

Adaptive Multimodality Sensing of Landmines

Lihan He, Shihao Ji, *Member, IEEE*, Waymond R. Scott, Jr., *Member, IEEE*, and Lawrence Carin, *Fellow, IEEE*

Abstract—The problem of adaptive multimodality sensing of landmines is considered based on electromagnetic induction (EMI) and ground-penetrating radar (GPR) sensors. Two formulations are considered based on a partially observable Markov decision process (POMDP) framework. In the first formulation, it is assumed that sufficient training data are available, and a POMDP model is designed based on physics-based features, with model selection performed via a variational Bayes analysis of several possible models. In the second approach, the training data are assumed absent or insufficient, and a lifelong-learning approach is considered, in which exploration and exploitation are integrated. We provide a detailed description of both formulations, with example results presented using measured EMI and GPR data, for buried mines and clutter.

Index Terms—Lifelong learning, multimodality landmine detection, partially observable Markov decision process.

I. INTRODUCTION

THERE ARE many sensing challenges for which the use of an unmanned autonomous sensing platform is desirable, vis-à-vis using humans to deploy the sensors by hand. One important application that fits this profile is ground-based sensing of landmines [1]. This problem represents a significant challenge for an autonomous agent that must control the platform position, while also deciding which of the possible multiple sensors to deploy. This challenge is exacerbated by the heterogeneous characteristics of the environment that may be encountered [2]. For example, there are many different types of landmines (metal, plastic, small, and large), and these multiple mines appear differently as sensed by typical sensors; ground-penetrating radar (GPR) and electromagnetic induction (EMI) sensors constitute the principal tools that are applied for handheld landmine detection [1]. The GPR and EMI signatures of landmines and clutter are also a strong function of the soil characteristics [3]–[8], which are heterogeneous and changing as a function of water content [2] (the electric and magnetic properties of soils are a strong function of the moisture content, which is locally varying and typically poorly known in practice).

For the problem considered here, we assume that GPR and EMI sensors are deployed on the same unmanned platform.

The task is for this system to navigate autonomously through a minefield, with the goal of detecting landmines, and doing so with a low false-alarm rate. The sensing “agent” must decide where to move the platform, which sensor (GPR or EMI) to deploy at a given point, and when to declare that a landmine is present or not. This task must be performed within a sensing budget, which is defined by the cost of deploying a sensor as well as the costs associated with making particular declarations (e.g., declaring the presence of a mine or clutter); as described in the section that follows, the cost associated with making classifications is performed within a Bayes-risk setting. To our knowledge, this paper is the first attempt at autonomous multimodality landmine detection; in practice, such sensing is currently performed by humans deploying hand-held sensors or via manned and unmanned sensor-mounted vehicles (with human control in both cases) [1].

The basic objective may be cast in the form of an adaptive sensor-management problem [9], [10] (here with two sensors, the GPR and EMI sensors), with the problem complicated significantly by the complexities of the landmine and clutter signatures and the dependence of such on (poorly known) environmental conditions. Here, we consider a partially observable Markov decision process (POMDP) formalism [11]. In the POMDP formulation, the environment under test is assumed to reside within a particular state s_E , and this state is not observable directly; the state of the environment, defined by the presence/absence of a mine in the region being sensed, is unchanged by the sensing itself. The state s_E is “partially” observable in the form of the measured sensor data. The agent has particular actions at its disposal, here, characterized by the opportunity to move to a new location, deploy either of the two (GPR and EMI) sensors, or classify a given region (make an inference with regard to s_E). Each of these actions has an expected immediate cost, as well as an impact on the long-term sensing cost. The POMDP constitutes a framework that balances the (discounted) infinite-horizon performance of this multisensor problem, i.e., it accounts for the immediate expected cost, as well as discounted future costs, over an infinite horizon [11].

The POMDP is employed to constitute a sensing policy, defining the optimal next action to take based upon the agent’s current belief about the environment under test [11]. The belief is defined in terms of a belief state, a probability mass function (pmf) that reflects the probability $p(s_E)$ for all environmental states in the set S_E [i.e., $p(s_E)$ for all $s_E \in S_E$], based upon all previous actions and observations [11]. To compute for the belief state, one requires a model of the environment under test [11]. For the work of interest here, the necessity of an underlying model is a serious limitation for the reasons previously discussed: The specific types of mines and clutter that may

Manuscript received May 10, 2006; revised February 12, 2007.

L. He, S. Ji, and L. Carin are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: lihan@ece.duke.edu; shji@ece.duke.edu; lcarin@ece.duke.edu).

W. R. Scott, Jr. is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: waymond.scott@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2007.894933

be encountered are typically unknown *a priori*, and even if these were known, the associated sensor signatures are a strong function of the soil properties, which are generally unknown and may change with variable weather.

Nevertheless, if we assume that we have access to measured data from the GPR and EMI sensors for targets and soils of interest, we may design the required statistical models through which a POMDP policy may be realized. As demonstrated in the succeeding text, such models are well characterized in terms of hidden Markov models (HMMs) [12], [13] with action-dependent state-transition probabilities. In this application, the set of target states associated with the HMM are defined by sensor positions that are relative to the target, and the sequence of target states visited is modeled as a Markov process, which is conditioned on the sensor-platform motion. In this setting, we must distinguish the overarching state of the environment under test $s_E \in S_E$ with the hidden underlying states of the target model (HMM), the latter defined by the set S_T . The set of environment states S_E is defined by the absence or presence of a landmine, while the set of target states S_T describes the hidden position of the sensor relative to the item under test; these distinctions are addressed more completely in Section III.

Given a set of GPR and EMI data, measured at a sequence of spatial positions relative to the target, we must now develop the underlying HMMs required for the POMDP. Issues that must be addressed include defining an appropriate set of HMM states; we must also define the appropriate number of codes [14] for quantization of the observations such that the observations are discrete, as required by the POMDP. To address these problems, we employ a variational-Bayes (VB) HMM analysis [15], [16], which yields a full posterior density function on the HMM parameter values. In addition, the VB formulation allows us to evaluate the “evidence” for each model type (defined by the number of states and codes) [17], from which the proper number of (data driven) states and codes may be defined.

Rather than assuming that we know which particular landmines, clutter and environmental (soil) conditions are under interrogation, the underlying POMDP model may be constituted to account for the full range of uncertainty with regard to these parameters [18]. Stated more precisely, the set of possible environments, mines and clutter, defined by S_E , may in principle be made large enough to cover the full range of known conditions that may be encountered. As the agent interrogates the actual environment with the multiple sensors, the belief state narrows down the conditions under test to those actually under interrogation (the belief state, defined by $p(s_E)$ for all $s_E \in S_E$, settles upon which environment under test is most probable). In addition, a new action is introduced with appropriate cost, which is characterized by calling an “oracle” [18] to reveal a label for an item under interrogation, thus allowing the underlying POMDP model to expand with the introduction of new mines, clutter, and/or environmental (soil) conditions. The use of an oracle is deemed appropriate for the landmine-sensing application, since oracle deployment entails excavation of the item under test, to determine whether it is a landmine or a clutter. There is a cost (for example in time) associated with such oracle deployment, and this cost is addressed within the algorithm.

It is computationally expensive to perform a framework of the type previously summarized (the number of environmental states in the set S_E must grow to account for the full range of possible mines, clutter, and soil conditions). To address this issue, an approximate algorithm has been proposed [18] in which the full distribution on target and environmental conditions is sampled to constitute a finite set of possible environments. As the environment is sensed, these models are pruned (analogous to a narrowing of the belief state, as previously discussed), and new models are introduced in their place through exploitation of the oracle [18]. In this paper, we adopt a modified form of this framework [18], with specific application to the landmine-sensing problem. As demonstrated in the examples, we use this approach to address multisensor (GPR and EMI) interrogation of a minefield, with no *a priori* knowledge assumed with regard to the mines, clutter, and soil conditions. This is termed “lifelong learning,” because the algorithm (agent) continually learns and refines its policy as it interacts with the environment.

In this paper, we first develop a POMDP formulation based on the (unrealistic) assumption that *a priori* and adequate training data are available for model development. This solution is used as a comparison for the “lifelong-learning” algorithm in which an oracle is employed, and the algorithm learns about its environment as it is sensed. Here, we employ measured GPR and EMI data for real mines and realistic clutter. To be specific, measured data were collected using actual EMI and GPR sensors on three different minefields (composed of inert mines), where the minefields were characterized by natural and man-made clutter, as well as landmines that are both plastic and metal. The measured data considered in this paper are available upon request, and therefore, it is hoped that it will evolve to a standard data set that researchers may use to test different adaptive sensor-management algorithms.

II. PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

In this section, we introduce POMDP basics, assuming that the underlying POMDP model is known, and in Section III, we discuss how the model may be learned based upon the training data. In Section IV, this is generalized further by assuming that the proper model is unknown, with model learning that is performed adaptively while sensing the environment (“lifelong” learning).

A POMDP model is represented by a six-element tuple $\{S, A, T, \Omega, O, R\}$ [11], where S is a finite set of discrete states, A is a finite set of discrete actions, and Ω is a finite set of discrete observations. The state-transition probability

$$T(s, a, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a) \quad (2.1)$$

describes the probability of transitioning from state s to state s' when taking action a . The observation function

$$O(a, s', o) = \Pr(o_{t+1} = o | a_t = a, s_{t+1} = s') \quad (2.2)$$

describes the probability of sensing observation o after taking action a and transiting to state s' . Finally, the reward function

$R(s, a)$ represents the immediate expected reward the agent receives by taking action a in state s .

Since the state is not observed directly, a belief state b is introduced. The belief state is a probability distribution over all states, representing the agent's probability of being in each of the states based on past actions and observations, assuming access to the correct underlying model. The belief state is updated by Bayes rule after each action and observation, based on the previous belief state

$$b_t(s') = \frac{1}{c} O(a, s', o) \sum_{s \in S} T(s, a, s') b_{t-1}(s) \quad (2.3)$$

with the normalizing constant

$$\begin{aligned} c &= \sum_{s' \in S} O(a, s', o) \sum_{s \in S} T(s, a, s') b_{t-1}(s) \\ &= \Pr(o|a, b). \end{aligned} \quad (2.4)$$

A POMDP policy is a mapping from belief states to actions, telling the agent which action to take based on the current belief state. The goal of the POMDP is to find an optimal policy by maximizing the expected discounted reward

$$V = E \left[\sum_{t=0}^{k-1} \gamma^t R(s_t, a_t) \right] \quad (2.5)$$

which is accrued over a horizon of length k ; in (2.5), the expectation is taken with respect to the states visited. The discount factor $\gamma \in (0, 1]$ describes the degree to which future rewards are discounted relative to immediate rewards. If k is finite, the optimal action depends on the distance from the horizon, and therefore, the policy is termed nonstationary. However, often, an appropriate k is not known, so we may consider an infinite-horizon policy, i.e., k goes to infinity for which we require $\gamma < 1$. An infinite horizon also implies a stationary policy that is independent of the agent's temporal position. When in belief state b , the maximum expected reward k steps from the horizon is expressed as

$$V^{(k)}(b) = \max_{a \in A} \left[\sum_s R(s, a) b(s) + \gamma \sum_o p(o|a, b) V^{(k-1)}(b_a^o) \right] \quad (2.6)$$

where b_a^o is the belief state after the agent takes action a and observes o , as updated in (2.3). The $V^{(k)}(b)$ represents the maximum expected discounted reward the agent will receive if it is in belief state b and takes actions according to the optimal policy for future steps. In this paper, policy design is performed using the point-based value iteration (PBVI) algorithm, with details provided in [19].

III. POMDP MODEL FOR LANDMINE DETECTION

The discussion in Section II assumed that the underlying POMDP model was available for subsequent policy design.

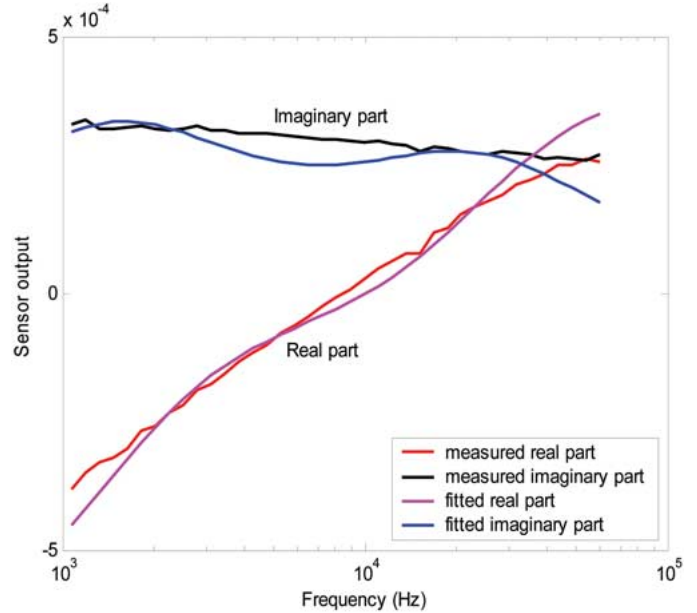


Fig. 1. EMI response and model fit when the sensor is above a metal mine.

