Continuous-Observation Partially Observable Semi-Markov Decision Processes for Machine Maintenance

Mimi Zhang, Matthew Revie

Partially observable semi-Markov decision processes (POSMDPs) provide a rich framework for planning under both state transition uncertainty and observation uncertainty. In this paper, we widen the literature on POSMDP by studying discrete-state, discrete-action yet continuous-observation POSMDPs. We prove that the resultant α-vector set is continuous and therefore propose a point-based value iteration algorithm. This paper also bridges the gap between POSMDP and machine maintenance by incorporating various types of maintenance actions, such as actions changing machine state, actions changing degradation rate, and the temporally extended action "do nothing". Both finite and infinite planning horizons are reviewed, and the solution methodology for each type of planning horizon is given. We illustrate the maintenance decision process via a real industrial problem and demonstrate that the developed framework can be readily applied to solve relevant maintenance problems.

***Index Terms*—Condition-based maintenance, degenerate distribution, imperfect maintenance, multi-state systems, point-based value iteration.**

Abbreviations & Acronyms

- SMDP: Semi-Markov Decision Process
- POMDP: Partially Observable Markov Decision Process
- POSMDP: Partially Observable Semi-Markov Decision Process
- ECD: Expected Cumulative Discounted
- RGF: Rapid Gravity Filter

Notation

- $\mathcal{S}$: State set
- $\mathcal{A}$: Action set
- $\mathcal{O}$: Continuous observation space
- U_z: Sojourn time between the zth and the $(z+1)$st transitions
- T_z: Time of the zth transition: $T_z = \sum_{k=0}^{z-1} U_k$
- $\ddot{S}_z$: Machine's state at epoch z
- S_t: Machine's state at time t: $S_t = \ddot{S}_z$ for $t_z \leq t < t_{z+1}$
- $\ddot{A}_z$: The maintenance action taken at epoch z
- A_t: The maintenance action taken at time t: $A_t = \ddot{A}_z$ for $t_z \leq t < t_{z+1}$
- $\ddot{O}_z$: The observation collected at epoch z
- $p_{ij}(a)$: Transition probability: if the machine is in state i and maintenance action a is taken on it, then it will transfer to state j with probability $p_{ij}(a)$
- $F_{ij}(u;a)$: Sojourn time distribution function: if the machine is in state i and will transfer to state j under maintenance action a, then the sojourn time follows $F_{ij}(u;a)$
- $f_{ij}(u;a)$: Density function of $F_{ij}(u;a)$
- $g_i(o;a)$: Density function of the observation: if the machine is in state i and maintenance action a is taken on it, then the observation has density function $g_i(o;a)$
- $\ddot{\boldsymbol{b}}_z$: Decision maker's belief state at epoch z: $\ddot{\boldsymbol{b}}_z = (\ddot{b}_z^1, \cdots, \ddot{b}_z^n)$
- $\ddot{\boldsymbol{b}}_t$: Decision maker's belief state at time t: $\ddot{\boldsymbol{b}}_t = \ddot{\boldsymbol{b}}_z$ for $t_z \leq t < t_{z+1}$
- $R_1(\ddot{S}_z, \ddot{A}_z)$: Immediate reward at epoch z
- $R_2(\ddot{S}_z, \ddot{A}_z)$: Reward with rate over the period $(t_z,\ t_{z+1})$
- $R(\boldsymbol{b}_t, \pi(\boldsymbol{b}_t))$: Instantaneous reward at time t for the SMDP model
- $\mathcal{R}_1(\ddot{S}_z, \ddot{A}_z)$: Immediate reward at epoch z for the POSMDP model
- $\mathcal{R}_2(\ddot{S}_z, \ddot{A}_z)$: Reward with rate over the period $(t_z,\ t_{z+1})$ for the POSMDP model
- $\mathcal{R}(\boldsymbol{b}_t, \pi(\boldsymbol{b}_t))$: Instantaneous reward at time t for the POSMDP model
- θ: discount factor
- w: Finite or infinite planning horizon
- $\pi(\cdot)$: Policy, mapping belief states into actions
- $V_\pi(\cdot)$: Value function
- $\{\alpha_z^k\}_k$: α-vector set at epoch z
- H: Bellman backup operator

Mimi Zhang and Matthew Revie are with the Department of Management Science, University of Strathclyde, Glasgow, G1 1XW, UK

I. Introduction

The emergence of technologically advanced data-collecting techniques, such as vibration monitoring, acoustics and physical condition monitoring, have been explored for improving reliability prediction and maintenance decision making. One popular choice of incorporating condition monitoring information into traditional lifetime data is through the proportional hazard model [1], [2], [3]. Another common choice is to directly model conditioning monitoring data by a stochastic process, e.g., the gamma process, [4], [5], [6]; failure is defined as the process exceeding a (random) threshold. However, in many practical applications, the physical condition of a machine[1] is characterized by a discrete set of states; see, e.g., [7] for road maintenance, [8] for power system management, [9] for production scheduling, and [10] for optimal replacement of wind turbines. Moreover, in many cases we are not permitted exact observations of the state of the machine. We can only model what is observable as probabilistically related to the true state of the machine; see, e.g., [11], [12], and [13]. In general, the partial observability stems from two sources. Firstly, different states can give the same observation. Secondly, sensor readings are noisy: observing the same state can result in different sensor readings.

For such applications, the decision-theoretic model of choice is a partially observable Markov decision process (POMDP). POMDPs provide a rich framework for planning under both state transition uncertainty and observation uncertainty. The POMDP model has been widely used for asset management under uncertainty; see [14] and the references therein. Note that POMDPs are not well suited for machine maintenance, as they are based on a discrete time step: the unitary action taken at time t affects the state and reward at time $t+1$. Yet, in machine maintenance, many maintenance actions take nontrivial time. For example, if we leave a machine operating for a period of time, then the action taken on the machine is “do nothing”, and the duration of “do nothing” is the time period over which the machine is operating. Hence, in the present paper, we introduce temporally extended actions into the POMDP model by studying the partially observable semi-Markov decision process (POSMDP). The POSMDP model was first proposed by [15] and then studied by, e.g., [16], [17] and [18].

Though the POSMDP model has existed for decades, there has been little effort in bridging the gap between POSMDP and machine maintenance. Moreover, the documented works on employing POSMDP in machine maintenance are all concerned with simple maintenance actions (e.g., perfect repair). A POSMDP problem is studied by [13] in which the state set contains only two elements, the maintenance action (repair) is simply perfect, and the observation space is discrete. In many practical problems, it is common for observations to be continuous, because sensors often provide continuous readings. Recently, [19], [20] and [21] coupled the POSMDP model with the Bayesian control chart. They assumed that, conditioned on the state of the machine, the observation vector follows a multivariate normal distribution. However, technically, the Bayesian control chart is introduced only to provide a threshold: if the posterior reliability of the machine drops to that threshold, a maintenance action will be performed. Moreover, the maintenance model they studied is rather simple: the state set only consists of two unobservable states and one observable failure state, and maintenance actions are assumed to be perfect. Another related work can be found in [22], who considered a POSMDP problem with continuous state space and continuous observation space. They then adopted a density projection method to convert the POSMDP problem into a semi-Markov decision process (SMDP) problem.

To date, the POSMDP model has not been developed for the case of discrete states, discrete actions and continuous observations. In addition, many of the applications of the POSMDP have failed to capture the subtleties of the maintenance actions available to decision makers. Other than perfect repair and minimal repair, maintenance engineers can often take an action which resets the machine to an earlier state but not to new, and/or reduces the rate of future degradation. An example of the former would be replacing some components of a complex machine and, for the latter, replacing the oil in an automobile to slow down deterioration of the gearbox. Motivated by the practical need of a maintenance-optimization tool for partially observable deteriorating systems, the current work proposes a POSMDP model with discrete state, discrete action yet continuous observation. The developed framework is generic and can be applied to a variety of cases: finite planning horizon, infinite planning horizon, and multi-dimensional observation space.

The remainder of the paper is organized as follows. In Section II, we derive the finite- and infinite-planning-horizon value functions of the continuous-observation POSMDP. In Section III, we study some properties of the value functions and develop a value-iteration algorithm. In Section IV, we illustrate, via an industrial problem, the incorporation of different maintenance actions (perfect, imperfect or minimal) into the POSMDP model. Section V gives numerical studies to show the feasibility and effectiveness of the proposed methods. Section VI concludes the paper.

