Emission Management for Low Probability Intercept Sensors in Network Centric Warfare

VIKRAM KRISHNAMURTHY
University of British Columbia

Sensor platforms with active sensing equipment such as radars may betray their existence, by emitting energy that can be intercepted by enemy surveillance sensors thereby increasing the vulnerability of the whole combat system. To achieve the important tactical requirement of low probability of intercept (LPI), requires dynamically controlling the emission of platforms. In this paper we propose computationally efficient dynamic emission control and management algorithms for multiple networked heterogenous platforms. By formulating the problem as a partially observed Markov decision process (POMDP) with an on-going multi-armed bandit structure, near optimal sensor management algorithms are developed for controlling the active sensor emission to minimize the threat posed to all the platforms. Numerical examples are presented to illustrate these control/management algorithms.

Manuscript received July 11, 2003; revised March 31 and July 29, 2004; released for publication September 15, 2004.

IEEE Log No. T-AES/41/1/844815.

Refereeing of this contribution was handled by J. P. Y. Lee.

This work was supported by an NSERC grant, and a British Columbia Advance Systems Institute Grant.

Author's address: Dept. of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, V6T 1Z4 Canada, E-mail: (vikramk@ece.ubc.ca).

I. INTRODUCTION

The Joint Vision 2010 [1] is the conceptual template for how the US Armed Forces will achieve dominance across the range of military operations through the application of new operational concepts. One of the fundamental themes underlying the Joint Vision 2010 is the concept of network centric warfare (NCW). The tenets of NCW are [1]: 1) a robustly networked force improves information sharing; 2) information sharing enhances the quality of information and shared situational awareness; 3) shared situational awareness enables collaboration and self-synchronization, and enhances sustainability and speed of command; 4) these, in turn, dramatically increase mission effectiveness.

The information for generating battlespace awareness in NCW is provided by numerous sources, for example, stand-alone intelligence, surveillance, and reconnaissance platforms, sensors employed on weapons platforms, or human assets on the ground. In the fundamental shift to network-centric operations, sensor networks emerge as a key enabler of increased combat power. The operational value or benefit of sensor networks is derived from their enhanced ability to generate more complete, accurate, and timely information than can be generated by platforms operating in stand-alone mode. Networked sensors have several advantages including decreased time to engagement, increased ability to detect low signature targets, improved track accuracy and continuity, improved target detection and identification and reduced sensor detectability to the enemy [10].

We focus here on this reduced sensor detectability aspect of NCW. We present decentralized sensor management algorithms for reducing the detectability of networked sensor platforms to the enemy. Recall that sensor management systems are an integral part of this command and control process in combat systems. Sensor management deals with how to manage, coordinate, and organize the use of scarce and costly sensing resources in a manner that improves the process of data acquisition while minimizing the threat due to radiation of sensors in various platforms. In this paper motivated by NCW applications, we consider the problem of how to dynamically manage and control the emission of active sensors in multiple platforms to minimize the threat posed to these platforms in combat situations. In the defense literature the acronym EMCON is used for emission control. Due to widespread use of sophisticated networked sensor platforms, there is increasing interest in developing a coordinated approach to control their usage to manage the emission and threat levels.

Emission management/control is emerging in importance due to the essential tactical necessity of sensor platforms satisfying a low probability of intercept (LPI) requirement. This LPI requirement is in response to the increase in capability of modern intercept receivers to detect and locate platforms that radiate active sensors. The emission management/control system needs to dynamically plan and react to the presence of an uncertain dynamic battlefield environment. The design of an EMCON system needs to take into account the following subsystems.

1) *Multiple Heterogeneous Networked Platforms of Sensors*: In a typical battlefield environment several sensor platforms are deployed (e.g., track vehicles, unmanned aerial vehicles (UAVs), ground-based radar) each with a variety of sophisticated sensors and weapons. A sensor platform can use both active sensors or passive sensors. Active sensors (e.g., radar) are typically linked with the deployment of weapon systems whereas passive sensors (e.g., sonar, imagers) are often used for surveillance. Typically, when a platform radiates active sensors (e.g., radars), the emission energy from these sensors can be picked up and monitored by the enemy's passive intercept receiver devices such as electronic support measures (ESMs), radar warning receivers (RWRs) and electronics intelligence (ELINT) receivers. These emissions can then betray the existence and location of the platform to the enemy and therefore increase the vulnerability of the platform. Note that different platform sensors provide different levels of quality of service (QoS) depending on the sophistication and accuracy of the sensors.

2) *Threat Evaluator*: The cumulative emission radiated from a platform and detected by enemy sensors directly affects the threat posed to the platform. This threat level posed to a platform can be indirectly measured by the response of the enemy system. A threat level evaluator for each platform consists of local sensors on the platform together with a network of surveillance sensors that monitor the activities of the enemy. Typically these surveillance sensors feed information to an AWACS (airborne warning and control system) aircraft. Based on the activities of the enemy, the combined threat evaluator (which includes both local sensors on the platform as well as a centralized threat evaluator) outputs an observed threat level, e.g., low, medium, or high threat level to each platform.

3) *Sensor Manager*: The sensor manager performs a variety of tasks (see [4] for a comprehensive description). Here we focus on the EMCON functionalities of the sensor manager to maintain an LPI. The sensor manager uses the observed threat level to perform emission control (it switches on or off the platform to decrease the threat level (minimize the emission impact)) and to initiate electronic countermeasures (ECMs) and/or deploy weapons which if successful can decrease the threat level.

The aim of this work is to answer the following question: How should the sensor manager achieve EMCON by dynamically deciding which platforms (or group of platforms) are to radiate active sensors at each time instant in order to minimize the overall threat posed to all the platforms while simultaneously taking into account the cost of radiating these sensors and the QoS they provide? Note that unlike platform centric warfare where scheduling of sensors is carried out within a platform, the above aim is consistent with the philosophy of NCW where given a network of several platforms, the sensor manager dynamically makes a local decision as to which platforms should radiate active sensors.

The main ideas in this paper are summarized as follows.

1) In Section II, we present a stochastic optimization formulation of the EMCON problem. The emission level impact (ELI) of a platform is modelled as a controlled finite state Markov chain and hence the observed threat level is a hidden Markov model (HMM). We then show that the EMCON problem can be naturally formulated as a controlled HMM problem which is also known as a partially observed Markov decision process (POMDP). POMDPs have recently received much attention in the area of artificial intelligence for autonomous robot navigation (see [7] for a nice web-based tutorial). They have also been used for optimal observer trajectory planning in bearings only target tracking (we refer the reader to [5] for an excellent exposition).

2) In general, solving POMDPs are computationally intractable apart from examples with small state and action spaces. In complexity theory [18] they are known as PSPACE hard problems requiring exponential memory and computation. For realistic EMCON problems involving several tens or hundreds of sensor platforms, the POMDP has an underlying state space that is exponential in the number of platforms—which is prohibitively expensive to solve. The main contribution of this paper is to formulate the EMCON problem as a POMDP with a special structure called an on-going multi-armed bandit [13]; see Section III for details. This multi-armed bandit problem structure implies that the optimal EMCON policy can be found by a so-called Gittins index rule, [13, 19]. As a result, the multi-platform EMCON problem simplifies to a finite number of single-platform optimization problems. Hence the optimal EMCON policy is indexable, meaning that at each time instant it is optimal to activate the sensors on the platform (or group of platforms) with the highest Gittins index. There are numerous applications of multi-armed bandit problems in the operations research and stochastic control literature, see [13] and [22].

3) Given the multi-armed bandit POMDP formulation and the indexable nature of the optimal

EMCON policy, the main issue is how to compute the Gittins index for the individual sensor platforms. While there are several algorithms available for computing the Gittins indices for fully observed Markov decision process bandit problems [3], our POMDP bandit problem is more difficult since underlying finite state Markov chain (actual threat level) is not directly observed. Instead the observations (observed threat levels) are a probabilistic function of the unobserved finite state Markov chain. The main contribution of Section IV is to present finite-dimensional algorithms for computing the Gittins index. We show that by introducing the retirement formulation [13] of the multi-armed bandit problem option, a finite-dimensional value iteration algorithm can be derived for computing the Gittins index of a POMDP bandit. The key idea is to extend the state vector to include retirement information.

4) A key feature of the multi-armed bandit formulation is that the EMCON algorithm for selecting which platforms should radiate active sensors can be fully decentralized. In Section V, we present a scalable decentralized optimal EMCON algorithm whose computational complexity is linear in the number of platforms. A suboptimal version of the multi-armed bandit based EMCON algorithm is presented using Lovejoy's approximation [16]. Lovejoy's approach proposed in the operation research literature in 1991 is an ingenious suboptimal method for solving POMDPs; here we adapt it to the multi-armed POMDP. We show how precedence constraints amongst the various sensor platforms can be considered. Also a two-time scale controller that can deal with slowly time-varying parameters is presented.

5) In Section VI numerical examples are presented of the multi-platform EMCON problem. The Gittins index for different types of platforms are computed. The performance of the suboptimal algorithm for computing the Gittins index based on Lovejoy's approximation is also illustrated.

II. MULTI-PLATFORM EMCON PROBLEM

The network centric multi-platform system we consider here consists of three subsystems: networked sensor platforms, a sensor manager which decides which platform (or group of platforms) should radiate active sensors, and a threat evaluator which yields information about the threat posed to the active platform. In this section we formulate probabilistic models for these sub-systems and formulate the EMCON problem as a POMDP. Fig. 1 shows the setup consisting of multiple platforms that are networked with the EMCON and threat evaluator. Actually, the EMCON algorithm we propose in Section V based on the multi-armed bandit theory is decentralized.

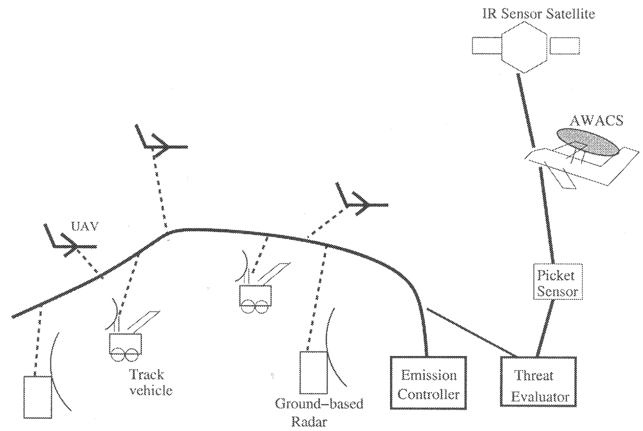

Fig. 1. Schematic setup consisting of 3 types of networked platforms (UAVs, track vehicles, and ground-based radar), Threat evaluator (IR sensor satellite, AWACS, picket sensors) and EMCON. All links shown are bidirectional. Threat level $\{y_k^{(p)}\}$ of platform p is determined by sensors in platform together with central threat evaluator.

Due to detailed modelling given below, it is worthwhile giving a glossary of the important global variables defined in this section that are used throughout this paper.

$p \in \{1,2,\ldots,P\}$ refers to platform p,

$s_k^{(p)}$ is ELI of platform p modelled as a Markov chain,

$A^{(p)}$ is transition probability matrix of $s_k^{(p)}$, see (2),

$u_k \in \{1,\ldots,P\}$ is platform radiating active sensors at time k,

$y_k^{(p)}$ is instantaneous incremental threat posed to platform p at time k, see (4),

$B^{(p)}(m)$ is observation likelihood matrix (7),

$x_k^{(p)}$ is HMM filter state estimate of $s_k^{(p)}$, also called information state, see (15),

$c(i,p)$ is cost of active platform p radiating sensors with ELI $s_k^{(p)} = i$, see (11),

$r(i,p)$ is cost of passive platform p with ELI $s_k^{(p)} = i$, see (11).

A. Heterogeneous Networked Sensor Platforms

Consider P heterogeneous sensor platforms indexed by $p = 1,\ldots,P$. We allow for heterogeneity of the platforms in two ways. First, the individual platforms (e.g., track vehicles, UAVs and ground-based radars) are themselves vastly different in their behaviour, see Fig. 1. Second, each sensor platform can deploy a wide variety of sophisticated flexible passive and active sensors. Active sensors (e.g., radar) are typically linked with the deployment of weapon systems whereas passive sensors (e.g., ESM, ELINT, COMINT (communications intelligence), FLIR (forward-looking infra-red radar), imagers) are often used for surveillance.

We assume that at each time instant only one platform (or group of platforms) is allowed to radiate active sensors and the other $P-1$ platforms can only use passive sensors. This assumption is not restrictive for the following two reasons.

1) Typically in a network of sensor platforms, certain groups of sensor platforms are always operated together. For example, multi-static radars consist of a group of networked distributed sensors. Within this multi-static sensor group, alternately one radar sensor transmits while all of the other distributed networked sensors are used as receivers simultaneously. Another example is a bistatic semi-active homing radar pair that is made up by a target illumination radar and the seeker head of a radar homing missile.

2) Due to the increased threat level posed to a platform that radiates active sensors (because of the possibility of its emission being picked up by enemy passive intercept receiver devices such as ESMs (e.g., RWRS or ELINT receivers)), it is often too risky to simultaneously allow several clusters of platforms to radiate active sensors. Indeed, to keep the overall threat within tolerable levels thus satisfying the LPI requirement, protocols for deploying sensor platforms often impose constraints that only a certain cluster of platforms can use active sensors at a particular time period, see Section VB.

It is the job of the EMCON functionality in the sensor manager to dynamically decide which platform (or group of platforms) should radiate active sensors at each time instant and which platforms can only use passive sensors to minimize the overall threat level posed to all the platforms (active and passive).