We now discuss how the model is generated for the landmine-sensing problem of interest here, assuming labeled multisensor data are available. Model construction involves defining S , A , Ω , and R and estimating the probabilities T and O .

A. Feature Extraction

We assume the EMI and GPR sensors reside on an autonomous platform (robot), and either an EMI or a GPR measurement may be made at any point. If appropriate, both types of measurements may be made sequentially. It is also assumed that the observed data are converted into associated features; the features are quantized using vector quantization [14], yielding the finite set of observations required for the POMDP.

1) *EMI Features*: The EMI measurements are performed in the frequency domain. A typical frequency-domain EMI response for the magnetic field $H(\omega)$ above a metal mine is shown in Fig. 1, where ω represents the angular frequency. The magnetic field induced by a metal target is represented as [20]

$$H(\omega) \propto a_1 + \frac{b_1 \omega}{\omega - j\omega_1} + \frac{b_2 \omega}{\omega - j\omega_2} \quad (3.1)$$

where a_1 , b_1 , and b_2 are related to the magnetic dipole moments of the target, and ω_1 and ω_2 represent the associated EMI resonant frequencies.

Features can be extracted from an EMI observation by fitting the measured data $Y(\omega)$ to the model in (3.1), assuming additive noise n in the observation, i.e., $Y(\omega) = H(\omega) + n$. The parameters $\{a_1, b_1, b_2, \omega_1, \omega_2\}$ are our EMI features, obtained via maximum-likelihood fitting under the assumption that n is an independently and identically distributed Gaussian noise [i.e., minimizing the mean-square error between the measured data and the model in (3.1)].

2) *GPR Features*: The GPR data for a given sensor position are assumed to be recorded in the time domain. Fig. 2(a)

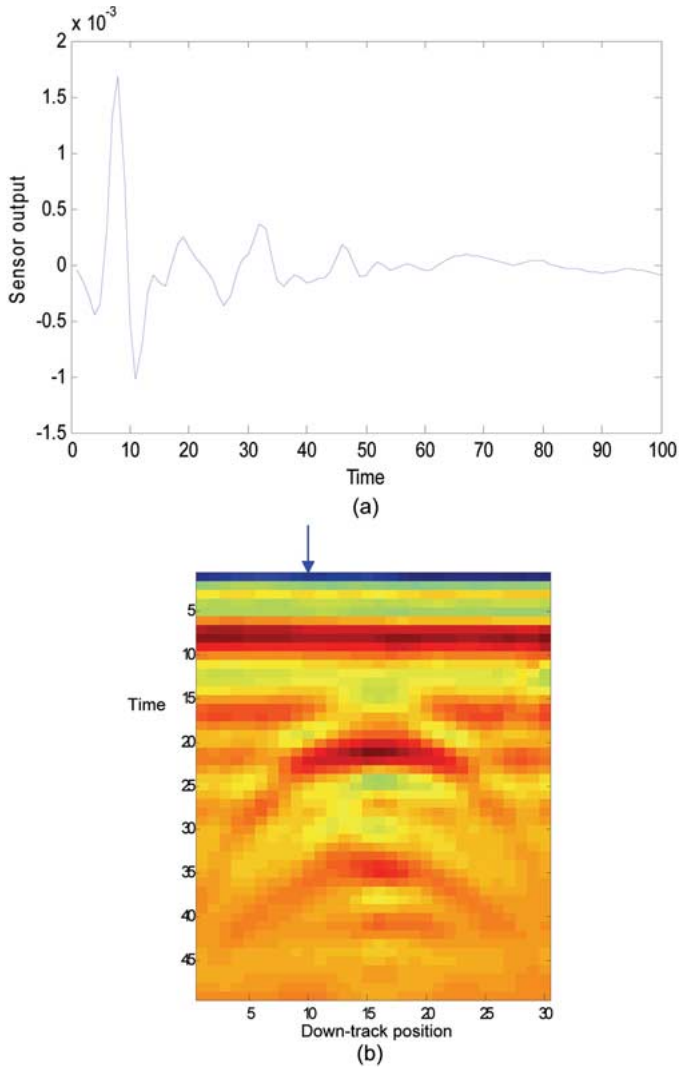


Fig. 2. GPR response when the sensor is above a plastic mine. (a) Amplitude versus time signal in one position. The time axis is sampled at a rate of 0.05 ns with the full waveform extending over 5 ns. The first peak corresponds to the reflection from the ground surface. (b) Two-dimensional scan of a plastic mine signature. The down-track positions are sampled at intervals of 2 cm. The arrow indicates the position where the sensor measured the signal in (a).

shows a typical GPR observation when the sensor is above a plastic mine, and Fig. 2(b) shows a 2-D scan of the landmine signature. Features extracted from a GPR observation include the raw moments (corresponding to energy features) and central moments (corresponding to fluctuation features) of the time series. Let $\{y_t\}_{t=1}^T$ denote the time series of a GPR observation, from which the raw and central moment features are

$$f_k^{(\text{raw})} = \frac{1}{T} \sum_{t=1}^T (y_t)^k, \quad \text{for } k = 1, 2, 3 \quad (3.2)$$

$$f_k^{(\text{cen})} = \frac{1}{T} \sum_{t=1}^T \left(y_t - f_1^{(\text{raw})} \right)^k, \quad \text{for } k = 2, 3 \quad (3.3)$$

respectively, in which $f_2^{(\text{cen})}$ reflects the variance of a GPR response, and $f_3^{(\text{cen})}$ reflects the degree of asymmetry of the wave. Moments higher than third order were found not to

contribute toward distinguishing target states, and therefore were not utilized. Details on the GPR and EMI sensors used to collect these data are provided in [21] and [22].

B. Specification of States S

In the discussion that follows, we introduce two types of states. The first type are called environment states, defined by the set S_E ; these states characterize the environment in which the agent operates. Five different environment states are considered here, corresponding to the presence of: 1) metal-class mine; 2) plastic-class mine; 3) Type-1 class clutter; 4) Type-2 class clutter; and 5) a clean subsurface. Type-1 clutter corresponds to large-sized metal items such as a soda can, while Type-2 clutter corresponds to small-sized metal items, including nails, shells, and screws. As shown when considering the results, some of the clutter is nonmetallic, but the associated signature has properties that may be characterized by the Type-1 and Type-2 classes, as previously discussed. The third type of nonmine corresponds to a “clean” region, which means that no mine or minelike objects are present in the vicinity of the sensor. More types of targets (mines and clutter) can be added to the model if desired. In Section IV, we generalize the framework to allow learning of the properties of new clutter and mines.

The particular (hidden) one of these environments under test defines the environment state $s_E \in S_E$. Many of these environment states yield complex sensor signals, as a function of the sensor position relative to the target; therefore, for each environment state $s_E \in S_E$, we define an associated set of target states S_T , with each target state representing where the sensor is relative to the target. An individual target state is one section of an annulus for a particular environment state (see Fig. 3). The target states are also unobservable but may be inferred along with s_E from the data. Let $|S_E|$ represent the number of environment states (here, $|S_E| = 5$), and let $|S_{T,m}|$ represent the number of underlying target states associated with the m th environment state; the total number of target states in the POMDP is $\sum_{m=1}^{|S_E|} |S_{T,m}|$. The state-transition and state-dependent observation probabilities in (2.1) and (2.2) are linked to the underlying target states. Because of the structure previously summarized, the state-transition-probability matrix defined by (2.1) is block diagonal, because the environment under test is fixed (but hidden), and therefore, as the sensor moves, state transitions are not allowed between target states of different environments. Below, we address the challenge of defining the set of target states S_T associated with each of the environment states $s_E \in S_E$.

In most cases, a landmine is cylindrically symmetric and buried with axis perpendicular or nearly perpendicular to the ground surface [see Fig. 3(a)]. A clutter item may not satisfy these properties, but the confusing clutter has a spatial signature that is similar to that of a mine. We also note that, even if the mine/clutter does not satisfy these burial and shape properties, the GPR and EMI sensors typically do not have sufficient resolution to explicitly discern the shape and orientation of the target, and therefore, the assumptions that follow are appropriate for most data to be considered.

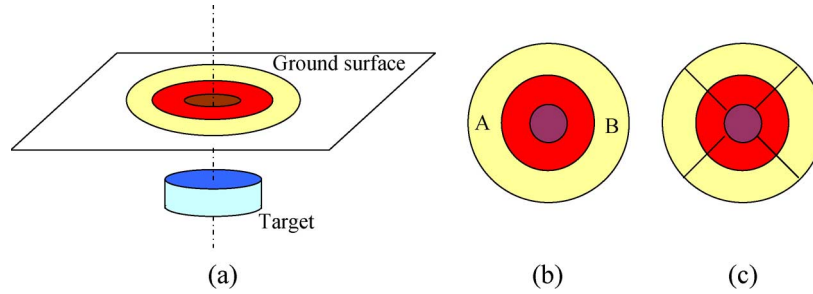


Fig. 3. Definition of the target state structure. (a) Illustration of the simple state definition. The states are concentric annuli on the ground surface with center above the center of a mine or clutter. (b) Platform of the simple state definition. Points A and B are in the same state, although the robot should have different actions for these two positions to best locate the landmine. (c) Updated state definition, where each annulus is divided into four sectors, corresponding to four directions.

TABLE I
DEFINITION OF ACTIONS A

Sensing actions	EMI sensing: 1. stay and sense with EMI 2. walk south and then sense with EMI 3. walk north and then sense with EMI 4. walk east and then sense with EMI 5. walk west and then sense with EMI	GPR sensing: 6. stay and sense with GPR 7. walk south and then sense with GPR 8. walk north and then sense with GPR 9. walk east and then sense with GPR 10. walk west and then sense with GPR
Declaration actions	11. declare "metal mine" 12. declare "plastic mine" 13. declare "Type-1 clutter" 14. declare "Type-2 clutter" 15. declare "clean"	

The robot is assumed to move on the (flat) 2-D ground surface. Considering that the energy of the signal response is a strong function of the distance from the object center, it is natural to define target states as concentric annuli on the ground surface, with center above the center of a mine or clutter, as shown in Fig. 3(a) and (b). Within each annulus, the sensor responses are considered relatively stationary. However, this simple state definition is not satisfying. The robot is assumed to move in four directions (forward, backward, left, and right), and we hope it can tell its position relative to an underground target by exploring the environment. For instance, the robot at points A and B in Fig. 3(b) should have different optimal actions to best locate the landmine. At point A, it should walk toward the right, while at point B, the best action is toward the left. The state definition in Fig. 3(b) does not allow the POMDP to distinguish this difference. Therefore, we divide each annulus into four sectors corresponding to the four directions (north, south, west, and east). The updated state definition is shown in Fig. 3(c). The representation in Fig. 3(c) motivates the basic target state structure considered here, and the remaining question is how many target states should we use to represent a given target; this is addressed in Section III-E.

C. Specification of Observations Ω

The discrete set of possible observations Ω is obtained as the codebook resulting from the vector quantization [14] of the continuous features. Each of the two sensors (EMI and GPR) generates its own codebook independently, resulting in two disjoint codebooks, the union of which defines Ω .

D. Specification of Actions A

The robot is assumed to have 15 actions, i.e., $A = \{1, 2, \dots, 15\}$, of which the first ten are sensing actions, and the rest are declaration actions; Table I provides a list of these actions. If a sensing action is applied, the robot first walks in one of the four directions for a distance δ (or it may stay at the same position), and then, it makes an EMI or GPR measurement according to the selected action. It is assumed that the robot always travels the same distance δ in each step in any direction, if it does not "stay." An adaptive step size could also be considered with an increase in complexity. The declaration actions declare the current position (where the robot currently is) to be one of the five types of mines or clutter buried underground (these define the "unobservable" environmental states S_E discussed in the Introduction).

E. Determination of the Number of Target States $|S_T|$ and the Codebook Size $|\Omega|$

The number of target states in the representation of a target and the size of the codebook are important issues in the POMDP model design. We address these issues by using the VB expectation-maximization (EM) method for model selection [17], which allows us to compute an approximation of the "evidence" for different $|S_T|$ and $|\Omega|$.

