to future belief state and measurement random variables.

Similarly to the value of a belief state when following policy π, it is possible to define the optimal value of a belief state as:

$$V_H^{*}(\mathbf{b}_k) = \max_{\mathbf{a}\in\mathscr{A}} \left(R(\mathbf{b}_k, \mathbf{a}) + \mathbb{E}\left[V_{H-1}^{*}(B_{k+1}^{\mathbf{a}})|B_k = \mathbf{b}_k\right]\right) \tag{1.6}$$

where $B_{k+1}^{\mathbf{a}}$ is a random variable representing the belief state in the next decision step that evolves when taking action $\mathbf{a}$, i.e. $B_{k+1}^{\mathbf{a}} = f_B(B_k, \mathbf{a}, Z_{k+1})$. Using the optimal value function, the optimal policy function can be defined, which is a description of an optimal perception-action cycle:

$$A^{*}(\mathbf{b}_k) = \arg\max_{\mathbf{a}\in\mathscr{A}} \left(R(\mathbf{b}_k, \mathbf{a}) + \mathbb{E}\left[V_{H-1}^{*}(B_{k+1}^{\mathbf{a}})|B_k = \mathbf{b}_k\right]\right) \tag{1.7}$$

The first term in (1.7) represents the expected reward associated with the current belief state and the chosen action, and is relatively easy to calculate. However, the

second term that represents the expected reward associated with future belief states in the time horizon is very difficult to calculate. Consequently, solving the optimal policy function is generally intractable. The majority of stochastic optimization approaches focus on approximate solutions to this optimal policy function.

Equation (1.3) is a multi-step objective function for the case when it is desired to optimize the expected rewards accumulated over the time horizon. Alternatively, the terminal reward may be of interest at the end of the time horizon. This can be accommodated by using an altered reward function that returns zero except for the last decision step in the time horizon. This section has described a problem with finite horizon H. An infinite horizon problem can be described in the same way, but requires the inclusion of a discounting factor.

1.5.2 Simplified Multi-Step Objective Functions

Finding policies that solve (1.7) is very challenging due to the need to evaluate the impact of the current action on expected future rewards, knowing only the current belief state. There are simplifications that are often performed that drastically reduce the complexity of the problem but result in an objective function that does not fully consider the uncertainty present in the problem. These simplifications are often applied in current cognitive radar techniques, as will be shown in Section 1.7.

1.5.2.1 Myopic Optimization

If the time horizon is taken as a single step, i.e. $H = 1$, then the problem of evaluating the impact of the action on expected future rewards is removed. Hence, the optimal policy function in (1.7) is significantly simplified to:

$$A^*(\mathbf{b}_k) = \arg\max_{\mathbf{a}\in\mathscr{A}} \left(R(\mathbf{b}_k, \mathbf{a})\right) \tag{1.8}$$

This approach is known as myopic or greedy optimization as it focuses on the immediate expected reward and ignores the impact of potential future rewards. This approach can represent a significant simplification of the problem that may result in poor action selection and hence a reduced accumulated reward. However, there may be problems in which the optimal myopic policy coincides with the optimal non-myopic policy. In which case, this simplification is completely justified.

1.5.2.2 Deterministic Optimization

A second common simplification is to perform a deterministic optimization based on expected values of the belief state and/or future measurements, instead of treating them as random variables and calculating the expected reward. An example of this approach would be to simplify the myopic reward function in (1.5) as:

$$R(\mathbf{b}_k, A^\pi(\mathbf{b}_k)) \approx r\left(\mathbf{b}_k, A^\pi(\mathbf{b}_k), \mathbb{E}\left[Z_{t+1}|B_k = \mathbf{b}_k\right]\right) \tag{1.9}$$

Whereas myopic optimization ignores the propagation of uncertainty into the future, deterministic optimization ignores the uncertainty in the belief state transition and measurement processes. However, by treating the optimization problem as being deterministic, it can be easier to solve. As the reward is now a deterministic mapping

from the actions to a real number, standard techniques to optimize functions can be used, for example numerical optimization methods, metaheuristics such as simulated annealing, or convex optimization.

1.5.2.3 Discussion

Stochastic optimization techniques aim to find a policy that closely matches the optimal policy function and therefore perform an action that is optimized considering the uncertainty in the future evolution of the system and the noisy measurement process. However, it should be clear that solving the optimal value and policy functions for realistic problems is intractable. Consequently, existing cognitive radar techniques often simplify the problem by performing myopic or deterministic optimization. However, advances in computational capability combined with the development of new algorithms mean that it is possible to move away from these simplifications and look towards designing perception-action cycles that fully consider the uncertainty in the problem. A subsequent and critical question for any problem is then: *which sources of problem uncertainty have a significant impact on performance and should therefore be incorporated into the optimization process?*

1.6 Policies and Perception-Action Cycles

Solving a stochastic optimization problem involves finding a policy that maps from belief states into actions and hence constitutes a perception-action cycle. This section gives an overview on methods for finding policies that are widely used in stochastic optimization. As mentioned earlier, [3] organizes the methods of finding policies into four classes that cover all approaches in the literature. The first two methods are policy search approaches and are referred to as policy function approximations (PFA) and cost function approximations (CFA). The second two approaches are lookahead approaches and are referred to as value function approximations (VFA) and direct lookahead. We discuss each of these in turn here, with the purpose of showing that established algorithmic strategies from the field of stochastic optimization can be valuable tools for designing perception-action cycles in a cognitive radar. More details on each of these methodologies can be found in [3] and the references therein.

1.6.1 Policy Search

The general approach to policy search is to find and tune a policy that matches or approximates the optimal policy function in (1.7). Generally, the optimal policy is unlikely to be found. Instead an approximation to the optimal policy function is sought in the form of a policy function approximation or a cost function approximation.

1.6.1.1 Policy Function Approximations

Policy function approximations (PFAs) attempt to find and tune a function that approximates the optimal policy function in (1.7). For example, we can consider a family of functions $\mathscr{F}$, where a function $f \in \mathscr{F}$ is parameterized by $\theta \in \Theta^f$. Our

goal is then to find a function and parameterisation f, θ so that the optimal policy function in (1.7) can be approximated as:

$$A_{PFA}^{f,\theta}(\mathbf{b}_k) = f(\mathbf{b}_k;\theta) \tag{1.10}$$

The optimal policy can be found if the optimal policy belongs to the family of functions and the corresponding parameter space. The goal of policy function approximations is not to find the optimal policy, but to find the best approximation within a class of function approximations. The function class may be any approach for approximating a function, such as an analytic function or a neural network.

An example from the radar literature is the work presented in [39]. Here the problem of optimizing the radar revisit times for a target track is considered. The authors define the concept of a *track sharpness* V_0, which is the major axis of the uncertainty ellipsoid in antenna coordinates (u-v space) relative to the beam width. The general strategy is to schedule a radar dwell to update the track once the track sharpness crosses a given threshold.