B. Emission Level Impact

Let $k = 0,1,2,\ldots,$ denote discrete time. At each time instant k the sensor manager decides which platform to activate. Let $u_k \in \{1,\ldots,P\}$ denote the platform that is activated by the sensor manager at time k. Denote the ELI of platform p at time k as $s_k^{(p)}$. The ELI of platform p is the cumulative received emission registered by the enemy sensors from platform p until time k:

$$s_{k+1}^{(p)} = s_k^{(p)} + e_{k+1}^{(p)}, \qquad p \in \{1,\ldots,P\}. \tag{1}$$

Here, $e_k^{(p)}$ denotes the instantaneous (incremental) emission registered at the enemy from platform p at time k. Note that the ELI is a surrogate measure for the effectiveness of the LPI feature of the sensor platform—the larger the ELI $s_k^{(p)}$, the worse the LPI feature of the sensor platform. Due to the uncertainty in our modelling of how the enemy registers the ELI, $\{e_k^{(p)}\}$ and hence $\{s_k^{(p)}\}$ are assumed to be random processes. Naturally, $e_k^{(p)}$ depends to a large extent on the actual emission originating from the platform p, e.g., $e_k^{(p)}$ is small when the platform does not emit radiation, i.e., $p \neq u_k$. Subsequently, $s_k^{(p)}$ is referred to as the state of platform p.

We assume that the ELI $s_k^{(p)}$ is quantized to a finite set $\{1,2,\ldots,\mathcal{N}_p\}$ where the values in the finite set correspond to physical ELI values, e.g., 1 is low, 2 is medium, and 3 is high.[1] Given that the ELI $s_k^{(p)}$ is finite state and at any time instant k depends on the ELI at the previous time instant (1), it is natural to model the evolution of $\{s_k^{(p)}\}$ probabilistically as a finite state Markov chain. It is clear from (1) that the ELI $s_k^{(u_k)}$ of the platform (or group of sensors) radiating active sensors at time k, evolves with time. The uncertainty (stochasticity) of $s_k^{(u_k)}$ depends largely on how the enemy registers the ELI. The ELI of the platforms that only use passive sensors remain approximately constant since the sensors do not emit energy that can be intercepted by the enemy, i.e, $e_k^{(p)}$ is small when $p \neq u_k$. We idealize this by the following controlled Markov model for the evolution of the ELI $s_k^{(p)}$: If $u_k = p$, the ELI $s_k^{(p)}$ evolves according to an $\mathcal{N}_p$-state homogeneous Markov chain with transition probability matrix

$$A^{(p)} = (a_{ij}^{(p)})_{i,j\in\mathcal{N}_p} = \mathbf{P}(s_{k+1}^{(p)} = j \mid s_k^{(p)} = i)$$
$$\text{if platform } p \text{ radiates active sensors at time } k. \tag{2}$$

The states of all the other $(P-1)$ platforms using passive only sensors are unaffected, i.e., $s_{k+1}^{(p)} = s_k^{(p)}$, if platform p only uses passive sensors at time k, or equivalently

$$A^{(p)} = I \qquad \text{if} \quad p \neq u_k. \tag{3}$$

In the above model (1), since the ELI is cumulative emission registered at the enemy sensors, it follows that the longer the sensors in a platform are active, the more probable its emissions are picked up by the enemy. Thus the quantized ELI $s_k^{(p)}$ in (2), (3) is a nondecreasing controlled Markov process that eventually reaches and remains at the highest level. Of course if our sensor manager exactly knows how the enemy registers the ELI, then $s_k^{(p)}$ would be a nondecreasing controlled deterministic process.

To complete our probabilistic formulation, assume the ELI of all platforms are initialized with prior distributions: $s_0^{(p)} \sim x_0^{(p)}$ where $x_0^{(p)}$ are specified initial distributions for $p = 1,\ldots,P$.

Model for Decreasing ELI: Although not essential, additional flexibility in the ELI model can be introduced by allowing for decreasing ELI as follows. Assume that at each time instant k, because the platform u_k that radiates active sensors incurs maximal risk from the enemy (compared with platforms using passive sensors), the sensor manager also deploys ECMs and possibly weapons to assist this platform.

[1] For convenience, we continue to use $s_k^{(p)}$ for the quantized ELI and $e_k^{(p)}$ for the quantized incremental ELI.

This can reduce the ELI of the platform. Another model is to assume that when a platform deploys weapons (such a platform is considered to be active since usually a platform deploying weapons emits radiation), the ELI can be reduced. In Section IIC (see (6)) we show how deployment of weapons and ECMs can also reduce the threat levels posed to all platforms—not just the active one. We assume here that information exchange between the networked platforms does not add to the ELI of an individual platform.

C. Threat Evaluator

In battlefield environments, the ELI $\{s_k^{(p)}\}$, $p = 1,\ldots,P$, registered by the enemy is not directly available to our sensor manager. We assume that local sensors on each platform p together with a centralized threat evaluation system share information over the network to compute an observed threat level posed to each platform $p = 1,\ldots,P$, which is a probabilistic function of the ELI as described below. The centralized threat evaluation system typically comprises an IR sensor satellite satellite, ground-based picket sensors, surveillance sensor network, and AWACS aircraft that observe the behaviour of the enemy. Fig. 1 shows the schematic setup. For example, if the enemy deploys a radar in the search mode, the observed threat level is typically low; if the enemy radar is in the acquisition mode or track mode, or if the enemy deploys an electronic attack (jamming), the observed threat level is medium. If the enemy commences weapon deployment, (such as precision guided munitions and antiradiation missiles) the observed threat level is high. These are detectable by the threat evaluator which uses warning sensors such as RWRs and IR warning systems that can readily detect the plume of a launched missile [4, pp. 135].

Let $z_k^{(p)}$ denote the observed cumulative threat posed to platform p at time k. Then the process $\{z_k^{(p)}\}$ evolves with time for each platform p as

$$z_{k+1}^{(p)} = z_k^{(p)} + y_{k+1}^{(p)}, \qquad p \in \{1,\ldots,P\} \tag{4}$$

where $y_k^{(p)}$ denotes the observed instantaneous (incremental) threat posed to platform p at time k. Clearly the threat posed to any platform p is a function of the ELI of the platform. Thus it is natural to model the instantaneous threat $y_k^{(p)}$ as a probabilistic function of the instantaneous emission $e_k^{(p)} = s_k^{(p)} - s_{k-1}^{(p)}$ (defined above). For example, one possible model for the instantaneous threat is

$$y_k^{(p)} = s_k^{(p)} - s_{k-1}^{(p)} + t_k^{(p)} + w_k^{(p)} \tag{5}$$

where $t_k^{(p)}$ is a positive valued incremental trend process which could be deterministic, e.g., $t_k^{(p)} = 1$ for all time k, or stochastic, in which case we assume it to be a stationary process that is statistically independent of $w_k^{(p)}$ (defined below) and $s_k^{(p)}$. As a result of the incremental trend process $t_k^{(p)}$, the cumulative threat $z_k^{(p)}$ posed to platform p in (4) typically monotonically increases with time k. For example, choosing $t_k^{(p)} = 1$ for all time k, makes the cumulative trend at time k proportional to k, and this causes the cumulative threat $z_k^{(p)}$ posed to platform p to increase linearly with time. In (5), $w_k^{(p)}$ denotes the observation noise and takes into account several factors such as measurement errors in the surveillance sensors and incomplete knowledge and uncertainty about the enemies behaviour.

A more general example than (5) is to model the instantaneous threat posed to platform p as

$$y_k^{(p)} = s_k^{(p)} - s_{k-1}^{(p)} + t_k^{(p)} + w_k^{(p)} + \delta(u_k - p)f(s_k^{(p)}) - v_k^{(p)} \tag{6}$$

i.e., the cumulative threat $z_k^{(p)}$ in (4) increases faster by some function $f(s_k)$ when the platform p is active, i.e., $u_k = p$, compared with when the platform is passive. In (6), $v_k^{(p)}$ denotes the reduction in threat level due to the deployment of ECMs and/or weapons. We assume that the process $\{v_k^{(p)}\}$ is a stationary Markov chain which is possibly a function of u_k and is statistically independent of $s_k^{(p)}$.[2]

In the sequel, for convenience we refer to the observation process $\{y_k^{(p)}\}$ as the observed threat posed to platform p. Note that observing $\{y_k^{(p)}\}$ is equivalent to observing the cumulative threat $\{z_k^{(p)}\}$ since the former is obtained by taking successive differences of the latter; see (4).

We assume $y_k^{(p)}$ is quantized to a finite set $\{1,2,\ldots,\mathcal{M}_p\}$ where, for example, 1 denotes a small increment, 2 a medium increment, and 3 a large increment in the threat level. The observed threat $y_k^{(p)}$ in (6) is a probabilistic function of the instantaneous emission $e_k^{(p)} = s_k^{(p)} - s_{k-1}^{(p)}$. This probabilistic relationship is summarized by the $(\mathcal{N}_p \times \mathcal{N}_p)$ likelihood matrices $B^{(p)}(1),\ldots,B^{(p)}(\mathcal{M}_p)$,

$$B^{(p)}(m) = (b_{ijm}^{(p)})_{i,j\in\mathcal{N}_p}$$

$$\text{where} \quad b_{ijm}^{(p)} \triangleq \mathbf{P}(y_{k+1}^{(p)} = m \mid s_k^{(p)} = i, s_{k+1}^{(p)} = j) \tag{7}$$

denotes the conditional probability (symbol probability) of the threat evaluator generating an observed threat symbol of m when the instantaneous emission is $e_k^{(p)} = s_{k+1}^{(p)} - s_k^{(p)}$. Notice that if the platform p is inactive, i.e., $p \neq u_k$, then since the

[2] Stationarity of $v_k^{(p)}$ and $t_k^{(p)}$ are required, since we are interested in devising a stationary scheduling policy that optimizes an infinite horizon discounted cost.

emission $e_k^{(p)} = s_k^{(p)} - s_{k-1}^{(p)}$ is zero in (6) it follows that $b_{ijm}^{(p)} = 0$ for $i \neq j$. Thus

$$B^{(p)}(m) = I \qquad \text{if} \quad p \neq u_k. \tag{8}$$

Let $Y_k = (y_1^{(u_0)},\ldots,y_k^{(u_{k-1})})$ denote the observed threat history up to time k. Let $U_k = (u_0,\ldots,u_k)$ denote the sequence of past decisions made by the EMCON functionality of the sensor manager on which platforms radiate active sensors from time 0 to time k.

The above formulation captures the essence of a network centric system—the sensor manager controls different sensors in different platforms. This is in contrast to the older concept of platform centric systems where individual platforms have their own sensor managers that operate independently of other platforms.

D. Network Sensor Manager and Discounted Infinite Horizon Cost

The above probabilistic model for the sensor platform, ELI and threat evaluator together constitute a well-known type of dynamic Bayesian network called an HMM [9]. The problem of state inference of an HMM, i.e., estimating the ELI $s_k^{(p)}$ given (Y_k, U_k) has been widely studied, e.g., see [9]. In this paper we address the deeper and more fundamental issue of how the sensor manager should dynamically decide which platform (or group of platforms) should radiate active sensors at each time instant to minimize a suitable cost function that encompasses all the platforms. Such dynamic decision making based on uncertainty (observed threat levels) transcends standard sensor level HMM state inference which is a well-studied problem.

The EMCON functionality of the sensor manager decides which platform to activate at time k, based on the optimization of a discounted cost function which we now detail. The instantaneous cost incurred at time k due to all the deployed platforms (both active and passive) is

$$C_k = c(s_k^{(u_k)}, s_{k-1}^{(u_k)}, y_k^{(u_k)}, u_k) + \sum_{p\neq u_k} r(s_k^{(p)}, s_{k-1}^{(p)}, y_k^{(p)}, p) \tag{9}$$

where $c(s_k^{(u_k)}, s_{k-1}^{(u_k)}, y_k^{(u_k)}, u_k)$ denotes the cost of radiating active sensors in the platform u_k, and $r(s_k^{(p)}, s_{k-1}^{(p)}, y_k^{(p)}, p)$ denotes the cost of using only passive sensors in platform p. Based on the observed threat history $Y_k = (y_1^{(u_0)},\ldots,y_k^{(u_{k-1})})$, and the history of decisions $U_{k-1} = (u_0,\ldots,u_{k-1})$, the sensor manager needs to decide which sensor platform to activate at time k. The sensor manager decides which platform to activate at time k based on the stationary policy $\mu : (Y_k, U_{k-1}) \to u_k$. Here μ is a function that maps the history of observed threat levels Y_k and past decisions U_{k-1} to the choice of which platform u_k is to radiate active sensors at time k. Let $\mathcal{U}$ denote the class of admissible stationary policies, i.e., $\mathcal{U} = \{\mu : u_k = \mu(Y_k, U_{k-1})\}$. The total expected discounted reward over an infinite time horizon is given by

$$J_\mu = \mathbf{E}\left[\sum_{k=0}^{\infty} \beta^k C_k\right] \tag{10}$$

where $\beta \in (0,1)$ denotes the discount factor, C_k is defined in (9) and $\mathbf{E}$ denotes mathematical expectation. The aim of the sensor manager is to determine the optimal stationary policy $\mu^* \in \mathcal{U}$ which minimizes the cost in (10).

The above problem of minimizing the infinite horizon discounted cost (10) of stochastic dynamical system (2) with noisy observations (7) is a partially observed stochastic control problem. Developing numerically efficient EMCON algorithms to minimize this cost is the subject of the rest of the work presented here.

It is well known, [6, p. 31] that by defining

$$\begin{aligned} c(i,p) &= \sum_{j=1}^{\mathcal{N}_p} \sum_{m=1}^{\mathcal{M}_p} c(i,j,m.p) a_{ij}^{(p)} b_{ijm}^{(p)} \\ r(i,p) &= \sum_{j=1}^{\mathcal{N}_p} \sum_{m=1}^{\mathcal{M}_p} r(i,j,m.p) a_{ij}^{(p)} b_{ijm}^{(p)} \end{aligned} \tag{11}$$

we use the equivalent cost $C_k = c(s_k^{(u_k)}, u_k) + \sum_{p\neq u_k} r(s_k^{(p)}, p)$ in (10) since this has the same expectation as C_k in (9). Therefore, since the ELIs $s_k^{(p)}$ of the passive platforms $p \neq u_k$ remain constant, their cost $r(s_k^{(p)}, p)$ is also constant. Of course the cost $c(s_k^{(u_k)}, u_k)$ of the active platform evolves with time, since $s_k^{(u_k)}$ evolves with time. This property is crucial in our subsequent continuing bandit formulation. Note that the only assumption made in obtaining (11) is the stationarity of the incremental trend $t_k^{(p)}$ and the weapons/ECM effectiveness $v_k^{(p)}$.