1) *VB EM Algorithm:* The EM algorithm is widely used in learning model parameters for incomplete data. In our problem, the data are incomplete because the states are partially observable. The traditional EM algorithm gives a maximum likelihood or maximum *a posteriori* point estimate, which does not express the posterior parameter uncertainty. Rather than

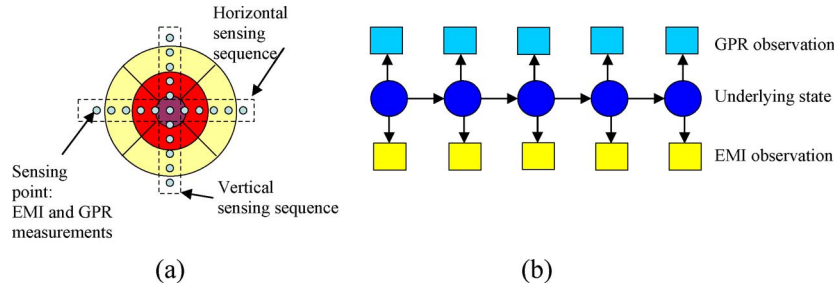


Fig. 4. HMM for model selection. (a) Illustration of sensing data positions over a target. The dots are sensing points. The horizontal and vertical sensing sequences pass through the center of the target. The concentric annular sectors with different colors represent different states. (b) The underlying HMM with two sets of observations (GPR and EMI).

a point estimate of the model parameters, we desire the full posterior via Bayes rule

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \quad (3.4)$$

where \mathbf{y} is measured data, $\boldsymbol{\theta}$ denotes model parameters, and $p(\boldsymbol{\theta})$ is the prior distribution over parameters. Given the measured data \mathbf{y} and several candidate models M_1, M_2, \dots, M_n with different structures, the goal of model selection is to decide which model fits the data best. One criterion for this selection involves comparing the marginal likelihood of the data \mathbf{y} for each model and choosing the model that has the highest likelihood. The marginal likelihood is also called the “evidence” [17] and is expressed as

$$p(\mathbf{y}|M) = \int p(\mathbf{y}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)d\boldsymbol{\theta}. \quad (3.5)$$

This is the denominator of the right-hand side of (3.4), except that the model M is now written explicitly.

By Bayes rule, the posterior distribution over the candidate models is given as

$$p(M_i|\mathbf{y}) = \frac{p(\mathbf{y}|M_i)p(M_i)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|M_i)p(M_i)}{\sum_i p(\mathbf{y}|M_i)p(M_i)}. \quad (3.6)$$

If we assume that all candidate models are equally probable, defined by the prior distribution $p(M_i)$, choosing the maximum marginal likelihood $p(\mathbf{y}|M_i)$ is equivalent to choosing the maximum posterior $p(M_i|\mathbf{y})$ over models.

The marginal likelihood (3.5) is difficult to evaluate because the integral is typically intractable analytically. The VB method [17] provides an approach to compute the lower bound of this marginal likelihood by introducing a factorized distribution $q(\mathbf{x}, \boldsymbol{\theta}) \approx q_x(\mathbf{x})q_\theta(\boldsymbol{\theta})$ to approximate the true distribution $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}, M)$, where \mathbf{x} denotes the hidden variables (states), and \mathbf{y} denotes the observed variables (measured data). Details with regard to VB applied to HMMs may be found in [16].

2) *Determining $|S_T|$ and $|\Omega|$* : We now address the problem of determining the number of target states and the codebook size by using the VB approximation of the model evidence (marginal likelihood). Suppose we are given the target position, and the associated EMI and GPR measurements, the horizontal or vertical sensing sequences passing through the center of a target are used as a training set to estimate the model param-

eters, as shown in Fig. 4(a). The data are assumed to be represented by an HMM with two sets of observations, as shown in Fig. 4(b). We assume that the GPR and EMI observations share the same underlying target states, which characterize the intrinsic physics of the target. The state-sequence statistics are assumed to be Markovian, i.e., for the moving platform, the target state sampled at time t depends only on the state sampled at $t - 1$; it is approximated to be independent of the states sampled before time $t - 1$. Given the current target state, the corresponding observation is independent of any other states or observations. In addition, the vertical and horizontal sequences [see Fig. 4(a)] are assumed to be equivalent since the target signature is assumed to be symmetric; this symmetry property of the signature is a good approximation for most landmines, and it is relevant for the type of clutter confused as a mine (for the resolution of the GPR and EMI sensors considered).

The HMM is used to model a target as a nonstationary stochastic process, as viewed by the sensors when they gradually approach the target, approach its center, and then leave it. The response signals (observations) are a function of the distance between the sensors and the target center: the smaller this distance, the stronger the response.

The candidate models have $|S_T| = 1, 5, 9, 13, \dots, 4K + 1$ target states, corresponding to $1, 2, 3, 4, \dots, K + 1$ annuli, respectively, and we consider codebook sizes $|\Omega| = 2, 3, 4, \dots, N$, where K and N define the range of the model structures we consider. The illustration of these candidate models for different numbers of states is shown in Fig. 5. The outer radius (15 cm) is the same for each of the candidates, and the different models are distinguished by the number of circular rings considered within.

To find the best $|S_T|$ and $|\Omega|$, we need to compute the model evidence for all the combinations of $|S_T|$ and $|\Omega|$, which are $N(K + 1)$ models in total. When N and K are large, this is computationally expensive. An alternative approach for large N and K is to iteratively optimize one parameter while fixing the other. We fix $|\Omega|$ and find the optimal $|S_T|$, and then fix $|S_T|$ as the optimal value from the last step and find the optimal $|\Omega|$. Note that in each iteration, $|S_T|$ or $|\Omega|$ is updated if and only if the model evidence corresponding to the new $|S_T|$ or $|\Omega|$ is larger than the old one. This search terminates when $|S_T|$ and $|\Omega|$ are both unchanged. Convergence is guaranteed since the model evidence increases monotonically, and there is an upper bound of the model evidence for all the combinations of $|S_T|$ and $|\Omega|$ considered. To avoid local minima, this procedure may

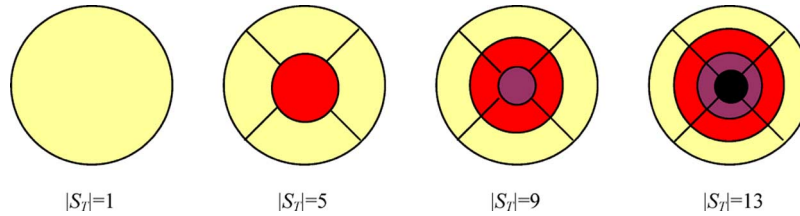


Fig. 5. Candidate models for different number of states $|S_T| = 1, 5, 9, 13$.

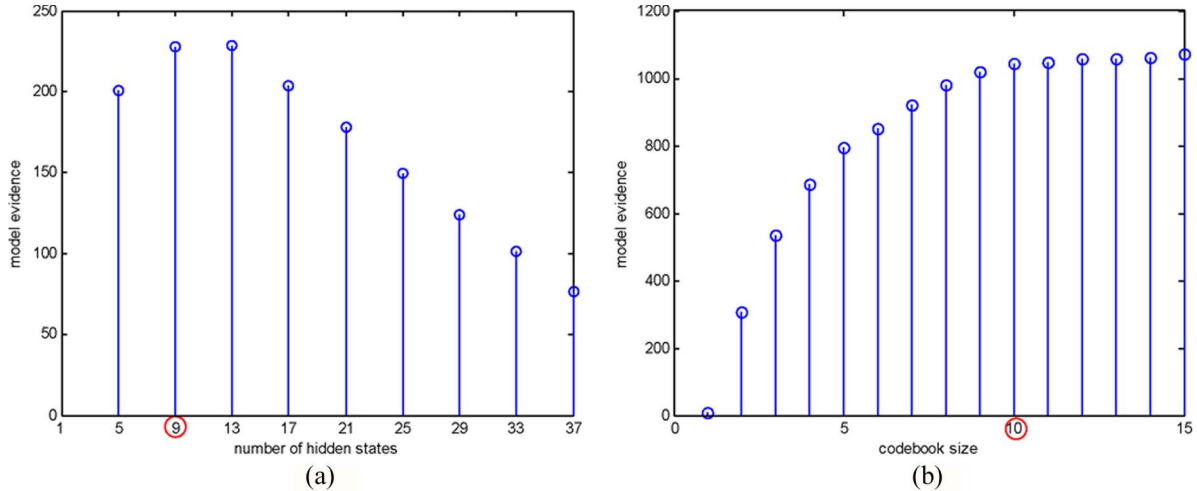


Fig. 6. Example of model evidence (marginal likelihood) for model selection. (a) Selection of the number of states when the codebook size is $|\Omega| = 10$. The maximum evidence occurs at $|S_T| = 9$. (b) Selection of the codebook size when the number of state is $|S_T| = 9$. The evidence remains stable after $|\Omega| = 10$. In both figures, the model evidence is the logarithm of the marginal likelihood apart from a constant.

be repeated several times starting from different initializations. Experiments suggest that this iterative approach obtains a satisfactory result relative to jointly searching over both parameters. An example of determining $|S_T|$ and $|\Omega|$ of metal mines using the iterative approach is shown in Fig. 6 (after convergence), in which we choose $|S_T| = 9$ for a fixed codebook size $|\Omega| = 10$ and choose $|\Omega| = 10$ for a fixed number of states $|S_T| = 9$. Note that in Fig. 6(b), the marginal likelihood remains stable when $|\Omega| \geq 10$; we select $|\Omega| = 10$ as the best choice according to Ockham's razor [23] which suggests that the simpler model is preferred for the same evidence.

For any given type of mine or clutter, the number of target states $|S_T|$ is determined as described previously. At the same time, the optimal state sequence of the target can also be determined by the Viterbi algorithm [12], which gives a maximum-likelihood estimate of the best state sequence and, hence, the size (radii) of the annuli. By estimating $|S_T|$ and the annulus radius for each of the five types of mines and clutter discussed previously, we define a total of 29 states as described in the next section. Similarly, we determine $|\Omega| = 12$ for both EMI data and GPR data.

F. Estimation of T and O

We cannot use the state-transition probabilities and the observation functions obtained from the VBHMM (discussed in the previous section) directly as the corresponding transition and observation functions in the POMDP model, because the VBHMM is a 1-D HMM model (robot moves in a 1-D hori-

zontal or vertical path passing the center of the target) without action selection, but the POMDP model is a 2-D model (robot moves in a 2-D plane) with multiple action selections. We now discuss how to estimate the transition and observation functions for the POMDP model using the $|S_T|$, $|\Omega|$, and the state definition obtained from the last section.

Across all five types of mines and clutter considered (for the five environment states considered), we define a total of 29 target states, i.e., $S_T = \{1, 2, \dots, 29\}$. The 29 states are divided into five disjoint subsets: $S_T = S_m \cup S_p \cup S_{t1} \cup S_{t2} \cup S_c$, denoting, respectively, the states of metal mines, plastic mines, Type-1 clutter, Type-2 clutter, and "clean"; the number of states in each of the five subsets are 9, 9, 9, 1, and 1, respectively. The definition of the states is illustrated in Fig. 7(a).

The belief state in (2.3) is represented across all of the target states in S_T , implying that the belief state here is a 29-D pmf. The probability that the item under test is in environment state $s_E = m$ is expressed as $p(s_E = m) = \sum_{s \in S_{T,m}} b(s)$, where $S_{T,m}$ is the set of target states associated with the m th environment. Probabilities of the form $p(s_E)$ are used to compute the immediate expected Bayes risk (cost) associated with declaring the region under test as being associated with environment state s_E , i.e., when a decision is made to stop sensing in a given region and make a declaration. Assuming that the algorithm declares that the region under test is a characteristic of environment $\hat{s}_E \in S_E$, in this case, the immediate expected reward is $\sum_{s_E \in S_E} R(\hat{s}_E, s_E)p(s_E)$, which is the negative of the Bayes risk of declaring $\hat{s}_E \in S_E$. The

