Adaptive Multi-Modality Sensing of Landmines

[1]Lihan He, [1]Shihao Ji, [2]Waymond Scott, Jr., and [1]Lawrence Carin

[1]Department of Electrical and Computer Engineering
Duke University, Durham, NC 27708
{lihan,shji,lcarin}@ece.duke.edu

[2]Department of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0250
waymond.scott@ece.gatech.edu

Abstract – The problem of adaptive multi-modality 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.

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 possibly 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 applied for handheld landmine detection. The GPR and EMI signatures of landmines and clutter are also a strong function of the soil characteristics [3,4], 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 a GPR and EMI sensor are deployed on the same unmanned platform. The task is for this system to navigate autonomously through a mine field, 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, 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 below, the cost associated with making classifications is performed within a Bayes-risk setting.

The basic objective may be cast in the form of an adaptive sensor-management problem [5,6] (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. We here consider a partially observable Markov decision process (POMDP) formalism [7]. 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 multi-sensor problem, *i.e.*, it accounts for the immediate expected cost, as well as discounted future costs, over an infinite horizon [7].

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 [7]. The belief is defined in terms of a belief state, a probability mass function (pmf) that reflects the probabilities $p(S_E)$ for all environmental states S_E, based upon all previous actions and observations [7]. To compute the belief state one requires an underlying model of the environment under test [7], characterized by a statistical representation of a sequence of actions and observations. For the work of interest here the necessity of an underlying model is a serious limitation, for the reasons discussed above: the specific types of mines and clutter that may 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 below, such statistical models are well characterized in terms of hidden Markov models (HMMs) [8,9] with action-dependent state-transition probabilities. In this application the target states S_T of the HMM are defined by sensor positions relative to the target, and the sequence of target states visited is modeled as a Markov process, conditioned on the sensor-platform motion; since the target position is unknown ("hidden"), the targets are modeled via an HMM. In this setting we must distinguish the overarching state of the environment under test S_E, which is to be inferred by the POMDP policy (via the belief state), *vis-à-vis* the hidden underlying states of the target model S_T, which are visited when performing the adaptive sensing.

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 number of target states S_T; we must also define the appropriate number of codes [10] for quantization of the observations, such that the observations are discrete, as required by the POMDP. To address the problem of determining the proper number of states S_T associated with a given target type, as well as the number of codes, we employ a variational-Bayes (VB) HMM analysis [11,12], 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) [13], from which the proper number of (data driven) states S_T 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 [14]. As the agent interrogates the environment with the multiple sensors, the belief state narrows down the conditions under test, to those actually under interrogation. This narrowing down of the belief state while sensing the environment is a form of exploration, with exploitation performed simultaneously [14]. A new action is introduced, with appropriate cost, characterized by calling an oracle to reveal a label for an item under interrogation, this allowing the underlying POMDP model to expand with the introduction of new mines, clutter and/or environmental (soil) conditions.

It is computationally expensive to perform an exploration-exploitation framework of the type summarized above (the number of environmental states 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 [14], also employing an oracle, 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, and new models are introduced in their place, through exploitation of the oracle [14]. We adopt a modified form of this framework in the work presented here, with specific application to the landmine-sensing problem. As demonstrated in the examples, we

use this approach to address multi-sensor (GPR and EMI) interrogation of a simulated mine field, 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. We here employ measured GPR and EMI data, for real mines and realistic clutter. The measured data considered in this study are available upon request, and therefore it is hoped that it will evolve to a standard data set 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 training data. In Section IV this is generalized further by assuming that the proper model is unknown, with model learning performed adaptively while sensing the environment ("lifelong" learning).

A POMDP model is represented by a six-element tuple $\{S, A, T, \Omega, O, R\}$ [7], 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

$$T(s, a, s') = \Pr(S_{t+1} = s' | S_t = s, A_t = a) \tag{2.1}$$

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

$$O(a, s', o) = \Pr(O_{t+1} = o \mid A_t = a, S_{t+1} = s') \tag{2.2}$$

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:

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

with the normalizing constant

$$c = \sum_{s' \in S} O(a, s', o) \sum_{s \in S} T(s, a, s') b_{t-1}(s) = \Pr(o \mid a, b). \tag{2.4}$$

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

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

which is accrued over a horizon of length *k*. 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 non-stationary. 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, independent of the agent's temporal position.

When in belief state *b*, the maximum expected reward *k* steps from the horizon $V^{(k)}$ is

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

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 PBVI algorithm, with details provided in [15].

III. The POMDP Model for Landmine Detection

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

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

3.1 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 [10], yielding the finite set of observations required for the POMDP.

3.1.1 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 [16]

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

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

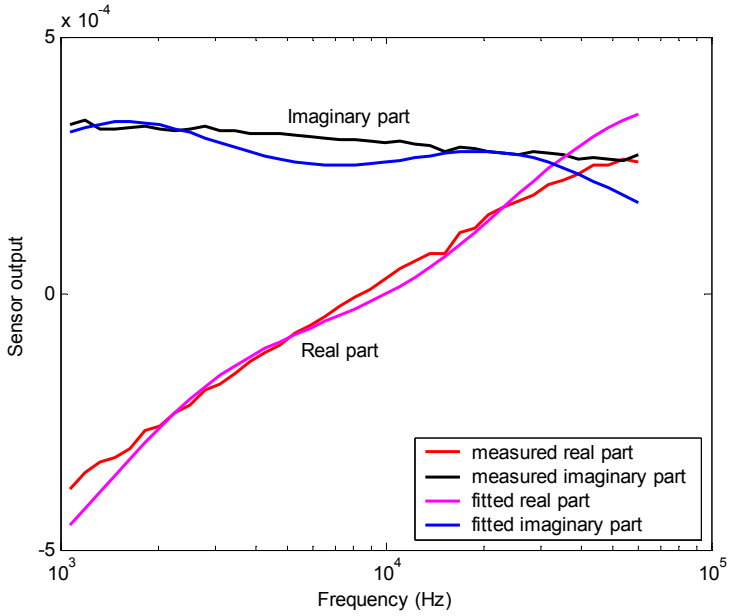


Fig.1. EMI response and model fit when the sensor is above a metal mine.

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 independently and identically distributed Gaussian noise (*i.e.*, minimizing the mean square error between the measured data and the model in (3.1)).

3.1.2 GPR features

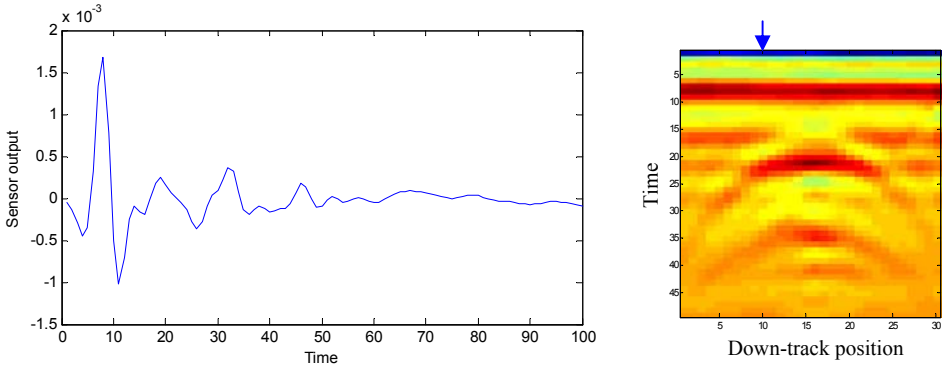
The GPR data for a given sensor position is assumed to be recorded in the time domain. Figure 2(a) shows a typical GPR observation when the sensor is above a plastic mine, and Fig. 2(b) is a 2-dimensional scan of the landmine signature. Features extracted from a GPR observation include the raw moments (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 moment and central moment features are

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

and

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

respectively, in which $f_2^{(cen)}$ reflects the variance of a GPR response, and $f_3^{(cen)}$ reflects the degree of



(a) (b)

Fig. 2. The GPR response when the sensor is above a plastic mine. (a) Amplitude vs. time signal in one position. The time axis is sampled at a rate of 0.05ns, with the full waveform extending over 5ns. The first peak corresponds to the reflection from the ground surface. (b) 2-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).

asymmetry of the wave. Moments higher than 3rd order were found to not 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 [17,18].