E. Examples of Cost Function

Overall threat minimization: If the aim was to minimize the overall threat to all platforms then choosing

$$c(s_k^{(p)}, s_{k+1}^{(p)}, y_k^{(p)}, p) = r(s_k^{(p)}, s_{k+1}^{(p)}, y_k^{(p)}, p) = y_k^{(p)} \qquad p = 1,\ldots,P \tag{12}$$

leads to the infinite horizon cost (10) $J_\mu = \sum_{k=0}^{\infty} \beta^k \sum_{p=1}^{P} \mathbf{E}\{y_k^{(p)}\}$ which is the total discounted cumulative threat posed to all the P platforms.

We now present several other examples of the cost C_k in (9) and (10). For convenience, we classify the cost incurred by a platform radiating active sensors as comprising of 4 components:

$$c(s_k^{(u_k)}, u_k) = -c_0(u_k) + c_1(u_k) + c_2(s_k^{(u_k)}, u_k) - c_3(u_k) \tag{13}$$

while the cost incurred by a sensor using only passive sensors $p \in \{1, 2, \ldots, P\} - \{u_k\}$ comprises of

$$r(s_k^{(p)}, p) = -r_0(p) + r_1(p) + r_2(s_k^{(p)}, p) - r_3(p). \tag{14}$$

The four components in the above costs (13), (14), are described as follows.

1) *Quality of service (QoS)*: $c_0(p)$, $r_0(p)$ denote the QoS of the platform p radiating active sensors and only passive sensors, respectively. Typically this QoS is the average mean square error (covariance) of the estimates provided by the sensors in the platform. Usually, the QoS from radiating active sensors in a platform is much higher than using only passive sensors, i.e., $c_0(p) > r_0(p)$, $p = 1, \ldots, P$. The minus signs in (14), (13), reflect the fact that the lower the QoS the higher the cost and vice versa. Often the platform processes the signals from its sensors. In this case, the QoS of the platform is determined both by the processing algorithm and inherent QoS of the sensor. For example, if a radar is used for a maneuvering target, and an IMM algorithm is used for tracking the target, the target and senor can be modelled as a jump Markov linear system. Estimates of the covariance of the resulting state estimate can be obtained via simulation; see [2]. If the sensor processing algorithm is a Kalman filter, the mean square error is given by the solution of the algebraic Ricatti equation.

2) *Sensor usage cost*: In (13), (14), $c_1(p)$ denotes the usage cost of radiating active sensors in platform p. Usually, the cost of radiating active sensors (e.g., radars) in a platform $c_1(p)$ is much more expensive than the cost of using passive sensors (e.g., sonar and imagers) $r_1(p)$.

3) *Threat and ELI minimization*: To minimize the overall threat as in (12) we can choose c_2 in (13) as the instantaneous threat in (12). Another example is to choose the overall ELI as the cost, i.e., $c_2(s_k^{(u_k)}, u_k) = s_k^{(u_k)}$, $r_2(s_k^{(p)}, p) = s_k^{(p)}$, $p \neq u_k$. Then (10) minimizes the overall discounted ELI of all platforms. Recall that the LPI characteristic of a sensor platform can be measured in terms of its ELI as described earlier.

4) *Defensive capability*: Typically a platform has a number of ECMs and weapons it can deploy. $c_3(p)$ denotes the effectiveness of these countermeasures and weapons the platform p can deploy when it radiates active sensors. $r_3(p)$ denotes the effectiveness of these countermeasures and weapons when the platform only uses passive sensors. The minus sign for $c_3(\cdot)$ and $r_3(\cdot)$ in (13), (14), reflect the fact that the higher the countermeasures and weapons capability of a platform, the lower the cost.

F. Information State Formulation

The above stochastic control problem (10) is an infinite horizon POMDP with a rich structure which considerably simplifies the solution, as is shown later. But first, as is standard with partially observed stochastic control problems, we convert the partially observed multi-arm bandit problem to a fully observed multi-arm bandit problem defined in terms of the information state, see [3] for a complete exposition. Roughly speaking, the idea is to convert a partially observed stochastic control problem (where the state $s_k^{(p)}$ is observed in noise) to a fully observed stochastic problem in terms of the filtered density of the state (called the information state). This filtered density is considered to be fully observed since it is exactly computable given the observations and past decisions. Of course, the information state space is continuous valued since the information state space is a conditional probability. Deriving a finite-dimensional EMCON algorithm on this continuous-valued state space is our main objective.

For each sensor platform p, the information state at time k, which we denote by $x_k^{(p)}$ (column vector of dimension $\mathcal{N}_p$) is defined as the conditional filtered density of the ELI $s_k^{(p)}$ given Y_k and U_{k-1}:

$$x_k^{(p)}(i) \triangleq \mathbf{P}(s_k^{(p)} = i \mid Y_k, U_{k-1}), \qquad i = 1, \ldots, \mathcal{N}_p. \tag{15}$$

The information state can be computed recursively by the HMM state filter (also known as the "forward algorithm" also known as "Baum's algorithm" [12]) as given in (18) below.

Using the smoothing property of conditional expectations, the EMCON cost (10) can be reexpressed in terms of the information state as follows:

$$J_\mu = \mathbf{E}\left[\sum_{k=0}^{\infty} \beta^k \left(c'(u_k) x_k^{(u_k)} + \sum_{p \neq u_k} r'(p) x_k^{(p)}\right)\right] \tag{16}$$

where $c(u_k)$ denotes the $\mathcal{N}_{u_k}$-dimensional reward vector $[c(s_k^{(p)} = 1, u_k), \ldots, c(s_k^{(p)} = \mathcal{N}_{u_k}, u_k)]'$, and $r(p)$ is the $\mathcal{N}_{u_k}$-dimensional reward vector $[r(s_k^{(p)} = 1, p), \ldots, c(s_k^{(p)} = \mathcal{N}_p, p)]'$. The aim of the EMCON problem is to compute the optimal policy $\arg\min_{\mu \in \mathcal{U}} J_\mu$.

In terms of the above information state formulation, the EMCON problem described above can be viewed as the following dynamic scheduling problem. Consider P parallel HMM state filters, one for each sensor platform. The pth HMM filter computes the ELI (state) estimate (filtered density) $x_k^{(p)}$ of the pth platform, $p \in \{1, \ldots, P\}$. At each time instant, only one of the P platforms radiates active sensors, say platform p. Let $y_{k+1}^{(p)}$ be its observed threat level. This is processed by the pth HMM state filter which updates its estimate of the sensor platform's ELI as

$$x_{k+1}^{(p)}(j) = \frac{\sum_{i=1}^{\mathcal{N}_p} a_{ij}^{(p)} b_{ij,y_{k+1}}^{(p)} x_k^{(p)}(i)}{\sum_{i=1}^{\mathcal{N}_p} \sum_{l=1}^{\mathcal{N}_p} a_{il}^{(p)} b_{il,y_{k+1}}^{(p)} x_k^{(p)}(i)}, \quad j = 1,\ldots,\mathcal{N}_p \quad \text{if} \quad p = u_k. \tag{17}$$

Note that due to the dependency of y_k on s_k and s_{k+1}, the above is slightly different to the standard HMM filter. Equation (17) can be written in matrix-vector notation as

$$x_{k+1}^{(p)} = \frac{B^{(p)\prime}(y_{k+1}^{(p)}) \square A^{(p)\prime} x_k^{(p)}}{\mathbf{1}' B^{(p)}(y_{k+1}^{(p)}) A^{(p)\prime} x_k^{(p)}} \qquad \text{if} \quad p = u_k \tag{18}$$

where for $y_{k+1}^{(p)} = m$, $B^{(p)}(m)$ is defined in (7), $\square$ denotes Hadamard product,[3] and 1 is an $\mathcal{N}_p$-dimensional column unit vector. (Note that throughout the paper we use $'$ to denote transpose).

The ELI estimates of the other $P-1$ platforms that use only passive sensors remain unaffected, i.e., since $B^{(q)}(m) = I$ and $A^{(q)} = I$ if $q \neq u_k$ (see (8), (3)), we have

$$x_{k+1}^{(q)} = x_k^{(q)} \qquad \text{if platform } q \text{ only uses passive sensors,} \quad q \in \{1,\ldots,P\}, \quad q \neq p. \tag{19}$$

Let $\mathcal{X}^{(p)}$ denote the state space of information states $x^{(p)}$ for sensor platforms $p \in \{1,2,\ldots,P\}$. That is

$$\mathcal{X}^{(p)} = \{x^{(p)} \in \mathbb{R}^{\mathcal{N}_p} : \mathbf{1}' x^{(p)} = 1,\ 0 < x^{(p)}(i) < 1 \ \text{for all } i \in \{1,\ldots,\mathcal{N}_p\}\}. \tag{20}$$

Note that $\mathcal{X}^{(p)}$ is an $\mathcal{N}_p - 1$-dimensional simplex. We subsequently refer to $\mathcal{X}^{(p)}$ as the information state space simplex of sensor platform p.

In terms of (18), (16) the multi-arm bandit problem reads thus: Design an optimal dynamic scheduling policy to choose which platform to radiate active sensors and hence which HMM Bayesian state estimator to use at each time instant. Note that there is a real-time computational cost of $O(\mathcal{N}_p^2)$ computations associated with running the pth HMM filter.

III. PARTIALLY OBSERVED ON-GOING BANDIT FORMULATION

As it stands the POMDP problem (18), (19), (16) or equivalently (10), (2), (7) has a special structure.

1) Only one Bayesian HMM state estimator operates according to (18) at each time k, or equivalently, only one platform (or group of platforms) radiates active sensors at a given time k. The remaining $P-1$ Bayesian estimates $x_k^{(q)}$ remain frozen, or equivalently, the remaining $P-1$ platforms only operate passive sensors.

2) The platform radiating active sensors incurs a cost depending on its current information state; see (11) and discussion below (11). The costs incurred by platforms using passive only sensors are frozen depending on the state when they were last active.

The above two properties imply that (18), (19), (16) constitute what Gittins [13] terms as an on-going multi-armed bandit. A standard multi-armed bandit formulation [3] would require that the platforms using passive sensors do not incur any cost, i.e., $r(s_k^{(p)},p) = 0$ in (10). Unlike the standard multi-armed bandit, the platforms using passive sensors do incur a cost $r(s_k^{(p)},p)$ making the problem an "on-going" bandit. It turns out that by a straightforward transformation an on-going bandit can be formulated as a standard multi-armed bandit. We quote this as the following result (see [13, p. 32] for a proof).

THEOREM 1 *The ongoing multi-armed bandit problem* (2), (7), (10) *has an identical optimal policy* μ^* *to the following standard multi-armed bandit: dynamics given by* (18), (19) *and only the platform radiating active sensors accrues an instantaneous reward*

$$R(i,u) = -\beta\left(c(i,u) - \sum_{j=1}^{\mathcal{N}_p} a_{ij}^{(u)} r(j,u)\right) \tag{21}$$

so that the discounted reward function to maximize is

$$J_\mu = \mathbf{E}\left\{\sum_{k=0}^{\infty} \beta^k R(s_k^{(u_k)}, u_k)\right\}. \tag{22}$$

Note that in the above theorem, we have, for convenience, made the objective function (22) a reward function (which is simply the negative of a cost function), so maximizing the reward is equivalent to minimizing the cost. We assume in the rest of this paper that the rewards $R(i,p) \geq 0$. If any $R(i,p)$ are negative, simply set $R(i,p) := R(i,p) - \min_{i,p} R(i,p)$ for all i,p, this is always nonnegative. Obviously, subtracting this constant $\min_{i,p} R(i,p)$ from all the rewards does not alter the solution to the EMCON problem, i.e., the optimal policy remains the same.

Finally, for notational convenience, with $R(i,u)$ defined in (21) define the vector

$$R(p) = (R(1,p),\ldots,R(\mathcal{N}_p,p))'. \tag{23}$$

We now summarize the main results of the rest of this paper. It is well known that the multi-armed bandit problem has a rich structure which results in the EMCON optimization (22) decoupling into P independent optimization problems. Indeed, from the theory of multi-armed bandits it follows that the optimal EMCON policy has an indexable rule [22]: for each platform p there is a function $\gamma^{(p)}(x_k^{(p)})$ called the Gittins index, which is only a function of the

[3] For square matrices A, B, C, the Hadamard product $C = A \square B$ has elements $c_{ij} = a_{ij} b_{ij}$.

platform p and the information state $x_k^{(p)}$, whereby the optimal EMCON policy at time k is to activate the platform with the largest Gittins index, i.e.,

$$\text{activate platform } q \text{ where } q = \max_{p\in\{1,\ldots,P\}} \{\gamma^{(p)}(x_k^{(p)})\}. \tag{24}$$

For a proof of this index rule for general multi-arm bandit problems see [22]. Thus computing the Gittins index is a key requirement for solving any multi-arm bandit problem. (For a formal definition of the Gittins index in terms of stopping times, see [13]. An equivalent definition is given in [3] in terms of the parameterized retirement cost M).

REMARKS The indexable structure of the optimal EMCON policy (24) is particularly convenient for the following three reasons.

1) *Scalability*: Since the Gittins index is computed for each platform separately of every other platform (and this is also done off-line), the EMCON problem is easily scalable in that we can handle several hundred platforms. In contrast without taking the multi-armed bandit structure into account, the POMDP has $\mathcal{N}_p^P$ underlying states making it computationally impossible to solve.

2) *Suitability for heterogeneous platforms*: Notice that our formulation of the platform dynamics allows for them to have different transition probabilities and likelihood probabilities. In particular, different platforms can even have different number of threat levels. Moreover, since the Gittins index of platform does not depend on other platforms, we can meaningfully compare different types of platforms. Note that each platform can have a variety of sophisticated sensors—we characterized them above by their overall quality of service.