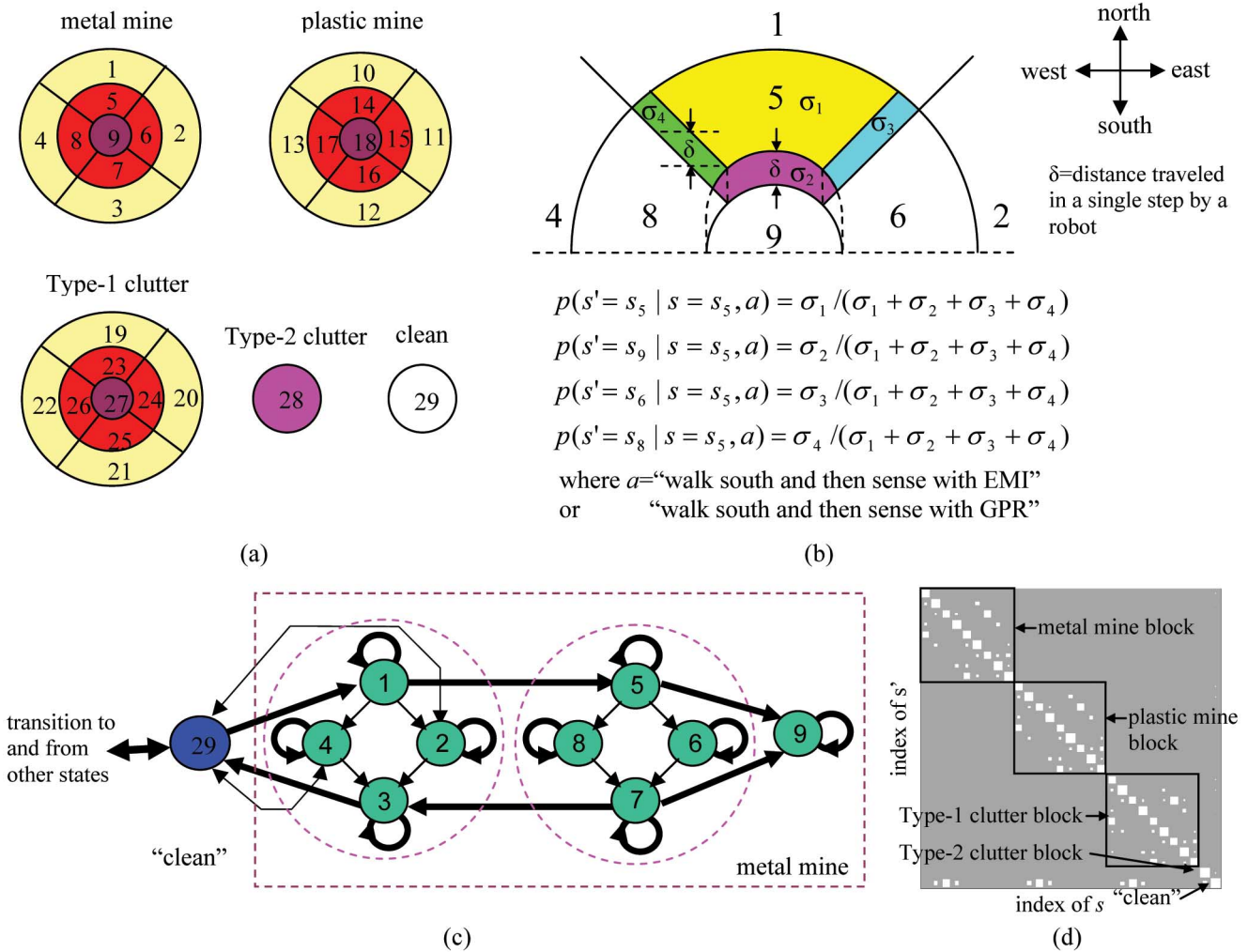


Fig. 7. State definition and transition-probability estimation for the landmine-detection problem. (a) Definition of the states. Metal mine, plastic mine, and Type-1 clutter (large-sized metal clutter) are each modeled by nine states, indexed 1 to 9, 10 to 18, and 19 to 27, respectively. Type-2 clutter (small-sized metal segment) is modeled by a single state (state 28); state 29 is used to indicate “clean” (i.e., there are no mine or minelike objects buried underground). (b) Illustration of the geometric method in computing the state-transition probabilities $T(s = 5, a, s')$ when a is one of the two sensing actions involving “walk south.” It is assumed that the robot travels the same distance in each step and that the robot’s position is uniformly distributed in any given state. $\sigma_1, \sigma_2, \sigma_3,$ and σ_4 denote the four regions of state 5 as well as their respective area metric. (c) Partial graph of state transitions of the model. This graph shows state transitions within states 1 to 9 (metal mine states) and state 29 (“clean”), when the action involves “walk south.” The bold arrows denote transitions with relatively high probability, while the thin arrows represent low-probability transitions. (d) Diagram showing the block diagonal structure of the transition probability matrix, where the white boxes represent nonzero entries, with box size proportional to probability; the action is “walk south” in this example.

algorithm must weigh the immediate risk/cost of making an immediate declaration of this type against the cost of future sensing actions and how these may impact the Bayes risk in the future. The matrix $R(\hat{s}_E, s_E)$ employed in the examples considered here is discussed in Section III-G. When a declaration action is taken, the expression $\sum_{s_E \in S_E} R(\hat{s}_E, s_E)p(s_E)$ is used to compute the first term to the right of the equality in (2.6).

The two sensing actions in which the robot does not move (the “stay” action) do not cause target state transitions; hence, $T(\cdot, a, \cdot)$ is an identity matrix when a is “stay and sense with GPR” or “stay and sense with EMI.” All remaining sensing actions can result in transitions from one target state to another. Assuming that the robot travels the same distance δ in each step and that the robot’s position is uniformly distributed in any given state, the probabilities of these transitions are directly

determined by using an elementary geometric computation. Fig. 7(b) illustrates how the transition probabilities $T(s = 5, a, s')$ for the two sensing actions involving “walk south” are computed. Let $\sigma_1, \sigma_2, \sigma_3,$ and σ_4 denote the four regions of state 5, as well as their respective area metric. The underlying state will transit to state 5, 9, 6, or 8, respectively, if the robot is in the region $\sigma_1, \sigma_2, \sigma_3,$ or σ_4 currently, and will walk south. The probabilities of transitioning to other states are zero. Given the traveling distance δ and the partition of the states, the regions $\sigma_1, \sigma_2, \sigma_3,$ and σ_4 can be easily determined by a geometric computation. Furthermore, the probability of transitioning to state 5, 9, 6, or 8 from state 5 is proportional to the area of the region $\sigma_1, \sigma_2, \sigma_3,$ or σ_4 .

Fig. 7(c) is a partial graph of the state transitions of the model, which only shows states 1 to 9 (metal mine states) and state 29 (“clean”), when the action involves “walk south.” If we

TABLE II
REWARD FUNCTION R , WHERE THE STATES HERE CORRESPOND TO THE POSSIBLE ENVIRONMENTAL STATES S_E

State \ Action	Sensing	Declare "metal mine"	Declare "plastic mine"	Declare "Type-1 clutter"	Declare "Type-2 clutter"	Declare "clean"
Metal mine	-1	+10	+5	-100	-100	-100
Plastic mine	-1	+5	+10	-100	-100	-100
Type-1 clutter	-1	-50	-50	+10	+5	+5
Type-2 clutter	-1	-50	-50	+5	+10	+5
Clean	-1	-50	-50	+5	+5	+10

assume that a mine or clutter is buried separately (no overlap), the transition-probability matrix $T(\cdot, a, \cdot)$ related to a "move and sense" action is block diagonal (i.e., a state transition happens only within each target) except that "clean" (state 29) could transit to or from the states of other types of targets (this is why Type-2 clutter is modeled by a single state; there is a possibility to transition to this state from "clean"). Fig. 7(d) shows the block-diagonal property of the state-transition matrix. From this point of view, the model can easily be expanded by adding more diagonal blocks. Each block corresponds to one target considered in the model. This property is important for the lifelong-learning algorithm considered in the next section.

The observation functions $O(a, s', o)$ are estimated similarly by using the geometric computation based on the state definitions and discrete observations resulting from vector quantization. The observation probability $O(a, s', o)$ is proportional to the number of the observations o in the state s' , so the computation is a counting process and then a normalization. The codebook size is 12 for both EMI and GPR data (24 codes in total). The sensing actions involving the same type of sensor share the same observation probability, independent of the directions in which the robot moves.

Computing $T(s, a, s')$ and $O(a, s', o)$ requires prior knowledge of the possible mines and clutter, and therefore, we assume access to examples of possible mines and clutter. The assumption of access to such a training set is removed in Section IV when addressing lifelong learning.

G. Specification of Reward R

The reward function R considered in the subsequent examples is defined in Table II. Note that the sensing actions are independent of the environment state $s_E \in S_E$ under test, while the declaration actions are a strong function of s_E . All sensing actions have a cost of -1 , although in general, we can set different costs for the two sensors.

For the declaration reward, intuitively, correctly recognizing a target should get a positive reward; partially correct declaration, which means the robot is confused between types of mines or between types of clutter but not between mines and clutter, gets a less positive reward; missing a landmine or declaring a landmine as clutter should have a very large penalty, and declaring a clutter as a landmine also has a large cost but less than a missed mine; these rewards are used when evaluating the immediate expected reward $\sum_{s_E \in S_E} R(\hat{s}_E, s_E)p(s_E)$ discussed previously. The units in Table II are arbitrary, and we note that a different reward structure may readily be considered, resulting in a new policy. In this sense, the manner in

which the reward structure is defined constitutes the subsequent policy.

IV. LIFELONG LEARNING: EXPLORATION AND EXPLOITATION

All discussions thus far assumed accurate knowledge of the POMDP model. It was therefore assumed that we have a complete training data set, which describes the properties of all mines and clutter (including the soil properties) that may be encountered. It has been assumed that a reliable POMDP model is built from the training data, from which the policy is learned, and this policy is exploited when sensing. Stated succinctly, the exploration and exploitation phases have been assumed to be separable and distinct. The system first obtains labeled training data (exploration) with which the POMDP model is learned, and the policy is designed. The policy is then exploited subsequently when detecting landmines, and this policy is not refined during this latter process.

The assumptions inherent to the POMDP setting are often not easily satisfied. In many scenarios, the training data cannot be provided in advance; the robot is required to learn the model and policy by exploring the environment itself. In this situation, the training phase and the detection phase become one overall process, with exploration and exploitation performed jointly. In other cases, even though a model and a policy could be learned beforehand, the model may not be good enough or appropriate for future sensing. For example, some new targets are frequently encountered in the detection phase; hence, the robot should consider adding a new target into the existing model and possibly de-emphasizing models of mines and clutter that are not observed when sensing. It is desirable for the robot to modify its understanding of the environment online during its detection phase; this is termed "lifelong learning" [18], [24], [25].

In this section, we investigate a method for learning the model by an online approach, i.e., the robot learns the model at the same time as it moves and senses in the minefield (combining exploration and exploitation). By this approach, the model size (number of states $|S|$ and discrete observations $|\Omega|$), model parameters (transition probability T and observation function O), and optimal policy are updated online during the learning process. The algorithm given in the next section is motivated by and modified from the MEDUSA algorithm [18].

A. Dirichlet Distribution

We first review the Dirichlet distribution, which is an important tool for the lifelong-learning algorithm that follows. The

Dirichlet distribution is the conjugate prior of the multinomial distribution [26]. The multinomial distribution is a discrete distribution that gives the probability of choosing a given collection of m items from a set of n items, without concern for order; the probabilities of the n items are given, respectively, by $\mathbf{p} = (p_1, \dots, p_n)$. Probabilities $\{p_i\}_{i=1}^n$ are the parameters of the multinomial distribution; $\{p_i\}_{i=1}^n$ are the random variables of the Dirichlet distribution, which will serve as a prior for \mathbf{p} .

The probability density of the Dirichlet distribution for random variables $\mathbf{p} = (p_1, \dots, p_n)$ with parameters $\mathbf{u} = (u_1, \dots, u_n)$ is defined by

$$p(\mathbf{p}) = \text{Dir}(\mathbf{p}; \mathbf{u}) = \frac{1}{c(\mathbf{u})} \prod_{i=1}^n p_i^{u_i-1} \quad (4.1)$$

with $p_1, \dots, p_n \geq 0$, $\sum_{i=1}^n p_i = 1$, $u_1, \dots, u_n \geq 0$, and the normalizing constant $c(\mathbf{u}) = [\prod_{i=1}^n \Gamma(u_i)] / [\Gamma(\sum_{i=1}^n u_i)]$.

The mean of the Dirichlet distribution is

$$E(p_i) = \frac{u_i}{\sum_{i=1}^n u_i}, \quad \text{for } i = 1, \dots, n. \quad (4.2)$$

Given i.i.d. data $\mathbf{y} = \{y_1, \dots, y_m\}$ drawn from a multinomial distribution with parameters \mathbf{p} , with prior on \mathbf{p} represented by $\text{Dir}(\mathbf{p}; \mathbf{u})$, the posterior distribution of \mathbf{p} is represented by an update of the Dirichlet distribution, $\text{Dir}(\mathbf{p}; \tilde{\mathbf{u}})$, which is computed by the counting process

$$\tilde{u}_i = u_i + \sum_{j=1}^m \text{indicator}(y_j = i), \quad \text{for } i = 1, \dots, n \quad (4.3)$$

where $\text{indicator}(z) = 1$, if z is true, and $\text{indicator}(z) = 0$, otherwise.

From a Bayesian view, the parameters u_i can be interpreted as prior observation counts for events governed by p_i . When u_i is large, the prior knowledge dominates the posterior distribution; alternatively, if u_i is a small number, we put more trust in the observed data.

B. Lifelong-Learning Algorithm

The lifelong-learning algorithm borrows ideas from Bayesian theory, in that we constitute a posterior distribution over possible POMDP models, based on prior intuition as to what models are appropriate and based on the observed data. A flowchart of the lifelong-learning algorithm is shown in Fig. 8.