3.2 Specification of states *S*

In most cases a landmine is cylindrically symmetric and buried with axis perpendicular or near-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.

The robot is assumed to move on the (flat) 2-dimensional 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 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 (b) does not allow the POMDP to distinguish this difference. Therefore, we divide each annulus into four sectors, corresponding to 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 state structure considered here, and the remaining question is how many states we should use to represent a given target, with this

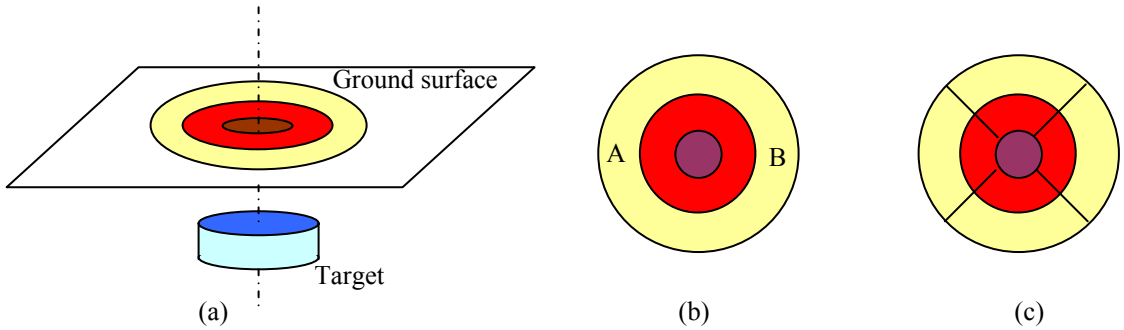


Fig. 3. Definition of state structure. (a) Illustration of the simple state definition. The states are concentric annuli on the ground surface, with center above the center of a mine or clutter. (b) Platform of the simple state definition. Points A and B are in the same state, although the robot should 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.

addressed in Section 3.5. We note that the states discussed above are the target states S_T from the Introduction, and there will be a set of these states for each type of overarching environmental state S_E (*e.g.*, a set of S_T states for a particular kind of mine, this mine defining an associated S_E state). The states s referred to in Sec. II correspond to the environmental states S_E, and the probability of being in any given S_E state is equal to the sum of the probabilities of being in the associated set of S_T states.

3.3 Specification of observations Ω

The discrete set of possible observations Ω is obtained as the codebook resulting from the vector quantization [10] 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.

3.4 Specification of actions A

In our experiments we consider five types of mines and clutter: metal mines, plastic mines, and three types of non-mines (clutter). 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 results, some of the clutter is non-metallic, but the associated signature has properties that may be characterized by the Type-1 and Type-2 classes discussed above. The third type of non-mine corresponds to a "clean" region, which means no mine or mine-like 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 robot is assumed to have 15 actions, *i.e.*, $A = \{1, 2, \ldots, 15\}$, of which the first 10 are sensing actions and the rest are declaration actions; Table 1 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

Table 1: 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"	

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).

3.5 Determination of the number of states $|\boldsymbol{S}|$ and the codebook size $|\boldsymbol{\Omega}|$

The number of 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 variational Bayesian (VB) expectation-maximization (EM) method for model selection [13], which allows us to compute an approximation of the "evidence" for different $|S|$ and $|\Omega|$.

3.5.1 Variational Bayesian 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 unobservable. The traditional EM algorithm gives a maximum likelihood (ML) or maximum *a posteriori* (MAP) point estimate, which does not express the posterior parameter uncertainty. Rather than a point estimate of the model parameters, we desire the full posterior via Bayes rule:

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{3.4}$$

where $\boldsymbol{y}$ is measured data, $\boldsymbol{\theta}$ denotes model parameters, and $p(\boldsymbol{\theta})$ is the prior distribution over parameters.

Given measured data $\boldsymbol{y}$ and several candidate models M_1, M_2, …, M_n with different structures, the goal of model selection is to decide which model fits the data best. One criterion for this selection is comparing the marginal likelihood of the data $\boldsymbol{y}$ for each model, and choosing the model that has the highest likelihood. The marginal likelihood is also called the "evidence" [13], and is expressed as

$$p(\boldsymbol{y} \mid M) = \int p(\boldsymbol{y} \mid \boldsymbol{\theta}, M) p(\boldsymbol{\theta} \mid M) d\boldsymbol{\theta} \,. \tag{3.5}$$

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

$$p(M_i \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid M_i)p(M_i)}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{y} \mid M_i)p(M_i)}{\sum_i p(\boldsymbol{y} \mid M_i)p(M_i)}. \tag{3.6}$$

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

The marginal likelihood (3.5) is difficult to evaluate because the integral is typically intractable analytically. The variational Bayesian (VB) method [13] provides an approach to compute the lower bound of this marginal likelihood, by introducing a factorized distribution $q(\boldsymbol{x},\boldsymbol{\theta}) \approx q_x(\boldsymbol{x})q_\theta(\boldsymbol{\theta})$ to approximate the true distribution $p(\boldsymbol{x},\boldsymbol{\theta} \mid \boldsymbol{y}, M)$, where $\boldsymbol{x}$ denotes the hidden variables (states) and $\boldsymbol{y}$ denotes the observed variables (measured data). Details with regard to VB applied to HMMs may be found in [12].

3.5.2 Determining |S| and |Ω|

We now address the problem of determining the number of states and the codebook size, 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 to estimate the model parameters, as shown in Fig. 4(a). The data are assumed to be represented by a hidden Markov model (HMM) with two sets of observations, as shown in Fig. 4(b). We assume that the GPR and EMI observations share the same underlying 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 sequence and horizontal sequence (see Fig. 4(a)) are

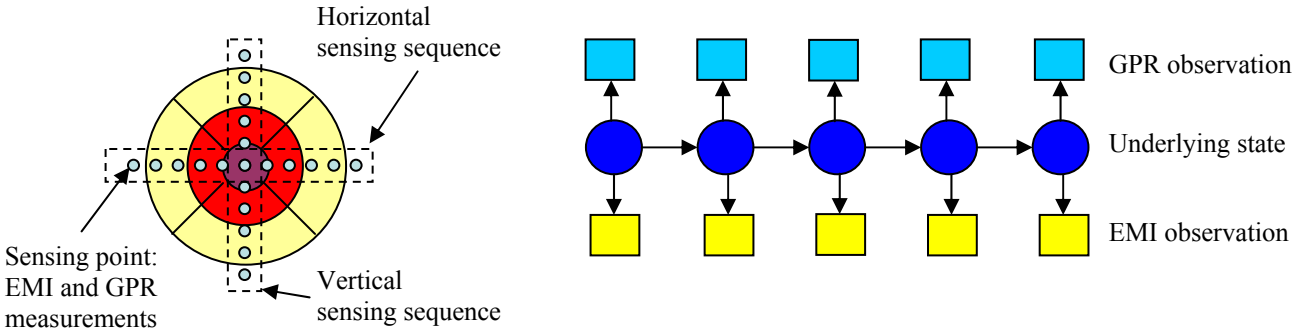


Fig. 4. HMM for model selection. (a) Illustration of sensing data positions over a target. The dots are sensing points. The horizontal and vertical sensing sequences pass through the center of the target. The concentric annular sectors with different colors represent different states. (b) The underlying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

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 non-stationary 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 $|S|=1,5,9,13,\ldots,4K+1$ target states, corresponding to $1,2,3,4,\ldots,K+1$ annuli, respectively, and we consider codebook sizes $|\Omega|=2,3,4,\ldots,N$, 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.