3) *Decentralized EMCON*: Since the Gittins index of a platform does not depend on other platforms, a fully decentralized EMCON can be implemented as described in Section V with minimal communication overhead between the platforms. Thus the valuable network bandwidth can be used for more important functionalities such as sensor data transfer, etc.

IV. VALUE ITERATION ALGORITHM FOR COMPUTING GITTINS INDEX

To simplify our terminology in this section a platform will be called active if it radiates active sensors, otherwise it will be called passive. The fundamental problem with (24) is that the Gittins index $\gamma^{(p)}(x_k^{(p)})$ for sensor platform p must be evaluated for each $x_k^{(p)} \in \mathcal{X}^{(p)}$, an uncountably infinite set. In contrast, for the standard finite state Markov multi-arm bandit problem considered extensively in the literature (e.g., [13]), the Gittins index can be straightforwardly computed.

In this section we derive a finite-dimensional algorithm for computing the Gittins index $\gamma^{(p)}(x_k^{(p)})$ for each platform $p \in \{1,2,\ldots,P\}$. We formulate the computation of the Gittins index of each platform as an infinite horizon dynamic programming recursion. A value-iteration based optimal algorithm[4] is given for computing the Gittins indices $\gamma^{(p)}(x_k^{(p)})$, for the platforms $p = 1,2,\ldots,P$. Then using the results of this section, in Section V, we use the results in Section IV to solve the EMCON problem.

As with any dynamic programming formulation, the computation of the Gittins index for each platform p is off-line, independent of the Gittins indices of the other $P-1$ platforms and can be done a priori.

For each platform p, let $M^{(p)}$ denote a positive real number such that

$$0 \le M^{(p)} \le \bar{M}^{(p)}, \qquad \bar{M}^{(p)} \triangleq \max_{i\in\mathcal{N}_p} R(s_k^{(p)} = i,\ u_k = p). \tag{25}$$

To simplify subsequent notation, we omit the superscript p in $M^{(p)}$ and $\bar{M}^{(p)}$, and the subindex k in $x_k^{(p)}$. The Gittins index [3], [13] of platform p with information state $x^{(p)}$ can be defined as

$$\gamma^{(p)}(x^{(p)}) \triangleq \min\{M : V^{(p)}(x^{(p)},M) = M\} \tag{26}$$

where $V^{(p)}(x^{(p)},M)$ satisfies the functional Bellman's recursion

$$\begin{aligned} V^{(p)}&(x^{(p)},M) \\ &= \max\left\{R'(p)x^{(p)} + \beta\sum_{m=1}^{\mathcal{M}_p} V^{(p)}\left(\frac{B^{(p)\prime}(m)\Box A^{(p)\prime}x^{(p)}}{\mathbf{1}'_{\mathcal{N}_p}B^{(p)\prime}(m)\Box A^{(p)\prime}x^{(p)}},M\right)\right. \\ &\qquad\left.\times\, \mathbf{1}'_{\mathcal{N}_p}B^{(p)\prime}(m)\Box A^{(p)\prime}x^{(p)},M\right\} \end{aligned} \tag{27}$$

where M denotes the parameterized retirement reward.

The Nth-order approximation of $V^{(p)}(x^{(p)},M)$ is obtained as the following value iteration algorithm $k = 1,\ldots,N$:

$$\begin{aligned} V_{k+1}^{(p)}&(x^{(p)},M) \\ &= \max\left[R'(p)x^{(p)} + \beta\sum_{m=1}^{\mathcal{M}_p} V_k^{(p)}\left(\frac{B^{(p)\prime}(m)\Box A^{(p)\prime}x^{(p)}}{\mathbf{1}'_{\mathcal{N}_p}B^{(p)\prime}(m)\Box A^{(p)\prime}x^{(p)}},M\right)\right. \\ &\qquad\left.\times\, \mathbf{1}'_{\mathcal{N}_p}B^{(p)\prime}(m)\Box A^{(p)\prime}x^{(p)},M\right]. \end{aligned} \tag{28}$$

Here $V_N(x^{(p)},M)$ is the value-function of an N-horizon dynamic programming recursion. Let $\gamma_N^{(p)}(x^{(p)})$ denote the approximate Gittins index computed via the value iteration algorithm (28), i.e.,

$$\gamma_N^{(p)}(x^{(p)}) \triangleq \min\{M : V_N^{(p)}(x^{(p)},M) = M\}. \tag{29}$$

[4]Strictly speaking the value iteration algorithm is near optimal, that is, it yields a value of the Gittins index that is arbitrarily close to the optimal Gittins index. However, for brevity we refer to it as optimal.

It is well known [17] that $V^{(p)}(x^{(p)},M)$ can be uniformly approximated arbitrarily closely by a finite horizon value function $V_N^{(p)}(x^{(p)},M)$ of (28). A straightforward application of this result shows that the finite horizon Gittins index approximation $\gamma_N^{(p)}(x^{(p)})$ of (29) can be made arbitrarily accurate by choosing the horizon N sufficiently large. This is summarized in the following corollary.

COROLLARY 1 *The (infinite horizon) Gittins index $\gamma^{(p)}(x^{(p)})$ of state $x^{(p)}$, can be uniformly approximated arbitrarily closely by the near optimal Gittins index $\gamma_N^{(p)}(x^{(p)})$ computed according to* (29) *for the finite horizon N. In particular, for any $\delta > 0$, there exists a finite horizon $\bar{N}$ such that:*

a) $\sup_{x^{(p)}\in\mathcal{X}^{(p)}} |\gamma_{\bar{N}-1}^{(p)}(x^{(p)}) - \gamma_{\bar{N}}^{(p)}(x^{(p)})| \le \delta$.

b) *For this $\bar{N}$,* $\sup_{x^{(p)}\in\mathcal{X}^{(p)}} |\gamma_{\bar{N}-1}^{(p)}(x^{(p)}) - \gamma^{(p)}(x^{(p)})| \le (2\beta\delta/(1-\beta))$.

Unfortunately, the value iteration recursion (28) does not directly translate into practical solution methodologies. The fundamental problem with (28) is that at each iteration k, one needs to compute $V_k^{(p)}(x^{(p)},M)$ over an uncountably infinite set $x^{(p)} \in \mathcal{X}^{(p)}$ and $M \in [0,\bar{M}]$. The main contribution of this section is to construct a finite-dimensional characterization for the value function $V_k^{(p)}(x^{(p)},M)$, $k = 1,2,\ldots,N$ and hence the near optimal Gittins index $\gamma_N^{(p)}(x^{(p)})$. We show that under a different coordinate basis $V_k^{(p)}(x^{(p)},M)$ can be expressed as a standard POMDP, whose value function is known to be piecewise linear and convex [20]. Then computing $\gamma_N^{(p)}(x^{(p)})$ in (29) simply amounts to evaluating $V_k^{(p)}(x^{(p)},M)$ at the hyper-planes formed by the intersection of the piecewise linear segments. Constructive algorithms based on this finite characterization are given in Section VI to compute the Gittins index for the information states of the original bandit process.

As described in [3, sec. 1.5], M can be viewed as a retirement reward. To develop a structural solution for the Gittins index, we begin by first introducing a fictitious retirement information state. Once the information state reaches this value, it remains there for all time accruing no cost. Define the $(\mathcal{N}_p + 1)$-dimensional augmented information state

$$\bar{x} \in \{[x',0]',[\mathbf{0}'_{\mathcal{N}_p},1]'\} \qquad \text{where} \quad x \in \mathcal{X}^{(p)} \qquad (30)$$

is as in (15). As described below, $\bar{x}_k = [\mathbf{0}'_{\mathcal{N}_p},1]'$ is interpreted as the "retirement" information state. Define an augmented observation process $\bar{y}_k \in \{1,\ldots,\mathcal{M}_p + 1\}$. Here $\mathcal{M}_p + 1$ corresponds to a fictitious observation which when obtained causes the information state to jump to the fictitious retirement state. Define the corresponding $(\mathcal{N}_p + 1) \times (\mathcal{N}_p + 1)$ transition and observation probability matrices as

$$\begin{aligned}
A_1^{(p)} &= \begin{bmatrix} A^{(p)} & \mathbf{0}_{\mathcal{N}_p} \\ \mathbf{0}'_{\mathcal{N}_p} & 1 \end{bmatrix} \\
B_1^{(p)}(m) &= \begin{bmatrix} B^{(p)}(m) & \mathbf{0}_{\mathcal{N}_p} \\ \mathbf{0}'_{\mathcal{N}_p} & 1 \end{bmatrix} \\
B_1^{(p)}(\mathcal{M}_p + 1) &= \begin{bmatrix} \mathbf{0}_{\mathcal{N}_p\times\mathcal{N}_p} & \mathbf{0}_{\mathcal{N}_p} \\ \mathbf{0}'_{\mathcal{N}_p} & 1 \end{bmatrix} \\
A_2^{(p)} &= \begin{bmatrix} \mathbf{0}_{\mathcal{N}_p\times\mathcal{N}_p} & \mathbf{1}_{\mathcal{N}_p} \\ \mathbf{0}'_{\mathcal{N}_p} & 1 \end{bmatrix} \\
B_2^{(p)}(m) &= I_{(\mathcal{N}_p+1)\times(\mathcal{N}_p+1)} \\
& m \in \{1,\ldots,\mathcal{M}_p + 1\}.
\end{aligned} \qquad (31)$$

To construct a finite-dimensional representation of $V^{(p)}(x^{(p)},M)$ we present a coordinate transformation under which $V^{(p)}(x^{(p)},M)$ is the value function of a standard POMDP (denoted $\bar{V}^{(p)}(\cdot)$ below), and $(x^{(p)},M)$ is an invertible map to the information state of this POMDP (denoted ($\pi^{(p)}$ below). To formulate this POMDP we need to express the variable M in (28) as an information state, i.e., express M in a similar form to (18). This can be done by defining the information state z as follows:

$$z \triangleq \begin{bmatrix} M/\bar{M} \\ 1 - M/\bar{M}, \end{bmatrix}, \qquad 0 \le M \le \bar{M}. \qquad (32)$$

Clearly, $0 \le z(1), z(2) \le 1$ and $z(1) + z(2) = 1$—so z can be viewed as an information state. Of course, M in (28) does not evolve, so we need to define a transition probability and observation probability matrix for z which keeps it constant.

Define the information state π and following coordinate transformation (where $\otimes$ denotes Kronecker product[5]):

$$\begin{aligned}
\pi &= z \otimes \bar{x} \\
\bar{A}_1^{(p)} &= I_{2\times2} \otimes A_1 = \begin{bmatrix} A_1^{(p)} & 0 \\ 0 & A_1^{(p)} \end{bmatrix} \\
\bar{A}_2^{(p)} &= I_{2\times2} \otimes A_2^{(p)} = \begin{bmatrix} A_2^{(p)} & 0 \\ 0 & A_2^{(p)} \end{bmatrix} \\
\bar{B}_1^{(p)}(m) &= I_{2\times2} \otimes B_1^{(p)}(m) \\
\bar{B}_2^{(p)}(m) &= I_{2\times2} \otimes B_2^{(p)}(m) \\
\bar{R}_1(p) &= [R'(p) \quad 0 \quad R'(p) \quad 0]' \\
\bar{R}_2(p) &= [\bar{M}\mathbf{1}'_{\mathcal{N}_p} \quad 0 \quad \mathbf{0}'_{\mathcal{N}_p} \quad 0]'.
\end{aligned} \qquad (33)$$

[5]For $m \times n$ matrix A, and $p \times q$ matrix B, the Kronecker product $C = A \otimes B$ is $(mp \times nq)$ with block elements $c_{ij} = a_{ij}B$.

It is easily shown that $\bar{A}_1^{(p)}$, $\bar{A}_2^{(p)}$ are transition probability matrices (their rows add to one and each element is positive) and $\bar{B}_1^{(p)}(m)$, $\bar{B}_2^{(p)}(m)$ are observation probability matrices. Also the $2(\mathcal{N}_p+1)$-dimensional vector $\pi^{(p)}$ is an information state since it belongs to $\Pi^{(p)}$ where

$$\Pi^{(p)} \triangleq \pi : \mathbf{1}'_{2(\mathcal{N}_p+1)}\pi^{(p)} = 1 \qquad \text{and} \qquad \pi^{(p)}(i) \geq 0, \qquad i = 1,2,\ldots,2(\mathcal{N}_p+1). \tag{34}$$

Finally, define the control variable $\nu_k \in \{1,2\}$ at each time k, where ν_k maps π_k to $\{1,2\}$ at each time k. Note $\nu_k = 1$ means continue and $\nu_k = 2$ means retire. Define the policy sequence $\nu = (\nu_1,\ldots,\nu_k)$. (The policy ν is used to compute the Gittins index of platform p. It is not to be confused with the policy μ defined in Section II which determines which platform to activate).

Consider now the following POMDP problem. Parameters $\bar{A}_2^{(p)}, \bar{B}_2^{(p)}, \bar{R}(2)$ defined in (33) form the transition probabilities, observation probabilities, and cost vectors of a two-valued control ($\nu_k \in \{1,2\}$) and objective

$$\max_{\nu} \mathbf{E}\left[\sum_{k=0}^{N} \beta^k \bar{R}'_{\nu_k}(p)\pi_k\right].$$

Here the vector $\pi^{(p)} \in \Pi^{(p)}$ is an information state for this POMDP and evolves according to

$$\pi_{k+1}^{(p)} = \frac{\bar{B}_{\nu_k}^{(p)'}(\bar{y}_{k+1})\Box(\bar{A}_{\nu_k}^{(p)})'\pi_k^{(p)}}{\mathbf{1}'_{2(\mathcal{N}_p+1)}\bar{B}_{\nu_k}^{(p)'}\Box(\bar{A}_{\nu_k}^{(p)})'\pi_k^{(p)}}$$

$$\nu_k \in \{1,2\}, \quad \bar{y}_{k+1} \in \{1,\ldots,\mathcal{M}_p+1\}$$

depending on the control ν_k chosen at each time instant. Note that $\nu_k = 2$ results in π_{k+1} attaining the retirement state $z \otimes [\mathbf{0}_{\mathcal{N}_p} \;\; 1]$.

