Optimal Planning and Learning in Uncertain Environments for the Management of Wind Farms

Milad Memarzadeh, A.M.ASCE[1]; Matteo Pozzi[2]; and J. Zico Kolter[3]

Abstract: Wind energy is a key renewable source, yet wind farms have relatively high cost compared with many traditional energy sources. Among the life cycle costs of wind farms, operation and maintenance (O&M) accounts for 25–30%, and an efficient strategy for management of turbines can significantly reduce the O&M cost. Wind turbines are subject to fatigue-induced degradation and need periodic inspections and repairs, which are usually performed through semiannual scheduled maintenance. However, better maintenance can be achieved by flexible policies based on prior knowledge of the degradation process and on data collected in the field by sensors and visual inspections. Traditional methods to model the O&M process, such as Markov decision processes (MDPs) and partially observable MDPs (POMDPs), have limitations that do not allow the model to properly include the knowledge available and that may result in nonoptimal strategies for management of the farm. Specifically, the conditional probabilities for modeling the degradation process and the precision of the observations are usually affected by epistemic uncertainty. Although MDPs and POMDPs are formulated for fixed transition and emission probabilities, the Bayes-adaptive POMDP (BA-POMDP) framework treats those conditional probabilities as random variables and is therefore suitable for including epistemic uncertainty. In this paper, a novel learning and planning method is proposed, called planning and learning in uncertain dynamic systems (PLUS), within the BA-POMDP framework that can learn from the environment, update the distributions of model parameters, and select the optimal strategy considering the uncertainty related to the model. Validating with synthetic data, the total management cost of a wind farm using PLUS is shown to be significantly less than costs achieved by a fixed policy or through the POMDP framework. The preliminary results show the promise of the proposed methodology for optimal management of wind farms. **DOI: 10.1061/(ASCE)CP.1943-5487.0000390.** *© 2014 American Society of Civil Engineers.*

Author keywords: Optimal planning and learning; Sequential decision making; Wind farm management; Markov Chain Monte Carlo; Reinforcement learning.

Introduction

Wind energy is playing an ever-increasing role worldwide as a renewable source. As a result, there will be an increasing demand for careful management of costs associated with operation and maintenance (O&M) of wind turbines, which on average account for approximately 25–30% of the overall energy generation costs (Marquez et al. 2012). To make wind energy sustainable and competitive with other sources, accurate risk assessment and effective management of wind farms are necessary. Farms are made up of many similar turbines, each of which is a complex system, including structural, mechanical, and electrical components; their conditions degrade because of aging, fatigue load, and exposure to environmental risks. Managing a wind farm includes selecting appropriate operation and maintenance levels for the turbines, scheduling visual inspections, and performing maintenance/repair/replacement actions. A rational manager has to find a reasonable tradeoff between a conservative maintenance policy, profitably exploiting the farm, and exploring the interaction of the turbines with the environment. Thus, a robust decision making tool is needed to automatically evaluate the uncertainties related to the environment. In this context, the overall goal is to find an optimal policy that maximizes the total expected reward of the system over a finite or infinite time horizon, making use of probabilistic models for predicting the degradation of the system and the effects of rehabilitations.

In the literature, methods based on the partially observable Markov decision process (POMDP) framework have been recently proposed for optimal management of wind farms (Byon et al. 2010; Byon and Ding 2010; Nielsen and Sorensen 2012), fixing the model parameters based on the historical data and finding the optimal policy based on them. The purpose of this paper is to address the main limitation of POMDPs. In a POMDP framework, the probabilities defining the state transitions and the accuracy of the observations are fixed as if known with certainty. On the contrary, in most real-world management problems, the transition probabilities (modeling the degradation process and the effectiveness of the maintenance actions) and the emission probabilities (modeling the precision of sensors and visual inspection) are themselves also affected by large uncertainty. However, these models can be learned from the data collected. Specifically, the framework of the Bayes-adaptive partially observable Markov decision process (BA-POMDP) (Ross et al. 2011) allows treatment of the state transition and emission probabilities as random variables, whose distribution can be learned and updated during the process of monitoring and management.

[1] Doctoral Candidate, Dept. of Civil and Environmental Engineering, Carnegie Mellon Univ., Porter Hall A7B, 5000 Forbes Ave., Pittsburgh, PA 15213 (corresponding author). E-mail: miladm@cmu.edu

[2] Assistant Professor, Dept. of Civil and Environmental Engineering, Carnegie Mellon Univ., Porter Hall 111, 5000 Forbes Ave., Pittsburgh, PA 15213. E-mail: mpozzi@cmu.edu

[3] Assistant Professor, School of Computer Science, Carnegie Mellon Univ., Gates Hillman Center 7115, 5000 Forbes Ave., Pittsburgh, PA 15213. E-mail: zkolter@cs.cmu.edu

Note. This manuscript was submitted on September 22, 2013; approved on March 4, 2014; published online on July 8, 2014. Discussion period open until December 8, 2014; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801/04014076(10)/$25.00.

This paper makes two contributions: (1) a BA-POMDP model for O&M processes is developed that accurately captures the epistemic uncertainties in this setting; and (2) a new algorithm is proposed for approximate learning and planning in the BA-POMDP framework, PLUS, which is demonstrated to perform better than existing methods in the present setting.

The remaining parts of the paper include the notation and the primary frameworks for sequential decision making, the proposed methodology, the application of the methodology to a numerical example, and conclusions.

Related Work

This section introduces the classical methodologies of robust decision making and reinforcement learning upon which the PLUS algorithm is built.

Markov Decision Process

A fundamental model for sequential decision making is the Markov decision process (MDP) [introductions are provided in the textbook of Bertsekas (1996) and Sutton and Barto (1998)]. In an MDP, the environment is modeled as a finite set of states and actions that an agent can take. The goal is to choose actions that maximize the total expected reward. A typical MDP model is shown in Fig. 1. Figs. 1, 2, and 4 are based on the "two time slice Dynamic Bayesian network" notation adopted by the textbook of Koller and Friedman (2009). The boxes with dashed edges include the variables for each time step. Subscript t indicates that a variable is referred to time t. Circles refer to random variables, squares to decision variables, diamonds to utility variables, and dots to model parameters. Variables outside the boxes are time-independent. Shaded nodes indicate variables observed by the agent. Arrows indicate dependencies among variables by reporting how each variable in the right box (i.e., at time $t+1$) is related to other variables. The notation effectively provides a template for unrolling the model into a graphical model, where the random variables, actions, and rewards are modeled at each time step.

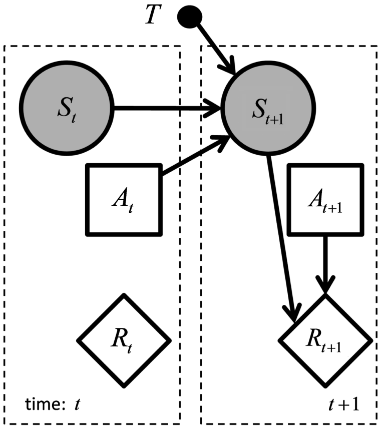

Fig. 1. MDP model

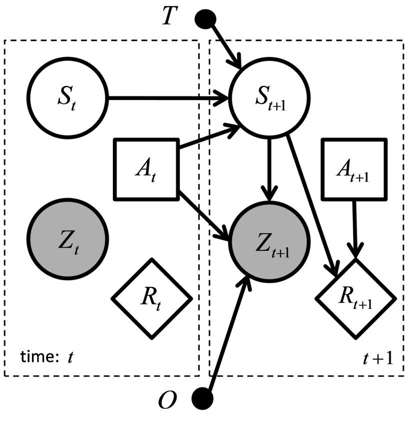

Fig. 2. POMDP model

An MDP is defined by a 5-tuple (S, A, T, R, γ), in which the following are true:

- S is a finite set of states of the system.
- A is a finite set of possible actions that an agent can take. These actions affect the system evolution and the reward that the agent receives.
- $T{:}S \times A \times S \rightarrow [0,1]$ is a transition probability function. It models the uncertainty in the prediction of the next state s_{t+1} given the current state s_t and action a_t. Formally, it is defined as $T(s,a,s') = P(s_{t+1} = s' | s_t = s, a_t = a)$, where $P(X|Y)$ indicates the conditional probability of event X given event Y. In the MDP, the Markov property holds the following: given the current state of the system and the action that an agent has taken, future states are independent of the past, so that $P(s_{t+1} = s' | \bar{a}_t, \bar{s}_t) = P(s_{t+1} = s' | a_t, s_t)$, where $\bar{s}_t = \{s_0, s_1, s_2, \ldots, s_t\}$ and $\bar{a}_t = \{a_0, a_1, a_2, \ldots, a_t\}$ indicate the sequence of states and actions, respectively, from the beginning of the process up to time t.
- $R{:}S \times A \rightarrow \mathbb{R}$ is a reward function. Based on the current state s and the action a, the agent receives reward $R(s,a)$. It can represent an immediate payoff (positive reward) or a cost (negative reward).
- $\gamma \in [0,1)$ is a discount factor that discounts the future rewards and relates them to present value.

In the MDP framework, the agent starts in an initial state, s_0. At any time step t, the agent observes the current state of the system, s_t, performs an action a_t, receives a reward $R(s_t, a_t)$, and moves to the next state s_{t+1} with probability $T(s_t, a_t, s_{t+1})$. This process is iterated up to a finite time horizon or indefinitely in the so-called infinite horizon problem.

A policy, $\pi{:}S \rightarrow A$, is a mapping from state space to actions. The value of a policy is the corresponding expected sum of discounted rewards when starting in some state and executing actions according to the policy. The optimal policy π^* is that achieving the maximum value. This paper focuses on the infinite horizon problem, in which the optimal policy is stationary and its value is described by Bellman's equation

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s') \right] \tag{1}$$

Optimal policy for MDPs can be identified by two classical methods: value iteration and policy iteration. The details of these algorithms can be found in Sutton and Barto (1998) and Russell and Norvig (2010).

Recently, researchers have been trying to incorporate uncertainty in the transition probabilities of the MDP framework directly in the formulations to find policies that are both optimal in terms of maximizing the total expected reward and robust to errors in the parameters (Bagnell et al. 2001; Iyengar 2005; Li and Si 2007; Nilim and Ghaoui 2005). Bagnell et al. (2001) have proposed a stochastic dynamic game to solve the problem of MDPs with uncertain transition probabilities. The proposed solution is equilibrium of the game that corresponds to the value function under the worst model. Li and Si (2007) have proposed a new optimality criterion that is a basis for development of robust policy iteration to solve MDPs with uncertain transition probabilities. Nilim and Ghaoui (2005) have solved the uncertain MDP problem in the context of finite and infinite horizon using robust value iteration. The authors' considers uncertainty in the model but does not assume that the state is completely observable.

Partially Observable Markov Decision Process

One of the primary limitations of a MDP is that it assumes that the state of the system is fully observable, which is not true in most real-world applications. A partially observable MDP (POMDP) is a generalization of the MDP (Smallwood and Sondik 1973; Sondik 1978). In the POMDP framework, the exact state of the system cannot be observed directly but can only be inferred by indirect observations. A typical POMDP model is shown in Fig. 2.

A POMDP is defined by an 8-tuple (S, A, Z, T, O, R, γ, b_0), in which S, A, T, R, and γ are defined as in MDP. The others are defined as follows:

- Z is a finite set of observations the agent has access to.
- O:$S \times A \times Z \rightarrow [0, 1]$ is the emission probability, which gives, for each action and resulting state, a probability distribution over the observations. $O(s, a, z) = P(z_t = z | s_t = s, a_{t-1} = a)$ defines the probability of observing z given that the agent has taken action a and landed in state s.
- $b_0 = P(s_0)$ is the initial belief state.

The belief state at time t is the posterior probability of the state of the system, given past and present observations, formally described as $b_t = P(s_t|\bar{a}_{t-1}, \bar{z}_t)$, where $\bar{z}_t = \{z_1, \ldots, z_t\}$ indicates the sequence of observations. It is assumed that, in the initial step $t = 0$, the agent does not receive any observation. The current belief state is a sufficient statistic for all past actions and observations so that action selection can be based only on the current belief state without loss of information. The belief state b_t belongs to B, the space of probability distributions over the states of the system S. Belief at time $t + 1$ can be updated from that at the previous step and observation z_{t+1}, using the Bayes rule

$$b_{t+1}(s') = \frac{O(s', a_t, z_{t+1}) \sum_{s \in S} T(s, a_t, s') b_t(s)}{\sum_{s'' \in S} O(s'', a_t, z_{t+1}) \sum_{s \in S} T(s, a_t, s'') b_t(s)} \quad (2)$$

A POMDP is equivalent to an MDP in the belief state, as shown by Aoki (1965) and Astrom (1965). In the context of POMDPs, a policy π:$B \rightarrow A$ is a mapping from the space of belief states to actions. Bellman's equation for optimal policy π^* can be formulated as

$$V^*(b) = \max_{a \in A} \left[\sum_{s \in S} b(s) R(s, a) + \gamma \sum_{z \in Z} P(z|b, a) V^*(b') \right] \quad (3)$$

where distribution b' is updated belief b_{t+1}, according to Eq. (2), with $a = a_t$, $z = z_{t+1}$, $b = b_t$, and conditional probability $P(z|b, a)$ is computed as

$$P(z|b, a) = \sum_{s' \in S} O(s', a, z) \sum_{s \in S} T(s, a, s') b(s) \quad (4)$$

In principle, a POMDP is solved by applying the methods to solve MDPs to the belief state. However, as the belief state is a probability distribution, it is defined on an infinite space, and so exact solution for the POMDP is not generally available. In reacting to the observations collected, an agent can select one conditional plan among the many available (Russell and Norvig 2010). The conditional plan can be interpreted as a policy function defined on the domain of the sequence of observations.

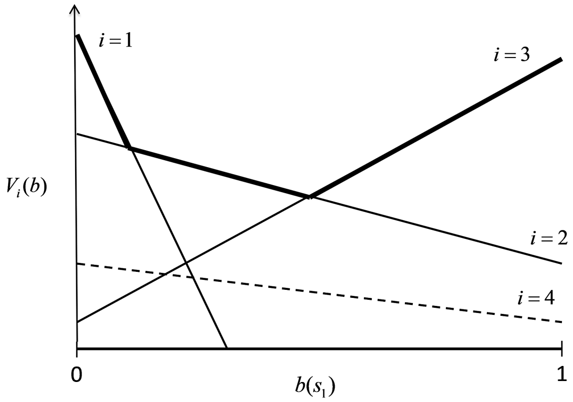

Fig. 3. Simple example of value function for two-state POMDP model (adapted from Kaelbling et al. 1998)

The number of possible conditional plans, n_c, grows exponentially with the time horizon assumed for the problem. Let $\alpha_i(s)$ define the value of executing the ith conditional plan starting from perfect knowledge that the system is in state s. The value of following that plan is linearly related to belief state b as $V_i(b) = \sum_s b(s) \alpha_i(s)$. Fig. 3, a graph inspired by Kaelbling et al. (1998) that refers to a simple example of a two-state POMDP, provides a better understanding of these concepts. Belief is completely described by a scalar value $b(s_1)$, as $b(s_2) = 1 - b(s_1)$. Fig. 3 reports the value for four conditional plans, and the bold line indicates the optimal value, depending on the belief state.