It is possible to cast the problem in [39] into the framework components in Section 1.3. The tracker provides parameters for the belief state $\mathbf{b}_k = [r\ \Theta\ \Sigma\ \sigma\ B]^T$, where r is the estimated target range, Θ, Σ are parameters of the Singer target dynamic model, σ is the measurement error standard deviation, and B is the radar half beamwidth. Note that σ and B are sensor parameters that may be dependent on the target kinematic parameters. The action space is a scalar representing the revisit interval time, i.e. $a_k \in \mathbb{R}_+$. where $\mathbb{R}_+$ denotes the positive real numbers. [39] proposes a function for finding the steady state revisit interval:

$$A_{PFA}^{V_0}(\mathbf{b}_k) = 0.4\left(\frac{r\sigma\sqrt{\Theta}}{\Sigma}\right)^{0.4}\frac{U^{2.4}}{1+0.5U^2} \tag{1.11}$$

where $U = BV_0/\sigma$ is the variance reduction ratio. Although it is not stated in [39], this can be considered as a policy function approximation, whereby the function in Equation (1.11) is parameterized by the track sharpness $\theta = V_0$. After proposing the policy function in Equation (1.11), [39] proceeds by finding the function parameterization, i.e. the value of V_0, that minimises the radar loading while also maintaining track on the target.

1.6.1.2 Cost Function Approximations

Instead of approximating the entire policy function as with a PFA, a cost function approximation (CFA) finds a functional approximation for the cost function, which is interchangeable with the reward function described in this paper. Consequently, the optimal policy function in (1.7) is replaced with:

$$A_{CFA}^{\pi,\theta}(\mathbf{b}_k) = \arg\max_{\mathbf{a}\in\mathscr{A}^\pi(\theta)}[\tilde{r}^\pi(\mathbf{b}_k,\mathbf{a};\theta)] \tag{1.12}$$

which comprises of the approximation to the cost function $\tilde{r}^\pi(\mathbf{b}_k,\mathbf{a}_k;\theta)$ as well as a potentially constrained action space $\mathscr{A}^\pi(\theta)$.

An example for such a cost function approximation can be found in the work of [40]. Here the track revisit interval is chosen to minimize the trace of the pre-

dicted conditional Bayesian information matrix (PC-BIM). Without any additional constraints such an optimization would always choose the minimum revisit interval. Therefore, a parameter K is introduced, which can be tuned to perform a trade-off between the information gain and the resource usage. Consequently, the following minimization is performed on a cost function approximation:

$$A^K_{CFA}(\mathbf{b}_k) = \arg\min_{a\in\mathscr{A}} \left(\mathrm{tr}(\mathbf{P}(a,\mathbf{b}_k)) + \frac{K}{a} \right) , \tag{1.13}$$

where $\mathscr{A} = \mathbb{R}_+$ is the set of all possible revisit intervals, and $\mathrm{tr}(\mathbf{P}(a,\mathbf{b}_k))$ is the trace of the PC-BIM after a measurement is produced with a revisit interval of a and conditioned on the current belief state realization $\mathbf{b}_k$.

1.6.2 Lookahead Approximations

Lookahead approximations differ from policy search as they attempt to evaluate the influence of an action on future rewards, instead of approximating the policy function. A lookahead approximation can be performed via a value function approximation or by simulating a direct lookahead.

1.6.2.1 Value Function Approximations

A value function approximation uses the optimal policy function in (1.7), but replaces the true optimal value of future belief states $V^*_{H-1}(B^{\mathbf{a}}_{k+1})$ with an approximation $\tilde{V}_{H-1}(B^{\mathbf{a}}_{k+1})$. In some cases the expectation in (1.7) may be difficult to calculate, in which case a value function approximation can be used to replace $\mathbb{E}\left[V^*_{H-1}(B^{\mathbf{a}}_{k+1})|B_k = \mathbf{b}_k\right]$.

The resulting policy for a value function approximation is:

$$X^{\tilde{V}}_{VFA}(\mathbf{b}_k) = \arg\max_{\mathbf{a}\in\mathscr{A}} \left(r(\mathbf{b}_k,\mathbf{a}) + \mathbb{E}\left[\tilde{V}(B^{\mathbf{a}}_{k+1})\right] \right) \tag{1.14}$$

Another variant is the approximation of the action value $\tilde{Q}$, which results in the policy

$$X^{\tilde{Q}}_{VFA}(\mathbf{b}_k) = \arg\max_{\mathbf{a}\in\mathscr{A}} \tilde{Q}(\mathbf{b}_k,\mathbf{a}) \, . \tag{1.15}$$

A famous algorithm from the literature that uses such policies is the Q-learning algorithm [41], whose variants have also been used for radar management [6]. A non-learning example in radar management can be found in [42]. Here the reward is based on the detection range of the radar, and the action $\mathbf{a}_k = [t\ ri]^T$ consists of the revisit interval ri and the dwell duration t. The problem is radar search, where the detection range of the radar should be optimized. To frame this problem in terms of Equation (1.15), $\tilde{Q}(\mathbf{b}_k,\mathbf{a})$ is the expected target detection range. Note that $\mathbf{b}_k$ here does not contain the estimate of a target track, as no track is detected yet. Instead, it contains observable quantities such as the platform's altitude and prior information about the expected target RCS and popup range. In [42] this value is approximated with a lookup table and used in a QoS resource allocation algorithm.

1.6.2.2 Direct Lookahead

For the cases when it is not possible to find an accurate value function approximation, the expected future value can be evaluated by simulating future system evolutions using the available models. As this process is computationally very costly, direct lookahead methods focus on making effective simplifications that still lead to accurate values. Common methods belonging to this class are deterministic lookaheads, Monte Carlo sampling, rollout policies and Monte Carlo tree search.

Myopic, single period lookahead policies are variants of this method, which are often used in the radar literature, e.g. [43,44]. A non-myopic lookahead method can be found in [45], which uses a policy rollout method to determine the optimal track revisit intervals. Note that this work also contains components of a cost function approximation, by encoding the performance to resource trade-off with a tune-able parameter in the cost function. Policy rollout is also used in [46] for solving the radar resource management problem.

1.6.3 Discussion

General methodologies for finding policies involve finding a function approximation to either the policy function, the cost function or the value function. The difference between these approaches is simply where the functional approximation is made, as illustrated in Figure 1.4. The effectiveness of these approaches depends on how well a function approximation can capture these respective relationships. All of these methodologies can be implemented with handcrafted models or using machine learning techniques. Although it is typical to perform offline training, these function approximations could be updated online as more data becomes available. Direct lookahead approaches are used when it is not possible to capture the structure of the problem with a function approximation.