The value iteration recursion for optimizing this POMDP over the finite horizon N is given by Bellman's dynamic programming recursion [17, eq. 2]

$$\begin{aligned}
&\bar{V}_{k+1}^{(p)}(\pi^{(p)}) \\
&\quad = \max\left[\bar{R}'_1(p)\pi^{(p)} + \beta\sum_{m=1}^{\mathcal{M}_p+1}\bar{V}_k^{(p)}\left(\frac{\bar{B}_1^{(p)'}(m)\Box(\bar{A}_1^{(p)})'\pi^{(p)}}{\mathbf{1}'\bar{B}_1^{(p)'}(m)\Box(\bar{A}_1^{(p)})'\pi^{(p)}}\right)\right. \\
&\qquad \times \mathbf{1}'\bar{B}_1^{(p)'}(m)\Box(\bar{A}_1^{(p)})'\pi^{(p)}, \\
&\qquad \bar{R}'_2(p)\pi^{(p)} + \beta\sum_{m=1}^{\mathcal{M}_p+1}\bar{V}_k^{(p)}\left(\frac{\bar{B}_2^{(p)'}(m)\Box(\bar{A}_2^{(p)})'\pi^{(p)}}{\mathbf{1}'\bar{B}_2^{(p)'}(m)\Box(\bar{A}_2^{(p)})'\pi^{(p)}}\right) \\
&\qquad \left.\times \mathbf{1}'\bar{B}_2^{(p')}(m)\Box(\bar{A}_2^{(p)})'\pi^{(p)}\right], \qquad k = 1,2,\ldots,N \\
&\bar{V}_0^{(p)}(\pi) = \max[\bar{R}'_1(p)\pi^{(p)}, \bar{R}'_2(p)\pi^{(p)}].
\end{aligned} \tag{35}$$

Here $\bar{V}_k^{(p)}(\pi^{(p)})$ denotes the value-function of the dynamic program,

$$\bar{V}_k^{(p)}(\pi) \triangleq \max \mathbf{E}\left[\sum_{t=N-k}^{N} \beta^t \bar{R}'_{\nu_t}(p)\pi_t \mid \pi_{N-k} = \pi\right]. \tag{36}$$

The two terms in the RHS of (35) depict the two possible actions $\nu_k \in \{1,2\}$.

The following is the main result of this section. It shows that the Gittins index can be computed by solving a standard two action POMDP.

THEOREM 2 *Under the coordinate basis defined in* (33)*, the following three statements hold:*

1) *The value function $V_k^{(p)}(x^{(p)}, M)$ in* (28) *for computing the Gittins index is identically equal to the value function $\bar{V}_k^{(p)}(\pi^{(p)})$ of the standard POMDP* (35)*.*

2) *At each iteration k, $k = 0,1,\ldots,N$, the value function $\bar{V}_k^{(p)}(\pi^{(p)})$ is piecewise linear and convex and has the finite-dimensional representation*

$$\bar{V}_k^{(p)}(\pi^{(p)}) = \max_{\lambda_{i,k}\in\Lambda_k^{(p)}} \lambda'_{i,k}\pi^{(p)}. \tag{37}$$

Here the $2(\mathcal{N}_p+1)$-dimensional vectors $\lambda_{i,k}$ belong to precomputable finite set of vectors $\Lambda_k^{(p)}$, see end of Section VA for computational algorithms.

3) *There always exists a unique vector in $\Lambda_k^{(p)}$ which we denote by $\lambda_{1,k} = [\bar{M}\mathbf{1}'_{\mathcal{N}_p} \;\; 0 \;\; \mathbf{0}'_{\mathcal{N}_p} \;\; 0]'$ with optimal control $\nu_k = 2$.*

4) *Denote the elements of each vector $\lambda_{i,k} \in \Lambda_k^{(p)} - \{\lambda_{1,k}\}$ as*

$$\lambda_{i,k} = [\lambda'_{i,k}(1) \quad \lambda_{i,k}(2) \quad \lambda'_{i,k}(3) \quad \lambda_{i,k}(4)]'$$
$$\text{where} \quad \lambda_{i,k}(1), \lambda_{i,k}(3) \in \mathbb{R}^{\mathcal{N}_p}, \quad \lambda_{i,k}(2), \lambda_{i,k}(4) \in \mathbb{R}. \tag{38}$$

Then at time $k = N$, for any information state $x^{(p)} \in \mathcal{X}^{(p)}$ of platform p, the near optimal Gittins index $\gamma_N^{(p)}(x^{(p)})$ is given by the finite-dimensional representation

$$\gamma_N^{(p)}(x^{(p)}) = \max_{\lambda_{i,N}\in\Lambda_N-\{\lambda_{1,N}\}} \frac{\bar{M}\lambda'_{i,N}(3)x^{(p)}}{\bar{M} + (\lambda_{i,N}(3) - \lambda_{i,N}(1))'x^{(p)}}. \tag{39}$$

REMARK Statement 1 of the above theorem shows that the value iteration algorithm (28) for computing the Gittins index $\gamma_k^{(p)}(x^{(p)})$ is identical to the dynamic programming recursion (35) for optimizing a standard finite horizon POMDP. Statement 2 says that the finite horizon POMDP has a finite-dimensional piecewise linear solution which is characterized by a precomputable finite set of vectors at each time instant. Statement 2 is well known in the POMDP literature and is easily shown by mathematical induction. It was originally proved by Sondik [20], see also [17] and [7] for a web-based tutorial. There are several linear programming based algorithms available for computing the finite set of vectors $\Lambda_k^{(p)}$ at each iteration k. Further details are given in Section VI.

Statement 4 with $\lambda_{1,N}$ defined in Statement 3, gives an explicit formula for the Gittins index of the HMM multi-armed bandit problem. Recall $x_k^{(p)}$ is the information state computed by the pth HMM filter at time k. Given that we can compute a set of vectors $\Lambda_N^{(p)}$, (39) gives an explicit expression for the Gittins index $\gamma_N^{(p)}(x_k^{(p)})$ at any time k for platform p. Note if all elements of $R(p)$ are identical, then $\gamma^{(p)}(x^{(p)}) = \bar{M}$ for all $x^{(p)}$.

PROOF The proof of the first statement is by mathematical induction. At iteration $k = 0$,

$$\begin{aligned}\bar{V}_0^{(p)}(\pi) &= \max[\bar{R}_1'(p)\pi^{(p)}, \bar{R}_2'(p)\pi^{(p)}]\\ &= \max[(\mathbf{1}\otimes R(p))'(z\otimes\bar{x}^{(p)}), M] = V_0^{(p)}(x^{(p)}, M).\end{aligned} \tag{40}$$

Assume that at time k, $\bar{V}_k^{(p)}(\pi) = V_k^{(p)}(x^{(p)}, M)$, and consider (35). Our aim is to show that the RHS of (35) is the same as the RHS of (28) which would imply that $\bar{V}_{k+1}^{(p)}(\pi) = V_{k+1}^{(p)}(x^{(p)}, M)$. Note that by construction of the costs in (33), we have for the terminal state

$$\bar{V}_k^{(p)}\left(z\otimes\begin{bmatrix}\mathbf{0}_{\mathcal{N}_p}\\ 1\end{bmatrix}\right) = 0, \qquad k = 0,1,2,\ldots. \tag{41}$$

From (30), and the definitions of π and $\bar{R}_1'(p)\pi^{(p)}$, $\bar{R}_2'(p)\pi^{(p)}$ in (33), it follows that

$$\bar{R}_1'(p)\pi^{(p)} = R(p)'x^{(p)}, \qquad \bar{R}_2'(p)\pi^{(p)} = M. \tag{42}$$

Now consider the terms within the summation of the RHS of (35). Since by our inductive hypothesis, $\bar{V}_k^{(p)}(\pi) = V_k^{(p)}(x^{(p)}, M)$, it is easily shown using standard properties of tensor products (recall $\bar{B}_1^{(p)}(m)$, $\bar{A}_1^{(p)}$ are defined in terms of tensor products in (33)) that for $m = 1,2,\ldots,\mathcal{M}_p$,

$$\begin{aligned}&\bar{V}_k^{(p)}\left(\frac{\bar{B}_1^{(p)'}(m)\square(\bar{A}_1^{(p)})'\pi^{(p)}}{\mathbf{1}'\bar{B}_1^{(p)'}(m)\square(\bar{A}_1^{(p)})'\pi^{(p)}}\right)\mathbf{1}'\bar{B}_1^{(p)'}(m)\square(\bar{A}_1^{(p)})'\pi^{(p)}\\ &\quad = V_k^{(p)}\left(\frac{B^{(p)'}(m)\square A^{(p)'}x^{(p)}}{\mathbf{1}'B^{(p)'}(m)\square A^{(p)'}x^{(p)}}, M\right)\mathbf{1}'B^{(p)'}(m)\square A^{(p)'}x^{(p)}.\end{aligned} \tag{43}$$

Because $\bar{B}_1^{(p)}(\mathcal{M}_p + 1) = \mathrm{diag}(\mathbf{0}_{\mathcal{N}_p}, 1)$, (see (31), (33)), and due to the structure of $\bar{A}_2^{(p)}$ it follows that

$$\begin{aligned}&\bar{V}_k^{(p)}\left(\frac{\bar{B}_1^{(p)'}(\mathcal{M}_p+1)\square(\bar{A}_1^{(p)})'\pi^{(p)}}{\mathbf{1}'\bar{B}_1^{(p)'}\square(\mathcal{M}_p+1)(\bar{A}_1^{(p)})'\pi^{(p)}}\right)\\ &\quad = \bar{V}_k^{(p)}\left(z\otimes\begin{bmatrix}\mathbf{0}_{\mathcal{N}_p}\\ 1\end{bmatrix}\right) = 0 \qquad \forall \quad \pi^{(p)}\in\Pi^{(p)}\\ &\bar{V}_k^{(p)}\left(\frac{\bar{B}_2^{(p)'}(m)\square\bar{A}_2^{(p)})'\pi^{(p)}}{\mathbf{1}'\bar{B}_2^{(p)'}(m)\square(\bar{A}_2^{(p)})'\pi^{(p)}}\right)\\ &\quad = \bar{V}_k^{(p)}\left(z\otimes\begin{bmatrix}\mathbf{0}_{\mathcal{N}_p}\\ 1\end{bmatrix}\right) = 0 \qquad \forall \quad \pi^{(p)}\in\Pi^{(p)}\\ &\qquad\qquad\qquad\qquad\qquad\qquad \forall \quad m\in\{1,\ldots,\mathcal{M}_p+1\}\end{aligned} \tag{44}$$

where $\Pi^{(p)}$ is defined in (34) and the last equality follows from (41). From (42), (43) and (44), it follows that the RHS of (35) is identical to the RHS of (28) implying that $\bar{V}_{k+1}^{(p)}(\pi) = V_{k+1}^{(p)}(x^{(p)}, M)$.

The third statement follows from (35) and the fact that $\bar{V}_k^{(p)}(\pi)$ is piecewise linear and convex. Indeed, from (35), $\bar{V}_{k+1}^{(p)}(\pi) = \max$[piecewise linear segments in $\pi, \bar{R}_2'(p)\pi$] and hence $\bar{R}_2(p) = [\bar{M}\mathbf{1}'_{\mathcal{N}_p}\ 0\ \mathbf{0}'_{\mathcal{N}_p}\ 0]'$ is one of the elements in $\Lambda_{N+1}^{(p)}$.

The fourth statement can be shown as follows:

$$\begin{aligned}\bar{V}_N^{(p)}(\pi^{(p)}) &= \max_{\lambda_{i,N}\in\Lambda_N^{(p)}} \lambda_{i,N}'\pi^{(p)}\\ &= \max\left\{\lambda_{1,N}'\pi^{(p)}, \max_{\lambda_{i,N}\in\Lambda_N^{(p)}-\{\lambda_{1,N}\}}\lambda_{i,N}'\pi^{(p)}\right\}.\end{aligned}$$

Substituting $\lambda_{N,1}'\pi = M$ yields

$$\bar{V}_N^{(p)}(\pi^{(p)}) = \max\left\{M, \max_{\lambda_{i,N}\in\Lambda_N^{(p)}-\{\lambda_{1,N}\}}\lambda_{i,N}'\pi^{(p)}\right\}. \tag{45}$$

From (29), and the statement 1 of the theorem, the Gittins index is $\gamma_N^{(p)}(x^{(p)}) = \min\{M : \bar{V}_N^{(p)}(\pi^{(p)}) = M\}$. With the aim of computing $\gamma_N^{(p)}(x^{(p)})$, let us examine more closely the set $\{M : \bar{V}_N^{(p)}(\pi^{(p)}) = M\}$. From (45), and using the fact that $\max(a,b) = a \Rightarrow b \le a$ for the second equality below yields

$$\begin{aligned}&\{M : \bar{V}_N^{(p)}(\pi^{(p)}) = M\}\\ &\quad = \left\{M : \max\{M, \max_{\lambda_{i,N}\in\Lambda_N^{(p)}-\{\lambda_{1,N}\}}\lambda_{i,N}'\pi^{(p)}\} = M\right\}\\ &\quad = \left\{M : \max_{\lambda_{i,N}\in\Lambda_N^{(p)}-\{\lambda_{1,N}\}}\lambda_{i,N}'\pi^{(p)} \le M\right\}.\\ &\gamma_N^{(p)}(x^{(p)}) = \min\{M : \bar{V}_N^{(p)}(\pi^{(p)}) = M\}\\ &\quad = \left\{M : \max_{\lambda_{i,N}\in\Lambda_N^{(p)}-\{\lambda_{1,N}\}}\lambda_{i,N}'\pi^{(p)} = M\right\}\\ &\quad = \left\{M : \max_{\lambda_{i,N}\in\Lambda_N^{(p)}-\{\lambda_{1,N}\}}\lambda_{i,N}'\begin{bmatrix}(M/\bar{M})\bar{x}^{(p)}\\ (1-M/\bar{M})\bar{x}^{(p)}\end{bmatrix} = M\right\}\end{aligned} \tag{46}$$