The optimal value function can be written as

$$V^*(b) = \max_i \sum_s b(s) \alpha_i(s) \quad (5)$$

where i is defined on the domain $\{1, 2, \ldots, n_c\}$. The proof of Eq. (5) can be found in the work by Smallwood and Sondik (1973), which shows that the optimal value function for any finite horizon POMDP is a piecewise-linear and convex function over the domain B. Eq. (5) cannot be solved explicitly, except for very short time horizon, because of the high value of n_c. However, as Fig. 3 clearly shows, some conditional plans are completely dominated (e.g., plan 4 in Fig. 3) and can be neglected (Russell and Norvig 2010). Each conditional plan begins with a specific first action, so Eq. (5) allows defining implicitly the optimal policy π^* as, for any belief state b, the optimal action is that to be executed as first one in the optimal conditional plan.

Kaelbling et al. (1998) have proposed the so-called witness algorithm for finding the exact solution to POMDPs through value iteration. However, this algorithm is not practical when the set of states, actions, and observations are large. An alternative approach is to discretize the belief space, using either a fixed grid (Lovejoy 1991) or a variable grid (Zhou and Hansen 2001). The value of any belief is then defined by interpolation of the points on the grid. However, in general, regular grids do not scale well in problems with high dimensionality, and nonregular grids suffer from expensive interpolation routines. Other point-based value iteration methods restrict the search to the beliefs that can be reached starting from the initial belief state (Pineau et al. 2003). In particular, one of the most effective point-based value iteration methods is successive approximations of the reachable space under optimal policies (SARSOP) (Kurniawati et al. 2008), which identifies the optimally reachable belief states and approximates the optimal value function using this set. SARSOP represents the state-of-the-art in solving POMDPs in terms of efficiency and accuracy. As all algorithms for POMDP, SARSOP formally solves the finite horizon problem, but it can be used as an approximation to also solve the infinite horizon case.

Bayes-Adaptive Partially Observable Markov Decision Process

The BA-POMDP framework is a generalization of POMDP, where the transition and emission probabilities, T and O, are unknown components of the system and are treated as random variables, with a prior distribution $P(T,O)$. The BA-POMDP model is shown in Fig. 4. Technically, the BA-POMDP model can be interpreted as a POMDP with a continuous state space and with an augmented belief state that also includes T and O. The augmented belief state at time t is now defined as $\tilde{b}_t = P(s_t,T,O|\bar{a}_{t-1},\bar{z}_t)$. In principle, the belief at time t can be expressed as a function of that at the previous step, as in the POMDP formulation reported in Eq. (2). However, as in most cases, any closed-form representation of the posterior cannot be found; in a BA-POMDP, it is easier to express the belief at any step by integrating the joint probability:

$$\begin{aligned}P(s_t,T,O|\bar{a}_{t-1},\bar{z}_t) &\propto P(\bar{z}_t,s_t|T,O,\bar{a}_{t-1})P(T,O)\\ &= P(T,O)\sum_{\bar{s}_{t-1}\in S^t} P(\bar{z}_t,\bar{s}_t|T,O,\bar{a}_{t-1})\\ &= P(T,O)\sum_{\bar{s}_{t-1}\in S^t} P(s_0)\left\{\prod_{s,a,s'\in[S\times A\times S]}[T(s,a,s')]^{N^a_{ss'}(\bar{s}_t,\bar{a}_{t-1})}\right\}\\ &\times\left\{\prod_{s,a,z\in[S\times A\times S]}[O(s,a,z)]^{N^a_{sz}(\bar{s}_t,\bar{a}_{t-1},\bar{z}_t)}\right\}\end{aligned} \qquad (6)$$

where S^t = set of possible sequences of states up to time t; $N^a_{ss'}(\bar{s}_t,\bar{a}_{t-1})$ = number of times the transition (s,a,s') appears in the process; and $N^a_{sz}(\bar{s}_t,\bar{a}_{t-1},\bar{z}_t)$ = number of times the emission (s,a,z) appears in the process.

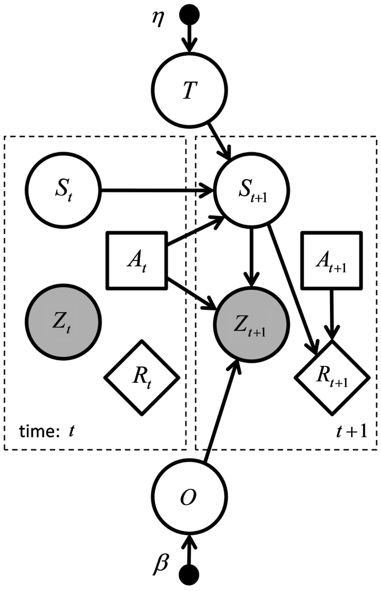

Fig. 4. BA-POMDP model

```
Algorithm 1- PLUS Algorithm
function PLUS $(\eta,\beta,b_0,\bar{a}_t,\bar{z}_t,R,\gamma,N)$
    ➢ Learning
    $\{T^{(k)},O^{(k)},b_t^{(k)}\}_N^{k=1} \leftarrow LEARNING(\eta,\beta,b_0,\bar{a}_t,\bar{z}_t,N)$
    ➢ Planning
    $a^* \leftarrow PLANNING\left(\{T^{(k)},O^{(k)},b_t^{(k)}\}_N^{k=1},R,\gamma\right)$
    return $a^*,\{T^{(k)},O^{(k)},b_t^{(k)}\}_N^{k=1}$
end function
```

Fig. 5. PLUS algorithm

```
Algorithm 2- PLUS Learning Algorithm
function LEARNING$(\eta,\beta,b_0,\bar{a}_{t-1},\bar{z}_t,N,n_b)$
    $T^{(0)} \sim \mathrm{Dir}(\eta)$
    $O^{(0)} \sim \mathrm{Dir}(\beta)$
    for $k = 1:(N+n_b)$ do
        $\left(\bar{s}_t^{(k)},b_t^{(k)}\right) \leftarrow \mathrm{FFBS}(T^{(k-1)},O^{(k-1)},b_0,\bar{a}_{t-1},\bar{z}_t)$
        $\eta' \leftarrow \mathrm{UpdateDirichlet}\left(\eta,\bar{s}_t^{(k)},\bar{a}_{t-1}\right)$
        $T^{(k)} \sim \mathrm{Dir}(\eta')$
        $\beta' \leftarrow \mathrm{UpdateDirichlet}\left(\beta,\bar{s}_t^{(k)},\bar{a}_{t-1},\bar{z}_t\right)$
        $O^{(k)} \sim \mathrm{Dir}(\beta')$
    end for
    return $\{T^{(k)},O^{(k)},b_t^{(k)}\}_{N+n_b}^{k=1+n_b}$
end function
```

Fig. 6. PLUS learning algorithm

```
Algorithm 3 – PLUS Planning Algorithm
function PLANNING$\left(\{T^{(k)},O^{(k)},b_t^{(k)}\}_N^{k=1},R,\gamma\right)$
    for $k = 1:N$ do
        $\{\alpha_h\}_m^{h=1} \leftarrow \mathrm{SARSOP}(T^{(k)},O^{(k)},b_t^{(k)},R,\gamma)$
        $[\alpha_1^*,\ldots,\alpha_A^*] = \mathrm{PRUNING}(\{\alpha_h\}_m^{h=1},b_t^{(k)})$
        for $j = 1:A$ do
            $Q_j^{(k)} \leftarrow \alpha_j^{*\mathrm{T}} \cdot b_t^{(k)}$
        end for
    end for
    for $j = 1:A$ do
        $Q_j \leftarrow \frac{1}{N}\sum_{k=1}^{N} Q_j^{(k)}$
    end for
    $a^* \leftarrow \operatorname{argmax}_j Q_j$
    return $a^*$
end function
```

Fig. 7. PLUS planning algorithm

Learning is a challenging task in the BA-POMDP framework, as the posterior is defined by the complicated formula in Eq. (6), and it is not possible to compute the posterior exactly because the number of possible sequences of states grows exponentially as the time horizon grows. It is convenient to adopt a Dirichlet distribution for the prior distribution $P(T,O)$ because that is the conjugate of a multinomial distribution. In this context, this implies that, after perfect observation of states $\bar{s}_t$, actions $\bar{a}_{t-1}$, and observations $\bar{z}_t$, the posterior $P(T,O|\bar{s}_t,\bar{a}_{t-1},\bar{z}_t)$ would still be in the Dirichlet family.