II. Model Formulation

We begin with a brief review of the SMDP model and then generalize the SMDP model to the POSMDP model with the observation space being continuous. The reader is referred to [23] for the SMDP model, and [24] for partially observable Markov processes.

[1] A machine in this context could be any piece of mechanical equipment which requires periodic maintenance due to deterioration of its internal components over time.

A. Semi-Markov Decision Process

The standard SMDP model consists of two finite sets:

- a finite set of n states, labelled by $\mathcal{S} = \{1, 2, \cdots, n\}$;
- a finite set of v actions, labelled by $\mathcal{A} = \{1, 2, \cdots, v\}$.

Here, both n and v are positive integers. The states of the SMDP model are indeed the states of the machine, and a higher state represents a higher deterioration level of the machine. Typically, state 1 represents an excellent condition of the machine, while state n represents failure. The action set is composed of all the available maintenance actions that can be taken on the machine, e.g., $\mathcal{A} = \{1 = \text{"do nothing"}, 2 = \text{"imperfect repair"}, \cdots, v = \text{"replace"}\}$.

The state of the machine is unveiled (e.g., by thorough inspection) at random time points. Let $\{\ddot{S}_z,\ z \in \mathbb{Z}\}$ denote the evolving process of the state, with $\ddot{S}_z$ taking values from the state set $\mathcal{S}$. Here, $\mathbb{Z}$ is the set of nonnegative integers. Let $\{\ddot{A}_z,\ z \in \mathbb{Z}\}$ denote the action process to control the deterioration of the machine. At epoch z ($\in \mathbb{Z}$), after knowing the state (i.e., the value of $\ddot{S}_z$), a decision maker chooses an action $\ddot{A}_z = a$ from the action set $\mathcal{A}$, making the state of the machine change at the next epoch $(z+1)$. The sojourn time between the zth and the $(z+1)$st epochs is a positive random variable, denoted by U_z. Let T_z denote the time of the zth transition: $T_z = \sum_{k=0}^{z-1} U_k$; that is, T_z is the time at which the state of the machine transits from $\ddot{S}_{z-1}$ to $\ddot{S}_z$. Let t_z (resp. u_z) denote, from the generic point of view, the value of T_z (resp. U_z). Let S_t denote the state of the machine at time t (≥ 0) and A_t denote the action taken at time t. Clearly, for $t_z \leq t < t_{z+1}$, we have $S_t = \ddot{S}_z$ and $A_t = \ddot{A}_z$. Figure 1 illustrates the maintenance decision process under the SMDP setting.

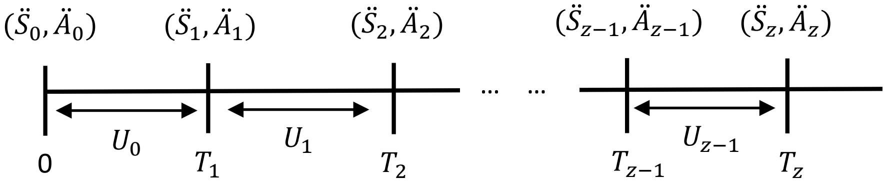
$(\ddot{S}_0, \ddot{A}_0)$
$(\ddot{S}_1, \ddot{A}_1)$
$(\ddot{S}_2, \ddot{A}_2)$
$(\ddot{S}_{z-1}, \ddot{A}_{z-1})$
$(\ddot{S}_z, \ddot{A}_z)$
U_0
U_1
U_{z-1}
0
T_1
T_2
T_{z-1}
T_z
··· ···

Fig. 1: At epoch z, we observe the state $\ddot{S}_z$. According to the state, we take action $\ddot{A}_z$. The machine remains at the present state for U_z units of time and then moves to state $\ddot{S}_{z+1}$ at time T_{z+1}.

The process $\{(S_t, A_t),\ t \geq 0\}$ is called an SMDP, if the following two Markovian properties are satisfied.

- Given the present state, states in the future do not depend on the states in the past.
- Given the present state and the next state, the distribution of the sojourn time does not depend on the states in the past.

The above two Markovian properties can be mathematically formulated as follows. Write $p_{ij}(a)$ ($i \in \mathcal{S}$, $a \in \mathcal{A}$ and $j \in \mathcal{S}$) for the law of motion of the SMDP at epoch z:

$$p_{ij}(a) = \Pr(\ddot{S}_{z+1} = j | \ddot{S}_0, \ddot{A}_0, U_0, \cdots, \ddot{S}_z = i, \ddot{A}_z = a).$$

Then we have

$$p_{ij}(a) = \Pr(\ddot{S}_{z+1} = j | \ddot{S}_0, \ddot{A}_0, U_0, \cdots, \ddot{S}_z = i, \ddot{A}_z = a) = \Pr(\ddot{S}_{z+1} = j | \ddot{S}_z = i, \ddot{A}_z = a).$$

Conditioned on the event that the next state is j, U_z has the distribution function $F_{ij}(u;\ a)$:

$$F_{ij}(u;\ a) = \Pr(U_z \leq u | \ddot{S}_0, \ddot{A}_0, U_0, \cdots, \ddot{S}_z = i, \ddot{A}_z = a, \ddot{S}_{z+1} = j) = \Pr(U_z \leq u | \ddot{S}_z = i, \ddot{A}_z = a, \ddot{S}_{z+1} = j),$$

with $u > 0$ and $F_{ij}(0;\ a) = 0$. For example, if the maintenance action taken at epoch z is "repair", then $F_{ij}(u;\ a)$ models the duration of the repair; if the maintenance action taken at epoch z is "do nothing", then $F_{ij}(u;\ a)$ models the time length of the machine staying in state i. Let $f_{ij}(u;\ a)$ denote the density function of U_z. By the law of total probability, we have

$$\Pr(U_z \leq u, \ddot{S}_{z+1} = j | \ddot{S}_z = i, \ddot{A}_z = a) = F_{ij}(u;\ a) p_{ij}(a).$$

That is, given that the current state is i and action a is taken on the machine, the state of the machine will, with probability $F_{ij}(u;\ a) p_{ij}(a)$, change to j within u units of time.

Remark 1. *A decision rule π_z at epoch z is a vector consisting of probabilities that are assigned to available actions. Specifically, an element $\pi_z(a)$ of π_z is the conditional probability of choosing action $a \in \mathcal{A}$ at the zth epoch: $\pi_z(a) = \Pr(\ddot{A}_z = a | \ddot{S}_0, \ddot{A}_0, U_0, \cdots, \ddot{S}_z = i)$. When concerned with finite (resp. infinite) horizon SMDPs, a policy π is a finite (resp. infinite) sequence of decision rules $\pi = \{\pi_0, \pi_1, \pi_2, \cdots\}$. π is called a Markov policy if $\pi_z(a)$ depends only on the present state: $\pi_z(a) = \Pr(\ddot{A}_z = a | \ddot{S}_z = i)$.*

Naturally, machine maintenance will bring about monetary costs and rewards. For the current work, we might treat a cost as a negative reward. At epoch z, after knowing the state $\ddot{S}_z$, the decision maker chooses an action $\ddot{A}_z$. Resulted from the action $\ddot{A}_z$ are an immediate reward $R_1(\ddot{S}_z, \ddot{A}_z)$ and a reward with rate $R_2(\ddot{S}_z, \ddot{A}_z)$ over the period $(t_z,\ t_{z+1})$. For example, if a wind turbine is shut down for maintenance, the immediate reward could be the cost of buying a new gearbox; the loss of the electricity income per day could be the reward rate. Let $R(S_t,A_t)$ denote the instantaneous reward function of the SMDP model:

$$R(S_t,A_t)=\begin{cases} R_1(\ddot{S}_z,\ddot{A}_z), & \text{if } t=t_z; \\ R_2(\ddot{S}_z,\ddot{A}_z), & \text{if } t_z<t<t_{z+1}. \end{cases}$$

B. Partially Observable semi-Markov Decision Process

Extending the SMDP setting, the POSMDP model further deals with uncertainty resulting from imperfect observations. It allows for maintenance scheduling for machines that are only partially observable to the decision maker. More formally, let $\{\ddot{O}_z,\ z\in\mathbb{Z}\}$ denote the information process, with $\ddot{O}_z$ taking values from a continuous observation space $\mathcal{O}$. The observation space $\mathcal{O}$ can be multi-dimensional. For example, an array of sensors (such as sonars, laser-range finders, video cameras and microphones) are equipped to provide partial information. Define a density function $g_i(o;a)$, modelling the probability of the event $\{\ddot{O}_z=o\}$ given that the action taken at epoch $(z-1)$ is a and that the current physical state of the machine is i:

$$\Pr(\ddot{O}_z\in\mathbb{O}|\ddot{A}_{z-1}=a,\ddot{S}_z=i)=\int_{\mathbb{O}}g_i(o;a)do, \tag{1}$$

where $\mathbb{O}$ is a subset of $\mathcal{O}$. Here and in the following, the symbol "d" denotes an infinitesimal change in a quantity. An assumption employed here is that the distribution of $\ddot{O}_z$ depends only on the last action and the present sate.