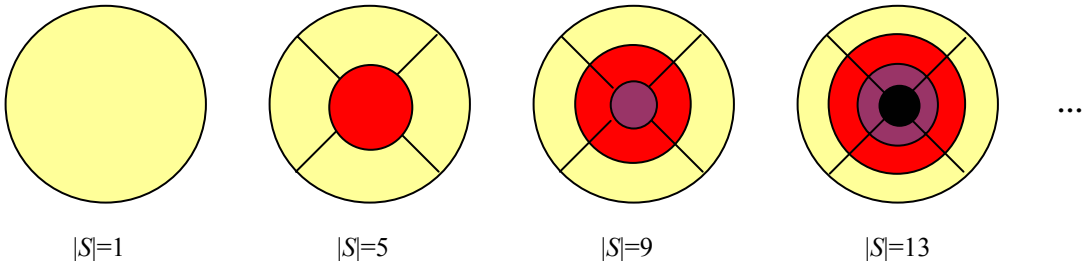
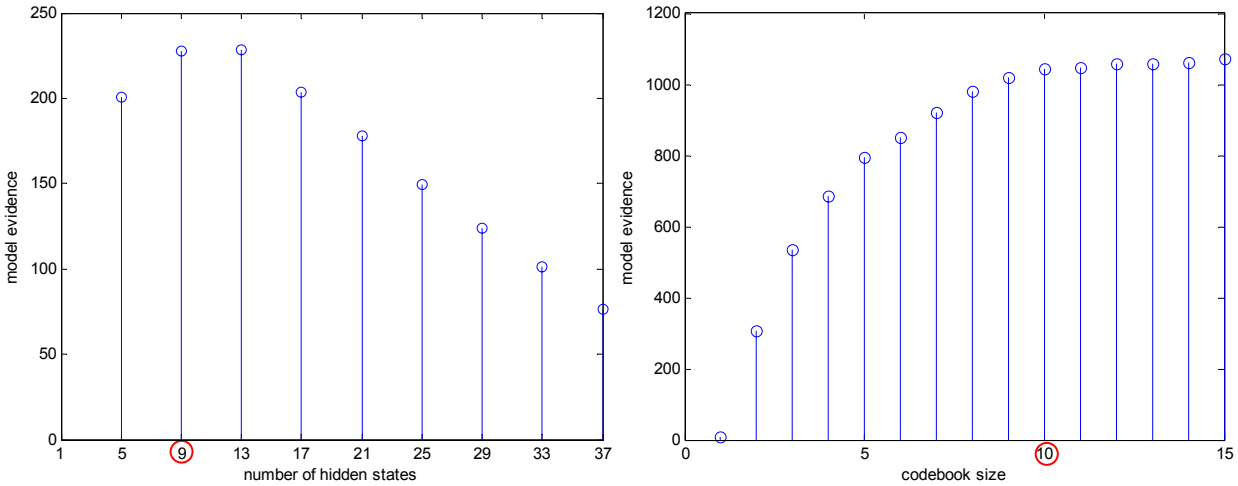


Fig. 5. Candidate models for different number of states $|S|=1,5,9,13,\ldots$.

To find the best $|S|$ and $|\Omega|$, we need to compute the model evidence for all the combinations of $|S|$ and $|\Omega|$, $N(K+1)$ 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 $|S|$, and then fix $|S|$ as the optimal value from the last step and find the optimal $|\Omega|$. This search terminates when $|S|$ and $|\Omega|$ are both unchanged. To avoid local minima, this procedure may 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 $|S|$ and $|\Omega|$ of metal mines using the iterative approach is shown in Fig. 6 (after convergence), in which we choose $|S|=9$ for a fixed codebook size $|\Omega|=10$ and choose $|\Omega|=10$ for a fixed number of states $|S|=9$. 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 [19] 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 $|S|$ is determined as described above. At the same time, the optimal state sequence of the target can also be estimated by the Viterbi algorithm [8], which gives an idea about the size (radii) of the annuli. By estimating $|S|$ and the annulus radius for each of the five types of mines and clutter discussed above, we define a total of 29 states as described in the next subsection. Similarly, we determine $|\Omega|=12$ for both EMI data and GPR data.



(a) (b)

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 $|S|=9$. (b) Selection of the codebook size when the number of state is $|S|=9$. The evidence remains stable after $|\Omega|=10$. In both figures, the model evidence is the logarithm of the marginal likelihood, apart from a constant.

3.6 Estimation of *T* and *O*

Across all five types of mines and clutter considered, we define a total of 29 target states, *i.e.*, $S=\{1,2,\ldots,29\}$. The 29 states are divided into 5 disjoint subsets: $S = S_m \bigcup S_p \bigcup S_{t1} \bigcup S_{t2} \bigcup S_c$, denoting respectively states of metal mines, plastic mines, Type-1 clutter, Type-2 clutter, and "clean"; the number of states in each of the five subsets is 9, 9, 9, 1 and 1, respectively. The definition of the states is illustrated in Fig. 7(a).

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. Figure 7(b) illustrates how the transition probabilities $T(s=5, a, s')$ for the two sensing actions involving

"walk south" are computed. Figure 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 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