Given the current target state s and an action a , the next state s' can be seen as a draw from a multinomial distribution with parameters

$$P_i^{T,s,a} = p(s' = s_i | s, a) = T(s, a, s_i), \quad \text{for } i = 1, \dots, |S| \quad (4.4)$$

with $\sum_{i=1}^{|S|} P_i^{T,s,a} = 1$. Similarly, for a given action a and state s' , the observation o is a draw from a multinomial distribution with parameters

$$\begin{aligned} P_i^{O,s',a} &= p(o = o_i | a, s') \\ &= O(a, s', o_i), \quad \text{for } i = 1, \dots, |\Omega|. \end{aligned} \quad (4.5)$$

The goal of lifelong learning is to learn these multinomial parameters in the state-transition probability T and observation function O . Based on the discussion in Section IV-A, for each state-action pair in the transition probability T or the observation function O , a Dirichlet prior is assigned

$$T(s, a, \cdot) \sim \text{Dir}(\mathbf{p}^{T,s,a}; \mathbf{u}_{T,s,a}) \quad (4.6)$$

$$O(a, s', \cdot) \sim \text{Dir}(\mathbf{p}^{O,s',a}; \mathbf{u}_{O,s',a}). \quad (4.7)$$

Learning is a process of continuously updating the hyperparameters $\mathbf{u}_{T,s,a}$ and $\mathbf{u}_{O,s',a}$. We observe that the transition and observation probabilities are drawn from respective Dirichlet distributions.

We also assume an ‘‘oracle’’ is available, which can provide the label (identity) of the underground target currently under interrogation, on request. The best ‘‘oracle’’ is implemented by excavating an item of interest (e.g., by a human operator). Note that in the landmine-sensing problem, for which the true labels can be acquired via excavation, the use of an ‘‘oracle’’ is practical, albeit expensive, with the cost of oracle deployment accounted for in the algorithm. If a new type of mine or clutter is excavated during this process, a new class of models is added (with associated target states and model learning performed, as discussed in Section III). If the mine/clutter type has been seen previously, the associated model is refined based upon the new measured data. We emphasize that the oracle only provides information about the environment state $s_E \in S_E$ of the excavated item, while the data are used to determine the associated target states.

An oracle query is performed if one of the following conditions is satisfied: 1) if the policy says that the oracle query is the optimal action at the current step; 2) if the agent finds a new observation that has never been seen before; this is a totally new knowledge and is unaccounted for in the existing model; and 3) if the agent has measured extensively in a subarea (the area from the last declaration position to the current sensing position within the lane, as defined in Section V). In this latter case, the number of sensing actions in the local region is larger than a threshold, and the agent still cannot make a decision about the underground target, which means the current task is too difficult for the agent. This phenomenon can happen if the expected cost of making a declaration, quantified by $-\sum_{s_E \in S_E} R(\hat{s}_E, s_E) p(s_E)$ is too high, and therefore, the agent would simply continue to perform relatively inexpensive sensing actions. By ‘‘too hard,’’ this implies that the pmf defined by $p(s_E)$, based on previous actions and observations, is not sufficiently ‘‘peaky’’ about one of the environments $s_E \in S_E$. In this case, the oracle is called upon to excavate the item without making a specific declaration as to the associated environment state $s_E \in S_E$.

When an oracle query is required, the robot senses its local area on a grid using the two sensors, such that it collects as much information of the unknown object as possible, and then, the label is revealed via the oracle (excavation). The size and position of the grid-sensing region is determined by the energy distribution in the local area. In general, if the energy and energy gradient are small in both the EMI signal and GPR

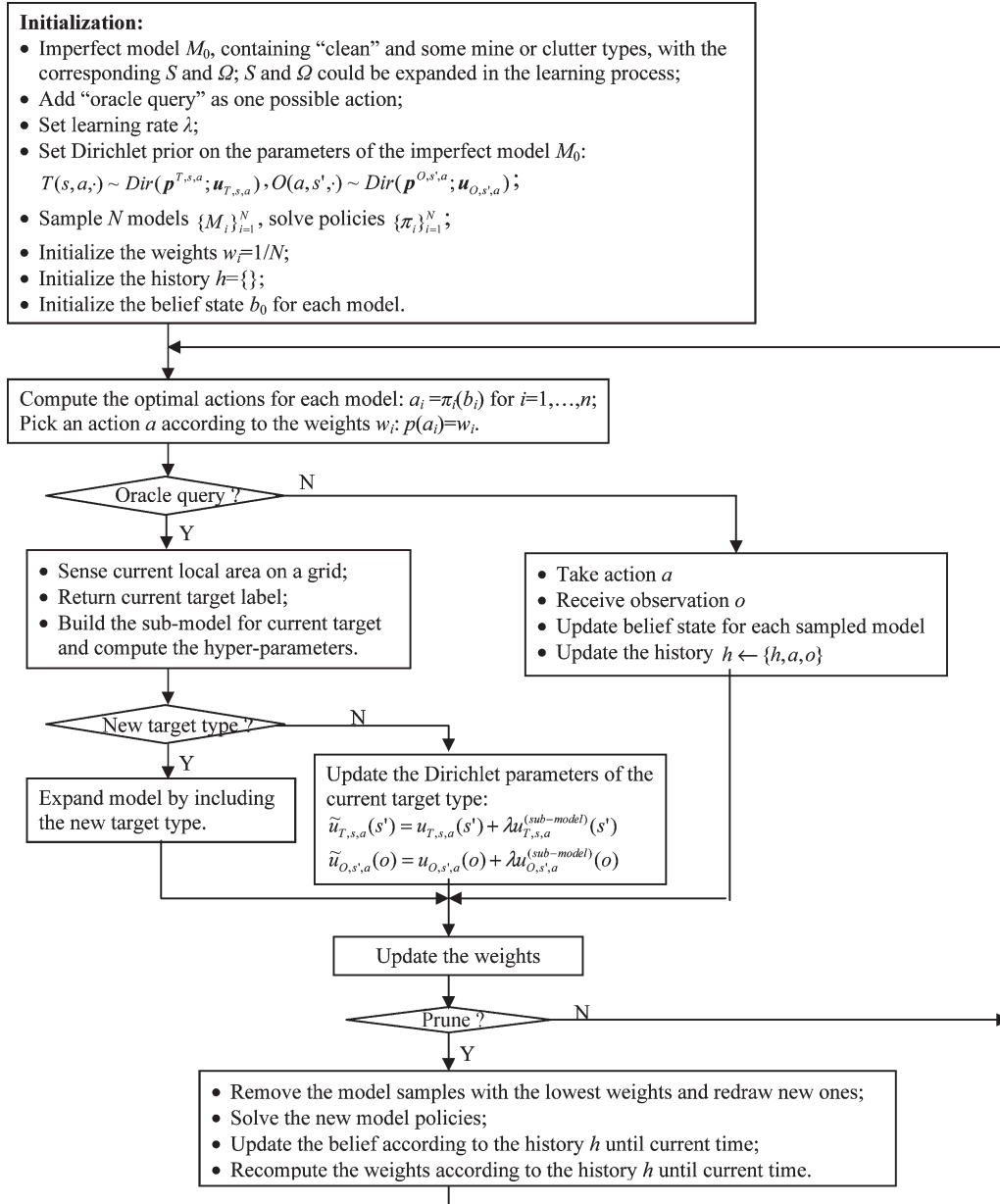


Fig. 8. Flowchart of lifelong learning in the landmine-detection problem.

signal, it is reasonable to consider the current point as the edge of an underground target and, hence, the edge of the grid-sensing region. The “label” includes the type of mine or clutter, size and position of the target, and the reward values when declaring it correctly or incorrectly. If the target class is new, a submodel is then built, which is similar to the discussion in Section III, using the grid-sensing data and the label information; the submodel is a subset of the entire model, which is composed of those states and observations related to the current target. Note that in the submodel, we learn the Dirichlet hyperparameters which represent the posterior distributions of $T^{(\text{submodel})}$ and $O^{(\text{submodel})}$ (as computed using VB).

There are two learning approaches in the proposed algorithm. When an oracle query reveals the target type to be a new one, the algorithm expands its model by adding the new target type into the existing model (the set of environment state S_E is

expanded by one). There is an expansion in the total set of target states S_T and in Ω (if necessary); this corresponds to adding a new diagonal block in the transition probability T (see Section III-F) and new states (and observations) in the observation function O . The associated hyperparameters are expanded at the same time. If according to an oracle query, the algorithm finds that the revealed target type is a familiar one, it updates the model hyperparameters for this target type at learning rate λ

$$\tilde{u}_{T,s,a}(s') = u_{T,s,a}(s') + \lambda u_{T,s,a}^{(\text{submodel})}(s') \quad (4.8)$$

$$\tilde{u}_{O,s',a}(o) = u_{O,s',a}(o) + \lambda u_{O,s',a}^{(\text{submodel})}(o) \quad (4.9)$$

where $u_{T,s,a}^{(\text{submodel})}$ and $u_{O,s',a}^{(\text{submodel})}$ are the submodel hyperparameters learned from the measured data for current target type,

TABLE III
GROUND TRUTH AND DETECTION RESULTS ON THREE MINEFIELDS

		Mine Field 1	Mine Field 2	Mine Field 3
Ground truth	Number of mines (metal + plastic)	5 (3+2)	7 (4+3)	7 (4+3)
	Number of clutter (metal + nonmetal)	21 (18+3)	57 (34+23)	29 (23+6)
Detection result	Number of mines missed	1	1	2
	Number of false alarms	2	2	2

$\mathbf{u}_{T,s,a}$ and $\mathbf{u}_{O,s',a}$ are the old hyperparameters for the existing model, and $\tilde{\mathbf{u}}_{T,s,a}$ and $\tilde{\mathbf{u}}_{O,s',a}$ are the updated hyperparameters for the posterior distributions. The two learning approaches previously discussed are based on the assumption that the state definition is exclusive for each target type, and the state transition happens only within each target type. The learning rate λ balances the importance between the prior knowledge (existing model) and the new observations from an oracle query. If $\lambda = 0$, the agent never updates the model, and if $\lambda \rightarrow \infty$, the posterior is entirely decided by the new observations.

Based upon the data observed thus far, we have posterior distributions for the transition probability T and observation function O across all targets observed thus far; these posteriors represent our state of knowledge about the scene under test and are quantified by the aforementioned Dirichlet distributions. To make the analysis practical, we now sample N sets of parameters from the posteriors, constituting N POMDP models that, for sufficiently large N , capture the uncertainty with regard to the properties of the mines and clutter. These sampled POMDP models are then used to constitute N associated policies. At each sensing step, the agent has N optimal actions $\{a_i\}_{i=1}^N$ to choose from, coming from the N policies. The agent picks one action among them as follows. Let the history $h = \{a_1, o_1, a_2, o_2, \dots\}$ record the action–observation sequence as the agent experiences the environment. Denote weights $\{w_i\}_{i=1}^N$ as the normalized likelihood of the history h for each of the N models, computed at each step using the forward–backward algorithm [12] and normalized such that $\sum_{i=1}^N w_i = 1$. Then, the agent randomly chooses an action to execute according to the weights $\{w_i\}_{i=1}^N$, i.e., $p(a_i) = w_i$, for $i = 1, \dots, N$. Furthermore, at regular intervals, the agent removes the model samples with the lowest weights (below a prescribed threshold) and draws new models according to the current model hyperparameters. During the processes of updating the hyperparameters by oracle queries, picking actions by the weights, and pruning the low-weight model samples, the agent gradually focuses on the model sample which best represents the true characteristics of the underlying environment.

The PBVI algorithm [19] is applied to the POMDP models built according to the methods discussed in Sections III and IV to learn the policies; when implementing PBVI, the belief samples are obtained by belief expansion once every 15 iterations, and a total of seven expansion phases result in approximately 3000 belief points for policy learning.

V. EXPERIMENTAL RESULTS

The data used in this paper were measured by actual EMI and GPR sensors [22] for three “minefields,” each of which is defined by man-made and natural clutter as well as plastic and metal (inert) landmines. These measured data are then used

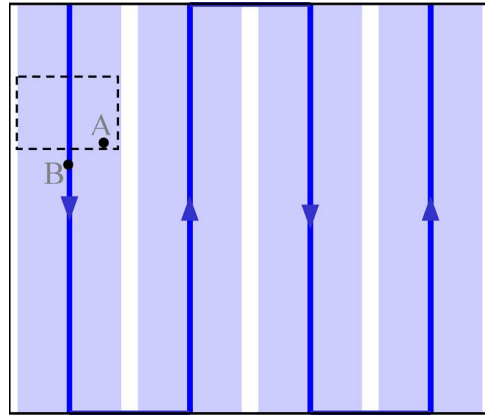


Fig. 9. Robot navigation path in a minefield. The dark blue curve is the “basic path,” which defines the lanes as indicated by light blue. The robot is restricted to move along the lanes by taking actions within the lanes. The “basic path” restrains the robot from moving across the lanes. A declaration is made in location A based on the exploration in a local region bounded by the dash-line box. After the declaration, the robot is reset to the next unexplored location along the basic path, denoted by location B .

to simulate a robot traversing the respective minefields. The EMI and GPR data are precollected over a $1.6 \times 1.6 \text{ m}^2$ per simulated minefield, with sensor data collected at a 2-cm sample rate in two coordinate dimensions [22]. The precollected data are used to simulate the data collected by an autonomous two-sensor agent, as it senses within the minefield. The three “minefields” are shown in Figs. 11, 13, 14, and Table III. Clearly, to avoid missing landmines, the robot should search almost everywhere in a given minefield. However, we hope that the robot can actively decide where to sense as well as which sensor to use, to minimize the sensing cost. Considering these two requirements together, we assign a “basic path” as shown in Fig. 9 (dark blue curve with arrows). The “basic path” defines the lanes as indicated by light blue in the figure, and the robot is restricted to take actions within the lanes. The “basic path” restrains the robot from moving across the lanes, and the robot defines sectors along each lane as being characterized by one of the mines/clutter, including “clean,” while moving in an overall direction consistent with the arrows in Fig. 9. The distance between two neighboring “basic paths” should be less than the diameter of a landmine signature. Furthermore, we assume the motion is not noisy, which means when taking actions, the robot can move from one location to its neighborhood in four directions without error; in future work, imprecise motion may be accounted for in the state-transition matrices.