Now the maintenance decision process proceeds as follows. At epoch z, the decision maker receives an observation $\ddot{O}_z$. According to the received information, the decision maker updates his belief regarding the current state $\ddot{S}_z$. According to the newly updated belief, the decision maker then chooses an optimal action to control or detect the deterioration of the machine, believing that the machine will remain at the present state for U_z units of time and then move to state $\ddot{S}_{z+1}$ at time T_{z+1}. At epoch $(z+1)$, the decision maker collects a new observation $\ddot{O}_{z+1}$, and the whole maintenance decision process repeats. Note that, for the SMDP model, the present state $\ddot{S}_z$ is exactly known, and the decision maker chooses an action based on the present state. However, for the POSMDP model, the decision maker can only form a subjective judgement on the state of the machine and chooses an action based on such judgement. The underlying true state at epoch $(z+1)$, i.e. $\ddot{S}_{z+1}$, need not be different from $\ddot{S}_z$.

The decision maker's judgement on the state of the machine can be represented by a vector of probabilities (called belief state): $\ddot{\boldsymbol{b}}_z=(\ddot{b}_z^1,\cdots,\ddot{b}_z^n)$, where $\ddot{b}_z^i$ $(1\le i\le n)$ is the decision maker's probability with which the condition of the machine at epoch z is in state i: $\ddot{b}_z^i=\Pr(\ddot{S}_z=i|\cdot)$. The conditional term to the right of the vertical slash will be detailed in Equation (2). We have $\ddot{b}_z^i\ge 0$ and $\sum_{i=1}^n\ddot{b}_z^i=1$. At time 0, the decision maker's belief state $\ddot{\boldsymbol{b}}_0$ characterizes the prior knowledge regarding the condition of the machine before the beginning of the sequential decision making. Let $\boldsymbol{b}_t$ denote the decision maker's belief state at time t: $\boldsymbol{b}_t=\ddot{\boldsymbol{b}}_z$ for $t_z\le t<t_{z+1}$. It is easy to prove that the whole information available by epoch z, i.e. $\{\ddot{\boldsymbol{b}}_0,\ddot{A}_0,U_0,\cdots,\ddot{O}_{z-1},\ddot{A}_{z-1},U_{z-1},\ddot{O}_z\}$, can be summarized by $\ddot{\boldsymbol{b}}_z$, and to calculate $\ddot{\boldsymbol{b}}_{z+1}$, it is sufficient to know $\ddot{\boldsymbol{b}}_z$ (see, e.g., [25]). Therefore, we can incorporate the information received over the period $(T_z,T_{z+1}]$, i.e., $\{\ddot{A}_z,U_z,\ddot{O}_{z+1}\}$, and the previous belief state $\ddot{\boldsymbol{b}}_z$ to calculate the posterior distribution of the state of the machine. This is accomplished by applying the Bayes theorem:

$$\ddot{b}_{z+1}^i=\Pr(\ddot{S}_{z+1}=i|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a,U_z=u,\ddot{O}_{z+1}=o)=\frac{\Pr(U_z=u,\ddot{O}_{z+1}=o,\ddot{S}_{z+1}=i|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)}{\Pr(U_z=u,\ddot{O}_{z+1}=o|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)}, \tag{2}$$

in which we have

$$\begin{aligned}
&\Pr(U_z=u,\ddot{O}_{z+1}=o,\ddot{S}_{z+1}=i|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)\\
&\quad=\Pr(U_z=u,\ddot{O}_{z+1}=o|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a,\ddot{S}_{z+1}=i)\Pr(\ddot{S}_{z+1}=i|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)\\
&\quad=\sum_{j=1}^n\ddot{b}_z^j\Pr(U_z=u,\ddot{O}_{z+1}=o|\ddot{S}_z=j,\ddot{A}_z=a,\ddot{S}_{z+1}=i)\Pr(\ddot{S}_{z+1}=i|\ddot{S}_z=j,\ddot{A}_z=a)\\
&\quad=\sum_{j=1}^n\ddot{b}_z^jf_{ji}(u;a)g_i(o;a)p_{ji}(a),
\end{aligned} \tag{3}$$

and

$$\begin{aligned}
\Pr(U_z=u,\ddot{O}_{z+1}=o|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)&=\sum_{i=1}^n\Pr(U_z=u,\ddot{O}_{z+1}=o,\ddot{S}_{z+1}=i|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)\\
&=\sum_{i=1}^n\sum_{j=1}^n\ddot{b}_z^jf_{ji}(u;a)g_i(o;a)p_{ji}(a).
\end{aligned} \tag{4}$$

In Equation (3), we have utilized the fact that the two conditional events, $\{U_z=u|\ddot{S}_z=j,\ddot{A}_z=a,\ddot{S}_{z+1}=i\}$ and $\{\ddot{O}_{z+1}=o|\ddot{A}_z=a,\ddot{S}_{z+1}=i\}$, are independent. Equation (4) is a direct result of Equation (3). From Equation (2), we know that we can calculate our belief sate $\ddot{\boldsymbol{b}}_{z+1}$ if we know the values of $\ddot{A}_z$, U_z and $\ddot{O}_{z+1}$. Let $\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o)$ denote the updated belief sate at epoch $z+1$, conditioned on executing $\ddot{A}_z = a$ and, after u units of time, observing $\ddot{O}_{z+1} = o$; that is, $\ddot{\boldsymbol{b}}_{z+1} = \boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o)$. Figure 2 illustrates the maintenance decision process under the POSMDP setting.

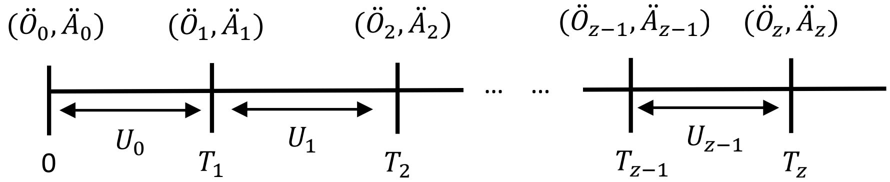
$(\ddot{O}_0, \ddot{A}_0)$
$(\ddot{O}_1, \ddot{A}_1)$
$(\ddot{O}_2, \ddot{A}_2)$
$(\ddot{O}_{z-1}, \ddot{A}_{z-1})$
$(\ddot{O}_z, \ddot{A}_z)$
U_0
U_1
U_{z-1}
0
T_1
T_2
T_{z-1}
T_z

Fig. 2: At epoch z, we receive an observation $\ddot{O}_z$. According to the value of ($\ddot{\boldsymbol{b}}_{z-1}$, $\ddot{A}_{z-1}$, U_{z-1}, $\ddot{O}_z$), we update the belief state to $\ddot{\boldsymbol{b}}_z$ and take action $\ddot{A}_z$. We will collect a new observation $\ddot{O}_{z+1}$ at time T_{z+1}, the time point at which we think the state of the machine will change to $\ddot{S}_{z+1}$.