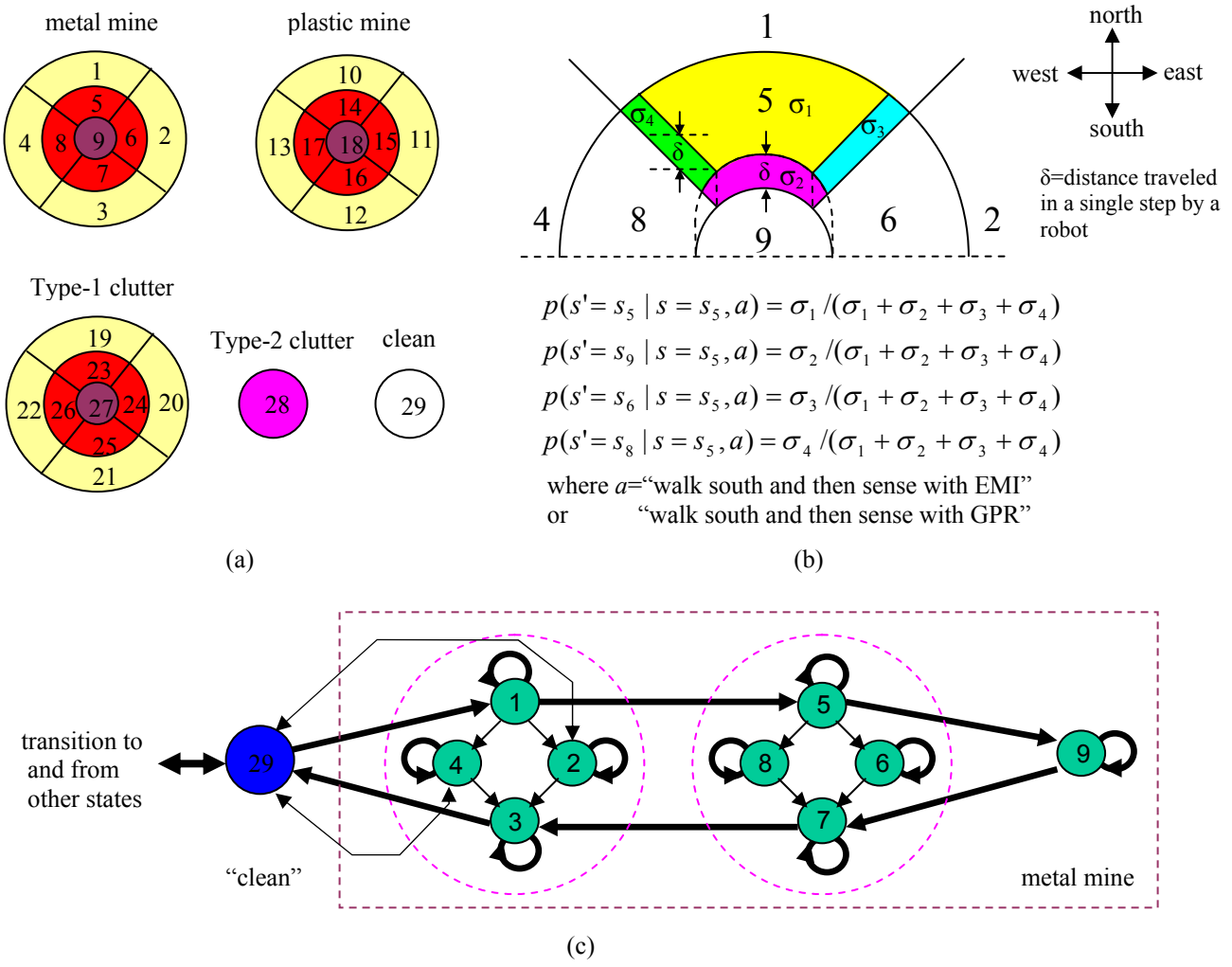


Fig. 7. State definition and transition-probability estimation for the landmine-detection problem. (a) Definition of the states. Metal mine, plastic mine and Type-1 clutter (large-sized metal clutter) are each modeled by 9 states, indexed 1 to 9, 10 to 18, and 19 to 27, respectively; Type-2 clutter (small-sized metal segment) is modeled by a single state (state 28); state 29 is used to indicate "clean" (*i.e.*, there are no mine or mine-like 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. σ_1, σ_2, σ_3 and σ_4 denote the 4 borders 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.

single state; there is a possibility to transition to this state from "clean"). 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 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.

3.7 Specification of reward *R*

The reward function *R* considered in the subsequent examples is defined in Table 2.

Table 2: Reward function *R,* where the states here correspond to the possible environmental states S_E.

Action / State	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

All sensing actions have a cost of -1, although in general, we can set different costs for the two sensors. In this paper we set them equal to see how the policy selects sensors for different type of targets, without the disturbance of cost differences.

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. The units in Table 2 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 termed "lifelong learning" [14,20,21].

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 mine field (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 below is motivated and modified from the MEDUSA algorithm [14].

4.1 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 [22]. 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, with repetitions; the probabilities of the n items are given respectively by $\boldsymbol{p}=(p_1,\ldots p_n)$. 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 $\boldsymbol{p}$.

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

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

with $p_1,\ldots,p_n \geq 0$, $\sum_{i=1}^{n} p_i = 1$, $u_1,\ldots u_n \geq 0$, and the normalizing constant $c(\boldsymbol{u}) = \frac{\prod_{i=1}^{n}\Gamma(u_i)}{\Gamma(\sum_{i=1}^{n} u_i)}$.

The mean of the Dirichlet distribution is

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

Given *iid* data $\boldsymbol{y}=\{y_1,\ldots,y_m\}$ drawn from a multinomial distribution with parameters $\boldsymbol{p}$, with prior on $\boldsymbol{p}$ represented by $Dir(\boldsymbol{p};\boldsymbol{u})$, the posterior distribution of $\boldsymbol{p}$ is represented by an update of the Dirichlet distribution, $Dir(\boldsymbol{p};\tilde{\boldsymbol{u}})$, which is computed by the counting process:

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

where $indicator(z)=1$ if z is true, and $indicator(z)=0$ 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.

4.2 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

$$p_i^{T,s,a} = p(s' = s_i \mid s,a) = T(s,a,s_i), \text{ for } i = 1,\ldots,|S|, \tag{4.4}$$

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

$$p_i^{O,s',a} = p(o = o_i \mid a, s') = O(a, s', o_i), \text{ for } i = 1,...,|\Omega|. \tag{4.5}$$

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 4.1, for each state-action pair in the transition probability T or the observation function O, a Dirichlet prior is assigned

$$T(s, a, \cdot) \sim Dir(\boldsymbol{p}^{T,s,a}; \boldsymbol{u}_{T,s,a}), \tag{4.6}$$

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

Learning is a process of continuously updating the hyper-parameters $\boldsymbol{u}_{T,s,a}$ and $\boldsymbol{u}_{O,s',a}$.

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. If the mine/clutter type has been seen previously, the associated model is refined based upon the new measured data.

An oracle query is performed if one of the following conditions is satisfied. First, if the policy says that the oracle query is the optimal action at the current step. Secondly, if the agent finds a new observation that has never been seen before; this is totally new knowledge and is unaccounted for in the existing model. Thirdly, if the agent has measured extensively in a sub-area (the area from the last declaration position to the current sensing position within the lane, as defined in Section V) and still cannot make a decision about the underground target, which means the current task is too difficult for the agent.

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 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 is new, a sub-model is then built, similar to the discussion in Section III, using the grid-sensing data and the label information; the sub-model is a subset of the entire model, composed of those states and observations related to the current target. Note that in the sub-model, we learn the Dirichlet hyper-parameters which represent the posterior distributions of $T^{(sub\text{-}model)}$ and $O^{(sub\text{-}model)}$ (as computed using variational Bayes).

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. This is done by increasing S and Ω (if necessary), and adding a new diagonal block in the transition probability T (see Section 3.6) and new states (and observations) in the observation function O. The associated hyper-parameters 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 hyper-parameters for this target type, at learning rate λ:

$$\widetilde{u}_{T,s,a}(s') = u_{T,s,a}(s') + \lambda u_{T,s,a}^{(sub-model)}(s'), \tag{4.8}$$

$$\widetilde{u}_{O,s',a}(o) = u_{O,s',a}(o) + \lambda u_{O,s',a}^{(sub-model)}(o), \tag{4.9}$$

where $\boldsymbol{u}_{T,s,a}^{(sub-model)}$ and $\boldsymbol{u}_{O,s',a}^{(sub-model)}$ are the sub-model hyper-parameters learned from the measured data for current target type, $\boldsymbol{u}_{T,s,a}$ and $\boldsymbol{u}_{O,s;,a}$ are the old hyper-parameters for the existing model, and $\widetilde{\boldsymbol{u}}_{T,s,a}$ and $\widetilde{\boldsymbol{u}}_{O,s;,a}$ are the updated hyper-parameters for the posterior distributions. The above two learning approaches 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 λ=0, the agent never updates the model, and if $\lambda \to \infty$, 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. 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 $h = \{a_1, o_1, a_2, o_2, ...\}$ record the action-observation sequence as the agent experiences the environment. Denote weights $\{w_i\}_{i=1}^N$ as the