Formally, the Dirichlet distribution (which is indicated as "Dir") is the multivariate extension of the beta distribution and defines a probability density over discrete distributions. Suppose $\mathbf{q} = [q_1, \ldots, q_k]^{\mathrm{T}}$ defines a k-dimensional discrete distribution. The density can be parameterized by count vector $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_k]^{\mathrm{T}}$, listing only non-negative entries. The Dirichlet probability density on $\mathbf{q}$ is defined as

$$\mathrm{Dir}(\mathbf{q}) = \frac{\prod_{i=1}^{K} \Gamma(\phi_i)}{\Gamma(\sum_{i=1}^{K} \phi_i)} \prod_{i=1}^{k} q_i^{\phi_i - 1}$$

The textbook of Murphy (2012) gives an introduction to the Dirichlet distribution. In the model presented in Fig. 4, it is assumed that s_0, T and O are marginally independent, and parameters η and β define the count parameters of the prior Dirichlet distributions of T and O, respectively.

The transition probability T can be represented by a three-dimensional matrix (of size $[S \times S \times A]$) and the parameters η of the Dirichlet prior probability $P(T)$ by a matrix of the same size as T. Let $\eta_{s_i,s_j}^{a_k}$ define the counts of transitioning from s_i to s_j after action a_k. After observing $\bar{s}_t$ and $\bar{a}_{t-1}$, the parameters of the posterior Dirichlet distribution $P(T|\bar{s}_t, \bar{a}_{t-1})$ are computed as $\eta_{s_i,s_j}^{a_k} + N_{s_i,s_j}^{a_k}(\bar{s}_t, \bar{a}_{t-1})$, i.e., by simply updating the counts with the observed state transitions. A similar analysis can be performed for the emission probability O. However, in the BA-POMDP framework, states are not observed directly. The augmented belief state, as expressed in Eq. (6), can also be derived as product $P(T, O|\bar{a}_{t-1}, \bar{z}_t)P(s_t|T, O, \bar{a}_{t-1}, \bar{z}_t)$: the first term is the posterior distribution of the model parameters given the observations and does not belong to the Dirichlet distribution. Rather, this density is a mixture of Dirichlet distributions, in which each component corresponds to the counts of one specific possible state sequence, weighted by the probability of that sequence (Fruhwirth-Schnatter 2006). The next section presents the method to perform approximate updating in that context.

In the BA-POMDP framework, the optimal policy may be suboptimal for any specific value of T and O as long as it maximizes the value for the entire state of epistemic uncertainty about the model. Furthermore, the value of a policy also includes the benefit of taking exploratory actions that help the agent reduce uncertainty in the model parameters themselves. Generally, a policy maps beliefs over (s, T, O) to actions. This suggests that the sequential decision problem of optimally behaving under state and model uncertainty can be modeled as a POMDP over the augmented state, including the actual states of the system and model parameters T and O. However, solving POMDPs over the infinite space of beliefs over this augmented state, including continuous components, is not computationally feasible. The next section proposes the PLUS algorithm, which is an approximate solution to this computational complexity.

Jaulmes et al. (2005a, b) have proposed an algorithm called MEDUSA to find the optimal policy for a POMDP when the model is not known or poorly specified. The algorithm tries to improve the POMDP incrementally using selected queries while still optimizing the total expected reward. The next sections introduce the proposed PLUS algorithm and compare its performance with MEDUSA and with the use of a POMDP with fixed parameters.

Proposed Methodology

An approximate method is proposed for optimally planning and learning in uncertain dynamic systems (PLUS) within the BA-POMDP framework. Fig. 5 shows the overall PLUS method, which is organized in two main parts: learning and planning. The algorithm can be called at any stage of the process. At time t, it represents the augmented belief state $\tilde{b}_t$ by a set of samples, and it suggests action a^*. In the algorithm, notation $x^{(k)}$ indicates the kth sample of variable x.

PLUS Learning Phase

The PLUS algorithm makes use of an approximate method based on Markov chain Monte Carlo (MCMC) Gibbs sampling (Carter and Kohn 1994). The present approach is a slight variation of the beam sampling approach used in the context of infinite hidden Markov models (Van Gael et al. 2008) and infinite POMDPs (Doshi-Velez 2010). Fig. 6 shows the details of the proposed algorithm for learning: the method samples T, ON instances of T, O, and belief state b_t from the joint posterior distribution. The procedure starts with sampling T, O from the corresponding prior Dirichlet distributions, then alternate between sampling state sequence $\bar{s}_t$, and sampling T and O. For each fixed T and O, a state sequence is drawn by forward filtering backward sampling (FFBS) (Fruhwirth-Schnatter 2006), as described in the next section. In turn, as mentioned previously, the posterior distribution given each sample $\bar{s}_t$ is still in the Dirichlet family. Parameter set η' defines the updated Dirichlet distribution for the transition probabilities, depending on sampled state sequence $\bar{s}_t$, whereas β' defines that of the emission probabilities, depending on $\bar{s}_t$ and observations $\bar{z}_t$. After appropriate burn-in phase, this proposed method is selecting samples from the true posterior distribution. In Fig. 6, n_b indicates the number of samples in the burn-in phase to be discarded (Murphy 2012), and the notation $x \sim p$ indicates that sample x is generated from distribution p.

Forward Filtering Backward Sampling

FFBS is a multimove sampling method for discrete systems (Fruhwirth-Schnatter 2006). The steps are as follows: (1) for each time step j ranging from 0 to t, derive the posterior probability $P(s_j|T, O, \bar{a}_{j-1}, \bar{z}_j)$, solving the so-called filtering problem; and (2) sample state s_t' from the last distribution and s_j', from time step $j = t - 1$ backward to $j = 0$, from distribution $F(s_j) \propto P(s_j|T, O, \bar{a}_{j-1}, \bar{z}_j)P(s_{j+1}'|T, s_j, a_j)$. The outcome of FFBS algorithm is the sequence of states $\{s_0', \ldots, s_t'\}$ sampled from distribution $P(\bar{s}_t|T, O, \bar{a}_{t-1}, \bar{z}_t)$.