The belief state $\ddot{\boldsymbol{b}}_z$ is a probability mass function over $\mathcal{S}$. All belief states are contained in an $(n-1)$-dimensional simplex Δ, implying that we can represent a belief state using $(n-1)$ numbers. For the SMDP model, a policy is a mapping from states to actions (see Remark 1). However, for the POSMDP model, due to the partial observability, a policy π is a mapping from belief states to actions. Hence, we might re-write the policy π into a functional form $\pi(\cdot) : \Delta \to \mathcal{A}$. In other words, given the belief state $\ddot{\boldsymbol{b}}_z$, the policy π will indicate which action to take at epoch z: $\ddot{A}_z = \pi(\ddot{\boldsymbol{b}}_z)$. If $\pi(\ddot{\boldsymbol{b}}_z) = a$ $(\in \mathcal{A})$, then, with probability $\ddot{b}_z^i$ $(1 \le i \le n)$, the decision maker can earn an immediate reward $R_1(i,a)$ and a reward with rate $R_2(i,a)$ over the period $(t_z,\ t_{z+1})$. Hence, the instantaneous reward function for the POSMDP model is given by

$$\mathcal{R}(\boldsymbol{b}_t, \pi(\boldsymbol{b}_t)) = \begin{cases} \mathcal{R}_1(\ddot{\boldsymbol{b}}_z, \pi(\ddot{\boldsymbol{b}}_z)), & \text{if } t = t_z, \\ \mathcal{R}_2(\ddot{\boldsymbol{b}}_z, \pi(\ddot{\boldsymbol{b}}_z)), & \text{if } t_z < t < t_{z+1}, \end{cases}$$

in which

$$\mathcal{R}_1(\ddot{\boldsymbol{b}}_z, \pi(\ddot{\boldsymbol{b}}_z)) = \sum_{i=1}^{n} \ddot{b}_z^i R_1(i, \pi(\ddot{\boldsymbol{b}}_z)), \quad \text{and} \quad \mathcal{R}_2(\ddot{\boldsymbol{b}}_z, \pi(\ddot{\boldsymbol{b}}_z)) = \sum_{i=1}^{n} \ddot{b}_z^i R_2(i, \pi(\ddot{\boldsymbol{b}}_z)).$$

Apparently, each policy will induce an expected cumulative (and possibly discounted by a discount factor) reward. The objective of a POSMDP problem is to work out a policy that will maximize the expected cumulative reward, which is the topic of Section II-C.

Now we can briefly define the POSMDP model which can be specified by a tuple $\langle \mathcal{S},\ \mathcal{A},\ \mathcal{O},\ p_{ij}(a),\ F_{ij}(u;a),\ g_i(o;a),\ \mathcal{R}(\boldsymbol{b}_t, \pi(\boldsymbol{b}_t)),\ \theta,\ w \rangle$, where θ $(\in (0,1))$ is a discount factor, and w (> 0) is a planning horizon. The discount factor, analogous to the interest rate, is for calculating the present value of future earnings. The planning horizon can be finite or infinite.

Remark 2. *Note that the parameters in $p_{ij}(a)$, $F_{ij}(u;a)$ and $g_i(o;a)$ have to be estimated from the observations of $\{\ddot{O}_1, \ddot{O}_2, \ddot{O}_3, \cdots\}$. This is a common parameter-estimation problem in the area of hidden Markov model. Efficient methods are available, e.g., the Baum-Welch algorithm proposed by [26]. We do not discuss parameter estimation here, as it is outside the scope of this paper.*

C. Value Function

For a POSMDP problem, the decision maker's objective is to work out a maintenance policy that optimizes a given reward criterion. Two criteria have been extensively used in the literature: the expected cumulative discounted (ECD) reward and the long-run average expected reward; see, e.g., [27]. For the current work, we only concentrate on the ECD reward. The ECD reward is easier to analyze and understand than the long-run average expected reward, since discounting lands itself naturally to economic problems in which future earnings are discounted by the interest rate. Moreover, the ECD reward criterion can be interpreted as putting more weight on initial decisions.

The quality of a policy, π, can be assessed by a value function $V_\pi(\cdot) : \Delta \to \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers. The function value $V_\pi(\boldsymbol{b})$ of the policy π is the ECD reward that will be earned over the planning horizon $[0, w]$ when starting in belief state $\boldsymbol{b} \in \Delta$. Among all candidate policies, if π yields the maximal function value $V_\pi(\boldsymbol{b})$ for all $\boldsymbol{b} \in \Delta$, then π is called the optimal policy; the optimal policy specifies the optimal action to execute at the current epoch, assuming that the decision maker will also act optimally in the future. In what follows, we let π^* denote the optimal policy.

At time t_z $(< w)$, given that the decision maker's belief state has been updated to $\ddot{\boldsymbol{b}}_z$, we then need to determine the optimal maintenance action $\ddot{A}_z$. Let $V_z(\ddot{\boldsymbol{b}}_z)$ denote the ECD reward (discounted to time t_z) the decision maker can obtain by following the optimal policy π^*:

$$V_z(\ddot{\boldsymbol{b}}_z) = E\left[\int_{t_z}^{w} \exp(-\theta(t - t_z)) \mathcal{R}(\boldsymbol{b}_t, \pi^*(\boldsymbol{b}_t)) dt \middle| \ddot{\boldsymbol{b}}_z \right],$$

where E is the conditional expectation operator for the processes $\{\ddot{S}_z,\ z \in \mathbb{Z}\}$, $\{U_z,\ z \in \mathbb{Z}\}$ and $\{\ddot{O}_z,\ z \in \mathbb{Z}\}$, given the a priori probability vector $\ddot{\boldsymbol{b}}_z$.

When the planning horizon is finite, i.e., $w < \infty$, define $\bar{w} = \max\{k : t_k \leq w\}$. The ECD reward can be further written into

$$\begin{aligned} V_z(\ddot{\boldsymbol{b}}_z) = & E\left[\mathcal{R}_1(\ddot{\boldsymbol{b}}_z, \pi^*(\ddot{\boldsymbol{b}}_z)) + \frac{1-\exp(-\theta U_z)}{\theta}\mathcal{R}_2(\ddot{\boldsymbol{b}}_z, \pi^*(\ddot{\boldsymbol{b}}_z)) \middle| \ddot{\boldsymbol{b}}_z\right] \\ & + E\left[\sum_{k=z+1}^{\bar{w}-1} \exp(-\theta \sum_{r=z}^{k-1} U_r) \left[\mathcal{R}_1(\ddot{\boldsymbol{b}}_k, \pi^*(\ddot{\boldsymbol{b}}_k)) + \frac{1-\exp(-\theta U_k)}{\theta}\mathcal{R}_2(\ddot{\boldsymbol{b}}_k, \pi^*(\ddot{\boldsymbol{b}}_k))\right] \middle| \ddot{\boldsymbol{b}}_z\right] \\ & + E\left[\exp(-\theta \sum_{r=z}^{\bar{w}-1} U_r) \left[\mathcal{R}_1(\ddot{\boldsymbol{b}}_{\bar{w}}, \pi^*(\ddot{\boldsymbol{b}}_{\bar{w}})) + \frac{1-\exp(-\theta(w-T_{\bar{w}}))}{\theta}\mathcal{R}_2(\ddot{\boldsymbol{b}}_{\bar{w}}, \pi^*(\ddot{\boldsymbol{b}}_{\bar{w}}))\right] \middle| \ddot{\boldsymbol{b}}_z\right]. \end{aligned}$$

If $\pi^*(\ddot{\boldsymbol{b}}_z) = a \ (\in \mathcal{A})$, we have

$$\begin{aligned} E\left[\exp(-\theta U_z)\middle|\ddot{\boldsymbol{b}}_z\right] &= \int_0^{w-t_z} \exp(-\theta u)\Pr(U_z = u|\ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)du \\ &= \int_0^{w-t_z} \sum_{i=1}^n \ddot{b}_z^i \sum_{j=1}^n \exp(-\theta u)\Pr(\ddot{S}_{z+1} = j, U_z = u|\ddot{S}_z = i, \ddot{A}_z = a)du \\ &= \int_0^{w-t_z} \sum_{i=1}^n \ddot{b}_z^i \sum_{j=1}^n \exp(-\theta u) f_{ij}(u;\ a) p_{ij}(a)du. \end{aligned}$$

Define an n-dimensional vector $\bar{R}_z^a$ with elements

$$\bar{R}_z^a(i) = R_1(i,a) + \frac{1}{\theta}\left\{1 - E\left[\exp(-\theta U_z)\middle|\ddot{\boldsymbol{b}}_z\right]\right\} R_2(i,a), \qquad 1 \leq i \leq n.$$

Then the ECD reward can be written in a recursive form:

$$V_z(\ddot{\boldsymbol{b}}_z) = \max_{a\in\mathcal{A}}\left\{\sum_{i=1}^n \bar{R}_z^a(i)\ddot{b}_z^i + \int_0^{w-t_z}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)\exp(-\theta u)V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))dodu\right\}. \tag{5}$$