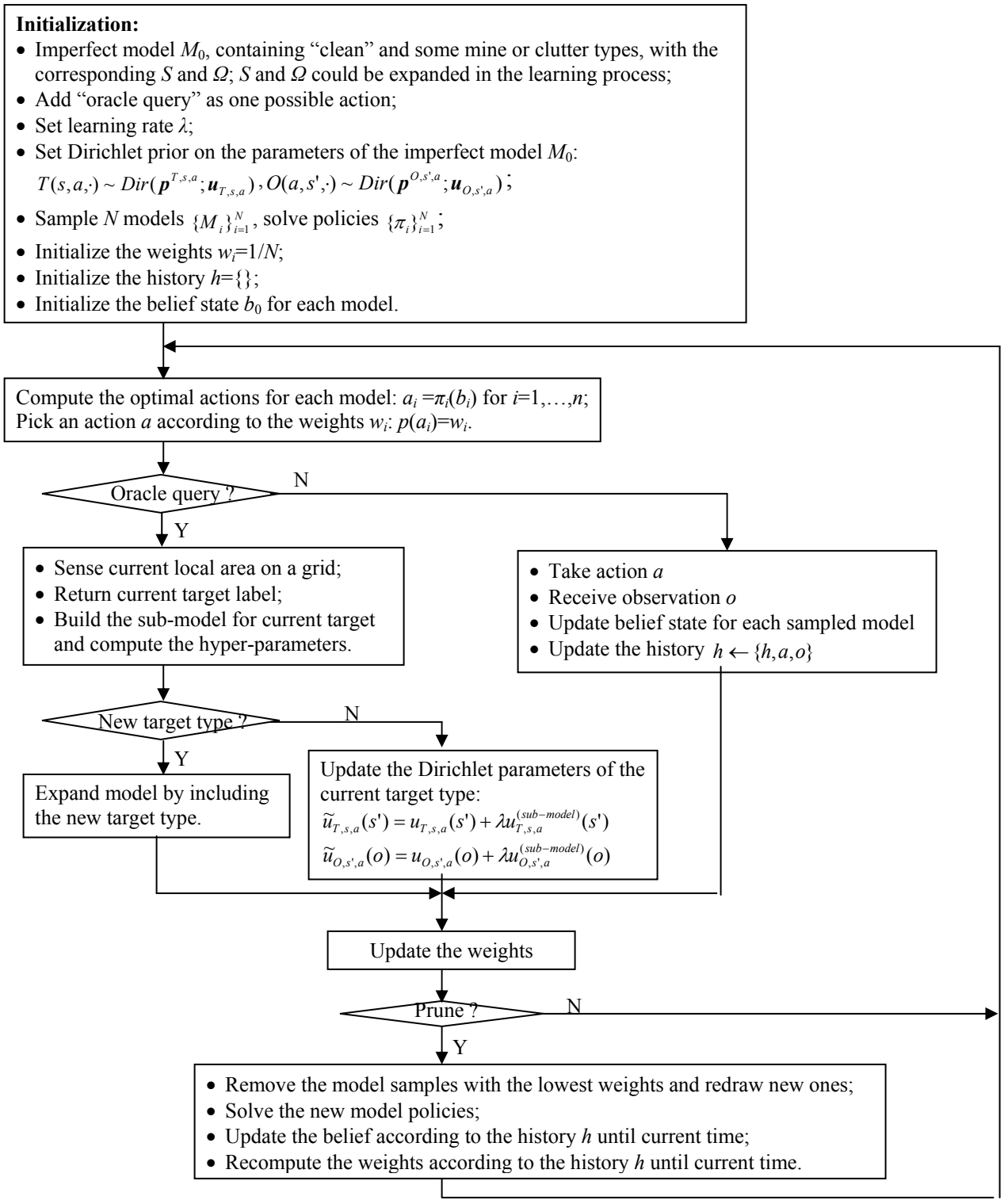


Fig. 8. Flowchart of lifelong learning in the landmine-detection problem.

normalized likelihood of the history h for each of the N models, computed at each step using the forward-backward algorithm [8] 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,…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 hyper-parameters. During the processes of updating the hyper-parameters 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 [15] 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 robot is navigated in three simulated mine fields. The EMI and GPR data are pre-collected over a $1.6\times1.6\text{m}^2$ per simulated mine field, with sensor data collected at a 2 cm sample rate in two coordinate dimensions. The pre-collected data are used to simulate the data collected by an autonomous two-sensor agent, as it senses within the mine field. The three simulated mine fields are shown below in Figs. 11, 13 and 14. Clearly, to avoid missing landmines the robot should search almost everywhere in a given mine field. However, we hope that the robot can actively decide where to sense as well as which sensor to use, to minimize the detection 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.

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 model we build does not fit the data in this area, possibly because our model does not include the current underground target. 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 make a reluctant declaration. We let

the robot declare “unknown” in this situation, while in the lifelong learning algorithm the “oracle” is employed.

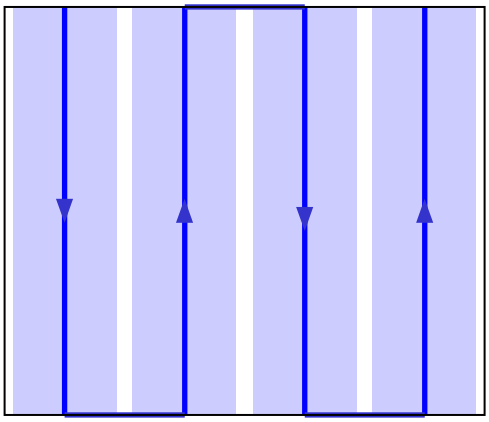


Fig. 9. Robot navigation path in a mine field. The dark blue curve is the “basic path”, which defines the lanes as indicated by light blue. The robot is restricted to move along the lanes by taking actions within the lanes. The “basic path” restrains the robot from moving across the lanes.

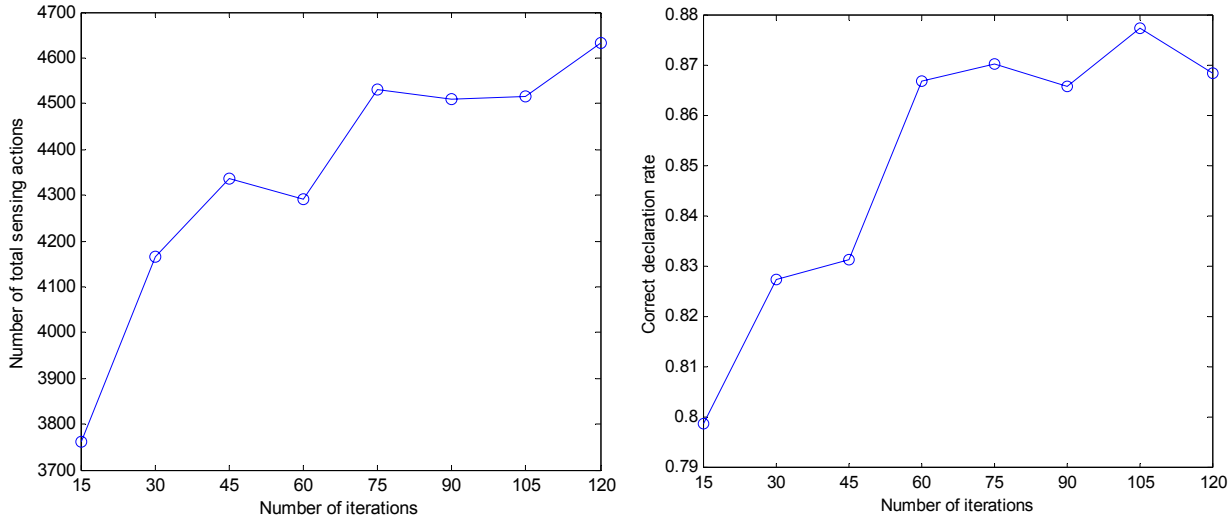
5.1 Detection performance of the offline-learning algorithm

In the offline-learning approach the training data are given in advance, and the training phase and test phase are separate. We use Mine Field 1 (Fig. 11) as the training data to learn the model and the policy, and then test our method on all three mine fields. The training data and test data match well in that the three mine fields 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.