PLUS Planning Phase

The planning method is based on two approximations. First, neglect the exploratory value of learning variables T, O, i.e., the system model parameters. The proposed method aims at identifying the optimal policy as that for transition and emission probabilities modeled by $P(T, O|\bar{a}_{t-1}, \bar{z}_t)$, neglecting the updating attributable to future observations. To formalize the second approximation, define $Q_{T,O}(a, b)$ as the quality of a belief-state-action (Q-value) for a POMDP, i.e., the value of starting from belief b, performing action a, and following the optimal policy after that for a model defined by T, O. Identify the optimal action a^* for the overall BA-POMDP by the following approximate formula:

$$a^* \cong \underset{a}{\mathrm{argmax}} \mathbb{E}_{T,O}[Q_{T,O}(a, b)] \tag{7}$$

where $\mathbb{E}_x$ indicates the statistical expectation, according to actual knowledge of variable x; and the belief state at time t is defined as in a POMDP as $b = P(s_t|T, O, \bar{a}_{t-1}, \bar{z}_t)$. Eq. (7) represents an approximation because it combines quantities related to optimal policies for different models. However, do not use the approximation to estimate the value of the policy but only to select the current optimal action. Computationally, the advantage of Eq. (7) is that

$Q_{T,O}(a,b)$ can be obtained from the results of a POMDP solver, i.e., SARSOP.

The Q-value of a belief-state-action can be related to the α-vectors presented in the POMDP section. For a model T, O and belief b, the optimal conditional plan starting with action a for each available action can be identified. Define $\alpha^*_{a,b,T,O}(s)$ as the component referring to state s of the corresponding α-vector, and the Q-value of a belief-state-action can be computed as

$$Q_{T,O}(a,b) = \sum_s b(s)\alpha^*_{a,b,T,O}(s) \tag{8}$$

Fig. 7 presents the scheme of the planning algorithm, which is based on Eqs. (7) and (8). At time t, after the learning phase, augmented belief state $\tilde{b}_t$ is represented by N samples. The next step is to solve the corresponding POMDP problem for each sample, using SARSOP (Kurniawati et al. 2008). The outcome of SARSOP is the set of m nondominated α-vectors. Among them, the algorithm selects one optimal α-vector per each action: this is the pruning routine mentioned in the algorithm. α^*_j refers to the optimal vector for the jth action; $Q^{(k)}_j$ to the Q-value of a belief-state-action for the kth sampled model under the jth action; and Q_j to the expected Q-value of a belief-state-action for the entire model space, which is computed by sample average. Action a^* is selected by identifying the maximum of Q_j among all possible actions.

Numerical Validation

To validate the proposed methodology, a numerical example of wind farm management is used. It is assumed that the condition state of each turbine can be modeled by a Markov process defined by a few states, and the observations collected can be classified within a few possible discrete values. Although PLUS can be applied to much more complicated problems, this simple setup allows the extensive investigation of the performance of the algorithm so it can be compared to other existing methods.

The condition state of the turbine degrades owing to fatigue and aging, potentially causing a structural failure and a relevant economical loss to the agent. In turn, the agent can perform repairs to avoid failures and inspections to refine the knowledge about each condition state. In detail, the farm is assumed to consist of 10 turbines of the same type, so that a unique value of transition and emission probabilities can be referred to. The cost in Figs. 8, 9, and 12 refers to the average per one turbine. Specifically, three condition states are assumed: $s = 1$ refers to an intact structure; $s = 2$ to a damaged one; and $s = 3$ to a collapsed turbine. The three actions are the following: $a = 1$ corresponds to *do nothing* (DN); $a = 2$ to *repair* (RE); and $a = 3$ to performing a *visual inspection* (VI). When DN is selected, the condition state evolves according to the degradation process. RE models a costly intervention, which is supposed to improve the condition state, whereas VI models an effort providing only information on the condition state, without affecting the degradation process. Each time step is assumed to be six months, and the agent takes one action per turbine at each time step.

Observations are classified in four discrete outcomes: $z = 1$ is intended as a reassuring output, suggesting that the turbine is undamaged; $z = 2$ and $z = 3$ indicate two symptoms of damage; after recording $z = 4$, the agent knows that the turbine is collapsed.

As indicated in "Proposed Methodology," the agent's prior knowledge is modeled on transition and emission probabilities by independent Dirichlet distributions with parameters η and β, respectively. Parameter η can be represented by three matrices: η_{DN}, η_{RE}, and η_{VI}, referring to the actions listed above

$$\eta_{DN} = \eta_{VI} = \begin{bmatrix} 8 & 4 & 2 \\ 0 & 4 & 2 \\ 0 & 0 & 1 \end{bmatrix} \qquad \eta_{RE} = \begin{bmatrix} 8 & 4 & 0 \\ 4 & 2 & 0 \\ 4 & 2 & 0 \end{bmatrix}$$

The transitions are assumed to be identical for actions DN and VI. The zeros in the matrix η_{DN} indicate that after any of these actions, the condition state cannot improve; therefore, for example, the turbine stays in a collapsed state after action DN. Generally, according to this matrix, the turbine in the intact state has a tendency to stay undamaged, but it can also become damaged or directly collapse, whereas a turbine in the damaged state has a tendency to stay there, but it can also collapse. After action RE, the turbine cannot be in a collapsed state, but it can still be damaged because the intervention is not known to be perfect and, even after a perfect repair, the turbine can transit to the damage state during the following period, considering the long time step (six months). As for any feature of the process, the effectiveness of such an intervention can be learnt by the agent during the management history. Knowledge about emissions, depending on the action, are modeled by the following values:

$$\beta_{DN} = \beta_{RE} = \begin{bmatrix} 8 & 4 & 2 & 0 \\ 2 & 8 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \beta_{VI} = \begin{bmatrix} 4 & 2 & 0 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As can be deduced from these matrices, the agent thinks that, as a tendency, States 1 and 2 generate Observations 1 and 2, respectively, under actions DN or RE. The visual inspection VI is regarded as possibly imperfect; again, its actual effectiveness can be discovered during the management process. Independent of the action, the collapse State 3 is univocally related to Observation 4, so that the agent is immediately aware of any failure event.

The reward function is the sum of three components: the costs for repairing, inspecting, and down-time. The agent pays US\$10,000 for any repair, \$500 for any visual inspection, and \$50,000 for any time step in which a turbine is in the collapsed state. The discount factor is assumed to be $\gamma = 0.95$.

The belief about the initial state is modeled as

$$b_0 = [0.8 \quad 0.2 \quad 0]$$

therefore, the agent believes that the turbines are in the intact state with 80% probability and in damaged state with 20% probability.

The behaviors of different turbines in the farm are assumed to be independent, and the agent refers the planning to the infinite horizon setting.

The method is validated by two sets of numerical experiments. First, transition and emission were fixed to a value compatible with the available knowledge, referring to this as the true model. The true model was assigned to each turbine in the farm, and the performance of alternative schemes for learning and planning were simulated. Next, the planning algorithm was tested for the range of all possible models representing the turbines.

In both experiments, four types of agents were considered: The true model agent has perfect knowledge about the true underlying transition and emission probabilities and adopts a POMDP model with correct value for T and O, making use of the SARSOP algorithm for planning: this represents an lower bound (in terms of cost to be minimized) to the performance of any planning strategy under uncertainty. The expected model agent derives the expected value of T and O from the prior Dirichlet distribution and again, adopts POMDP solved by SARSOP: it represents the simplest and most common approach to solve the planning problem under model uncertainty. The MEDUSA agent makes use of the algorithm described in Jaulmes et al. (2005a, b), whereas the PLUS agent adopts the method that was presented in "Proposed Methodology."

Two different metrics are used to validate the methods. First, the immediate and cumulative management cost for assessing the performance of the planning methods is evaluated because they are directly related to what each agent is trying to optimize. For additional validation of the learning process itself, evaluate the Kullback-Leibler (KL) divergence (Cover and Thomas 2006) between the transition (or emission) probabilities as modeled by the posterior distribution and in the true model. The KL divergence is a nonsymmetric measure of the differences between two probability distributions. Specifically, the KL divergence of distributions Q from distribution P (both are distributions defined on n discrete values), denoted as $D_{KL}(P||Q)$, is a measure of information lost when Q is used to approximate P, and is defined as

$$D_{KL}(P||Q) = \sum_{i=1}^{n} \ln\left(\frac{P(i)}{Q(i)}\right) P(i) \tag{9}$$

where ln = natural logarithm. In computing the KL divergence between two transition (or emission) models, the results referring to the average over all values of s_t and a_t.