Here, $V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))$ is the ECD reward (discounted to time t_{z+1}) when starting in belief state $\ddot{\boldsymbol{b}}_{z+1} = \boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o)$. The expression for the probability $\Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)$ is given by Equation (4). When the planning horizon is infinite, i.e., $w = +\infty$, we have

$$\begin{aligned} V_z(\ddot{\boldsymbol{b}}_z) = & E\left[\mathcal{R}_1(\ddot{\boldsymbol{b}}_z, \pi^*(\ddot{\boldsymbol{b}}_z)) + \frac{1-\exp(-\theta U_z)}{\theta}\mathcal{R}_2(\ddot{\boldsymbol{b}}_z, \pi^*(\ddot{\boldsymbol{b}}_z)) \middle| \ddot{\boldsymbol{b}}_z\right] \\ & + E\left[\sum_{k=z+1}^{+\infty} \exp(-\theta \sum_{r=z}^{k-1} U_r) \left[\mathcal{R}_1(\ddot{\boldsymbol{b}}_k, \pi^*(\ddot{\boldsymbol{b}}_k)) + \frac{1-\exp(-\theta U_k)}{\theta}\mathcal{R}_2(\ddot{\boldsymbol{b}}_k, \pi^*(\ddot{\boldsymbol{b}}_k))\right] \middle| \ddot{\boldsymbol{b}}_z\right] \\ = & \max_{a\in\mathcal{A}}\left\{\sum_{i=1}^n \bar{R}_z^a(i)\ddot{b}_z^i + \int_0^{\infty}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)\exp(-\theta u)V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))dodu\right\}. \end{aligned} \tag{6}$$

The optimal action to take at epoch z should be the one that maximizes $V_z(\ddot{\boldsymbol{b}}_z)$:

$$\pi^*(\ddot{\boldsymbol{b}}_z) = \arg\max_{a\in\mathcal{A}} V_z(\ddot{\boldsymbol{b}}_z).$$

From Equation (5) we know that V_z is a function of V_{z+1}. We might simply write the recursion (5) as $V_z = HV_{z+1}$:

$$(HV_{z+1})(\ddot{\boldsymbol{b}}_z) = \max_{a\in\mathcal{A}}\left\{\sum_{i=1}^n \bar{R}_z^a(i)\ddot{b}_z^i + \int_0^{w-t_z}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)\exp(-\theta u)V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))dodu\right\}. \tag{7}$$

H is called the Bellman backup operator [28]. This notation is defined here only to facilitate the proof for Proposition 2. Note that, if the planning horizon w is finite, the value function $V_{z+1}(\cdot)$ changes with the value of U_z. Yet, if the planning horizon is infinite, the value function $V_{z+1}(\cdot)$ does not change with u_z. As will be proved in Proposition 2, both $V_z(\cdot)$ and $V_{z+1}(\cdot)$ are identical to the underlying true value function, denoted by $V(\cdot)$. The main objective of Section III is to approximate this true value function. When $w = +\infty$, the recursion (6) can be written in the Bellman functional form $V = HV$:

$$(HV)(\ddot{\boldsymbol{b}}_z) = \max_{a\in\mathcal{A}}\left\{\sum_{i=1}^n \bar{R}_z^a(i)\ddot{b}_z^i + \int_0^{\infty}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)\exp(-\theta u)V(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))dodu\right\}. \tag{8}$$

III. A Value-Iteration Algorithm

In the literature, there are several algorithms for computing an optimal policy: value iteration, policy iteration and linear programming; see, e.g., [27]. Herein, we develop a value-iteration algorithm. Value-iteration algorithms compute a sequence of value functions in a backward manner starting from a lower bound on the true value function. We first concentrate on the theoretical basis on which to develop the value-iteration algorithm.

Proposition 1. *Let $< \cdot, \cdot >$ denote the inner product operator. The optimal value function $V_z(\cdot)$ at time t_z can be expressed as*

$$V_z(\boldsymbol{b}) = \sup_{\{\alpha_z^k\}_k} < \alpha_z^k,\ \boldsymbol{b} >, \qquad \boldsymbol{b} \in \Delta.$$

Here, $\{\alpha_z^k\}_k$ is a continuous set of vectors, and $\alpha_z^k = (\alpha_z^k(1), \cdots, \alpha_z^k(n))$.

Proof. We here only prove that $V_z(\cdot)$ can be written into the inner-product form, assuming that $\{\alpha_z^k\}_k$ is a continuous set for $z < \bar{w}$. The continuity of the set $\{\alpha_z^k\}_k$ is proved in Appendix A.

The proof is done via induction. At epoch $\bar{w}$ we have

$$V_{\bar{w}}(\ddot{\boldsymbol{b}}_{\bar{w}}) = \mathcal{R}_1(\ddot{\boldsymbol{b}}_{\bar{w}}, \pi^*(\ddot{\boldsymbol{b}}_{\bar{w}})) + \frac{1}{\theta}\left\{1 - E\left[\exp(-\theta(w - T_{\bar{w}}))\middle|\ddot{\boldsymbol{b}}_{\bar{w}}\right]\right\}\mathcal{R}_2(\ddot{\boldsymbol{b}}_{\bar{w}}, \pi^*(\ddot{\boldsymbol{b}}_{\bar{w}})) = \max_{a \in \mathcal{A}} < \bar{R}_{\bar{w}}^a,\ \ddot{\boldsymbol{b}}_{\bar{w}} > .$$

Then we define the set $\{\alpha_{\bar{w}}^k\}_k = \{\bar{R}_{\bar{w}}^a\}_{a \in \mathcal{A}}$ and have $V_{\bar{w}}(\ddot{\boldsymbol{b}}_{\bar{w}}) = \max_{\{\alpha_{\bar{w}}^k\}_k} < \alpha_{\bar{w}}^k,\ \ddot{\boldsymbol{b}}_{\bar{w}} >$.

Now assume that $V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))$ can be written into the inner-product form:

$$V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o)) = \sup_{\{\alpha_{z+1}^k\}_k} < \alpha_{z+1}^k,\ \boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o) > .$$

From Equation (2) we have

$$V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o)) = \sup_{\{\alpha_{z+1}^k\}_k} \sum_{i=1}^n \alpha_{z+1}^k(i) l_a^i(\ddot{\boldsymbol{b}}_z, u, o) = \frac{\sup_{\{\alpha_{z+1}^k\}_k} \sum_{i=1}^n \alpha_{z+1}^k(i) \sum_{j=1}^n \ddot{b}_z^j f_{ji}(u;a) g_i(o;a) p_{ji}(a)}{\Pr(U_z = u, \ddot{O}_{z+1} = o | \ddot{\boldsymbol{b}}_z, \ddot{A}_z = a)}.$$

Substitute $V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z, u, o))$ into Equation (5):

$$\begin{aligned} V_z(\ddot{\boldsymbol{b}}_z) &= \max_{a \in \mathcal{A}} \left\{ < \bar{R}_z^a, \ddot{\boldsymbol{b}}_z > + \int_0^{w-t_z} \int_{\mathcal{O}} \exp(-\theta u) \sup_{\{\alpha_{z+1}^k\}_k} \left\{ \sum_{i=1}^n \alpha_{z+1}^k(i) \sum_{j=1}^n \ddot{b}_z^j f_{ji}(u;a) g_i(o;a) p_{ji}(a) \right\} do du \right\} \\ &= \max_{a \in \mathcal{A}} \left\{ < \bar{R}_z^a, \ddot{\boldsymbol{b}}_z > + \int_0^{w-t_z} \int_{\mathcal{O}} \exp(-\theta u) \sup_{\{\alpha_{z+1}^k\}_k} \left\{ \sum_{j=1}^n \left[\sum_{i=1}^n \alpha_{z+1}^k(i) f_{ji}(u;a) g_i(o;a) p_{ji}(a) \right] \ddot{b}_z^j \right\} do du \right\}. \end{aligned} \tag{9}$$

Let $\delta_k^a(u,o) = (\delta_{k1}^a(u,o), \cdots, \delta_{kn}^a(u,o))$ denote a vector-valued function:

$$\delta_{kj}^a(u,o) = \sum_{i=1}^n \alpha_{z+1}^k(i) f_{ji}(u;a) g_i(o;a) p_{ji}(a), \qquad 1 \le j \le n. \tag{10}$$