After each declaration, the robot is assumed to be reset to a new location to explore the next local region. Without this resetting mechanism, the robot would stop at a mine and declare it as a mine forever (getting a significant reward, if right, every-time). To avoid this, resetting is assigned to the next unexplored location along the basic path (see Fig. 9 for further explanation).

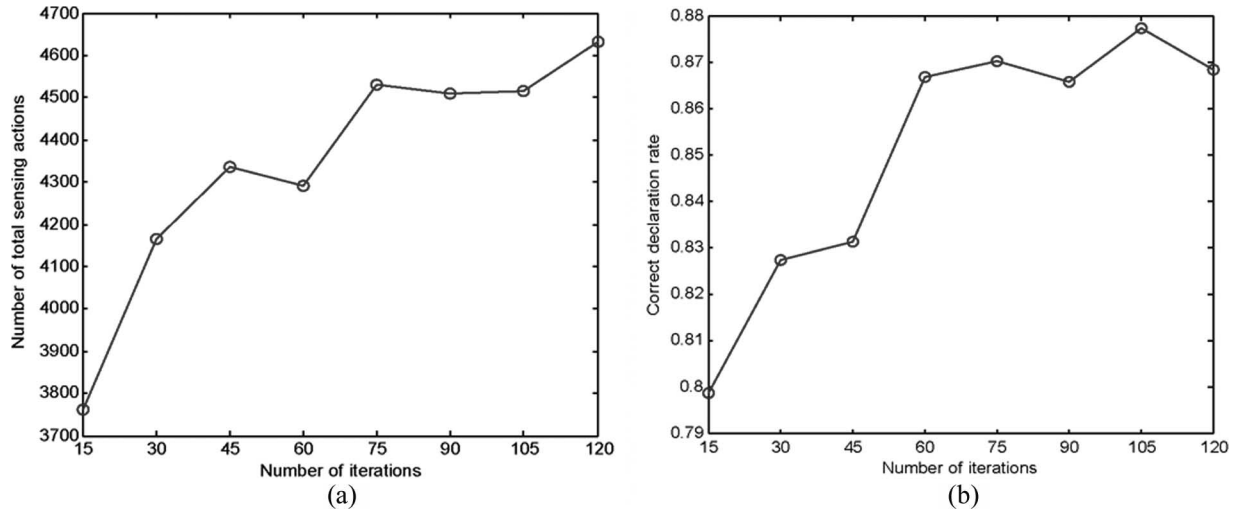


Fig. 10. Detection performance as a function of number of iterations when learning the policy. (a) Number of total sensing actions. (b) Correct declaration rate.

Assume the robot has explored the region bounded by the dash-line box since the last declaration and made the current declaration at location *A* in Fig. 9. By the resetting mechanism, the robot starts a new search from location *B*, outside, and neighboring of the previously explored region.

It is possible that after many measurements in one local area, the agent still cannot make a declaration. For example, this can occur if the belief state becomes oscillatory, possibly because our model does not include the current underground target or the model we build does not fit the data in this area well. More measurements do not help to make a better decision. If this happens, it is better to say “I do not know” rather than continue sensing or making a reluctant declaration. We let the robot declare “unknown” in this situation, while in the lifelong learning algorithm, the third query condition is met and the “oracle” is employed.

A. Detection Performance of the Offline-Learning Algorithm

In the offline-learning approach, the training data are given in advance, and the training phase and test phase are separate. We use Minefield 1 (Fig. 11) as the training data to learn the model and the policy, and then test our method on all three minefields; we emphasize that the EMI and GPR data were collected separately on three separate “minefields”—the data are not constituted by stitching together individual mine and clutter signatures. The training and test data match well in that the three minefields contain almost the same types of metal mines, plastic mines, and clutter. The clutter includes metal clutter (soda can, shell, nail, coin, screw, lead, rod, and ball bearing) and nonmetal clutter (rock, bag of wet sand, bag of dry sand, and a CD). Note that we consider many types of clutter items, and these all fall within the broad classes discussed in Section III.

1) *Model Training and Policy Design*: Using Minefield 1 as the training data set, the POMDP model is built as discussed in Section III, and the policy is learned by PBVI. Other policy learning algorithms such as region-based value iteration (RBVI) [27], Perseus [28], or Q-MDP [29] can also

be considered. The number of sensing actions and the correct declaration rate as a function of value iteration number (as discussed in [19]) when determining the policy are plotted in Fig. 10. The correct declaration rate is defined as the ratio of the number of correct declarations relative to the number of all declarations. Note that the correct rate is not equivalent to probability of detection since one landmine could be declared multiple times, and the correct declaration of clutter or “clean” is also counted in the correct rate. However, it does reflect the detection performance by comparing declaration position and ground truth. From Fig. 10, after 75 iterations and five belief expansion phases, the PBVI-learned policy becomes stable.

2) *Landmine Detection Results*: The stationary policy from the last section is then used to navigate the robot in the three minefields. The ground truth and detection results are summarized in Table III. As an example, the layout of Minefield 1, the declaration result, and a zoom-in of sensor choices are shown in Fig. 11. Note that one target may be declared several times.

Missed landmines are usually caused by one of the following two reasons: The mine has very weak signal in both EMI and GPR responses, such as a small antipersonnel mine which is a low-metal content mine; or the mine is very close to a large metal clutter so that the clutter’s strong response hides the weak signal of the mine.

From Fig. 11(c), we see that the policy selects GPR sensors to interrogate plastic mines, while it prefers EMI sensors when metal mines are present. This verifies the policy to some degree since the EMI sensor is almost useless for detecting plastic mines but is good for detecting metal mines. We also see that on the “clean” area or at the center of a landmine, a declaration is made only based on very few sensing actions, usually two or three, since it is relatively easy for the robot to estimate its current states. However, at the edge of a landmine, where there is an interface between two objects (the landmine and the “clean”), the robot usually requires many sensing actions to make a declaration.

The robot requires, on average, approximately 4500 sensing actions in one minefield; the correct declaration rate is about

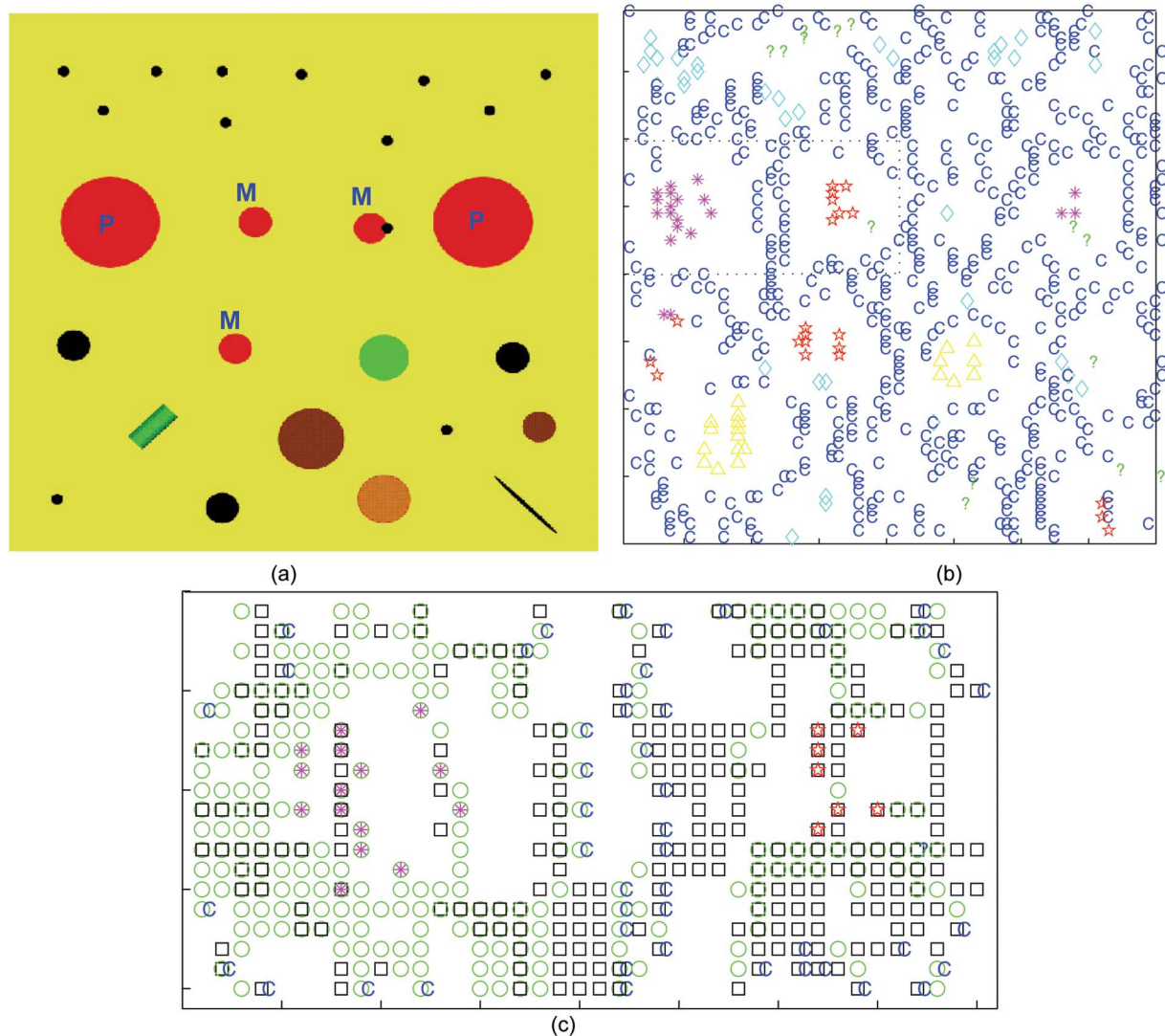


Fig. 11. Ground truth and detection details in Minefield 1. (a) Ground truth. The red circles are landmines, with “M” and “P” indicating metal mine and plastic mine, respectively; the other symbols represent clutter. Black dots are small metal segments, and the rest are large-sized metal or nonmetal clutter. (b) Declaration result. The blue “C” means a declaration of “clean,” the green “?” means “unknown,” and other marks with various colors represent declarations of mines or clutter. Red pentagram: Metal mine; pink star: Plastic mine; yellow triangle: Type-1 clutter; cyan diamond: Type-2 clutter. (c) Sensor choice in the broken-lined rectangular area shown in (b). The black square means sensing with EMI sensor, and the green circle means GPR sensor. It can be seen that the policy prefers the GPR sensor for plastic mine [left half in (c)] and the EMI sensor for metal mine [right half in (c)].

0.87 (see Fig. 10). As a comparison, if a myopic policy is applied, where the agent considers maximizing its rewards of only one step ahead to select actions [equivalent to $k = 1$ in (2.5)], a total of around 8000 sensing actions are needed, and a correct declaration rate of 0.82 is achieved. Note that if one senses on every grid point using both sensors, the total number of measurements is $2 \times 800^2 = 12\,800$.

B. Detection Performance of the Lifelong-Learning Algorithm

In the lifelong-learning approach, the training and the test phases are integrated, and the model and the policy are updated online during the combination of exploration and exploitation. We let the robot move in Minefield 1, navigated by the policies within the lanes defined by the “basic path.” We set the learning rate as $\lambda = 1$, the number of model samples as $N = 10$, and the cost of the oracle query as $r = -80$. The other reward values

are the same as discussed in Section III-G. At the beginning, the imperfect model includes only the “clean” situation, i.e., one state and several observations; we, therefore, assume no knowledge of the mines or clutter. The results of the lifelong learning in Minefield 1 are shown in Fig. 12, where Fig. 12(a) shows the positions of the oracle queries and the other declarations when the robot explores and exploits the environment, and Fig. 12(b) is the average error of the model learned by the lifelong learning relative to an “ideal model.” Here, the “ideal model” is assumed to be the one we obtained by offline approach in Section V-A1; the average error is defined as the average value of the absolute differences between the learned model parameters (the means of the parameter distributions) and the corresponding “ideal model” parameters.