Learning and Planning Validation

The first numerical campaign is devoted to the validation of the planning and learning algorithms. To do so, a model is fixed and assigned to all turbines. This is called the true model, and it is defined by transition T^* and emission O^*, as listed in the following:

$$T^*_{DN} = T^*_{VI} = \begin{bmatrix} 0.9 & 0.08 & 0.02 \\ 0 & 0.9 & 0.1 \\ 0 & 0 & 1 \end{bmatrix} \qquad T^*_{RE} = \begin{bmatrix} 1 & 0 & 0 \\ 0.9 & 0.1 & 0 \\ 0.9 & 0.1 & 0 \end{bmatrix}$$

$$O^*_{DN} = O^*_{RE} = \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0 \\ 0.05 & 0.9 & 0.05 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad O^*_{VI} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This specific model describes a turbine that is more reliable than that defined by the expected value of the distribution reported in the previous section. These models were selected by adapting examples from the literature (Byon et al. 2010; Byon and Ding 2010; Nielsen and Sorensen 2012) after discussion with industry experts from EverPower Wind Holdings (Pittsburgh, PA). For example, the probability of a collapse in one 6-month period, for an intact turbine, is only 2%. The emissions related to the visual inspection models perfect information on the condition state.

For each agent, the management of the wind farm is simulated 20 times, and the average outcome is plotted in Fig. 8. In each simulation, the initial state is sampled according to the distribution b_0. Fig. 8(a) reports the average immediate cost versus the time step. The dashed line represents the true model agent, the line represents the expected model agent, and the dash-dotted line represents the PLUS agent, whereas other lines refer to the MEDUSA algorithm, with a learning rate (LR) of 0.1, 0.5 and 1.

Each agent starts with a low cost in the first steps owing to the good state of the turbines as assumed by the initial belief state. The true model and the expected model agents adopt a stationary policy, and the corresponding immediate cost converges to a constant value, which is approximately \$2,200/6 months for the former, and \$3,500/6 months for the latter agent. Fluctuations are attributable to randomness in the average of the small set of simulations. Agents adopting the MEDUSA and the PLUS algorithm, on the other hand, adopt nonstationary policies because of the learning process. At each time, the knowledge about the model is affected by processing the previous observations, and the policy varies accordingly. Ideally, if sufficient information is collected, the policies (and consequently the immediate cost) of these agents should converge to that of the true model agent. As expected, it is apparent from the figure that the immediate cost grows in the first phase (i.e., the first 10–20 steps) and then is reduced in time because of the effect of learning. The PLUS algorithm also performs well in the first phase because of the robust algorithm for planning. After 30 steps, the immediate cost is approximately \$2,600/6 months. In this simulation, the MEDUSA algorithm achieves a higher cost for a range of different learning rates. The benefit of the PLUS algorithm over the expected model approach can be quantified as approximately \$1,000/6 months.

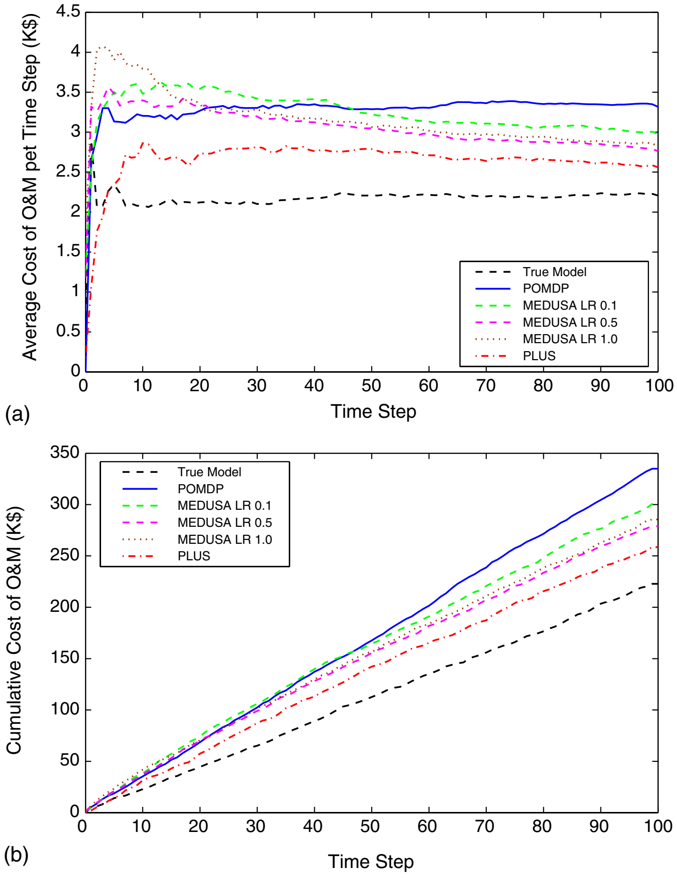

Fig. 8. Costs for O&M of a wind farm versus time for six agents: (a) immediate; (b) cumulative

Fig. 8(b) shows the cumulative costs of O&M, computed as the integral in time of the curves plotted in Fig. 8(a). This representation is useful for assessing the long-term benefit of adopting alternative schemes. In a 100-step period (corresponding to 50 years), the true model agent expects a cost of approximately \$220,000; the expected model agent a cost of approximately \$350,000, whereas the PLUS agent expects a cost of approximately \$250,000. Thus, the benefit of adopting PLUS is quantifiable to approximately \$100,000 for this period. These costs and savings are for a single turbine, and the costs and savings regarding the entire farm is ten times higher.

Fig. 9 shows the cumulative costs for O&M of a wind farm for PLUS, true model, and POMDP agents, including the 95% confidence intervals.

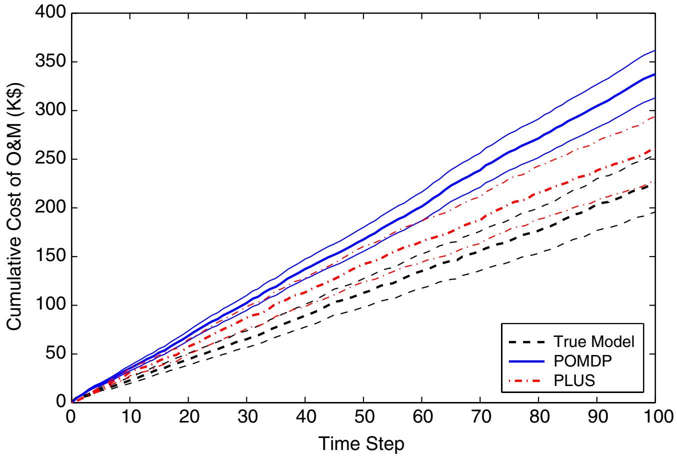

Fig. 9. Cumulative costs for O&M of a wind farms versus time for three agents, including the 95% confidence intervals

Fig. 10 focuses on the learning process, showing the evolution of the KL divergence between the posterior distribution of the model, as formulated by each agent, and the true model. Fig. 10(a) plots the transition probabilities and Fig. 10(b) the emission probabilities. The expected model agent does not learn during the process; and consequently, the KL divergence is constant. The agents using MEDUSA or PLUS update their knowledge during the management process, and the KL divergence is expected to go to zero when the information encoded in the collected observations is sufficient to identify the model. For these agents, the KL divergence is computed as the average from a set of samples generated according to the posterior distribution (as illustrated in "PLUS Learning Phase," PLUS algorithm requires the generation of samples, so this further computation is straightforward). This study used 10 samples in this simulation. As shown in the figure, the learning is fast in the initial phase, but it becomes slow as more and more observations have been already collected. According to this simulation, the MEDUSA agents learn the transition probabilities well but not the emission probabilities [Fig. 10(b)]. MEDUSA learns the emission probabilities poorly, perhaps because of the different planning approach compared with PLUS, and may need more data. However, in the long run, provided that sufficient exploration is performed, MEDUSA is conjectured to asymptotically learn the true model. Generally, MEDUSA and PLUS are different in terms of the tradeoff between computational cost and accuracy: MEDUSA is computationally cheaper and easier to scale; however, it provides less accurate solutions compared with PLUS.