Note that, for fixed u and o, the cardinality of $\{\delta_k^a(u,o)\}$ is exactly the cardinality of $\{\alpha_{z+1}^k\}$, and $\delta_k^a(u,o)$ is independent of $\ddot{\boldsymbol{b}}_z$. Then we have

$$V_z(\ddot{\boldsymbol{b}}_z) = \max_{a \in \mathcal{A}} \left\{ < \bar{R}_z^a, \ddot{\boldsymbol{b}}_z > + \int_0^{w-t_z} \int_{\mathcal{O}} \exp(-\theta u) \sup_{\{\delta_k^a(u,o)\}_k} < \delta_k^a(u,o),\ \ddot{\boldsymbol{b}}_z > do du \right\}.$$

We can write

$$\begin{aligned} V_z(\ddot{\boldsymbol{b}}_z) &= \max_{a \in \mathcal{A}} < \bar{R}_z^a + \int_0^{w-t_z} \int_{\mathcal{O}} \exp(-\theta u) \arg \sup_{\{\delta_k^a(u,o)\}_k} < \delta_k^a(u,o),\ \ddot{\boldsymbol{b}}_z > do du,\ \ddot{\boldsymbol{b}}_z > \\ &= \max_{a \in \mathcal{A}} < \delta_a(\ddot{\boldsymbol{b}}_z),\ \ddot{\boldsymbol{b}}_z >, \end{aligned}$$

where

$$\delta_a(\ddot{\boldsymbol{b}}_z) = \bar{R}_z^a + \int_0^{w-t_z} \int_{\mathcal{O}} \exp(-\theta u) \arg \sup_{\{\delta_k^a(u,o)\}_k} < \delta_k^a(u,o),\ \ddot{\boldsymbol{b}}_z > do du. \tag{11}$$

Then we can simply define the set $\{\alpha_z^k\}_k$ as

$$\{\alpha_z^k\}_k = \bigcup_{\boldsymbol{b}\in\Delta}\{\arg\max_{\{\delta_a(\boldsymbol{b})\}_{a\in\mathcal{A}}} <\delta_a(\boldsymbol{b}),\ \boldsymbol{b}>\}.$$

Therefore, $\{\alpha_z^k\}_k$ is a continuous set of vectors parameterized in the action set; that is, each vector is associated with an action, which is the optimal action for the belief state that has such vector as the maximizing one. Consequently, $V_z(\boldsymbol{b})$ can be put in the desired form

$$V_z(\boldsymbol{b}) = \sup_{\{\alpha_z^k\}_k} <\alpha_z^k,\ \boldsymbol{b}> .$$

□

The elements in $\{\alpha_z^k\}_k$ are usually called α-vectors. Let Ω_z denote the set of α-vectors: $\Omega_z = \{\alpha_z^k\}_k$. Using Proposition 1 we can readily prove that the value function $V_z(\boldsymbol{b})$ is convex. Since the inner product operator is linear in its two arguments, the convex property is given by the fact that $V_z(\boldsymbol{b})$ is defined as the supreme of a set of convex (linear) functions.

Proposition 2. *Let $||\cdot||$ denote the supreme norm. For the Bellman backup operator H given by Equation (8) and two value functions V_{z+1} and U_{z+1}, it holds that $||U_z - V_z|| = ||HU_{z+1} - HV_{z+1}|| < ||U_{z+1} - V_{z+1}||$. Moreover, if $V_{z+1} \geq U_{z+1}$, then $V_z \geq U_z$. That is, the backup operator H is a contractive and isotonic mapping.*

The proof is given in Appendix B. According to Proposition 1, the space of value functions is closed under addition and scalar scaling. The contraction property further ensures that this space is complete. Therefore, the space of value functions together with the supreme norm form a Banach space. The Banach fixed-point theorem ensures the existence of a single fixed point and that the value iteration always converges to this point. The isotonic property ensures that value iteration converges monotonically.

Propositions 1 and 2 indicate that value iteration is a promising method for policy optimization. Equations (10) and (11) provide a constructive way to define the set of vectors $\{\alpha_z^k\}_k$. The computation of the α-vector for a given belief point $\boldsymbol{b}$ is called a backup:

$$\text{backup}(\boldsymbol{b}) = \arg\sup_{\{\alpha_z^k\}_k} <\alpha_z^k,\ \boldsymbol{b}> = \arg\max_{\{\delta_a(\boldsymbol{b})\}_{a\in\mathcal{A}}} <\delta_a(\boldsymbol{b}),\ \boldsymbol{b}> .$$

Using the backup operator, the value of $V_z(\boldsymbol{b})$ is simply $V_z(\boldsymbol{b}) = <\text{backup}(\boldsymbol{b}), \boldsymbol{b}>$. A backup for the whole belief space requires the computation of all the α-vectors. However, a whole backup is impossible when the observation space is continuous, because the set $\{\alpha_z^k\}_k$ is continuous, having infinitely many vectors. Consequently, we employ the idea of point-based value iteration (PBVI), which performs backup on a finite set of belief points [29]. The α-vectors for the restricted set of belief points form an approximation to the set $\{\alpha_z^k\}_k$, and they can be used to approximate the true value function for any belief point. Many extensions have been suggested to the idea of PBVI. [30] provided a comprehensive overview of existing point-based solvers. In this paper we employ, with some modifications, the Perseus algorithm developed by [31].

An intuitive approach to selecting belief points would be to maintain a regular grid of belief points. One downside of such an approach is that it is highly probable that many of these belief points are not reachable. The Perseus algorithm starts with collecting a set B of reachable belief points by using Algorithm 1, in which the initial belief point $\boldsymbol{b}$ is provided by the decision maker. Then the Perseus algorithm proceeds to value-function updating, performed only on the belief points in B. Particularly, given a set Ω_{z+1} of finite α-vectors obtained at epoch $z+1$, [31] developed Algorithm 2 for approximating the value function $V_z(\cdot)$. Starting from epoch $\bar{w}$, repeatedly applying Algorithm 2 until we arrive at $z=0$ (for $w<+\infty$) or until the value function is stable (for $w=+\infty$). Note that, for finite planning horizon, if the observation space is discrete (namely, the α-vector sets are all discrete), then we need not use Algorithm 2; we can exactly derive every element in each α-vector set. However, since the α-vector sets for a continuous-observation POSMDP are all continuous (except the starting one), when employing the idea of point-based value iteration, Algorithm 2 will give a better approximation to each true value function.

Algorithm 1 Random exploration

```
$B = \{\boldsymbol{b}\}$.
while the cardinality of $B$ is smaller than a threshold do
    Randomly simulate a state, denoted by $j$, according to the distribution $\boldsymbol{b}$.
    Uniformly simulate an action from $\mathcal{A}$, denoted by $a$.
    Randomly simulate a state, denoted by $i$, according to $(p_{j1}(a), p_{j2}(a), \cdots, p_{jn}(a))$.
    Randomly simulate a sojourn time, denoted by $u$, from the distribution $f_{ji}(u;a)$.
    Randomly simulate an observation, denoted by $o$, from the distribution $g_i(o;a)$.
    Calculate the posterior distribution $\boldsymbol{l}_a(\boldsymbol{b},u,o)$ according to Equation (2).
    Add $\boldsymbol{l}_a(\boldsymbol{b},u,o)$ to $B$ and initialize $\boldsymbol{b}$ to $\boldsymbol{l}_a(\boldsymbol{b},u,o)$.
end while
```

Algorithm 2 Value-function updating

```
Set $\Omega_z$ to be empty and initialize $\tilde{B}$ to $B$.
while $\tilde{B}$ is not empty do
    Randomly sample a belief point $\boldsymbol{b}$ from $\tilde{B}$ and compute $\alpha = \text{backup}(\boldsymbol{b})$ based on the $\alpha$-vectors in the set $\Omega_{z+1}$.
    if $<\alpha,\ \boldsymbol{b}>$ is larger than $V_{z+1}(\boldsymbol{b})$ then
        Add $\alpha$ to the set $\Omega_z$. Remove from $\tilde{B}$ all the belief points that can be improved by $\alpha$.
    else
        Add $\hat{\alpha} = \arg\max_{\{\alpha^k_{z+1}\}_k} <\alpha^k_{z+1},\ \boldsymbol{b}>$ to the set $\Omega_z$. Remove from $\tilde{B}$ all the belief points that can be maximized by $\hat{\alpha}$.
    end if
end while
```

When w is infinite, the Perseus algorithm requires that the value function $V_z(\cdot)$ be a lower bound on $V(\cdot)$. To this end, we only need to make $V_{\ddot{w}}(\cdot)$ be a lower bound. We define $R = \min_{s\in\mathcal{S},a\in\mathcal{A}}\{R_2(s,a)\}$ and set $\{\alpha^k_{\ddot{w}}\}_k = \{\alpha\}$:

$$\alpha(i) = \int_0^\infty \exp(-\theta u) R du = \frac{R}{\theta}, \qquad i = 1, \cdots, n.$$

$\alpha(i)$ is equivalent to the present value of the cash flow in which we only receive the minimal possible reward rate (no immediate reward).