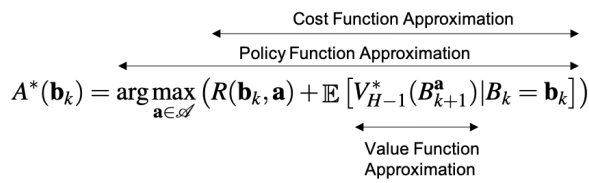


Figure 1.4 Different function approximation types for the optimal policy function.

1.7 Relationship between Cognitive Radar and Stochastic Optimization

In the previous sections, the general framework of stochastic optimization, as well as possible solution techniques are described. This framework models the problem of selecting the best action under uncertainty, to maximize a reward. Cognitive radar is an application domain, which falls under the assumptions of this framework.

Knowledge about the true state is only received by noisy measurements, and state and belief state transitions are non-deterministic. A radar controller must select the optimal sensing actions to maximize performance of the system, which can be formalized by a suitable reward function. In the following, this view of the cognitive radar task as a stochastic optimization problem is mapped out.

1.7.1 Problem Components

A representative set of cognitive radar problems for different applications can be found in the references. Although it may not always be explicitly stated, these problems can be characterized as stochastic optimization problems that possess the framework components described in Section 1.3. The components are sometimes explicitly stated or can be inferred.

In the case of target tracking [2, 37, 39, 44, 47–51], the belief state characterizes a posterior probability density function defined on the system state space. Typical belief states are the mean and covariance matrix of the distribution or a set of particles. The belief state transition function incorporates the Bayesian prediction and update processes. The exogenous information is some noisy function of the system state that maps to radar measurements, thus the system state is partially observable. Often, the likelihood function is a Gaussian approximation of the true measurement errors. Adaptive tracking [2, 39, 48] methods select actions in the form of revisit interval times as well as the waveform energy for the next measurement, in order to minimize resource usage while maintaining track. An early approach [39] was to use a function that mapped measurement and track accuracies, and Singer manoeuvre parameters to a revisit interval time. In the context of the methods described in Section 1.6, this can be thought of as an empirically derived policy function approximation. Another strand of work has focussed on waveform selection and adaptation [44, 47, 50], whereby the action space comprised different waveform modulations that were selected in order to minimize track RMSE.

The framework components are easy to identify for tracking problems, because the framework is essentially an extension to the standard Bayesian tracking process. However, other radar functions and applications can also be cast into the framework. For a search problem, the belief state can parameterize an undetected target posterior density. In target detection [52–55], the system state is the state of the clutter, interference, and noise environment. Typical belief states include the clutter, interference, and noise covariance matrix or a posterior distribution on a spectrum occupancy state. For imaging and classification [56] the belief state characterizes a posterior probability mass function. Typical belief states are the pairwise likelihood ratios or the posterior probabilities themselves. Some works also consider a combination of radar functions [43, 57, 58].

Generally, the action space is some set of parameters that characterize the radar transmission and reception, including transmit and receive sensor selection and scheduling, transmit frequency, bandwidth, time, duration, power, and waveform design. The exogenous information is some noisy function of the system state, thus the system state is partially observable. Generally, reward functions differ widely, but can be categorized according to the classes in Section 1.4.

1.7.2 Typical Cognitive Radar Solution Methodologies

A variety of solution methodologies have been applied to cognitive radar problems, which can be compared with the strategies described in Section 1.6. The majority of the reference works formulate myopic optimization problems, which represent a simplification with respect to the general non-myopic multi-step objective function. Depending on the problem, this can be a very valid approach to reduce the complexity of the optimization, especially if it is clear that the current action does not influence future rewards. However, it is worthwhile to explicitly consider how the myopic and non-myopic solutions differ, as there are certainly problems where considering the future rewards associated with the current action can significantly improve performance.

There are also cases in the reference works where an optimization is performed on an expected value of the belief state and/or an expected future measurement, instead of treating the system state and future measurements as random variables and calculating the expected reward. This approach has the benefit of enabling deterministic optimization methods to be applied and is a valid approximation if the reward function is not sensitive in the region of significant probability as described by the posterior and expected measurement PDFs. However, this approach ignores or under-utilizes the uncertainty in the future state evolution and corresponding measurements, which could significantly impact performance.

The cognitive radar methodologies in the reference works generally attempt to solve an optimization problem online by performing numerical optimizations or searches over the action space. However, the strategies described in Section 1.6 first attempt to identify structure in the policy, cost or value function and attempt to use specific models or machine learning to produce a functional approximation. This is a particularly attractive approach because it can reduce the complexity of the online optimization problem, or remove the need to perform an online optimization, depending on the functional approximation type. This approach is underrepresented in the reference works, but can be identified in [51], where a neural network is used to learn the policy function that an optimizer with more complexity would generate.

1.7.3 Cognitive Radar Objective Functions

Although cognitive radar problems and approaches can be cast into the framework described in Section 1.3, there are some key differences. A main difference comes in form of the objective function.

In Section 1.5 the objective of the framework is described as finding a policy with maximal accumulated reward. This is a common formulation in many fields of stochastic optimization, for example, in control theory or reinforcement learning. Almost all papers in the reinforcement learning literature demonstrate the development of these accumulated rewards (or costs) over the time of the training, consequently allowing a summary of the performance of a policy in a single metric. Therefore, it is possible to directly compare two policies and decide which is better.

On the other hand, such a statement is rare in the cognitive radar literature. Typically several metrics are considered in the evaluation. Although these metrics

are used in the performance evaluation, they are not always stated explicitly as part of the objective function. We acknowledge that the evaluations yield important insights, however, it is useful to also clearly state the true objective in a single quantifiable way. For example "in the given evaluation scenario, we want to minimize the sum of the tracking errors over the whole scenario time". A common occurrence in the radar literature is the usage of surrogate functions (e.g. SNR, mutual information, etc.). Generally, it should be clear whether this is the true objective or actually a surrogate for the harder to evaluate true objective function.

We note that finding representative and quantifiable objective functions is a non-trivial challenge, which is a justification for using simplified or surrogate objective functions that are then evaluated against the actual true objectives. For example, it is very challenging to find appropriate objective functions for multi-function radars, which are required to balance the conflicting demands of different functions. Generally, the radar designer may be less interested in the result of optimization as the all-round performance of the radar in realistic situations, which may not be possible to evaluate until after the optimization has been performed. Regardless, differences between the objective function, and the actual objective and hence evaluation metrics, is an indicator that the objective function may not be truly representative of the problem to be solved. We identify the construction of representative objective functions as an important challenge in cognitive radar research.

1.8 Simulation Examples

In this section, we present two simulation examples that demonstrate the influence of different sources of uncertainty on the control process.

1.8.1 Adaptive Tracking Example

This section presents an adaptive tracking example, whereby it is necessary to decide on the next revisit interval for tracking a target with an agile beam radar. As the radar steers the beam to the estimated target position, a beam positioning loss occurs that is dependent on the difference between the beam pointing direction and the true target direction, which in turn depends on the accuracy of the track. Overall, it is desired to use as few resources as possible to track the target whilst also aiming to prevent the target from escaping the radar beam.