Fig. 10 shows that initially the KL divergence of the expected model agent is lower than that of the PLUS agent. This is a random effect owing to the selection of the true model in this simulation. The expected model agent adopts the mean transition and emission. Depending on the actual model of the turbine, it may be the case that the KL divergence can be arbitrarily small, and possibly much smaller than that of the PLUS agent. In other words, it may be the case that the model assumed by the expected model agent is actually the correct one; and therefore, no learning is needed. Generally, the performance of the alternative methods depends on the specific actual model. In the next section, a validation of the planning algorithm is performed for all possible models.

Fig. 11 shows the same results in Fig. 10(a), including the 95% confidence intervals for the learning process of PLUS agent.

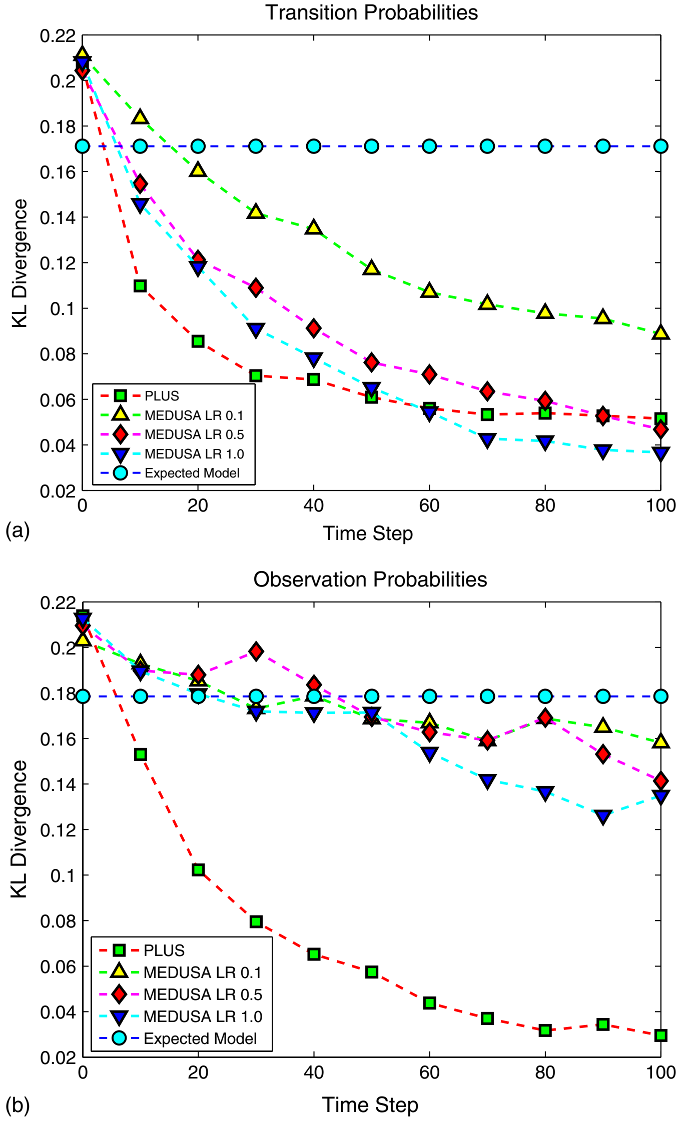

Fig. 10. Performance of the proposed learning methodology (PLUS) compared with MEDUSA (with different learning rates) and POMDP (does not involve learning); the graphs show the KL divergence between each model and the true model: (a) transition; (b) emission probabilities

Planning Approach Validation

The outcomes of the previous section highlight that the PLUS algorithm outperforms the expected model agent. It is possible, however, that this benefit derives only from the learning process. The second experiment aims specifically to validate the planning algorithm only, removing the learning effect. The expected benefit of a method cannot be assessed by referring only to a single model, so in this campaign a set of samples are drawn from the prior distributions of models, and the performance of the methods are averaged across these scenarios.

Fig. 12 reports the immediate and cumulative costs of O&M, for the true model, the expected model, and the PLUS agents. Again, the true model agent represents a lower bound, leading to an immediate cost of approximately $2,900/6 months, whereas the expected model agent achieves a cost of approximately $8,300/6 months, and the PLUS agent a cost of approximately $7,700/6 months. The difference between these latter

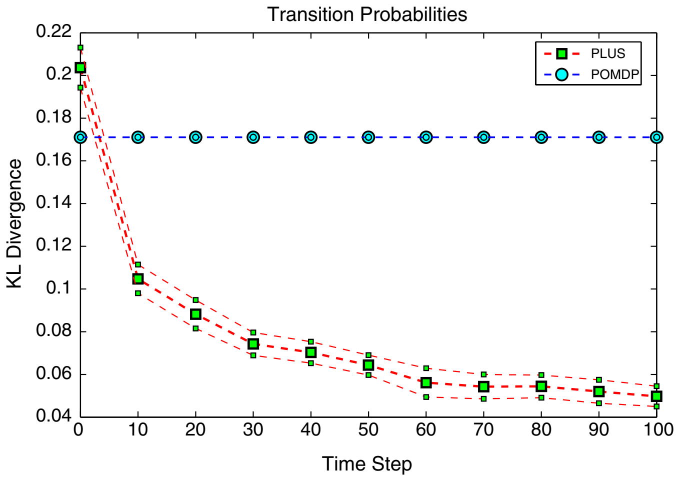

Fig. 11. Performance of the proposed learning methodology (PLUS) compared with POMDP (does not involve learning) including the 95% confidence intervals; the graphs show the KL divergence between each model and the true model parameters

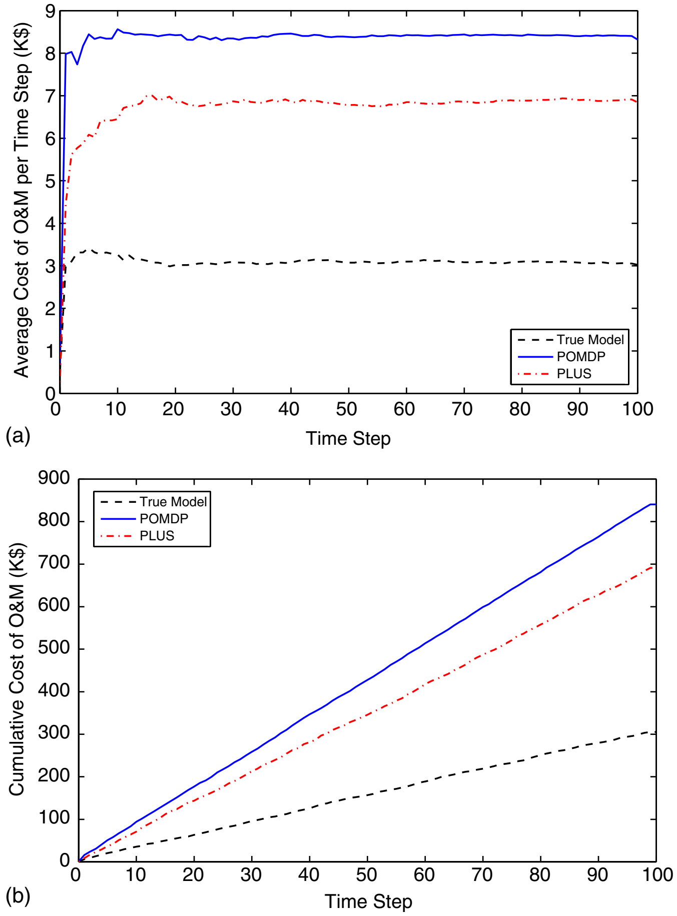

Fig. 12. Planning performance of the PLUS algorithm compared with POMDP and true model agents; costs for O&M of a wind farm versus time: (a) immediate; (b) cumulative