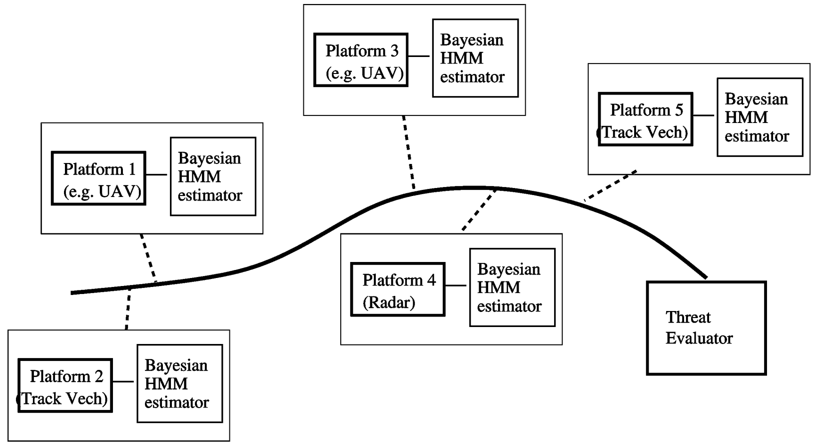

Fig. 2. Decentralized EMCON for networked platforms. Each platform has Bayesian estimator to compute information state $x_k^{(p)}$. Gittins index $\gamma(x_l^{(p)})$ is transmitted via network to other platforms. Platform with largest Gittins index activates its sensors. All links are bidirectional.

where in the last equality above we have used (33) to substitute $\pi^{(p)} = [M/\bar{M}, 1 - M/\bar{M}]' \otimes \bar{x}^{(p)}$. Let M_i, $i = 2, \ldots, |\Lambda_N^{(p)}|$ denote the solution of the $|\Lambda_N^{(p)}| - 1$ algebraic equations

$$\lambda_{i,N}' \begin{bmatrix} (M/\bar{M})\bar{x}^{(p)} \\ (1 - M/\bar{M})\bar{x}^{(p)} \end{bmatrix} = M.$$

Using the structure of $\lambda_{i,N}$ in (38) to solve the above equation for M_i yields

$$M_i = \frac{\bar{M}\lambda_{i,N}'(3)x^{(p)}}{\bar{M} - (\lambda_{i,N}(3) - \lambda_{i,N}(1))'x^{(p)}}.$$

Then (46) yields $\gamma_N^{(p)}(x^{(p)}) = \max\{M_2, \ldots, M_{|\Lambda_N^{(p)}|}\}$ which is equivalent to (39).

V. DECENTRALIZED SCALABLE EMCON ALGORITHM FOR MULTIPLE PLATFORMS

In the previous section we showed that the Gittins index for each platform p, can be computed by solving a POMDP associated with platform p. Thus instead of solving a POMDP comprising of $\mathcal{N}_1 \times \cdots \times \mathcal{N}_P$ states and P actions (which would be the brute force solution), due to the bandit structure, we only need to solve P independent POMDPs, each comprising of $2(\mathcal{N}_p + 1)$ states and 2 actions. This makes the EMCON problem tractable. However, it should be noted that even with the bandit formulation, solving a $2(\mathcal{N}_p + 1)$ state POMDP can still be expensive for large $\mathcal{N}_p$. As mentioned in Section I, POMDPs are PSPACE hard problems—in worst case the number of vectors in Λ_k can grow exponentially with k.

In this section we outline the multi-armed bandit based EMCON algorithm, describe a decentralized implementation, present a suboptimal algorithm to compute the Gittins index based on Lovejoy's approximation, and finally describe how precedence constraints for the various sensor platforms can be taken into account.

Fig. 2 shows the setup and optimal solution. The EMCON algorithm consists of P Bayesian state inference filters (HMM filters), one for each sensor platform. Suppose that sensor platform 1 is the optimal platform to radiate active sensors at time $k-1$, i.e., $u_{k-1} = 1$. The HMM filter 1 receives observed threat level $y_k^{(1)}$ of platform 1 from the threat evaluator and updates the filtered density (information state) $x_k^{(1)}$ of the ELI of platform 1 according to the HMM filter (18). The corresponding Gittins index of this state is computed using (39). For the platforms using passive sensors, their ELI and thus their information states remain unchanged ($x_k^{(2)} = x_{k-1}^{(2)}$, $x_k^{(3)} = x_{k-1}^{(3)}$) and hence the Gittins indices $\gamma^{(2)}(x_k^{(2)})$, $\gamma^{(2)}(x_k^{(3)})$ remain unchanged. The Gittins indices of the states at time k of the P platforms are then compared. The multi-armed bandit theory then specifies that the optimal choice u_k at time k is to radiate active sensors in the platform with the smallest Gittins index as shown in Fig. 2.

A. Optimal EMCON Algorithm

The complete EMCON algorithm based on the multi-armed bandit theory of the previous section is given in Algorithm 1; see also Fig. 2.

ALGORITHM 1 *Algorithm for Real-Time EMCON*

Input for each platform $p = 1, \ldots, P$:
$A^{(p)}$ *{ELI transition probability matrix}*, $B^{(p)}$ *{Observation threat likelihood matrix}*, $R(p)$ *{Reward vector}*,
$x_0^{(p)}$ *{A priori state estimate at time 0}*, N *{Horizon size (large)}*, β *{discount factor}*
Off-line Computation of Gittins indices: *Compute finite set of vectors* $\Lambda_N^{(p)}$

for $p = 1,\ldots,P$ **do**
compute $\Lambda_N^{(p)}$ *according to Section IV*
end
Initialization: *At time* $k = 0$ *compute* $\gamma_N^{(p)}(x_0^{(p)})$ *according to* (39).

Real Time EMCON over horizon N
while *time* $k < N$ **do**
$\{$*Radiate active sensors on platform q with largest Gittins index.*$\}$
Activate platform $q = \min_{p\in\{1,\ldots,P\}}\{\gamma^{(p)}(x_k^{(p)})\}$ *(see* 24*)*
Obtain threat level measurement $y_{k+1}^{(q)}$
Update ELI estimate of qth platform using the HMM filter (18)

$$x_{k+1}^{(q)} = \frac{B^{(q)'}(m)\Box A^{(q)'}x_k^{(q)}}{\mathbf{1}'B^{(q)'}(m)\Box A^{(q)'}x_k^{(q)}}$$

$\gamma_N^{(q)}(x_{k+1}^{(q)}) = \min_{\lambda_{i,N}\in\Lambda_N^{(q)}} \lambda_{i,N}'(x_{k+1}^{(q)} \otimes x_{k+1}^{(q)})$ *according to* (39)
$\{$*For other* $P-1$ *platforms* $p = 1,\ldots,P,\ p \neq q$, *ELI estimates remain unchanged*$\}$
$\gamma_N^{p}(x_{k+1}^{(p)}) = \gamma_N^{(p)}(x_k^{(p)})$
$k = k+1$
end.

Real-Time Computation Complexity: Given that the vector set $\Lambda_N^{(p)}$ is computed off-line, the real time computations required in the above algorithm at each time k are:

1) computing the HMM estimate $x_{k+1}^{(q)}$ (18) which involves $O(\mathcal{N}_p^2)$ computations,

2) computing $\gamma_N^{(q)}(x_{k+1}^{(q)})$, (39) requires $O(|\Lambda_N^{(p)}|\mathcal{N}_p^2)$ computations.

Given the finite-dimensional representation of the Gittins index in (39) of Theorem 2, there are several linear programming based algorithms in the POMDP literature such as Sondik's algorithm, Monahan's algorithm, Cheng's algorithm [17], and the Witness algorithm [7] that can be used to compute the finite set of vectors $\Lambda_N^{(p)}$ depicted in (37). In the numerical examples below we used the "incremental-prune" algorithm recently developed in the artificial intelligence community by Cassandra, et al. in 1997 [8, 6] (the C++ code can be freely downloaded from the website [7]).

B. Precedence Constraints

In many cases, there are precedence constraints as to when the active sensors of the platform currently only using passive platform can be activated depending on the ELI of the currently active platform. If the estimated ELI of the active platform is high, it may be required to continue to keep the platform active until this estimated ELI is reduced to medium by using ECMs and weapons, after which another platform can be activated.

For example, suppose that an active platform can only be made passive if the probability of the estimated ELI level being high is smaller than δ, $0 < \delta < 1$. Such precedence constraints are easily taken into account in the multi-armed bandit formulation as follows: see [13] for details.

With the ELI $s_k^{(p)} \in \{\texttt{low},\texttt{medium},\texttt{high}\}$, the component $x_k^{(q)}(3)$ of the active platform's information state defined in (18) denotes the probability that the ELI is high given all the available information. Then in the above EMCON algorithm 1, continue with the current platform radiating active sensors until time τ where $\tau = \arg\min_{t>k} x_\tau^{(q)}(t)$. At this time τ, compare the Gittins indices of the various platforms according to Algorithm 1 to decide on the next platform to activate.

C. Decentralized EMCON Protocol

Algorithm 1 can be implemented in completely decentralized fashion as follows (note that we do not take into account network delays due to communication between platforms). Assuming that any time instant k, every platform stores the P-dimensional vector (u_k,γ), where u_k denotes the platform radiating active sensors at time k, and γ is the vector of Gittins indices of the $P-1$ platforms that use passive sensors, arranged in descending order, i.e.,

$$\begin{aligned}\gamma &= \sigma(\gamma^{(p)}(x_k^{(p)}), p = 1,\ldots,P, p\neq u_k)\\ &= (\gamma^{(\sigma(1))}(x_k^{(\sigma(1))}), \gamma^{(\sigma(2))}(x_k^{(\sigma(2))}),\ldots,\gamma^{(\sigma(P-1))}(x_k^{(\sigma(P-1))})).\end{aligned}$$

Here $\sigma(\cdot)$ denotes the permutation operator on the set of platforms $\{1,2,\ldots,P\} - \{u_k\}$ so that they are ordered according to decreasing order of Gittins index, i.e., at time k, $\gamma^{(\sigma(1))}(x_k^{(\sigma(1))})$ is the passive platform with the highest Gittins index, while $\gamma^{(\sigma(P-1))}(x_k^{(\sigma(P-1))})$ is the passive platform with the lowest Gittins index.

At time k, the platform radiating active sensors receives observed threat level $y_{k+1}^{(u_k)}$ and updates $x_{k+1}^{(u_k)}$ locally using the Bayesian HMM filter as described in Algorithm 1.

If $\gamma^{(u_k)}(x_{k+1}^{(u_k)}) \geq \gamma^{(\sigma(1))}(x_k^{(\sigma(1))})$ then set $k = k+1$, $u_{k+1} = u_k$, i.e., continue with the same active platform.

Else if $\gamma^{(u_k)}(x_{k+1}^{(u_k)}) < \gamma^{(\sigma(1))}(x_k^{(\sigma(1))})$ then platform u_k broadcasts $\gamma^{(u_k)}(x_{k+1}^{(u_k)})$ over the network and shuts off its active sensors.

On receiving this broadcast, platform $\sigma(1)$ (which has the highest Gittins indices of all the passive platforms) activates its sensors.

All the platforms update the vector (u_{k+1},γ) where now $\gamma^{(u_k)}(x_{k+1}^{(u_k)}$ is one of the elements of γ.

The above implementation is completely decentralized and requires minimal communication overheads (bandwidth) over the network. The platform currently radiating active sensors only broadcasts its Gittins index over the network when it is less than that of another platform, thus signifying that the platform will shut down its active sensors and the new platform will activate its sensors. In particular, note that the platforms radiating passive sensors never need to broadcast their Gittins indices over the network. This minimal communication overhead of only 1 broadcast when the active platform is changed, saves the network bandwidth for other important functionalities of the sensor manager.

D. Suboptimal Algorithm based on Lovejoy's Approximation

In general the number of linear segments that characterize the $\bar{V}_k^{(p)}(\pi)$ of (36) and hence the Gittins indices $\gamma_N^{(p)}(\cdot)$ can grow exponentially; indeed the problem is PSPACE hard (i.e., involves exponential complexity and memory). In 1991 Lovejoy proposed an ingenious suboptimal algorithm for POMDPs. Here we adapt it to computing the Gittins index of the POMDP bandit. It is obvious that by considering only a subset of the piecewise linear segments that characterize $\bar{V}_k^{(p)}(\pi)$ and discarding the other segments, one can reduce the computational complexity. This is the basis of Lovejoy's [16] lower bound approximation. Lovejoy's algorithm [16] operates as follows: Initialize $\bar{\Lambda}'_0 = \Lambda_0^{(p)}$, i.e., according to (40).

Step 1 Given a set of vectors $\bar{\Lambda}_k^{(p)}$, construct the set $\tilde{\Lambda}_k^{(p)}$ by pruning $\bar{\Lambda}_k^{(p)}$ as follows. Pick any R points, $\pi_1,\pi_2,\ldots,\pi_R$ in the information state simplex $\Pi^{(p)}$. (In the numerical examples below we picked the R points based on a uniform Freudenthal triangularization of $\Pi^{(p)}$, see [16] for details). Then set $\tilde{\Lambda}_k^{(p)} = \{\arg\min_{\lambda\in\bar{\Lambda}_k^{(p)}} \lambda'\pi_r,\ r = 1,2,\ldots,R\}$.

Step 2 Given $\tilde{\Lambda}_k^{(p)}$, compute the set of vectors $\bar{\Lambda}_{k+1}^{(p)}$ using a standard POMDP algorithm.

Step 3 $k \to k+1$.