1.8.1.1 Problem Components

This problem can be described in terms of the problem components introduced in Section 1.3.2.

System State and State Transition Function - The underlying system state is the position and velocity of the target in antenna (i.e. u-v) coordinates. The system state transitions according to a constant velocity, continuous white noise acceleration model, with a specific process noise intensity.

Actions and Action Space - The action is the revisit interval, which is the interval between the current time and the time of the next track update. Revisit intervals

between 0.1 s and 5 s are allowed and this continuous range is discretized into 50 possible revisit interval values.

Measurements and Measurement Space - The radar produces measurements of the target angle in antenna (i.e. u-v) coordinates. Therefore, $\mathbf{z}_{k+1} \in [-1,1]^2$. A measurement occurs if the signal amplitude exceeds the detection threshold assuming Swerling 1 radar cross section fluctuations. The detection and measurement processes depend on an SNR value which is influenced by the beam positioning loss. This beam positioning loss occurs as the beam is directed to the angle given by the estimate in the track, which may differ from the true target angle. The beam positioning loss is modelled as a Gaussian loss function matched to the radar beamwidth.

Measurement Likelihood Function - Each measurement dimension is corrupted by independent Gaussian noise with standard deviation depending on the SNR:

$$\sigma_{u,v} = \frac{B}{2\sqrt{SNR}} \tag{1.16}$$

where B is the radar 3-dB beamwidth. Consequently, $\mathscr{L}(\mathbf{x}_k|\mathbf{z}_k,\mathbf{a}_{k-1}) \equiv \mathscr{N}(\mathbf{z}_k;\mathbf{H}\mathbf{x}_k,\mathbf{R}_k)$ where $\mathbf{H}$ is the observation matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{1.17}$$

and $\mathbf{R}_k = \text{diag}([\sigma_{u,v}^2 \sigma_{u,v}^2])$ is the measurement error covariance.

Information State - As described in Section 1.3.2, the information state is the collection of previous actions and measurements. As the belief state described next is a sufficient statistic of the information state, it is not necessary to maintain the information state.

Belief State - The belief state comprises an estimate of the target angles in antenna coordinates and the associated covariance matrix. Additionally, the belief state contains the known mean target radar cross section and the process noise intensity for the system state transition model.

Belief State Transition Function - The belief state transition function incorporates the standard Kalman filter prediction and update steps.

Reward Function - If a detection occurs, then the reward is the revisit interval value that was selected. If no detection occurs, then zero reward is achieved, leading to the reward function:

$$r(\mathbf{b}_k,\mathbf{a}_k,\mathbf{z}_{k+1}) = \begin{cases} 0 & \text{if } \mathbf{z}_{k+1} = \{\} \\ a_k & \text{otherwise} \end{cases} \tag{1.18}$$

Consequently, the controller is motivated to maximise the revisit interval whilst also ensuring a detection occurs and hence that the target does not escape the beam. The reward is normalised by the number of actions in the horizon in order to allow an easy comparison between different horizon lengths.

1.8.1.2 Control Methods

The problem described above is solved using an exhaustive direct lookahead that evaluates the expected reward for every possible action. For a time horizon of multi-

ple time steps, all possible action sequences are evaluated. The reward function is a function of several random variables that can be considered as sources of uncertainty. The state uncertainty is represented by the belief state, and results in an SNR loss due to the uncertain target angle differing from the estimated target angle. The measurement uncertainty results from Swerling 1 radar cross section fluctuations that impact on the SNR, the stochastic detection process, and stochastic angular measurement errors.

For this analysis we use different methods that account for different sources of uncertainty. When a source of uncertainty is considered by the controller, then the expected reward is evaluated using Monte Carlo sampling for the respective random variable. We compare the following lookahead strategies for evaluating the expected reward:

- *Expected Value State/Expected Value Measurement* (EVS/EVM) - The expected values of the state and the target radar cross section are used. The SNR is scaled by the non-zero expected beam positioning loss. An angular measurement is generated with no measurement noise. The reward is scaled by the probability of detection.
- *Randomly Sampled State/Expected Value Measurement* (RSS/EVM) - Samples of the state are drawn from the belief state, leading to samples of the beam positioning loss and consequently samples of the SNR based on the mean radar cross section. For each sample, an expected angular measurement is generated with no measurement noise. The reward is scaled by the probability of detection.
- *Expected Value State/Randomly Sampled Measurement* (EVS/RSM) - The expected value of the state and the beam positioning loss is used. The radar cross section and hence SNR is sampled and the detection process simulated for each sample. If a detection occurs, noisy angular measurements are generated according to the standard deviation of the respective SNR sample.
- *Randomly Sampled State/Randomly Sampled Measurement* (RSS/RSM) - Samples of the state and the radar cross section are drawn, leading to samples of the beam positioning loss and consequently samples of the SNR. The detection process is simulated for each sample. If a detection occurs, noisy angular measurements are generated according to the standard deviation of the respective SNR sample.

As RSS/RSM evaluates the expected reward considering all the modelled sources of uncertainty, it can be considered as the true reward value, under the assumption that the underlying models match to the reality.

1.8.1.3 Results

These results are produced using the parameter values in Table 1.1.

The expected rewards associated with the possible actions using a single step horizon are illustrated in Figure 1.5. It can be seen that the different consideration of the sources of uncertainty in the lookahead strategies lead to different expected rewards. As the action with the greatest expected reward is selected, not considering certain sources of uncertainty can lead to sub-optimal action selections. In this result,

Parameter	Value
SNR	111 (20 dB)
Probability of False Alarm	10^{-6}
Mean RCS	1 m^2
Beamwidth	1 (degrees)
Track Sharpness	0.15 (beamwidths)
Process Noise	$(0.004)^2$

Table 1.1 Simulation parameters unless otherwise stated in the results. SNR is the SNR for the mean radar cross section and without beam positioning losses.

measurement sampling (RSM) does not influence the expected reward and instead an expected measurement can be used. This is logical, as the reward function for a one step horizon is not impacted by the different measurement values or the associated measurement noise covariances. The reward function is impacted by the detection probability, however, it is not necessary to simulate the actual detection process. In Figure 1.5 it can be seen that sampling the state results in significantly different expected rewards. The state uncertainty is the source that has the greatest influence on the expected reward. When considering RSS/RSM to be the true expected reward, not considering the state uncertainty leads to the selection of a 3.1 s revisit interval instead of 2 s, which results in a 19% reduction of the expected reward from 1.294 to 1.048.