5.1.1 Model training and policy design

Using Mine Field 1 as the training data set the POMDP model is built according to Section III, and the policy is learned by PBVI. The number of sensing actions and the correct declaration rate as a function of iteration number 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.

5.1.2 Landmine detection results



(a) (b)

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.

The stationary policy from the last subsection is then used to navigate the robot in the three simulated mine fields. The ground truth and detection results are summarized in Table 3. As an example, the layout of Mine Field 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.

Table 3: Ground truth and detection results on three mine fields

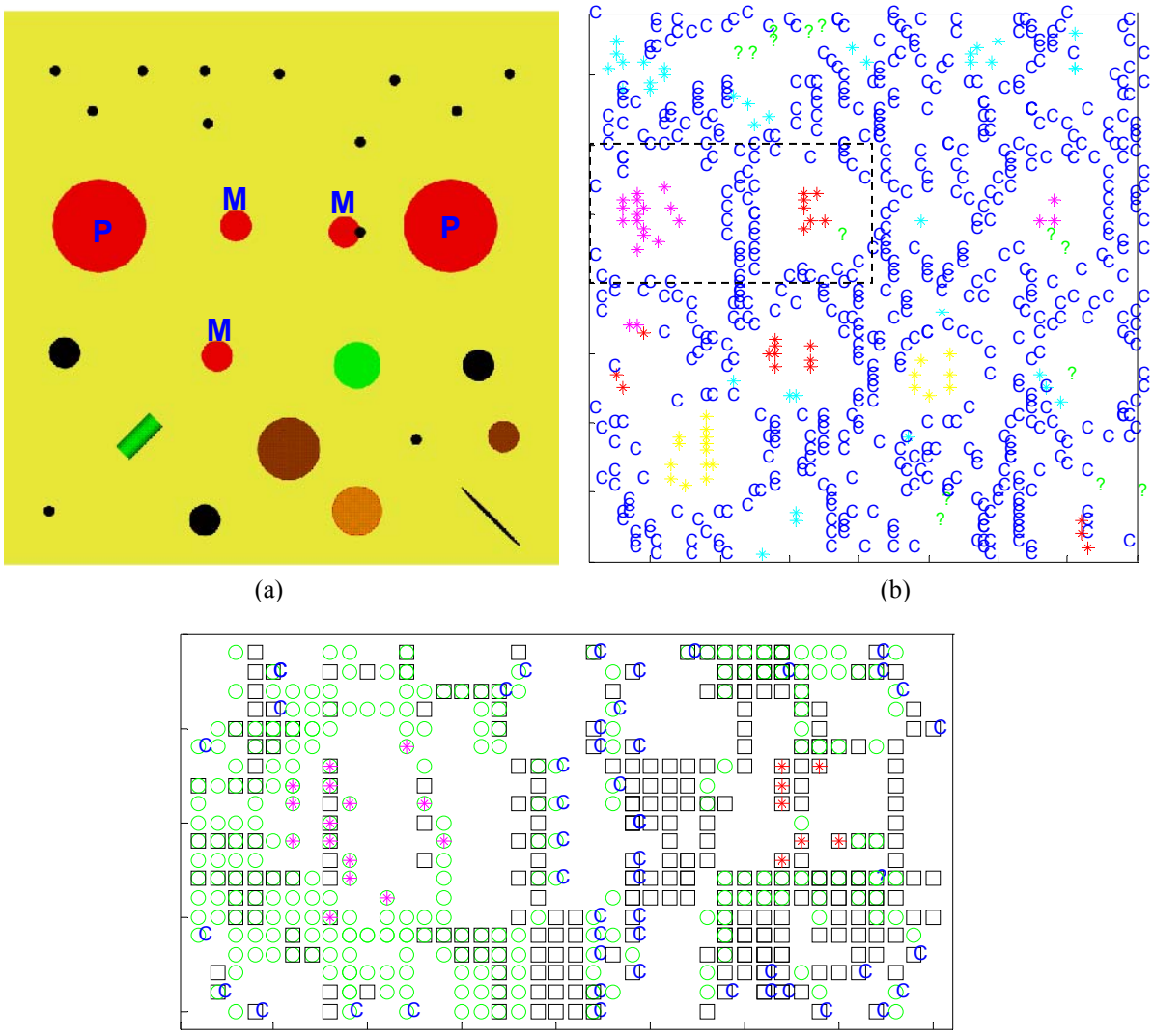
		Mine Field 1	Mine Field 2	Mine Field 3
Ground truth	Number of mines (metal + plastic)	5 (3+2)	7 (4+3)	7 (4+3)
	Number of clutter (metal + nonmetal)	21 (18+3)	57 (34+23)	29 (23+6)
Detection result	Number of mines missed	1	1	2
	Number of false alarms	2	2	2

Missed landmines are usually caused by one of the following two reasons: the mine has very weak signal in both EMI and GPP responses, such as a small anti-personnel 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 mine field; the correct declaration rate is about 0.87 (see Fig. 10). As a comparison, if a myopic policy is applied, where the agent considers only one step ahead to select actions, 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



(a) (b)

(c)

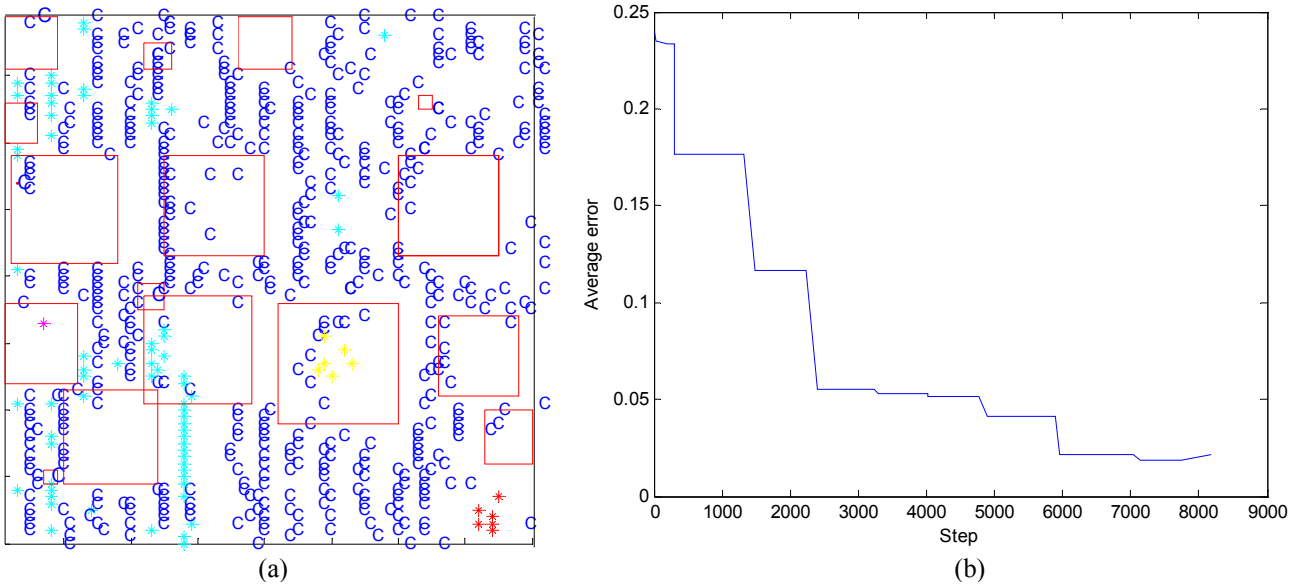
Fig. 11. Ground truth and detection details in Mine Field 1. (a) Ground truth. The red circles are landmines, with "M" and "P" indicating metal mine and plastic mine, respectively; the other symbols represent clutter. Black dots are small metal segments and the rest are large-sized metal or nonmetal clutter. (b) Declaration result. The blue "C" means a declaration of "clean", the green "?" means "unknown", and the stars with various colors represent declarations of mines or clutter. Red star: metal mine; pink star: plastic mine; yellow star: Type-1 clutter; cyan star: 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)).

measurements is $2\times800^2 = 12800$.