Notice that $\tilde{V}_k^{(p)}(\pi) = \max_{\lambda\in\tilde{\Lambda}_k^{(p)}} \lambda'\pi$ is represented completely by R piecewise linear segments. Lovejoy [16] shows that for all k, $\bar{V}_k^{(p)}(\pi)$ is a lower bound to the optimal value function $V_k^{(p)}(\pi)$, i.e., $V_k^{(p)}(\pi) \geq \bar{V}_k^{(p)}(\pi)$ for all $\pi \in \mathcal{P}$. Lovejoy's algorithm gives a suboptimal EMCON scheduling policy at a computational cost of no more than R evaluations per iteration k. Lovejoy [16] also provides a constructive procedure for computing an upper bound to $\limsup_{\pi\in\mathcal{P}} |V_k^{(p)}(\pi) - \bar{V}_k^{(p)}(\pi)|$. In Section VI it is shown that Lovejoy's approximation yields excellent performance.

E. Two-Time Scale EMCON

So far, we have assumed that the parameters (a priori ELI estimate $x_0^{(p)}$, transition probabilities $A^{(p)}$ for ELI, threat observation probabilities $B^{(p)}$, transition probabilities for trend process $\{t_k^{(p)}\}$ and weapons effectiveness $\{v_k^{(p)}\}$, statistics of the noise process $w_k^{(p)}$, and costs) remain constant over time. For convenience, let us group all these parameters into a vector denoted θ. Based on the assumption that θ is constant over time, we presented in previous sections a bandit formulation for computing the stationary policy for an infinite horizon discounted cost. We now consider the case where θ evolves with time but on a time scale much slower than signals $s_k^{(p)}$, $y_k^{(p)}$, $t_k^{(p)}$, $v_k^{(p)}$, $w_k^{(p)}$, etc. We use the notation θ_k to denote this time-evolving parameter vector. The time variation reflects practical situations where the parameters in a battlefield situation are quasi-stationary either due to changing circumstances or as a result of other functionalities operating in the sensor manager. It also allows us to consider cases where the multi-armed bandit assumptions hold over medium length batches of time. For example instead of requiring that the ELI $s_k^{(p)}$ remain constant when platform p only uses passive sensors, we can allow $s_k^{(p)}$ to vary as a slow Markov chain with transition probability matrix $I + \epsilon Q$ where ϵ is a small positive constant defined below.

Using stochastic averaging theory [15, 21] a two-time scale EMCON algorithm can be designed as outlined below. The intuitive idea behind stochastic averaging is this: on the fast time scale, the slowly time-varying parameter θ_k can be considered to be a constant, and the multi-armed bandit solution proposed in previous section applies. On the slow time scale, the variables evolving over the fast time scale behaves according to their average, and it suffices for the slow time scale controller to control this average behaviour. The result presented below is based on weak convergence two-time scale arguments in [14, ch. 5]; we refer the reader to [14] for technical details.

We start with the following average cost problem: Compute $\inf_\mu J_\mu^T$ where

$$\mathcal{J}_\mu^T = \mathbf{E}\left\{\frac{1}{T}\sum_{k=1}^{T} C_k(\theta_k)\right\}. \tag{47}$$

Here T is a large positive integer, and $C_k(\theta_k)$ is defined as in (9) except that it now depends on a slowly time-varying parameter θ_k. Note that (47) can be rewritten as

$$\mathcal{J}_\mu^\epsilon = \mathbf{E}\left\{\epsilon\sum_{k=1}^{1/\epsilon} C_k(\theta_k)\right\} \tag{48}$$

where $\epsilon = 1/T$ is a suitable small positive constant. In stochastic control taking $\epsilon \to 0$, or equivalently

$T \to \infty$, yields the so-called "infinite horizon average cost" problem. We consider such a formulation below. It is important to note that unlike a discounted cost problem, the existence of an optimal stationary control for an average cost POMDP problem requires assumptions on the ergodicity of the information state $x_k^{(p)}$; see [11]. We do not dwell on these technicalities here.

Next assume that the quasi-stationary parameter vector θ_k evolves slowly according to

$$\theta_{k+1} = \theta_k + \epsilon h(\theta_k, n_k) \tag{49}$$

where the step size $\epsilon > 0$ is the same as the ϵ in (48), n_k denotes a random ergodic process (it can model a noisy observation or a supervisory control that controls the evolution of the parameters). $h(\cdot,\cdot)$ is assumed to be a uniformly bounded function.

The following is the main result. For any $x \in \mathbb{R}$, let $\lfloor x \rfloor$ denote the largest integer $\leq x$.

THEOREM 3 *With $T = \lfloor 1/\epsilon \rfloor$ and $\{\theta_k\}$ evolving according to* (49)*, the average cost* (48) *in the limit as $\epsilon \to 0$ behaves as follows:*

$$\mathcal{J}_\mu \stackrel{\Delta}{=} \lim_{\epsilon\to 0} \mathcal{J}_\mu^\epsilon = \lim_{T_1\to\infty} \frac{T_1}{T} \sum_{\tau=1}^{\lfloor T/T_1 \rfloor} J_{\mu_\tau}(\theta_{\tau T_1}). \tag{50}$$

Here $\tau = 1,2\ldots,\lfloor T/T_1 \rfloor$ denotes the index of batches of length T_1, where $T_1/T \to 0$ and $T_1 \to \infty$, $J_{\mu_\tau}(\theta_{\tau T_1})$ is defined as in (10) *with frozen parameter $\theta_{\tau T_1}$ over the batch of length T_1. Indeed,*

$$\begin{aligned} J_{\mu_\tau}(\theta_{\tau T_1}) &= \lim_{T_1\to\infty} \mathbf{E}\left\{ \frac{1}{T_1} \sum_{k=\tau T_1}^{(\tau+1)T_1 - 1} C_k(\theta_{\tau T_1}) \right\} \\ &= \lim_{T_1\to\infty} \lim_{\beta\to 1} (1-\beta) \mathbf{E}\left\{ \sum_{k=\tau T_1}^{(\tau+1)T_1 - 1} \beta^{(k-\tau T_1)} C_k(\theta_{\tau T_1}) \right\}. \end{aligned} \tag{51}$$

Thus, the optimal policy $\inf_\mu \mathcal{J}_\mu$ *decomposes into a sequence of individual bandit problems:*

$$\inf_\mu \mathcal{J}_\mu = \{\inf_{\mu_1} J_{\mu_1}(\theta_{T_1}), \inf_{\mu_2} J_{\mu_2}(\theta_{2T_1}), \ldots, \inf_{\mu_\tau} J_{\mu_\tau}(\theta_{\tau T_1}), \ldots\}. \tag{52}$$

The above theorem says the following. Suppose we decompose the entire time length T into batches, each of size T_1. The condition $T \to \infty$, $T_1 \to \infty$ but $T_1/T \to 0$, means that the batch size T_1 grows to infinity, but T grows to infinity much faster so that the number of batches $\lfloor T/T_1 \rfloor$ are still infinite. Such a condition is typical in two-time scale stochastic control [14, 15]. For example. choose $T_1 = \sqrt{T}$ or more generally $T_1 = T^\alpha$, $0 < \alpha < 1$. Under this condition, the theorem states that the slowly time-varying parameter θ_k can be replaced over each batch $k \in [\tau T_1, \ldots, (\tau+1)T_1 - 1]$ by the frozen parameter $\theta_{\tau T_1}$, where τ denotes the index of the batch. Equation (50), and the first equality of (51) give an explicit representation of how the average cost $\mathcal{J}_\mu^\epsilon$ as $\epsilon \to 0$ can be decomposed into the sum of averaged costs each over batch length T_1. Equation (50) is proved in [14] using weak convergence techniques. The second equality in (51) says that the average cost over the τth batch of length T_1 is equivalent to a discounted infinite horizon cost obtained by setting the discount factor β close to 1. This is a well-known result, it simply relates a discounted average to an arithmetic average [3]. Finally, (50) says that on the slow time scale, it suffices simply to consider the average behaviour, namely $J_{\mu_\tau}(\theta_{\tau T_1})$ of the fast time scale, where $J_{\mu_\tau}(\theta_{\tau T_1})$ is explicitly defined in (51). As a result, (52) says that the average cost problem (47) decomposes into $\tau = 1,2,\ldots,\lfloor T/T_1 \rfloor$ bandit problems that can be solved sequentially; solving the τth bandit problem with parameter θ_τ yields the optimal EMCON policy $\inf_{\mu_\tau} J(\theta_{\tau T_1})$.

From a practical point of view, i.e., for finite but large T, this leads to the following two-time scale EMCON algorithm. Suppose T is a large positive integer, $T_1 = \sqrt{T}$, $\epsilon = 1/T$.

ALGORITHM 2 *Algorithm for Two-Time Scale EMCON Control*

Step 1 *Update parameters on slow time scale as* $\theta_{(\tau+1)T_1} = \theta_{\tau T_1} + \epsilon \sum_{k=\tau T_1}^{(\tau+1)T_1 - 1} h(\theta_k, n_k)$*; see* (49)*. Here n_k can be a supervisory control.*

Step 2 *Use Algorithm* 1 *to compute optimal EMCON policy μ_τ.*

Set $\tau = \tau + 1$ and go to Step 1.

VI. NUMERICAL EXAMPLES

Here we present numerical examples that illustrate the performance of the optimal and suboptimal EMCON algorithms presented in Section V. When the ELI $s_k^{(p)}$ of each platform evolves according to a two state Markov chain, the Gittins index of each platform can be graphically illustrated meaning that a complete discussion of the algorithm behaviour can be given. For this reason, in this section we consider two state ELIs. For higher state examples, while the Gittins indices and hence optimal and suboptimal EMCON algorithms can be straightforwardly computed, it is not possible to graphically plot and visualize the Gittins indices.

Scenario and Parameters: The ELI[6] $s_k^{(p)} \in$ {`low`,`high`} of each platform is modeled as a two state Markov chain, i.e., $\mathcal{N}_p = 2$. In all cases, each platform has access to the threat evaluator and possibly data

[6]We suitably abuse notation here for clarity. More precisely, $s_k^{(p)} \in \{1,2\}$ where 1 denotes `low` and 2 denotes `high`. Similar notational abuse is used for $y_k^{(p)}$ and the platform index $p \in$ {`Track`,`Radar`} in this section.

from other sensors to evaluate the threat level posed to each platform. The observed incremental threat $y_k^{(p)} \in \{=, +, -\}$, i.e., $\mathcal{M}_p = 3$. $y_k^{(p)}$ is a noisy function of the incremental ELI $s_k^{(p)} - s_{k-1}^{(p)}$ see (4), where "=" means that the cumulative threat $z_k^{(p)}$ increases linearly with time, "+" means that $z_k^{(p)}$ increases faster than linearly with time and "−" means that $z_k^{(p)}$ increases slower than linearly with time. (Note that as described in Section IIC), the cumulative threat levels of all platforms can be modelled to increase with time).

The combat scenario below involves several platforms (possibly up to several tens or few hundreds) belonging to two types as outlined below.

1) *Armoured Track Vehicle Group*: Each platform consists of a group of vehicles (e.g., armored personnel carriers, tanks, armored recovery vehicles).
Parameters: (see (2), (7) for definition of A, $B(\cdot)$)

$$A^{(\texttt{Track})} = \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix}$$

$$B^{(\texttt{Track})}(=) = \begin{bmatrix} 0.8 & 0.05 \\ 0.05 & 0.8 \end{bmatrix}$$

$$B^{(\texttt{Track})}(+) = \begin{bmatrix} 0.1 & 0.9 \\ 0.05 & 0.1 \end{bmatrix}$$

$$B^{(\texttt{Track})}(-) = \begin{bmatrix} 0.1 & 0.05 \\ 0.9 & 0.1 \end{bmatrix}$$

$$c_0(\texttt{Track}) = 40, \qquad c_1(\texttt{Track}) = 45$$

$$c_2(\texttt{Track}, \texttt{low}) = 10, \qquad c_2(\texttt{Track}, \texttt{high}) = 40$$

$$c_3(\texttt{Track}) = 40, \qquad r_0(\texttt{Track}) = 20$$

$$r_1(\texttt{Track}) = 20, \qquad r_2(\texttt{Track}, \texttt{low}) = 5$$

$$r_2(\texttt{Track}, \texttt{high}) = 10, \quad r_3(\texttt{Track}) = 20$$

implying that the reward vector, see (23) is $R(\texttt{Track}) = (10.80, -15.75)'$.
The transition probability 0.5 means that if the track vechicle group has a high ELI, the weapons and ECM are effective in mitigating the ELI to low with probability 0.5. The $(1,1)$ elements of the three $B^{(\texttt{Track})}$ matrices model the observed threat probabilities given that $s_k = s_{k+1} = \texttt{low}$ (i.e., the ELI is constant)—meaning that with 0.8 probability a "=" observation is obtained when the ELI is constant. Similarly for (2,2) elements which consider $s_k = s_{k+1} = \texttt{high}$. Finally, the $(1,2)$ elements model the observation probabilities given $s_k = \texttt{low}$, $s_{k+1} = \texttt{high}$—meaning that with 0.95 probability a "+" observation is obtained when the ELI increases.
Active sensor: mobile medium range 3D radar (e.g., Raytheon Firefinder radar) which yields high QoS ($c_0 = 40$) but is expensive to use (c_1) and high emission impact $c_2(,\cdot,\texttt{high})$.
Passive sensors: imagers, information from other platforms, command and control, and threat evaluation system.
The weapons effectiveness $c_3 = 40$ is high when active sensors are radiated. The track vehicle group can typically deploy missiles, anti-aircraft weapons, artillery launchers, etc. The weapons effectiveness when passive sensors are operated is lower ($r_3 = 20$).

2) *Ground-Based Sensor Platform*: The platform has:
Parameters:

$$A^{(\texttt{Radar})} = \begin{bmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{bmatrix}$$

$$B^{(\texttt{Radar})}(=) = \begin{bmatrix} 0.95 & 0.025 \\ 0.025 & 0.95 \end{bmatrix}$$

$$B^{(\texttt{Radar})}(+) = \begin{bmatrix} 0.025 & 0.95 \\ 0.025 & 0.025 \end{bmatrix}$$

$$B^{(\texttt{Radar})}(-) = \begin{bmatrix} 0.025 & 0.025 \\ 0.95 & 0.025 \end{bmatrix}$$

$$c_0(\texttt{Radar}) = 62, \qquad c_1(\texttt{Radar}) = 60$$

$$c_2(\texttt{Radar}, \texttt{low}) = 14.5, \qquad c_2(\texttt{Radar}, \texttt{high}) = 60$$

$$c_3(\texttt{Radar}) = 60, \qquad r_0(\texttt{Radar}) = 38$$

$$r_1(\texttt{Radar}) = 35, \qquad r_2(\texttt{Radar}, \texttt{low}) = 5$$

$$r_2(\texttt{Radar}, \texttt{high}) = 15, \quad r_3(\texttt{Radar}) = 40.$$

Hence the reward vector (see (23)) is $R(\texttt{Radar}) = (11.25, -28.80)'$.
Active sensor: Multi-function radar providing surveillance, acquisition, tracking, discrimination, fire control support, and kill assessment.
Passive sensors: ELINT, information from other platforms and command and control.