values, i.e., $600/6 months, quantifies the benefit of adopting the robust planning approach presented in "PLUS Planning Phase" for the specific example. Naturally, adding the learning process as well would make PLUS perform much closer to the true model, but this experiment highlights the value of uncertainty-aware planning in and of itself. The costs and savings regarding the entire farm are ten times higher.

Conclusion

A method named PLUS is proposed for learning and planning within the BA-POMDP framework and applicable to the context of wind farm management. The BA-POMDP framework overcomes one of the primary limitations of the POMDP framework by treating the transition and emission probabilities as random variables, whose distributions can be updated during the learning process. The PLUS algorithm uses Markov chain Monte Carlo simulations to find an approximate solution for the BA-POMDP problem. The approach allows for a rational treatment of data collected by sensors and visual inspections, a reliable tracking of the condition states of turbines, and robust decision-making support.

The PLUS algorithm has been validated with synthetic data and is shown to out-perform state-of-the-art reinforcement learning approaches, such as MEDUSA. MEDUSA was originally proposed for applications of robot navigation and it scales easier than PLUS, requiring less computational effort. However, for application to wind farms, it is believed that the computational drawback of PLUS is not a significant concern because the computational cost is low with respect to the direct costs for operation and maintenance of a wind farm. On the contrary, in this context it is necessary to achieve a rational and robust selection of the management policy, making use of the knowledge available at any state of the process. PLUS allows this; it also allows the agent to learn, during the management, the statistics of the degradation process (transition probabilities) and the performance and reliability of the monitoring system (emission probabilities).

Acknowledgments

The authors would like to thanks Stephane Ross for his valuable discussion on the BA-POMDP framework and Ada Zhang and Jingwei Zhang for their valuable discussion and contribution. The authors would like to also especially thank EverPower Wind Holdings for invaluable collaboration.

References

Aoki, M. (1965). "Optimal control of partially observable Markov systems." *J. Franklin Inst.*, 280(5), 367–386.

Astrom, K. J. (1965). "Optimal control of Markov decision processes with incomplete state estimation." *J. Math. Anal. Appl.*, 10(1), 174–205.

Bagnell, A., Ng, A., and Schneider, J. (2001). "Solving uncertain Markov decision processes." *Proc., Neural Information Processing Systems*, Carnegie Mellon Univ., Pittsburgh, PA.

Bertsekas, D. P. (1996). *Dynamic programming and optimal control*, Vol. 1–2, Athena Scientific, Belmont, MA.

Byon, E., and Ding, Y. (2010). "Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process." *IEEE Trans. Power Syst.*, 25(4), 1823–1834.

Byon, E., Ntaimo, L., and Ding, Y. (2010). "Optimal maintenance strategies for wind turbine system under stochastic weather conditions." *IEEE Trans. Reliab.*, 59(2), 393–404.

Carter, C. K., and Kohn, R. (1994). "On Gibbs sampling for state space models." *Biometrika*, 81(3), 541–553.

Cover, T. M., and Thomas, J. A. (2006). *Elements of information theory*, Wiley, New York.

Doshi-Velez, F. (2010). “The infinite partially observable Markov decision process.” *Proc., Neural Information Processing Systems*, Vol. 22, 477–485.

Fruhwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*, Springer, New York.

Iyengar, D. (2005). “Robust dynamic programming.” *Math. Oper. Res.*, 30(2), 257–280.

Jaulmes, R., Pineau, J., and Precup, D. (2005a). “Active learning in partially observable Markov decision processes.” *European Conf. on Machine Learning*, Porto, Portugal, 601–608.

Jaulmes, R., Pineau, J., and Precup, D. (2005b). “Learning in non-stationary partially observable Markov decision processes.” *European Conf. on Machine Learning Workshop on Reinforcement Learning in Non-Stationary Environments*, Porto, Portugal.

Kaelbling, L. P., Littman, M. L., and Cassnadra, A. R. (1998). “Planning and acting in partially observable stochastic domain.” *J. Artif. Intell.*, 101(1–2), 99–134.

Koller, D., and Friedman, N. (2009). *Probabilistic graphical models*, MIT, Cambridge, MA.

Kurniawati, H., Hsu, D., and Lee, W. (2008). “SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces.” *Robotics: Science and Systems*, ETHZ, Zurich, Switzerland.

Li, B., and Si, J. (2007). “Robust dynamic programming for discounted infinite horizon Markov decision processes with uncertain stationary transition matrices.” *IEEE Symp. on Approximate Dynamic Programming and Reinforcement Learning*, Honolulu, HI, 96–102.

Lovejoy, W. S. (1991). “Computationally feasible bounds for partially observed Markov decision process.” *Oper. Res.*, 39(1), 162–175.

Marquez, F. P. G., Tobias, A. M., Perez, J. M. P., and Papaelias, M. (2012). “Condition monitoring of wind turbines: Techniques and methods.” *Renewable Energy*, 46, 169–178.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*, MIT, Cambridge, MA.

Nielsen, J. S., and Sorensen, J. D. (2012). “Maintenance optimization for offshore wind turbines using POMDP.” *Proc., 16th Conf. of Int. Federation for Information Processing on Reliability and Optimization of Structural Systems*, 175–182.

Nilim, A., and Ghaoui, L. E. (2005). “Robust solutions to Markov decision problems with uncertain transition matrices.” *Oper. Res.*, 53(5), 780–798.

Pineau, J., Gordon, G., and Thrun, S. (2003). “Point-based value iteration: An anytime algorithm for POMDPs.” *Proc., Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico.

Ross, S., Pineau, B., Chaib-draa, B., and Kreitmann, P. (2011). “A Bayesian approach for learning and planning in partially observable Markov decision process.” *J. Mach. Learn. Res.*, 12, 1729–1770.

Russell, S. J., and Norvig, P. (2010). *Artificial intelligence: A modern approach*, 3rd Ed., Prentice Hall, Upper Saddle River, NJ.

Smallwood, R. D., and Sondik, E. J. (1973). “The optimal control of partially observable Markov processes over a finite horizon.” *Oper. Res.*, 21(5), 1071–1088.

Sondik, E. J. (1978). “The optimal control of partially observable Markov processes over the infinite horizon.” *Oper. Res.*, 26(2), 282–304.

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement learning: An introduction*, MIT, Cambridge, MA.

Van Gael, J., Saatchi, Y., The, Y. W., and Ghahramani, Z. (2008). “Beam sampling for infinite hidden Markov model.” *Int. Conf. on Machine Learning*, Helsinki, Finland, 25.

Zhou, R., and Hansen, E. A. (2001). “An improved grid-based approximation algorithm for POMDPs.” *Proc. Int. Joint Conf. on Artificial Intelligence*, Seattle, Washington.