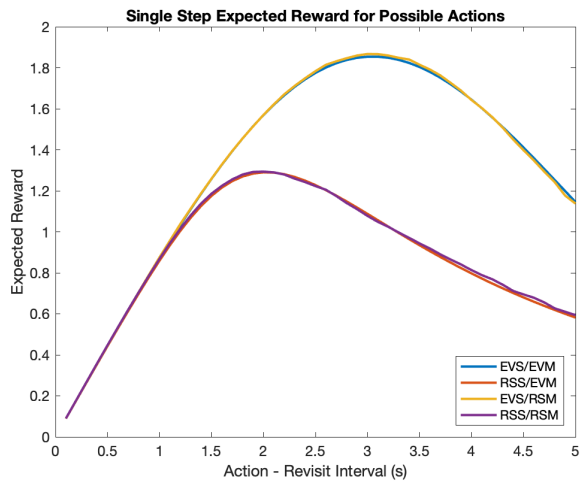


Figure 1.5 The expected rewards using the different lookahead strategies for a one step horizon.

Figure 1.6 shows the best action selected for different values of the process noise intensity and track sharpness. Generally, lower revisit intervals are selected for larger process noise intensities and initial track sharpness. Not considering the state uncertainty leads to sub-optimal action selection, regardless of whether measurement sampling is performed or not. As to be expected based on Figure 1.5, EVS/EVM and EVS/RSM selected optimistically long revisit intervals, because they do not adequately evaluate the impact of beam positioning loss on the expected reward. In Figure 1.6 it can be identified that there are simple functional relationships between the parameters of the belief state and the selected action. Consequently, it is possible to perform regression on the results from the exhaustive direct lookahead to produce a policy function approximation that has negligible online computation.

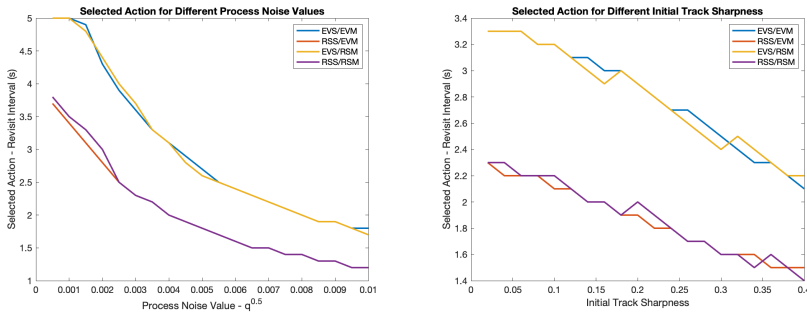


Figure 1.6 The selected actions using the different lookahead strategies for a one step horizon.

Figure 1.7 illustrates the expected reward of the possible actions using the different lookahead strategies for a two step horizon length. In contrast to Figure 1.5, all sources of uncertainty impact on the evaluation of the expected reward. Now, measurement sampling in the first step influences the probability of detection and hence the reward in the second step. However, just performing state sampling and not measurement sampling still results in expected rewards that are closer to the optimum of RSS/RSM. By analysing this figure it can be seen that EVS/EVM, RSS/EVM, EVS/RSM result in a loss of expected reward of 22.68%, 1.06% and 8.35% respectively. For this example, a single step horizon instead of a two step horizon results in a loss of expected reward of 2.33%. Although a longer time horizon improves performance, considering the sources of uncertainty has a greater impact on the expected reward and hence action selection.

Figure 1.8 shows the best action sequence selected for different values of the process noise intensity and track sharpness. Again it can be seen that larger process noise intensities and track sharpness lead to lower revisit intervals. A general recognisable strategy is to schedule a short revisit interval followed by a long revisit interval, especially in the cases of low process noise intensities as well as large initial track sharpness. As seen with the single step horizon, a basic functional relationship between the belief state parameters and the selected action can be seen.

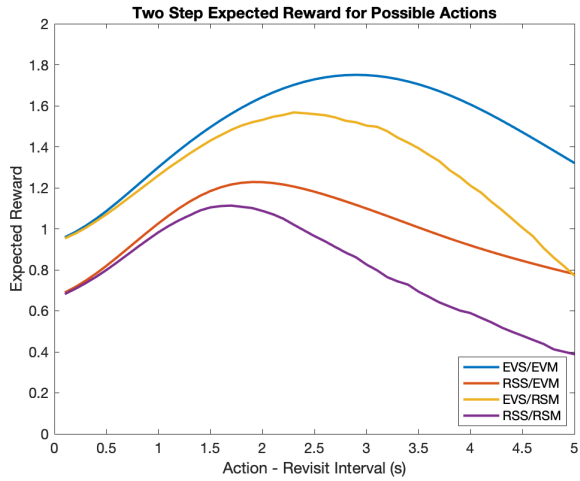


Figure 1.7 The expected rewards using the different lookahead strategies for a two step horizon.

Consequently, a policy function approximation can be produced using regression to approximate the result of this exhaustive direct lookahead, which required significant computation even for this simple example.

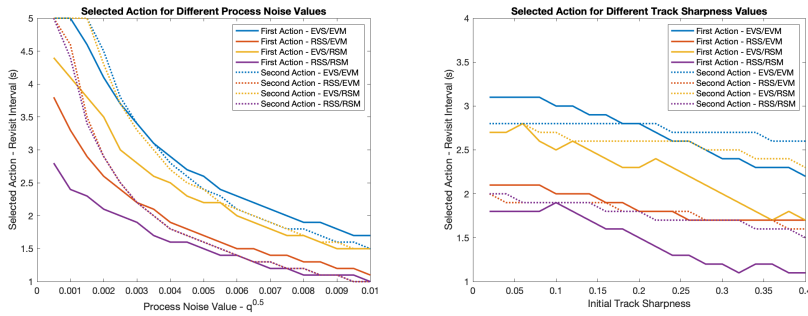


Figure 1.8 The selected action sequence using the different lookahead strategies for a two step horizon.

This analysis of the different lookahead strategies assumed that the underlying models match the reality and that the reward function matches the objective of the radar. Although the choice of the reward function is intuitively appealing, a radar engineer is likely to start wondering how this control strategy performs in terms of other performance measures, such as track losses and track accuracies. This highlights the difficulty in creating truly representative reward functions, as discussed in Section 1.7.3. Additionally, RSS/RSM can be considered as the true expected re-

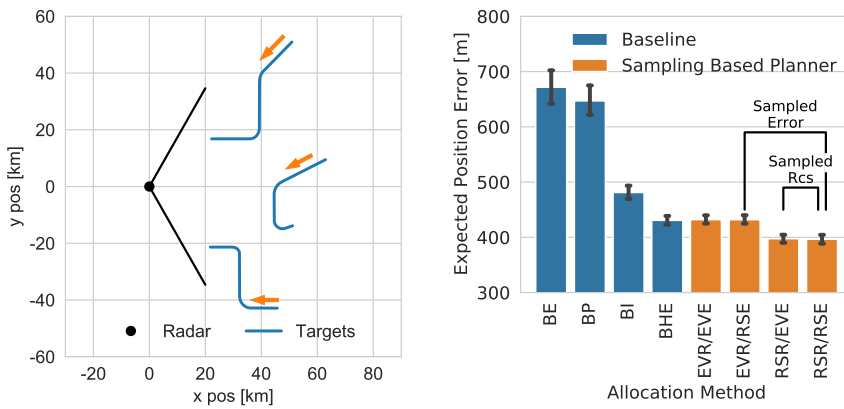
ward when the models are true. However, it is not the case that an object exhibits continuous white noise acceleration motion in antenna coordinates. An evaluation of these strategies on real trajectories will result in different accumulated rewards to those predicted by the lookahead strategies. This motivates the use of learning techniques for learning policy, cost or value function approximations based on realistic conditions.