The high QoS c_0 in the active mode (due to powerful nature of radar) is counter-balanced by the high usage cost c_1 (due to human operators, strategic importance of radar). The ELI in the active mode is high ($c_2(\texttt{Radar}, \texttt{high}) = 60$). The ground-based radar has powerful support weapons and ECM both in the active and passive mode ($c_3(\texttt{Radar}), r_3(\texttt{Radar})$). The transition probability of 0.6 reflects the fact that the ECM and weaponry is quite effective in mitigating the ELI from high to low.
Throughout we chose the discount factor $\beta = 0.9$ in the discounted cost function (10).

Procedure: Note that in a typical network of sensors there are several platforms of each of the above types, for example, say 20 groups of armoured track vechicles and 3 ground-based radars. Then the total number of platforms is 23. Without the multi-armed bandit approach, the resulting POMDP would involve 2^{23} states which is computationally

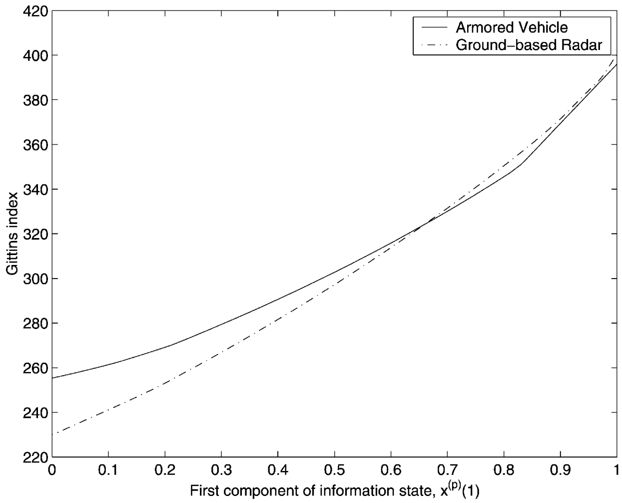

Fig. 3. Gittins indices for 2 types of platforms.

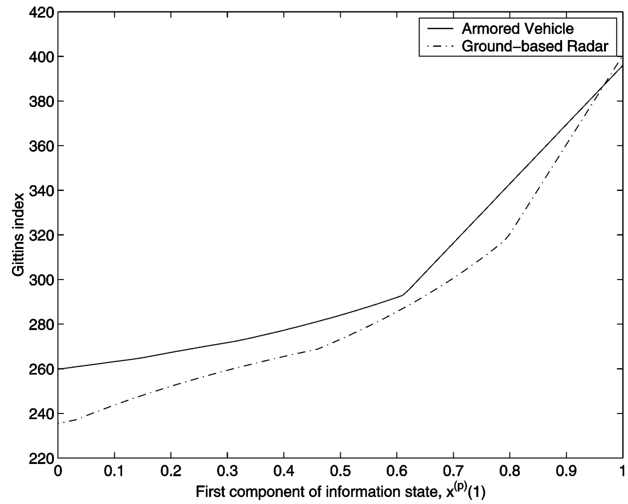

Fig. 4. Approximate Gittins indices computed using Lovejoys's approximation with triangulation $R = 3$ for 2 types of platforms.

intractable. Due to the multi-armed bandit structure, computing the Gittins index which yields the optimal scheduling solution only requires solving 2 POMDPs (since there are only 2 types of platforms) each with 6 states (since each POMDP has $2(\mathcal{N}_p + 1)$ states, see (34)). The various steps of the EMCON Algorithm 1 of Section V are implemented as follows.

1) *Off-line Computation of Gittins Index*: The Gittins indices of the 2 types of platforms were computed as follows. First, $\min_{i,p} R(i,p) = -28.8$ was subtracted from all $R(i,p)$ to make the rewards nonnegative; see discussion above (23).

Then we used the POMDP program from the website [7] to compute the set of vectors $\Lambda_N^{(\texttt{Track})}$, $\Lambda_N^{(\texttt{Radar})}$, for horizon $N = 20$. The POMDP program allows the user to choose from several available algorithms. We used the "Incremental Pruning" algorithm developed by Cassandra, et al. in 1997 [8]. This is currently one of the fastest known algorithms for solving POMDPs; see [7] for details.

A numerical resolution of $\epsilon = 10^{-2}$ yields 1129 vectors for $\Lambda_{20}^{(\texttt{Track})}$ (requiring 3773 s) and 2221 vectors for $\Lambda_{20}^{(\texttt{Radar})}$ (requiring 22472 s). Using these computed vectors $\Lambda_N^{(\texttt{Track})}$, $\Lambda_N^{(\texttt{Radar})}$, the Gittins index for the two types of platforms $\gamma_N^{(\texttt{Track})}(x)$, $\gamma_N^{(\texttt{Radar})}$ computed using (39) are plotted in Fig. 3. (Because $\mathcal{N}_p = 2$, and $x(1) + x(2) = 1$, it suffices to plot $\gamma_N^{(1)}(x)$ versus $x(1)$.)

2) *Real Time EMCON*: After computing $\Lambda_N^{(\cdot)}$ as described above, for all the platform, HMM filters can be implemented as outlined in Algorithm 1.

Lovejoy's Suboptimal Algorithm: Although the above computation of the Gittins indices is off-line—it takes substantial computational time. This motivates the use of Lovejoy's suboptimal algorithm of Section VD to compute the Gittins indices. For an $R = 3$ point uniform triangularization of the information state space (lower dashed line), Fig. 4 shows the computed Gittins indices. $\tilde{\Lambda}_{20}^{(\texttt{Track})}$ has 14 vectors and $\tilde{\Lambda}_{20}^{(\texttt{Radar})}$ has 16 vectors. The total computational time which includes Step 1 (see Section VD implemented in Matlab is less than 15 s.

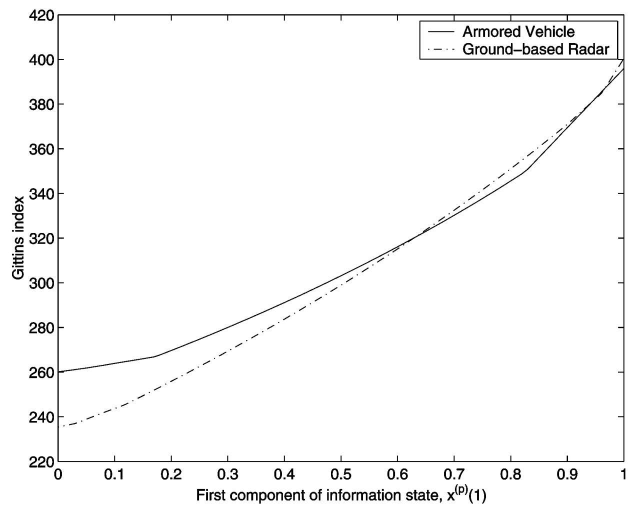

Fig. 5. Approximate Gittins indices computed using Lovejoys's approximation with triangulation $R = 5$ for 2 types of platforms.

For an $R = 5$ point uniform triangularization of the information state space (lower dashed line), Fig. 4 shows the computed Gittins indices, Fig. 5 shows the computed Gittins indices. $\tilde{\Lambda}_{20}^{(\texttt{Track})}$ has 51 vectors and $\tilde{\Lambda}_{20}^{(\texttt{Radar})}$ has 55 vectors. The total computational time is approximately 300 s.

By comparing Fig. 4 and Fig. 5 with Fig. 3, it can be seen that Lovejoy's lower bound algorithm provides an excellent estimate of the Gittins index with a relatively low computational complexity. For $R \geq 7$ (not shown), Lovejoy's algorithm yields estimates of $\gamma^{(p)}(x)$ which are virtually indistinguishable from the solid lines.

Numerical experiments not presented here show that for problems with platforms having up to 5 ELI levels, the incremental prune algorithm and Lovejoy's lower bound algorithm can be satisfactorily used.

REMARK The C++ code for implementing the POMDP value iteration algorithm was downloaded from [7]. The Matlab code for computing the Gittins indices and implementing Lovejoy's algorithm are freely available from the author at vikramk@ece.ubc.ca.

VII. CONCLUSION

We have presented EMCON algorithms for networked heterogeneous multiple platforms. The aim was to dynamically regulate the emission from the platforms to satisfy an LPI requirement. The problem was formulated as a POMDP with an on-going multi-armed bandit structure. Such bandit problems have an indexable (decomposable) solutions. A novel value iteration algorithm was proposed for computing these Gittins indices.

As shown in Section V, the main advantage of the above POMDP multi-armed bandit formulation is the scalability and decentralized nature of the resulting EMCON algorithm. With minimum communication overheads over the network, the platforms can dynamically regulate their emission and hence decrease their threat levels. As a result the bandwidth in the network can be utilized for other important functionalities in NCW. It is important to note that this paper has addressed only one aspect of NCW, namely EMCON. In future work we will consider hierarchical bandits for other aspects of NCW. For large scale problems the multi-armed bandit formulation (or approximation) appears to be the only feasible methodology for designing computationally feasible algorithms. We assumed here that the network over which the platforms exchange information does not have random delays and communication of the network does not increase the risk posed to a platform. In future work it is worthwhile considering these aspects in the design of scheduling algorithms in NCW.

REFERENCES

[1] Department of Defense
Network Centric Warfare: Department of Defense Report to U.S. Congress. Mar. 2001.
http://www.defenselink.mil/nii/NCW/.

[2] Bar-Shalom, Y., and Li, X. R.
Multitarget Multisensor Tracking: Principles and Techniques.
Storrs, CT: YBS Publishing, 1995.

[3] Bertsekas, D. P.
Dynamic Programming and Optimal Control, Vol. 1 and 2.
Belmont, MA: Athena Scientific, 1995.

[4] Blackman, S., and Popoli, R.
Design and Analysis of Modern Tracking Systems.
Dedham, MA: Artech House, 1999.

[5] Le Cadre, J. P., and Trémois, O.
Bearings-only tracking for maneuvering sources.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 1 (Jan. 1998), 179–193.

[6] Cassandra, A. R.
Exact and approximate algorithms for partially observed Markov decision process.
Ph.D. dissertation, Brown University, Providence, RI, 1998.

[7] Cassandra, A. R.
Tony's POMDP page.
http://www.cs.brown.edu/research/ai/pomdp/index.html.

[8] Cassandra, A. R., Littman, M. L., and Zhang, N. L.
Incremental pruning: A simple fast exact method for partially observed Markov decision processes.
In *Proceedings of 13th Annual Conference on Uncertainty in Arficial Intelligence (UAI-97)*, Providence, RI, 1997.

[9] Ephraim, Y., and Merhav, N.
Hidden Markov processes.
IEEE Transactions on Information Theory, **48** (June 2002), 1518–1569.

[10] Gagnon, G.
Network-centric special operations—exploring new operational paradigms.
Air and Space Power Chronicles, Feb. 2002, http://www.airpower.maxwell.af.mil/.

[11] Hernández-Lerma, O., and Laserre, J. B.
Discrete-Time Markov Control Processes: Basic Optimality Criteria.
New York: Springer-Verlag, 1996.

[12] James, M. R., Krishnamurthy, V., and LeGland, F.
Time discretization of continuous-time filters and smoothers for HMM parameter estimation.
IEEE Transactions on Information Theory, **42**, 2 (Mar. 1996), 593–605.

[13] Gittins, J. C.
Multi-armed Bandit Allocation Indices.
New York: Wiley, 1989.

[14] Kushner, H. J.
Weak Convergence and Singularly Perturbed Stochastic Control and Filtering Problems.
Boston: Birkhauser, 1990.

[15] Kushner, H. J., and Yin, G.
Stochastic Approximation Algorithms and Applications.
New York: Springer-Verlag, 1997.

[16] Lovejoy, W. S.
Computationally feasible bounds for partially observed Markov decision processes.
Operations Research, **39**, 1 (Jan.–Feb. 1991), 162–175.

[17] Lovejoy, W. S.
A survey of algorithmic methods for partially observed Markov decision processes.
Annals of Operations Research, **28** (1991), 47–66.

[18] Papadimitriou, C. H.
Computational Complexity.
Reading, MA: Addison-Wesley, 1995.

[19] Ross, S.
Introduction to Stochastic Dynamic Programming.
San Diego, CA: Academic Press, 1983.

[20] Smallwood, R. D., and Sondik, E. J.
Optimal control of partially observable Markov processes over a finite horizon.
Operations Research, **21** (1973), 1071–1088.

[21] Solo, V., and Kong, X.
Adaptive Signal Processing Algorithms—Stability and Performance.
Englewood Cliffs, NJ: Prentice Hall, 1995.

[22] Whittle, P.
Multi-armed bandits and the Gittins index.
Journal of the Royal Statistical Society B, **42**, 2 (1980), 143–149.

Vikram Krishnamurthy (S'90—M'91—SM'99—F'05) was born in 1966. He received his bachelor's degree in electrical engineering from the University of Auckland, New Zealand in 1988, and Ph.D. from the Australian National University, Canberra in 1992. Since 2002, he has been a professor and Canada research chair at the Department of Electrical Engineering, University of British Columbia, Vancouver, Canada. Prior to this he was a chaired professor at the Department of Electrical and Electronic Engineering, University of Melbourne, Australia. His research interests span several areas including stochastic scheduling and network optimization, biological nanotubes, statistical signal processing and wireless telecommunications.

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