There are a number of potential disadvantages to Perseus though; see [30]. One is that the random exploration is not optimal in that it may not encounter most of the same points as the optimal policy. We therefore propose to start with a relatively small size of B, e.g., 500 belief points. When the whole value-function updating procedure is completed, we add into B new belief points by utilizing the newly computed optimal policy (see Algorithm 3). We iteratively apply Algorithms 2 and 3 until the cardinality of B is larger than a pre-determined threshold.

Algorithm 3 Expand B

```
Initialize $\ddot{B}$ to $B$
for each $\boldsymbol{b} \in B$ do
    Randomly simulate a state, denoted by $j$, according to the distribution $\boldsymbol{b}$.
    Using the newly computed optimal policy to derive the optimal action for $\boldsymbol{b}$: the action associated with the vector that maximizes $\boldsymbol{b}$.
    Randomly simulate a state, denoted by $i$, according to $(p_{j1}(a), p_{j2}(a), \cdots, p_{jn}(a))$.
    Randomly simulate a sojourn time, denoted by $u$, from the distribution $f_{ji}(u;a)$.
    Randomly simulate an observation, denoted by $o$, from the distribution $g_i(o;a)$.
    Calculate the posterior distribution $\boldsymbol{l}_a(\boldsymbol{b},u,o)$ according to Equation (2)
    If $\boldsymbol{l}_a(\boldsymbol{b},u,o) \notin \ddot{B}$, add $\boldsymbol{l}_a(\boldsymbol{b},u,o)$ to $\ddot{B}$
end for
Return $\ddot{B}$
```

IV. Incorporating Different Maintenance Actions

The generic POSMDP model in Section II-B implies that maintenance actions (either changing machine state or changing deteriorating rate) can be fully characterized by the state transition matrices and sojourn time distributions. Perhaps a better way to explain how to characterize maintenance efficiency is through a heuristic example. In Section IV-A, we introduce a real industrial problem that involves different types of maintenance actions. In Section IV-B, we derive the corresponding state transition matrices to characterize the involved maintenance actions. In Appendix C, we present a detailed procedure of the belief point backup for the following industrial problem.

A. An Industrial Problem

To provide water service, a water utility operates water treatment works spreading over the UK. Raw water enters into a water treatment works, and clean water exits from the water treatment works. A water treatment works is a complex system with different components. Rapid gravity filters (RGFs) have been identified as the key components for purifying water. RGFs purify water by filtering water through a particular media. Experts within the water utility classify the condition of an RGF (according to the condition of the media) into four different states: {“good”, “acceptable”, “poor”, “awful”}. The condition of an RGF is not directly observable but can be revealed by a thorough inspection, requiring an expenditure of time and personnel while rendering the RGF unproductive for the duration of the inspection.

To guarantee that high water quality is achieved, the water utility records information on many features. An important feature is the turbidity in the water. Turbidity is the haziness of a fluid caused by individual particles and hence is naturally treated as a continuous random variable. The level of turbidity is recorded at key stages of the water treatment process (e.g., upon entry, pre-post processing at RGFs, on exit). When the ratio of outgoing to incoming water turbidity is close to zero, then the RGF is likely to be in a good condition. On the other hand, when the ratio is close to one, then the RGF is likely to be in a poor or even awful condition. Therefore, by comparing the turbidity in the water entering and exiting an RGF, we can infer from the probabilistic point of view the condition of the RGF.

To achieve better reliability and availability of the RGFs, experts in the water utility take maintenance actions on the RGFs. Typical maintenance actions are "do nothing", "backwash", "dose chemicals" and "replace". Chemical dosing is to change the state of the RGF, while backwash will slow down the deterioration of the RGF. By replacing the media, the RGF will be renewed to the good state. Experts in the water utility backwash an RGF regularly, e.g., every other day, irrespective of the condition of the RGF, which is obviously not economical. If an RGF fails, the maintenance work will not be conducted until related funding is approved. During the downtime of the failed RGF, other RGFs in the water treatment works will bear the extra working load, leading to increased deterioration of the other RGFs or additional operational costs. Evidently, the current maintenance practice in the water utility is not efficient, and there is an economic need to better plan maintenance activities.

B. Characterizing Different Maintenance Actions

The state set of the RGF has four elements: $\mathcal{S} = \{1=\text{"good"}, 2=\text{"acceptable"}, 3=\text{"poor"}, 4=\text{"awful"}\}$. The action set is defined as $\mathcal{A} = \{1=\text{"do nothing"}, 2=\text{"backwash"}, 3=\text{"dose chemicals"}, 4=\text{"replace"}\}$. Apparently, action 1 is minimal; actions 2 and 3 are imperfect; action 4 is perfect. We now discuss how to determine the state transition matrix for each action.

The state transition matrix of action 1 depends on the duration of action 1. For example, the transition probability $p_{12}(a=1)$ when action 1 lasts for one day will be different from $p_{12}(a=1)$ when action 1 lasts for one month. In order to determine the state transition matrix we have to first determine the duration of action 1, which is explained as follows. If the RGF is not maintained (or, equivalently, if the action taken is always "do nothing"), then the RGF will deteriorate gradually from state i to state $i+1$ for $i=1,2,3$. Let random variable T_i^1 denote the time length of the RGF staying in state i. For example, the RGF will stay in state 1 for T_1^1 units of time and then move to state 2. In practice, action 1 will be taken if the RGF is in the good state, and will last until the RGF is in the acceptable state. Hence, we let the duration of action 1 be fixed at u_1^* such that the probability $\Pr(T_1^1 < u_1^*, T_1^1 + T_2^1 \geq u_1^*)$ is maximized. In other words, if the RGF is in state 1 and we do nothing for u_1^* units of time, then by the end of action 1, the RGF will be in state 2 with the maximal probability. Consequently, the state transition matrix can be derived:

$$
\begin{aligned}
& P(a=1) \\
= & \begin{bmatrix}
\Pr(T_1^1 \geq u_1^*) & \Pr(T_1^1 < u_1^*, T_1^1 + T_2^1 \geq u_1^*) & \Pr(\sum\limits_{i=1}^{2} T_i^1 < u_1^*, \sum\limits_{i=1}^{3} T_i^1 \geq u_1^*) & 1 - \sum\limits_{i=1}^{3} p_{1i}(a=1) \\
0 & \Pr(T_2^1 \geq u_1^*) & \Pr(T_2^1 < u_1^*, T_2^1 + T_3^1 \geq u_1^*) & 1 - \sum\limits_{i=1}^{3} p_{2i}(a=1) \\
0 & 0 & \Pr(T_3^1 \geq u_1^*) & \Pr(T_3^1 < u_1^*) \\
0 & 0 & 0 & 1
\end{bmatrix}.
\end{aligned}
$$

Before determining the state transition matrix for action 2, we need to modify action 2, the reason for which is explained as follows. At epoch z, after updating the belief state, the decision maker decides to backwash the RGF. The duration of backwash is constant (not a random variable), and is denoted by u_W. Because backwash does not change the state of the RGF, at time $t_{z+1} = t_z + u_W$ we have $S_{z+1} = S_z = i$. Then, at time t_{z+1}, the decision maker decides to take action 1. The RGF stays in state i for U_{z+1} units of time and then transits to state j. However, as the backwash slows down the deterioration of the RGF, U_{z+1} no longer follows the distribution $F_{ij}(u; a=1)$. In other words, the sojourn time distribution $F_{ij}(u;a)$ is no longer stationary, but depends on z. In such case, the theoretical work developed in Section III cannot be applied for policy optimization. Note that the effect of backwash is temporary; that is, backwash at epoch z only affects the distribution of U_{z+1}. If the action taken at epoch $z+2$ is not backwash, then U_{z+2} again follows the distribution $F_{ij}(u;a)$. Moreover, compared to the lifetime of the RGF, the duration of backwash is negligible. Since action 2 is always followed by action 1, we propose to merge the two consecutive actions $\ddot{A}_z = 2$ and $\ddot{A}_{z+1} = 1$ into one action. The action of backwashing the RGF and then doing nothing is called "backwash and watch"; the action set is now $\mathcal{A} = \{1=\text{"do nothing"}, 2=\text{"backwash and watch"}, 3=\text{"dose chemicals"}, 4=\text{"replace"}\}$. At epoch z, we take action "backwash and watch", which lasts for a random period of time U_z. Then at time $t_{z+1} = t_z + u_z$, the state of the RGF transits to state j. By replacing "backwash" with "backwash and watch", the resulting decision process is now back to stationary. In the following, by action 2, we always mean "backwash and watch".