In Fig. 12(a), each rectangle represents an oracle query and its grid-sensing region. It can be seen that at the beginning [left part of the Fig. 12(a)] of the learning, many wrong “Type-2

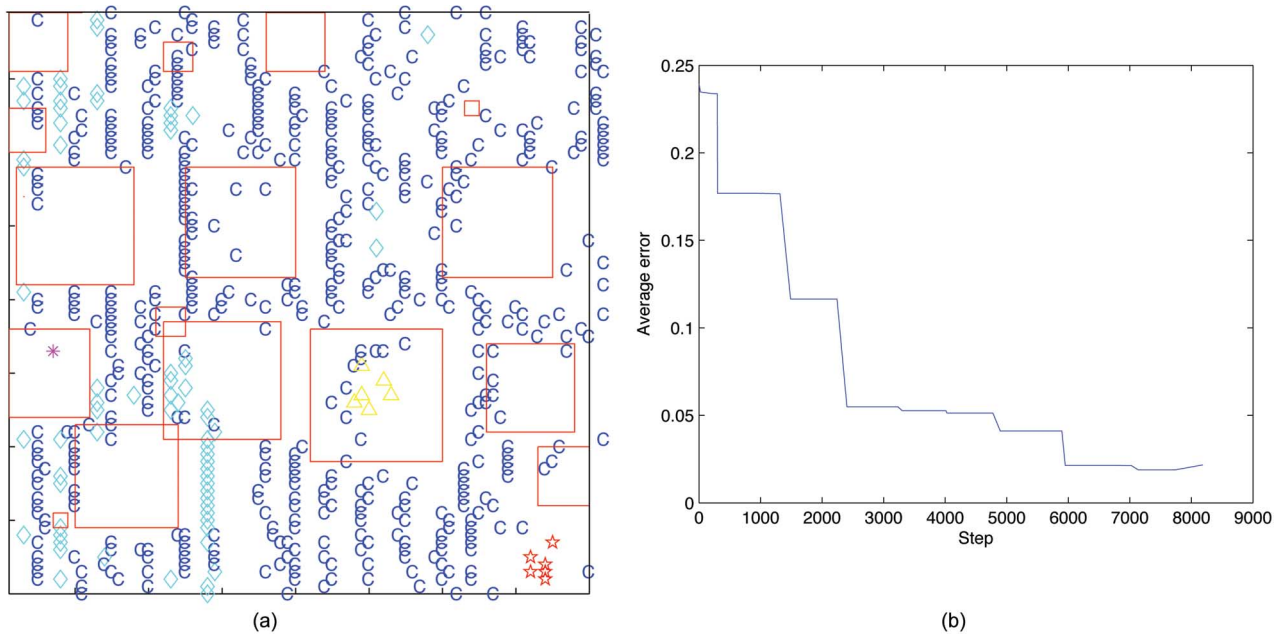


Fig. 12. Detection results of the lifelong learning in Minefield 1. (a) Oracle queries and other declarations. Each red rectangle represents an oracle query and the corresponding grid-sensing region. Other marks are declarations: Blue “C”—“clean,” red pentagram—metal mine, pink star—plastic mine, yellow triangle—Type-1 clutter, and cyan diamond—Type-2 clutter. The ground truth of the minefield is shown in Fig. 11(a). (b) Average error between the learned model and the model obtained by offline learning. The three big error drops at steps around 300, 1500, and 2000 correspond to finding new target types and adding them to the model.

clutter” declarations are made. After learning more, there are fewer wrong declarations. The model is expanded by adding two mine types, two clutter types, and more observations in the earlier period of the learning. This is also demonstrated by the big decrease in the average error in Fig. 12(b). Later, the model hyperparameters are updated when necessary, according to oracle queries, and the model becomes increasingly accurate. Note that the learning process does not end even though the robot finishes exploring all of Minefield 1. When a new task comes, the robot continues to modify the model parameters if the old model does not fit the new minefield.

Assume that the robot meets Minefield 2 and then Minefield 3 after it learned the model in Minefield 1. Minefields 2 and 3 contain the same types of mines and clutter learned previously. Fig. 13(a) is the ground truth of Minefield 2, and Fig. 13(b) shows the associated detection results. With one missed mine and two false alarms (the same as in Table III), the results demonstrate the performance of the lifelong learning. The detection result of Minefield 3 (see Fig. 14) also yields a result similar to the offline approach in Table III.

Finally, we discuss the immediate reward. It is assumed that during the exploitation in a minefield, the agent does not know the immediate reward after each declaration. Under this assumption, all the reward values the agent knows come from the initial model and oracle queries. This agrees with a practical situation for which the robot does not know if its decision is correct or incorrect immediately after each declaration. Note that if we discard this assumption, the learning will be more efficient, since the agent could evaluate its performance by checking the immediate reward it received and adjust its learning strategy.

For example, if the error rate is high, the agent could consider taking more oracle queries to improve the model. If a certain declaration often causes a penalty, the agent should be careful that the model for this target might be poor.

C. Importance of Setting the Reward Function

Setting the reward function is important in producing a good policy. An inappropriate reward function causes a poor policy and thus unsuccessful detection, even if the model is perfect. In a simulated experiment, the reward function can be set using a trial-and-error method or by experience. In a real problem, the reward value can be estimated by its real cost, although it is often very difficult to quantify costs or rewards.

We consider the critical role of the penalty when the robot misses a landmine. Refer to the previous setting in Section III-G where this penalty is -100 . We suppose to keep the other reward values the same but let this penalty vary from -10 to -1000 . Given Minefield 1 as the training data, the model and the corresponding policies are learned, and the robot executes the policies when detecting in the same minefield. The plots of the cumulative reward, the correct declaration rate, and the average number of sensing actions to make one declaration as a function of the penalty are shown in Fig. 15. From these figures, the cumulative reward reaches a peak value when the penalty is around -100 . The correct declaration rate increases as the penalty increases, achieves maximum at a penalty of around -100 , and then slightly decreases when the penalty is higher. The average number of sensing actions for one declaration increases monotonically. These results are consistent with our intuition. If the penalty is too low, the agent does not care if

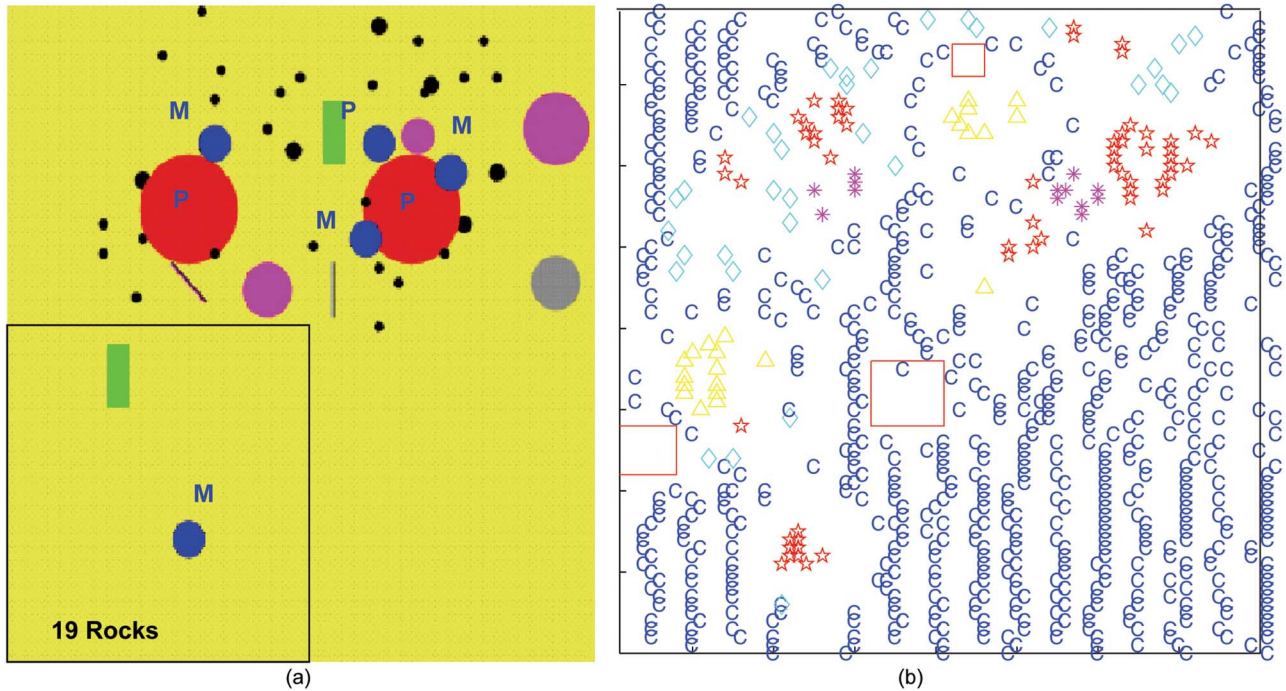


Fig. 13. Detection results of the lifelong learning in Minefield 2 after the algorithm has learned the model from Minefield 1. (a) Ground truth. The red and blue circles are landmines with “M” and “P” indicating metal and plastic mines, respectively; the other symbols represent clutter. Black dots are small metal segments, and the rest are large-sized metal or nonmetal clutter. (b) Oracle queries and other declarations. Each red rectangle represents an oracle query and the corresponding grid-sensing region. Other marks are declarations: Blue “C”—“clean,” red pentagram—metal mine, pink star—plastic mine, yellow triangle—Type-1 clutter, and cyan diamond—Type-2 clutter.

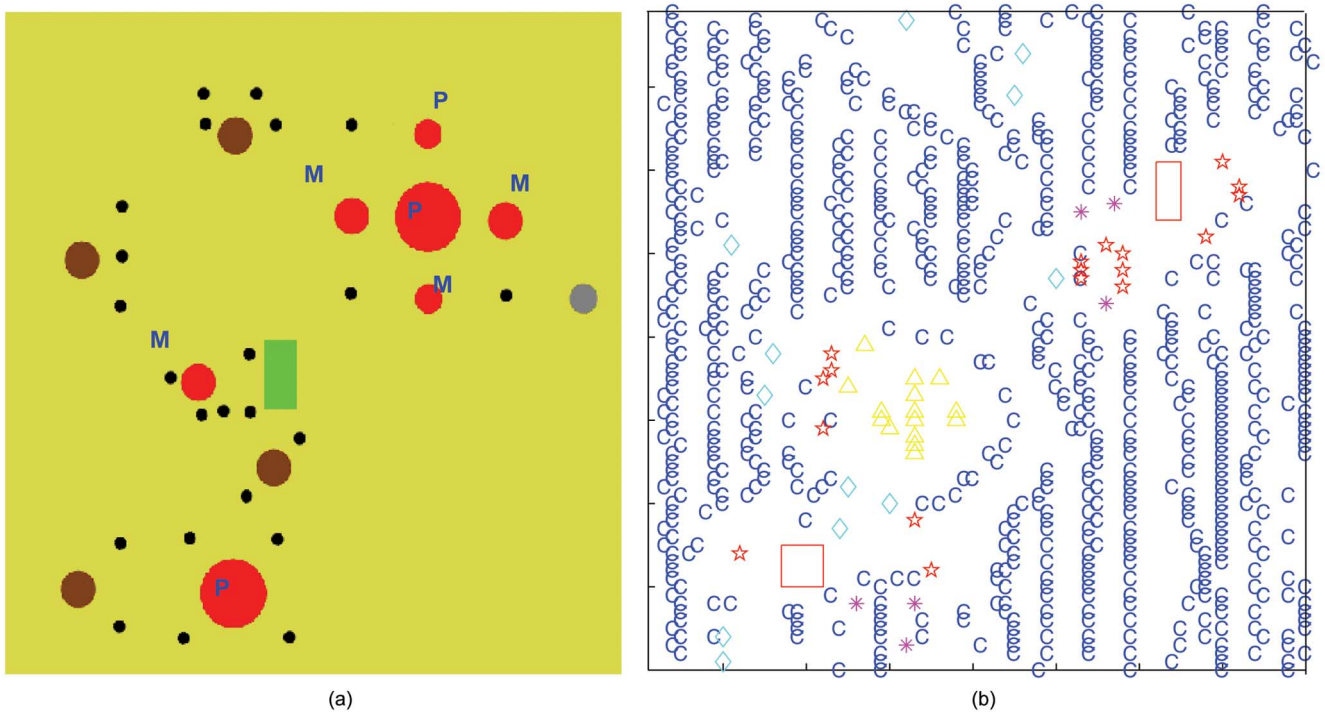


Fig. 14. Detection results of the lifelong learning in Minefield 3. (a) Ground truth. The red circles are landmines with “M” and “P” indicating metal and plastic mines, respectively; the other symbols represent clutter. Black dots are small metal segments, and the rest are large-sized metal or nonmetal clutter. (b) Detection result. Each red rectangle represents an oracle query and the corresponding grid-sensing region. Other marks are declarations: Blue “C”—“clean,” red pentagram—metal mine, pink star—plastic mine, yellow triangle—Type-1 clutter, and cyan diamond—Type-2 clutter.

a landmine declaration is correct or wrong, so it makes many wrong declarations with few sensing actions. On the contrary, if the penalty is high, the number of sensing actions must increase

to make the declarations more accurate. In addition, missing a landmine is more costly compared to other declarations, and hence, the agent would rather declare an object as a landmine