5.2 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 Mine Field 1, navigated by the policies within the lanes defined by the "basic path". We set the learning rate as λ=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 3.7. 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 Mine Field 1 are shown in Fig. 12, where (a) shows the positions of the oracle queries and the other declarations when the robot explores and exploits the environment, and (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 5.1.1; 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

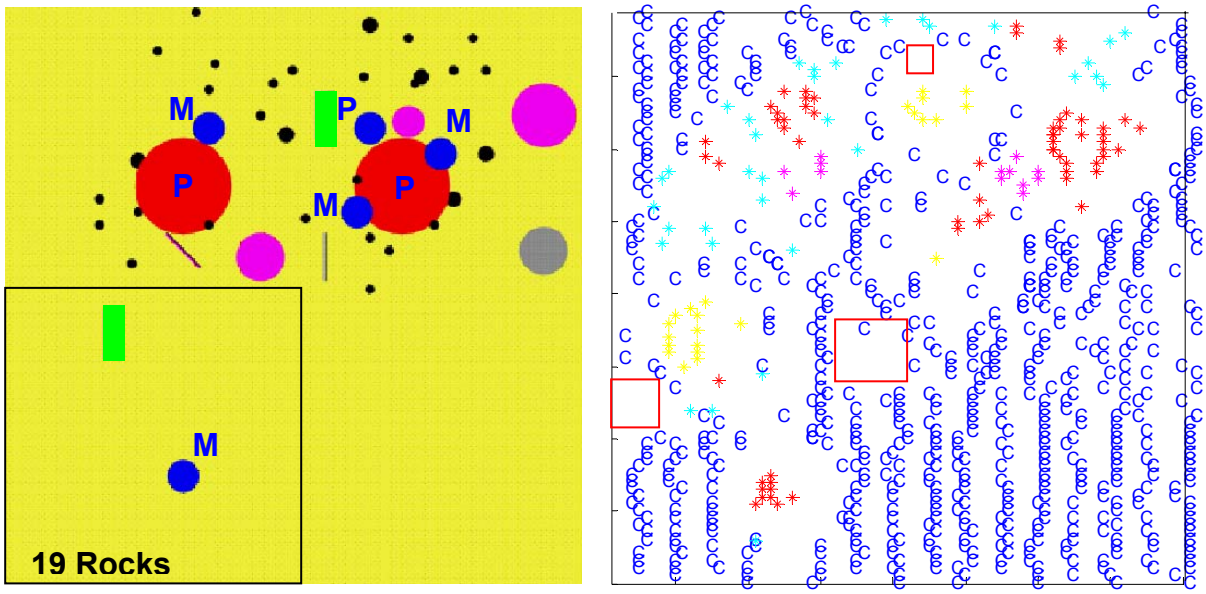


(a) (b)

Fig. 12. Detection results of the lifelong learning in Mine Field 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 star -- metal mine, pink star -- plastic mine, yellow star -- Type-1 clutter, and cyan star -- Type-2 clutter. The ground truth of the mine field 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.

beginning (left part of the Fig. 12(a)) of the learning, many wrong "Type-2 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 hyper-parameters 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 Mine Field 1. When a new task comes, the robot continues to modify the model parameters if the old model does not fit the new mine field.

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

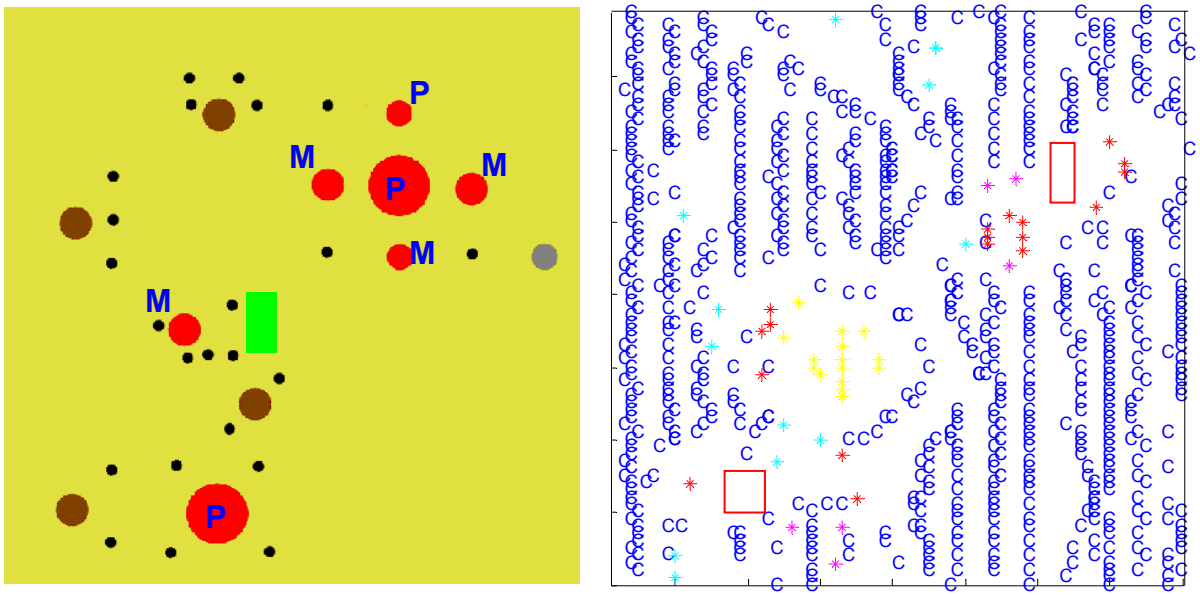


(a) (b)

Fig. 13. Detection results of the lifelong learning in Mine Field 2 after the algorithm has learned the model from Mine Field 1. (a) Ground truth. The red and blue 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) 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 star -- metal mine, pink star -- plastic mine, yellow star -- Type-1 clutter, and cyan star -- Type-2 clutter.

Finally, we discuss the immediate reward. It is assumed that during the exploitation in a mine field, 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.



(a) (b)

Fig. 14. Detection results of the lifelong learning in Mine Field 3. (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) Detection result. Each red rectangle represents an oracle query and the corresponding grid-sensing region. Other marks are declarations: blue "C" -- "clean", red star -- metal mine, pink star -- plastic mine, yellow star -- Type-1 clutter, and cyan star -- Type-2 clutter.

5.3 The importance of setting the reward function

Setting the reward function is important to 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.