1.8.2 Target Resource Allocation Example

This example uses a myopic lookahead method for allocating radar resources between multiple targets.

1.8.2.1 Scenario

The scenario contains a stationary radar, which tracks three airborne targets. The targets exhibit Swerling I radar cross section (RCS) fluctuations and follow trajectories 1-3 from [59]. Figure 1.9a shows the geometry of the scenario, which has a duration of 140 seconds. Additional parameters are given in Table 1.2. Nominal radar parameters in Table 1.2 specify the radar SNR performance, the actual SNR is calculated based on the actual parameter values by scaling the nominal SNR.



(a) (b)

Figure 1.9 Myopic planning example. (a) Scenario. The black lines show the field of view of the radar, and the orange arrows the movement direction of the targets. (b) Expected position error for different allocation methods.

The radar uses a 2000 Hz low pulse repetition frequency waveform and allocates a fixed time budget of 10% to the task of tracking the three targets. For every time step of 1200 ms, it uses 120 ms which is equivalent to 240 pulses for tracking. At each time step, the controller must make the decision of how to allocate those pulses to each of the targets. The targets are tracked using an IMM-EKF tracker, which a nearly constant velocity and a maneuver model. As the simulation focuses on track-

Parameter	Value
Nominal radar range	50 km
Nominal radar SNR	20 (13 dB)
Nominal RCS	1 m^2
Nominal number of pulses	100
Nominal pulse length	2 μs
Pulse length	2 μs
Pulse repetition frequency	2000 Hz
Signal bandwidth	1 MHz
Probability of a false alarm	10^{-6}
Wavelength	0.03 m
Target RCS	1 m^2
No-maneuver model noise factor q_0	10.0
Maneuver model noise factor q_1	1000.0
Simulation length	140 s

Table 1.2 Simulation Parameters.

ing and not search, tracks are initialized with the ground truth state at the beginning of the simulation. The tracker does not drop tracks during the simulation and it uses an ideal measurement to track association.

Since tracks are not dropped, the beam is always steered towards the true position of the target in order to avoid track divergence. While a track might still diverge, it would receive measurements when it gets resources allocated again. This is of course a simplifying assumption, and a real system would need some kind of reacquisition method for lost tracks. However, the implementation of such a method would make the simulation more complex, and distract from its purpose of comparing the sampling uncertainties.

1.8.2.2 Objective

The objective is to minimize the uncertainty of the tracks. We quantify this using just the position part of the covariance matrices for the tracks, leading to a negative reward (cost) at each stage k of

$$r(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1}) = -\sum_{t \in T} \sqrt{\mathrm{tr}\left(\mathbf{P}^{\mathrm{pos}}_{(k+1)t}\right)}\,, \tag{1.19}$$

where T is the set of targets, $\mathbf{P}^{\mathrm{pos}}_{(k+1)t}$ the 3x3 Cartesian position part of the covariance matrix of target t at time step $k+1$, updated with $\mathbf{z}_{k+1}$, and `tr` is the trace function.

1.8.2.3 Control Methods

We use four baseline control methods, which use simple heuristics

- (BE) equally allocates the same amount of pulses to each target
- (BP) allocates the pulses to targets such that all targets achieve the same SNR

- (BI) uses all the pulses for a single target and iterates through the targets between decision steps
- (BHE) uses all pulses on the target with the currently highest expected error, i.e. the target whose contribution to the reward is the highest

The planner is a myopic one-step lookahead planner. It considers a discrete action set A of possible allocations, and selects the action

$$\mathbf{a}_k = \arg\max_{\mathbf{a}\in A} \mathbb{E}\left[r_k(\mathbf{b}_k, \mathbf{a}, Z_{k+1}) \mid \mathbf{a}\right] \tag{1.20}$$

The actions consists of possible allocations of multiples of 60 pulses, leading to 15 different possible actions. For example, 240/0/0 or 60/120/60 are two possible actions. The control algorithm evaluates the expected value using Monte Carlo sampling. We sample two different random variables in the measurement generation process: the RCS of the targets, and the measurement errors. The RCS fluctuation influences the measurement SNR and therefore also the detection probability and measurement covariance. The measurement error is distributed according to the covariance of the measurement and influences not only the point estimate of the target, but also the likelihood of the different maneuver models. As we want to compare the influence of these two factors, we either perform Monte Carlo sampling or take the expected value. Each action is sampled 66 times, leading to a full computing budget of slightly below 1000 samples. We use Common Random Numbers when comparing the actions in order to reduce the sampling variance.

1.8.2.4 Results

Figure 1.9b shows the performance of the different methods after 100 Monte Carlo runs. The names of the different planner instantiations consists of combinations of randomly sampled (RS) or expected value (EV) of the RCS (R) or the error (E). For example, the rightmost entry results from a randomly sampled RCS and a randomly sampled error.

The results are given as the average from the covariance per target and per decision step. Note, that because the tracks are not dropped during the scenario and consequently the number of targets does not change, this scaled representation is proportional to the negative sum of the rewards $r_k(\mathbf{b}_k, \mathbf{a}_k, \mathbf{z}_{k+1})$ over the whole scenario. However, a scaled representation results in a more interpretable error metric.

We can see multiple effects in Figure 1.9b. Firstly, it seems to pay off to focus the pulses on a single target. Both baseline heuristic methods that spread the pulses (BE and BP) are significantly worse than those that focus them on a single target (BI and BHE). Secondly, the random variable that is sampled in the measurement process for the planner has a clear influence. When using the expected RCS, the planner is approximately as good as the strongest baseline. However, when sampling the RCS, the planner surpasses the baseline. On the other hand, whether it samples the measurement error or simply the takes expected measurement has no significant influence on performance. In this scenario, the RCS sampling has mostly an effect on the topmost target, which is furthest away. When taking the expected value of the RCS, the planner assumes that there is no detection chance at all and never allocates resources to this target for the first 25 seconds of the scenario. However, when

sampling the RCS, it recognizes that the RCS fluctuations provide a chance of a detection.