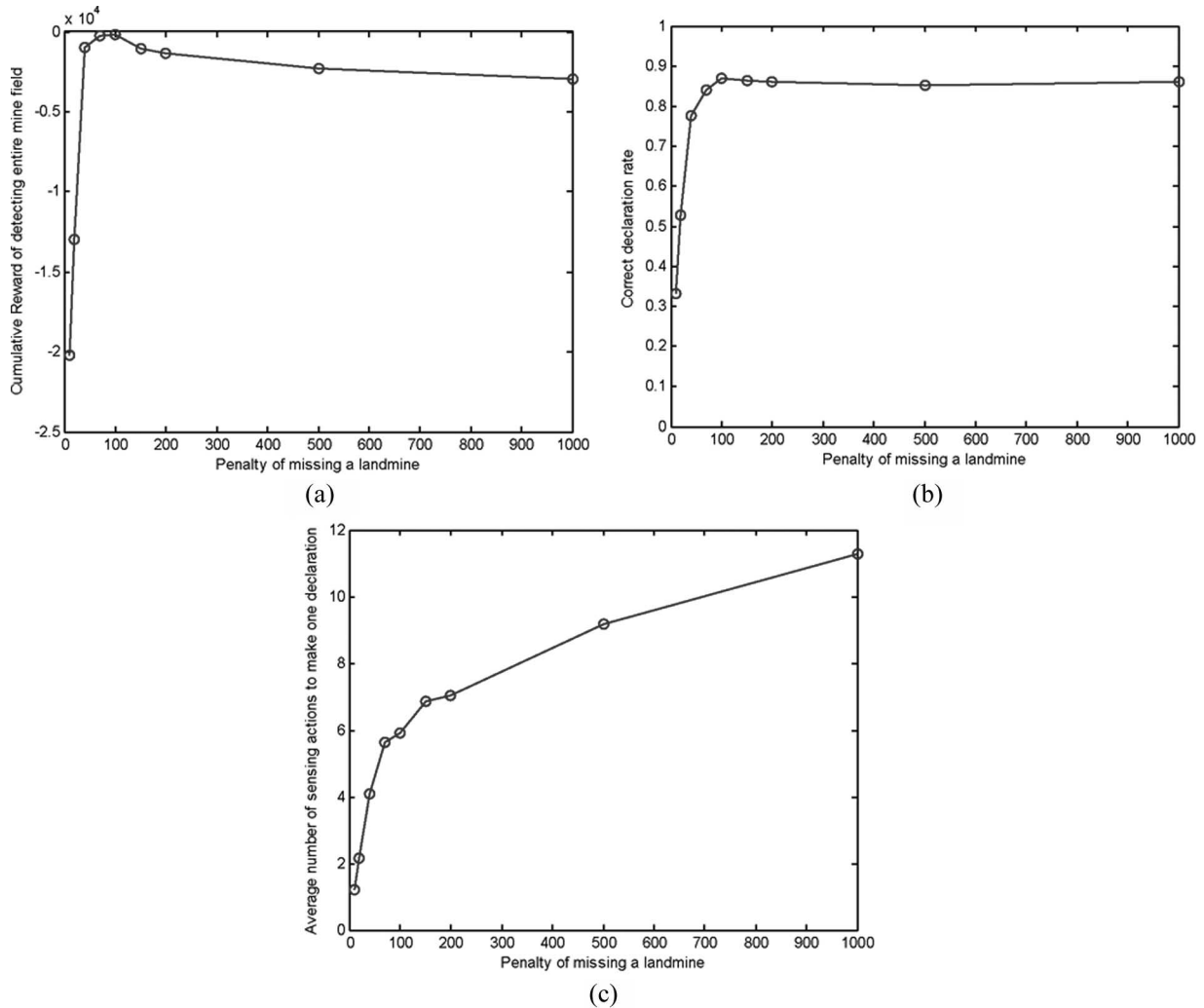


Fig. 15. Importance of setting an appropriate penalty of missing a landmine. (a) Cumulative reward of detecting the entire minefield. (b) Correct declaration rate. (c) Average number of sensing actions to make one declaration. In all the three figures, the values are evaluated at penalties $\{10, 20, 40, 70, 100, 150, 200, 500, 1000\}$.

than miss it, thus causing an increase of false alarms. The sensing cost and the false alarm penalty both reduce the cumulative reward.

VI. CONCLUSION

We have addressed the problem of employing GPR and EMI sensors placed on a single platform, with the objective of performing adaptive and autonomous sensing of landmines. The problem has been formulated in a POMDP setting, under two distinct assumptions. In the first case, we have assumed adequate and appropriate data for learning of the underlying POMDP models, with which policy design can be effected. The assumption that such data are available is often inappropriate, and therefore, we have also considered a lifelong-learning algorithm in which little if any *a priori* information is assumed with regard to the mines, clutter, and soil conditions. The formulation considered for this latter case has been based on the recently developed MEDUSA algorithm [18]. The work reported here is distinct from the original MEDUSA work in the following two principal ways: 1) In MEDUSA, the total

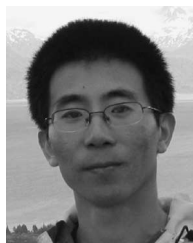
number of states is known in advance, and therefore, the model size is fixed; the algorithm only refines the model parameters based on the oracle query. In this paper, the total number of states is not fixed. The model size increases by adding more states and observations when new targets are revealed by the oracle query. The learning includes updating model parameters and increasing model size (if necessary). 2) In MEDUSA, it is assumed that the oracle reveals the underlying state directly, and the model is updated for only the involved single state. In this paper, the oracle reveals the target label instead of one single state. One target may involve several states, and the states are hidden. The target constructs a subset (a diagonal block in the state transition matrix) of the entire model. This subset (sub-model) is learned by the method discussed in Section III, since by grid sensing before the oracle reveals the label, sufficient training data are available for this target. Finally, the model update or expansion is performed on the subset of the model, not a single state.

The principal limitation of the approach developed here is the computational cost of implementing the POMDP policy. For the lifelong-learning algorithm addressed in Section IV-B,

we sampled $N = 10$ POMDP models; these characterized on average by 29 target states, 24 discrete observations, and 16 actions. The PBVI policy design required, on average, 58 min of CPU for each of these models (on a 3.06-GHz personal computer). Therefore, the principal challenge going forward is found in increasing the computational speed of policy design; there have been many recent improvements in POMDP policy design that will significantly accelerate the speed of policy design (see [27]–[29] and the references therein).

REFERENCES

- [1] J. MacDonald *et al.*, *Alternatives for Landmine Detection*. Santa Maria, CA: RAND's Sci. Technol. Policy Inst., 2003.
- [2] L. Carin, N. Geng, M. McClure, J. Sichina, and L. Nguyen, "Ultra-wideband synthetic aperture radar for mine field detection," *IEEE Antennas Propag. Mag.*, vol. 41, no. 1, pp. 18–33, Feb. 1999.
- [3] T. Yu and L. Carin, "Analysis of the electromagnetic inductive response of a void in a conducting-soil background," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1320–1327, May 2000.
- [4] T. P. Montoya and G. S. Smith, "Land mine detection using a ground-penetrating radar based on resistively loaded Vee dipoles," *IEEE Trans. Antennas Propag.*, vol. 47, no. 12, pp. 1795–1806, Dec. 1999.
- [5] P. Church, J. E. McFee, S. Gagnon, and P. Wort, "Electrical impedance tomographic imaging of buried landmines," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2407–2420, Sep. 2006.
- [6] J. Y. Song, Q. H. Liu, P. Torriero, and L. Collins, "Two-dimensional and three-dimensional NUFFT migration method for landmine detection using ground-penetrating radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1462–1469, Jun. 2006.
- [7] D. Potin, P. Vanheeghe, E. Duflos, and M. Davy, "An abrupt change detection algorithm for buried landmines localization," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 260–272, Feb. 2006.
- [8] Q. Zhu and L. M. Collins, "Application of feature extraction methods for landmine detection using the Wichmann/Niitek ground-penetrating radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 1, pp. 81–85, Jan. 2005.
- [9] K. Kastella, "Discrimination gain to optimize detection and classification," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 27, no. 1, pp. 112–116, Jan. 1997.
- [10] A. A. Abdel-Samad and A. H. Tewfik, "Search strategies for radar target localization," in *Proc. Int. Conf. Image Process.*, Oct. 1999, vol. 3, pp. 862–866.
- [11] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1, pp. 99–134, May 1998.
- [12] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *IEEE Trans. Signal Process.*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [13] P. D. Gadar, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 6, pp. 1231–1244, Jun. 2001.
- [14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
- [15] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, Univ. College London, London, U.K., 2003.
- [16] S. Ji, B. Krishnapuram, and L. Carin, "Variational Bayes for continuous hidden Markov models and its application to active learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 522–532, Apr. 2006.
- [17] M. J. Beal and Z. Ghahramani, "The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures," in *Bayesian Statistics 7*. London, U.K.: Oxford Univ. Press, 2003, pp. 453–464.
- [18] R. Jaulmes, J. Pineau, and D. Precup, "Active learning in partially observable Markov decision processes," in *Proc. ECML*, 2005, pp. 601–608.
- [19] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithms for POMDPs," in *Proc. IJCAI*, 2003, pp. 1025–1032.
- [20] Y. Zhang, L. Collins, H. Yu, C. Baum, and L. Carin, "Sensing of unexploded ordnance with magnetometer and induction data: Theory and signal processing," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 5, pp. 1005–1015, May 2003.
- [21] W. Scott, Jr., K. Kim, and G. Larson, "Investigation of a combined seismic, radar, and induction sensor for landmine detection," *J. Acoust. Soc. Amer.*, vol. 115, no. 5, p. 2415, May 2004.
- [22] W. Scott, Jr., K. Kim, G. Larson, A. Gurbuz, and J. McClellan, "Combined seismic, radar and induction sensor for landmine detection," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Sep. 2004, pp. 1613–1616.
- [23] W. H. Jefferys and J. O. Berger, "Sharpening Ockham's razor on a Bayesian stop," Dept. Statistics, Purdue Univ., West Lafayette, IN, Tech. Rep. 91-44C, 1991.
- [24] S. Thrun and L. Y. Pratt, Eds., *Learning to Learn*. Norwell, MA: Kluwer, 1998.
- [25] S. Thrun, "A lifelong learning perspective for mobile robot control," in *Proc. IEEE/RSJ/GI Conf. Intell. Robots and Syst.*, 1994, pp. 23–30.
- [26] M. H. DeGroot, *Probability and Statistics*, 2nd ed. Reading, MA: Addison-Wesley, 1986.
- [27] H. Li, X. Liao, and L. Carin, "Region-based value iteration for partially observable Markov decision processes," in *Proc. ICML*, 2006, pp. 561–568. [Online]. Available: <http://www.ee.duke.edu/~lcarin/Papers.html>
- [28] M. T. J. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *J. Artif. Intell. Res.*, vol. 24, pp. 195–220, 2005.
- [29] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 362–370.



Lihan He received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering at Duke University, Durham, NC.

His current research interests include Bayesian statistics, machine learning and data mining, decision making under uncertainty, and robot mapping and state estimation.



Shihao Ji (M'06) received the B.S. and M.S. degrees from Xidian University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree from Duke University, Durham, NC, in 2006, all in electrical engineering.

His research interests include sequential data processing with hidden Markov models, Bayesian inference, planning under uncertainty, and statistical signal processing. He is currently with Duke University.



Waymond R. Scott, Jr. (M'00) was born in Calhoun, GA, on April 6, 1958. He received the B.E.E., M.S.E.E., and Ph.D. degrees from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 1980, 1982, and 1985, respectively.

From 1979 to 1980, he was a Student Assistant and Graduate Research Assistant with the Georgia Tech Research Institute, and from 1980 to 1985, he was a Graduate Research Assistant with the School of Electrical Engineering at Georgia Tech, where he is currently a Professor of electrical and computer engineering. His research interests include methods for detecting buried objects using both electromagnetic and acoustic waves, measurement of the electromagnetic properties of materials, transient electromagnetic fields, and numerical methods, including the finite-element and the finite-difference time-domain techniques.

Lawrence Carin (SM'96-F'01) was born in Washington, DC, on March 25, 1963. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1985, 1986, and 1989, respectively.

In 1989, he joined the Electrical Engineering Department, Polytechnic University Brooklyn, NY, as an Assistant Professor and became an Associate Professor in 1994. In September 1995, he joined the Electrical and Computer Engineering Department, Duke University, Durham, NC, where he is currently the William H. Younger Distinguished Professor. He has been the Principal Investigator on several large research programs, including two Multidisciplinary University Research Initiative programs. He is the Cofounder of the small business Signal Innovations Group (SIG), which was purchased in 2006 by Integrian, Inc. He is the Director of Technology at SIG, which is now a subsidiary of Integrian. His current research interests include signal processing and machine learning for sensing applications.

Dr. Carin was an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION from 1996 to 2001. He is a member of the Tau Beta Pi and Eta Kappa Nu honor societies.