Consider the critical role of the penalty when the robot misses a landmine. Refer to the previous setting in Section 3.7, where this penalty is -100. Suppose we keep the other reward values the same, but let this penalty

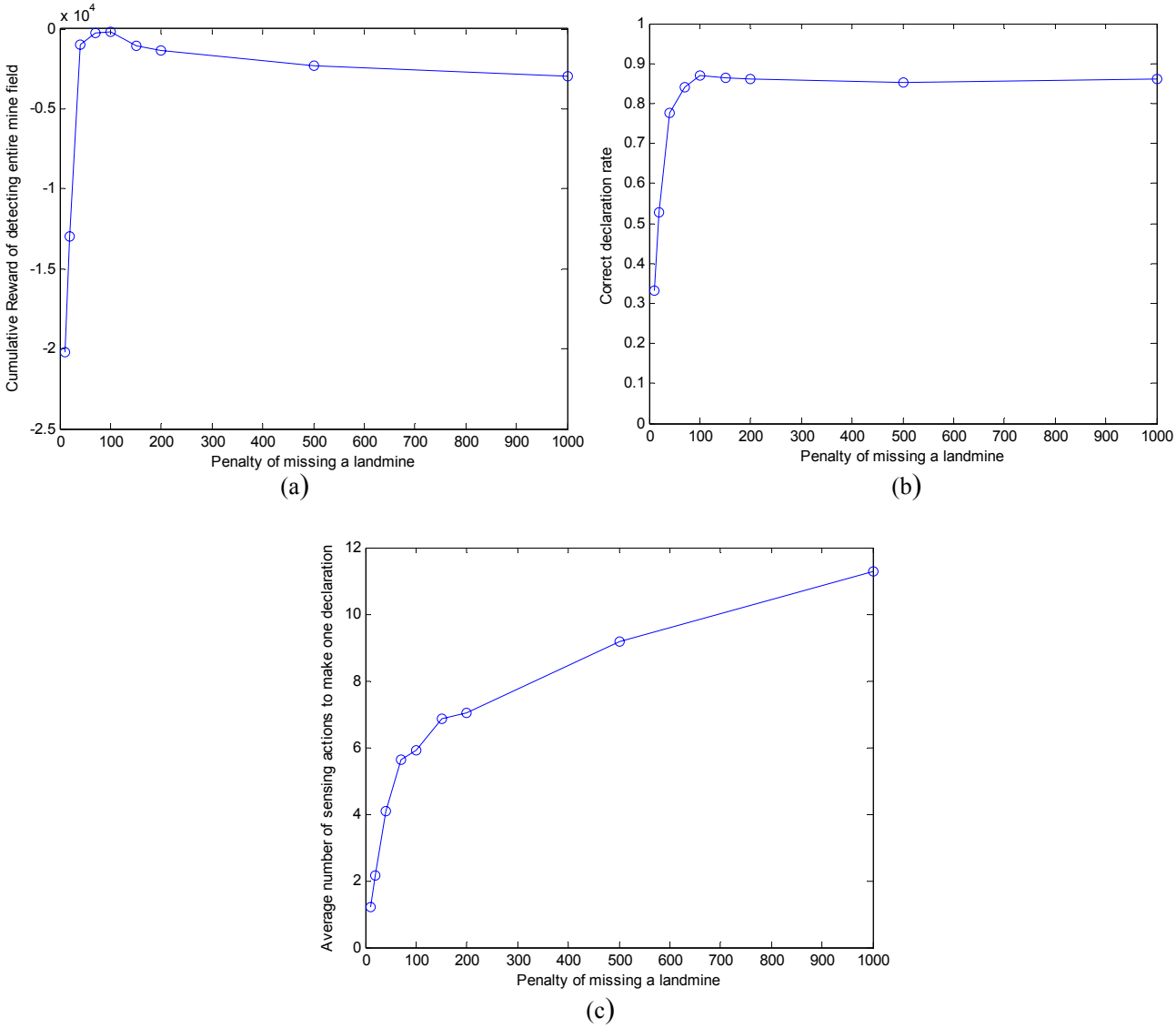


Fig 15. The importance of setting an appropriate penalty of missing a landmine. (a) Cumulative reward of detecting the entire mine field. (b) Correct declaration rate. (c) Average number of sensing actions to make one declaration. In all the three figures, the values are evaluated at penalties {10, 20,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

vary from -10 to -1000. Given Mine Field 1 as the training data, the model and the corresponding policies are learned and the robot executes the policies when detecting in the same mine field. 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 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 than miss it, thus causing an increase of false alarms. The sensing cost and the false alarm penalty both reduce the cumulative reward.

VI. Conclusions

We have addressed the problem of employing ground-penetrating radar (GPR) and electromagnetic induction (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 partially observable Markov decision process (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 [14]. The algorithms have been tested, with encouraging performance, on measured EMI and GPR data from simulated mine fields.

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 4.2, 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 minutes of CPU for each of these models (on a 3.06 GHz PC). 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 [23] and the references therein).

References

[1] J. MacDonald *et al.*, *Alternatives for Landmine Detection*, RAND's Science and Technology Policy Institute, 2003.

[2] L. Carin, N. Geng, M. McClure, J. Sichina, and L. Nguyen, "Ultra-wideband synthetic aperture radar for mine field detection," *IEEE Antennas and Propagation Magazine*, vol. 41, pp. 18-33, Feb. 1999.

[3] T. Yu and L. Carin, "Extended-Born method for the modeling of buried voids," *IEEE Trans. Geoscience and Remote Sensing*, vol. 38, 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 Propagat., vol.* 47, pp. 1795-1806, Dec. 1999.

[5] K. Kastella, "Discrimination gain to optimize detection and classification," *IEEE Trans. Syst., Man,*

Cybernetics – Part A: System and Humans, vol. 27, pp. 112-116, Jan. 1997.

[6] A.A. Abdel-Samad and A.H. Tewfik, "Search strategies for radar target localization," *Proc. 1999 International Conf. Image Proc*., vol. 3, pp. 862-866, Oct. 1999.

[7] L. P. Kaelbling, M. L. Littman and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99-134, 1998.

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *IEEE Trans. Signal Processing*, vol. 77, pp. 257--286, Feb. 1989.

[9] P. D. Gadar, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *Geoscience and Remote Sensing*, vol. 39, pp. 1231-1244, Jul. 2001.

[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Press/Springer, 1992.

[11] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*, PhD thesis, 2003.

[12] S. Ji, B. Krishnapuram and L. Carin, "Variational Bayes for continuous hidden Markov models and its application to active learning," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 28, pp. 522-532, Apr. 2006.

[13] M. J. Beal and Z. Ghahramani, "The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures," *Bayesian Statistics 7*, pp. 453-464, Oxford University Press, 2003.

[14] R. Jaulmes, J. Pineau, and D. Precup "Active learning in partially observable Markov decision processes," *Proceedings of ECML*, pp. 601-608, 2005.

[15] J. Pineau, G. Gordon and S. Thrun, "Point-based value iteration: an anytime algorithms for POMDPs," *Proceedings of IJCAI*, pp. 1025–1032, 2003.

[16] 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. Geoscience and Remote Sensing*, vol. 41, pp. 1005-1015, May 2003.

[17] W. Scott, Jr., K. Kim and G. Larson, "Investigation of a combined seismic, radar, and induction sensor for landmine detection," *J. Acoust. Soc. Am*., vol. 115, p. 2415, May 2004.

[18] W. Scott, Jr., K. Kim, G. Larson, A. Gurbuz and J. McClellan, "Combined seismic, radar and induction sensor for landmine detection," *Proc. IEEE Geosc. Remote Sens. Symp*., pp. 1613-1616, Sept. 2004.

[19] W. H. Jefferys and J. O. Berger, "Sharpening Ockham's razor on a Bayesian strop", *Technical Report # 91-44C*, Purdue University Department of Statistics, 1991.

[20] S. Thrun and L.Y. Pratt, editors, *Learning To Learn*, Kluwer Academic Publishers, 1998.

[21] S. Thrun, "A lifelong learning perspective for mobile robot control," *Proceedings of the IEEE/RSJ/GI Conference on Intelligent Robots and Systems*, pp. 23-30, 1994.

[22] M. H. DeGroot, *Probability and statistics* (2nd ed.), Addison-Wesley Pub. Co., 1986.

[23] H. Li, X. Liao and L. Carin, "Region-based value iteration for partially observable Markov decision processes," to appear at *Int. Conf. Machine Learning* (ICML), 2006. This reference is available at http://www.ee.duke.edu/~lcarin/Papers.html