The keen-eyed reader has likely realized that we did not consider a third source of uncertainty, the actual position of the target. Instead, we only used the expected value of the track. In theory, we have a probability distribution over the target given by the mean and covariance in the track. However, it is very rare that the target is really distributed according to this probability distribution function as the control output of few pilots is truly Gaussian noise, as assumed by many models. A Gaussian process model works well to make the tracker stable, but is not necessarily suited for lookahead planning. In our experiments sampling the state had sometimes even detrimental effects, as the planner thought the target could be very far away and it would not reach a detection, for example in situations involving manoeuvres that led to a large covariance. This highlights the value of higher fidelity target dynamic models [60] in the context of planning. In the given setup we could also not find benefits from a non-myopic planning. Note that this is not necessarily surprising, given the performance guarantees of greedy strategies in comparison to non-myopic controllers [61].

This example shows that it is important to consider the source of uncertainty in the planning step. Some sources may have have a large impact on performance and some may have little impact. This consideration can be incorporated into the design of a planning algorithm using the stochastic optimization framework described in this chapter. For example, the results above would indicate that replacing the sampling process by just two outcomes - detection and non-detection, weighted by their analytical probabilities, is likely sufficient. The influence of other sources of uncertainties, e.g. the target state, might have an influence, however, care must be taken that the sampling distribution actually corresponds to the true possible states of the targets.

1.9 Conclusion

Many cognitive radar techniques are emerging that tackle different applications or sub-problems in a radar system. This chapter has presented a common framework for describing these cognitive radar problems in terms of a stochastic optimization problem. By doing so, the cognitive radar problem can be addressed using existing algorithmic strategies from the field of stochastic optimization. Specifically, the strategy of finding functional approximations for the optimal policy, cost or value function using machine learning techniques is an attractive approach. In general, learning techniques can be adopted to tackle a stochastic optimization problem, when models in the problem are difficult to describe analytically. Consequently, both control-theoretic methods and learning methods fall under the same framework.

Traditionally, cognitive radar and radar management has performed myopic and deterministic optimizations. However, advances in computing and algorithmic capabilities can enable the more general stochastic optimization problem to be tackled, which fully considers uncertain measurements and state transitions as well as the impact of action selection on future rewards. However, it is important to consider

which sources of uncertainty actually impact on the performance. Since increasing the consideration of uncertainty in a planning algorithm ultimately results in increased computation, focusing on just the critical sources of uncertainty enables an efficient and performant algorithm.

References

[1] Gurbuz SZ, Griffiths HD, Charlish A, et al. An Overview of Cognitive Radar: Past, Present, and Future. IEEE Aerospace and Electronic Systems Magazine. 2019;34(12):6–18.

[2] Charlish A, Hoffmann F. Cognitive radar management. In: Novel Radar Techniques and Applications Volume 2: Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion. Institution of Engineering and Technology; 2017. p. 157–193.

[3] Powell WB. A unified framework for stochastic optimization. European Journal of Operational Research. 2019;275(3):795 – 821.

[4] Howard WW, Thornton CE, Martone AF, et al. Multi-player Bandits for Distributed Cognitive Radar. In: 2021 IEEE Radar Conference (RadarConf21); 2021. .

[5] Boer TD, Schöpe MI, Driessen H. Radar Resource Management for Multi-Target Tracking Using Model Predictive Control. In: IEEE 24th International Conference on Information Fusion (FUSION). International Society of Information Fusion (ISIF); 2021. .

[6] Thornton CE, Kozy MA, Buehrer RM, et al. Deep Reinforcement Learning Control for Radar Detection and Tracking in Congested Spectral Environments. IEEE Transactions on Cognitive Communications and Networking. 2020;6(4):1335–1349.

[7] Charlish A, Bell K, Kreucher C. Implementing Perception-Action Cycles using Stochastic Optimization. In: 2020 IEEE Radar Conference (RadarConf20); 2020. p. 1–6.

[8] Hero AO, Castañon DA, Cochran D, et al. Foundations and Applications of Sensor Management. Springer; 2007.

[9] Musick S, Malhotra R. Chasing the Elusive Sensor Manager. Proceedings of NAECON. 1994 May;p. 606–613.

[10] Liu J, Cheung P, Guibas L, et al. A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks. ACM International Workshop on Wireless Sensor Networks and Applications. 2002 September;.

[11] Lumelsky VJ, Mukhopadhyay S, Sun K. Dynamic Path Planning in Sensor-Based Terrain Acquisition. IEEE Transactions on Robotics and Automation. 1990 August;6(4):462–472.

[12] Popoli R. The Sensor Management Imperative. In: Bar-Shalom Y, editor. Multitarget-Multisensor Tracking: Advanced Applications. vol. II. Artech House; 1992. p. 325–392.

[13] Hintz KJ, McVey ES. Multi-Process Constrained Estimation. IEEE Transactions on Man, Systems, and Cybernetics. 1991 January/February;21(1):434–442.

[14] Hintz KJ. A Measure of the Information Gain Attributable to Cueing. IEEE Transactions on Systems, Man and Cybernetics. 1991;21(2):237–244.

[15] Schmaedeke W, Kastella K. Event-averaged maximum likelihood estimation and information-based sensor management. Proceedings of SPIE. 1994 June;2232:91–96.

[16] Kastella K. Discrimination Gain for Sensor Management in Multitarget Detection and Tracking. IEEE-SMC and IMACS Multiconference CESA. 1996;1:167–172.

[17] Kastella K. Discrimination Gain to Optimize Classification. IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans. 1997 January;27(1).

[18] Mahler R. Global Optimal Sensor Allocation. Proceedings of the Ninth National Symposium on Sensor Fusion. 1996;1:167–172.

[19] Zhao F, Shin J, Reich J. Information-Driven Dynamic Sensor Collaboration. IEEE Signal Processing Magazine. 2002 March;p. 61–72.

[20] Chong E, Kreucher C, Hero A. Partially observable Markov Decision Process Approximations for Adaptive Sensing. Discrete Event Dynamic Systems. 2009 September;19(3):377–422.

[21] Chong E, Kreucher C, Hero A. POMDP Approximation Using Simulation and Heuristics. In: Hero A, Castañon D, Cochran D, et al., editors. Foundations and Applications of Sensor Management. Springer; 2008. p. 95–120.

[22] Krishnamurthy V, Evans D. Hidden Markov Model Multiarm Bandits: A Methodology for Beam Scheduling in Multitarget Tracking. IEEE Transactions on Signal Processing. 2001 December;49(12):2893–2908.

[23] Krishnamurthy V. Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors. IEEE Transactions on Signal Processing. 2002 June;50(6):1382–1397.

[24] Bertsekas D, Castañon D. Rollout Algorithms for Stochastic Scheduling Problems. Journal of Heuristics. 1999;5(1):89–108.

[25] Castañon D. Approximate Dynamic Programming for Sensor Management. Proceedings of the 1997 IEEE Conference on Decision and Control. 1997;.

[26] Castañon D. Optimal Search Strategies for Dynamic Hypothesis Testing. IEEE Transactions on Systems, Man, and Cybernetics. 1995;25(7):1130–1138.

[27] Malhotra R. Temporal Considerations in Sensor Management. Proceedings of the IEEE 1995 National Aerospace and Electronics Conference, NAECON 1995. 1995 May;1:86–93.

[28] Chhetri A, Morrell D, Papandreou-Suppappola A. The Use of Particle Filtering with the Unscented Transform to Schedule Sensors Multiple Steps Ahead. Procedings of International Conference on Acoustics, Speech, and Signal Processing 2004. 2004;.

[29] Shin J, Guibas L, Zhao F. A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks. Proceedings of 2nd International Workshop on Information Processing in Sensor Networks. 2003 April;.

[30] Aoki EH, Bagchi A, Mandal P, et al. A theoretical look at information-driven sensor management criteria. In: 14th International Conference on Information Fusion; 2011. p. 1–8.

[31] Sundaresan R. A measure of discrimination and its geometric properties. In: Proceedings of the 2002 IEEE International Symposium on Information Theory. IEEE; 2002. p. 264.

[32] Liese F, Vajda I. On divergences and informations in statistics and information theory. IEEE Transactions on Information Theory. 2006;52(10):4394–4412.

[33] Van Trees HL, Bell KL, editors. Bayesian Bounds for Nonlinear Filtering/Tracking. Wiley; 2007.

[34] Ashok Kumar M, Mishra KV. Information Geometric Approach to Bayesian Lower Error Bounds. In: 2018 IEEE International Symposium on Information Theory (ISIT); 2018. p. 746–750.

[35] Ashok Kumar M, Mishra KV. Cramér–Rao lower bounds arising from generalized Csiszár divergences. Information Geometry. 2020;3:33–59.

[36] Hansen JP, Ghosh S, Rajkumar R, et al. Resource management of highly configurable tasks. In: 18th International Parallel and Distributed Processing Symposium. Santa Fe, New Mexico; 2004. p. 116.

[37] Mitchell AE, Smith GE, Bell KL, et al. Cost function design for the fully adaptive radar framework. IET Radar, Sonar, and Navigation. 2018;12(12):1380–1389.

[38] Yuan Y, Yi W, Kirubarajan T, et al. Scaled accuracy based power allocation for multi-target tracking with colocated MIMO radars. IEEE Journal on Selected Topics in Signal Processing. 2019;158:227–240.

[39] van Keuk G, Blackman SS. On phased-array radar tracking and parameter control. IEEE Transactions on Aerospace and Electronic Systems. 1993;29(1):186–194.

[40] Christiansen JM, Olsen KE, Smith GE. Fully adaptive radar for track update-interval control. In: 2018 IEEE Radar Conference, RadarConf 2018. IEEE; 2018. p. 400–404.

[41] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, Massachusetts, USA: MIT Press; 2018.

[42] Hoffmann F, Charlish A. A Resource Allocation Model For The Radar Search Function. In: International Radar Conference. Lille, France: IEEE; 2014. p. 1–6.

[43] Bell KL, Baker CJ, Smith GE, et al. Cognitive Radar Framework for Target Detection and Tracking. IEEE Journal on Selected Topics in Signal Processing. 2015;9(8):1427–1439.

[44] Sira SP, Li Y, Papandreou-Suppappola A, et al. Waveform-agile sensing for tracking. IEEE Signal Processing Magazine. 2009;26(1):53–64.

[45] Charlish A, Hoffmann F. Anticipation in Cognitive Radar using Stochastic Control. In: 2015 IEEE Radar Conference (RadarCon). Arlington, Virginia, USA: IEEE; 2015. p. 1692–1697.

[46] Schöpe MI, Driessen H, Yarovoy A. Multi-Task Sensor Resource Balancing Using Lagrangian Relaxation and Policy Rollout. In: 2020 IEEE 23rd International Conference on Information Fusion (FUSION); 2020. p. 1–8.

[47] Kershaw DJ, Evans RJ. Waveform selective probabilistic data association. IEEE Transactions on Aerospace and Electronic Systems. 1997;33(4):1180–1188.

[48] Kirubarajan T, Bar-Shalom Y, Blair WD, et al. IMMPDAF for radar management and tracking benchmark with ECM. IEEE Transactions on Aerospace and Electronic Systems. 1998;34(4):1115–1134.

[49] Chong EKP, Kreucher CM, Hero AO. Monte-Carlo- based partially observable Markov decision process approximations for adaptive sensing. In: 9th Intl. Wkshp. on Discrete Event Systems; 2008. p. 173–180.

[50] Haykin S. Cognitive Dynamic Systems: Perception-action Cycle, Radar and Radio. Cambridge University Press; 2012.

[51] John-Baptiste P, Smith GE. Utilizing neural networks for fully adaptive radar. In: IEEE Radar Conf.; 2019. p. 1–6.

[52] Guerci JR. Cognitive Radar: The Knowledge-Aided Fully Adaptive Approach. Artech House; 2010.

[53] Aubry A, De Maio A, Piezzo M, et al. Radar waveform design in a spectrally crowded environment via nonconvex quadratic optimization. IEEE Transactions on Aerospace and Electronic Systems. 2014;50(2):1138–1152.

[54] Stinco P, Greco M, Gini F. Cognitive radars in spectrally dense environments. IEEE Aerospace and Electronic Systems Magazine. 2016;31(10):20–27.

[55] Martone AF, Ranney KI, Sherbondy K, et al. Spectrum allocation for noncooperative radar coexistence. IEEE Transactions on Aerospace and Electronic Systems. 2018;54(1):90–105.

[56] Goodman NA, Venkata PR, Neifeld MA. Adaptive Waveform Design and Sequential Hypothesis Testing for Target Recognition With Active Sensors. IEEE Journal of Selected Topics in Signal Processing. 2007;1(1):105–113.

[57] Kreucher C, Hero AO, Kastella K. A Comparison of Task Driven and Information Driven Sensor Management for Target Tracking. In: 44th IEEE Conference on Decision and Control; 2005. p. 4004–4009.

[58] Charlish A, Katsilieris F. Array Radar Resource Management. In: Novel Radar Techniques and Applications: Real Aperture Array Radar, Imaging Radar, and Passive and Multistatic Radar. Institution of Engineering and Technology; 2017. p. 135–171.

[59] Blair WD, Watson GA, Kirubarajan T, et al. Benchmark for radar allocation and tracking in ECM. IEEE Transactions on Aerospace and Electronic Systems. 1998;34(4):1097–1114.

[60] Jung S, Schlangen I, Charlish A. A Mnemonic Kalman Filter for Non-Linear Systems With Extensive Temporal Dependencies. IEEE Signal Processing Letters. 2020;27:1005–1009.

[61] Williams JL. Information Theoretic Sensor Management. Massachusetts Institute of Technology; 2007.