The state transition matrix of action 2 also depends on the duration of action 2. Hence, we first need to determine the optimal duration of action 2. If the action taken is always "backwash and watch", then the RGF will also deteriorate gradually from state i to state $i+1$ for $i=1,2,3$. Let T_i^2 denote the time length of the RGF staying in state i. Owing to backwash, T_i^2 should be statistically larger than T_i^1. In practice, action 2 will be conducted if the RGF is in the acceptable state, and will last until the RGF is in the poor state. Hence, we let the duration of action 2 be fixed at u_2^* such that the probability $\Pr(T_2^2 < u_2^*, T_2^2 + T_3^2 \geq u_2^*)$ is maximized. In other words, if the RGF is in state 2, we backwash it and then do nothing for u_2^* units of time; by the end of action 2, the RGF will be in state 3 with the maximal probability. Consequently, the state transition matrix can be derived:

$$\begin{aligned} & P(a=2) \\ = & \begin{bmatrix} \Pr(T_1^2 \geq u_2^*) & \Pr(T_1^2 < u_2^*, T_1^2 + T_2^2 \geq u_2^*) & \Pr(\sum_{i=1}^{2} T_i^2 < u_2^*, \sum_{i=1}^{3} T_i^2 \geq u_2^*) & 1 - \sum_{i=1}^{3} p_{1i}(a=2) \\ 0 & \Pr(T_2^2 \geq u_2^*) & \Pr(T_2^2 < u_2^*, T_2^2 + T_3^2 \geq u_2^*) & 1 - \sum_{i=1}^{3} p_{2i}(a=2) \\ 0 & 0 & \Pr(T_3^2 \geq u_2^*) & \Pr(T_3^2 < u_2^*) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

The effect of action 3 can also be readily incorporated by specifying the corresponding state transition matrix. For example, $p_{21}(a=3) = 0.5$ indicates that, if the machine's current state is 2 and we take action 3, then the machine will change to state 1 with probability 0.5.

The effect of action 4 is simple: via replacement, the machine will be renewed to state 1, irrespective of its current state. Mathematically, we have $p_{ij}(a=4) = I(j=1)$, where $I(\cdot)$ is the indicator function and $1 \leq i, j \leq 4$.

Remark 3. *We may also face the case where the effect of "backwash" is transient. In other words, the effect of "backwash" will vanish before the machine changes its state. This could happen when the sojourn times usually take large values. In such case, we still employ the action "backwash and watch"; the only difference is that the duration of "backwash and watch" is no longer fixed at an optimal value, but follows a pre-determined distribution.*

We may have noticed that the sojourn time distributions, $F_{ij}(u; a=1)$ and $F_{ij}(u; a=2)$, are now degenerate distributions. It is common in machine maintenance that an action takes a fixed time. For example, backwashing an RGF is controlled by a computer and hence takes a fixed time.

V. Numerical Study

Here we give two numerical examples to illustrate the decision making process, and to show the feasibility and effectiveness of the proposed value-iteration algorithm. The following two examples are based on the industrial problem discussed in Section IV.

A. *Infinite Planning Horizon*

The parameter configuration for the infinite-planning-horizon POSMDP problem is specified as follows. Consider an RGF with four states: $\mathcal{S} = \{$1="good", 2="acceptable", 3="poor", 4="awful"$\}$. The action set is defined as $\mathcal{A} = \{$1="do nothing", 2="backwash and watch", 3="dose chemicals", 4="replace"$\}$. The ratio of outgoing to incoming water turbidity provides partial information on the state of the RGF; hence the observation space is the $[0,1]$ interval. Let $W(c,r)$ denote the Weibull distribution with the density function given by

$$f_W(u; c, r) = r\frac{u^{r-1}}{c^r}\exp(-(\frac{u}{c})^r), \qquad u > 0.$$

We assume that $\{T_1^1, T_2^1, T_3^1\}$ all follow the Weibull distribution with $c = 60$ and $r = 3$; $\{T_1^2, T_2^2, T_3^2\}$ all follow the Weibull distribution with $c = 65$ and $r = 3$. For action 1, the maximal value of the probability $\Pr(T_1^1 < u_1^*, T_1^1 + T_2^1 \geq u_1^*)$ is 0.7413; the corresponding optimal duration is $u_1^* = 78.7433$. For action 2, the maximal value of the probability $\Pr(T_2^2 < u_2^*, T_2^2 + T_3^2 \geq u_2^*)$ is also 0.7413; the corresponding optimal duration is $u_2^* = 85.3052$. Then the probability transition matrices are

$$P(a=1) = P(a=2) = \begin{bmatrix} 0.1043 & 0.7413 & 0.1493 & 0.0051 \\ 0 & 0.1043 & 0.7413 & 0.1544 \\ 0 & 0 & 0.1043 & 0.8957 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The duration of action 3 is fixed: $u_3^* = 3$; the state transition matrix is

$$P(a=3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.25 & 0.7 & 0.05 & 0 \\ 0.2 & 0.55 & 0.2 & 0.05 \end{bmatrix}.$$

For action 4, we have $p_{ij}(a=4) = I(j=1)$, $1 \leq i, j \leq 4$; the duration of replacement follows the truncated Gaussian distribution with mean $\mu = 10$, standard deviation $\sigma = 1.5$ and $u_4^* > 0$. Recall that we collect new observations whenever we think the

RGF's state changes. Hence, we collect the turbidity information when each maintenance action is completed. The observation function depends only on the RGF's state, not on the action. Let $f_B(o;\varepsilon,\beta)$ denote the density function of the beta distribution:

$$f_B(o;\varepsilon,\tau)=\frac{\Gamma(\varepsilon+\tau)}{\Gamma(\varepsilon)\Gamma(\tau)}o^{\varepsilon-1}(1-o)^{\tau-1},\qquad 0<o<1,$$

where $\Gamma(\cdot)$ is the gamma function. We have $g_i(o;a)=f_B(o;\varepsilon_i,\tau_i)$ in which $(\varepsilon_1,\cdots,\varepsilon_4)$=$(2,6,18,18)$ and $(\tau_1,\cdots,\tau_4)$=$(18,18,18,6)$. Let R_1 (resp. R_2) denote the reward matrix with $R_1(i,a)$ (resp. $R_2(i,a)$) representing the immediate reward (resp. reward rate) for taking action a when the RGF's state is i. We have

$$R_1=\begin{bmatrix} 0 & -100 & -200 & -500\\ 0 & -100 & -200 & -500\\ 0 & -100 & -200 & -500\\ 0 & -100 & -200 & -500 \end{bmatrix},\ \text{and}\ R_2=\begin{bmatrix} 500 & 500 & -100 & -100\\ 250 & 250 & -100 & -100\\ -300 & -300 & -100 & -100\\ -500 & -500 & -100 & -100 \end{bmatrix}.$$

The value of the discount factor, θ, is 0.01.

We first use Algorithm 1 to randomly simulate 1000 belief points. Then we employ Algorithm 2 to approximate the true value function $V(\cdot)$; we stop Algorithm 2 when two consecutive value functions are close enough: $||V_z-V_{z+1}||<0.01$. Expand the belief-point set B by Algorithm 3, and employ Algorithm 2 on the expanded belief-point set until $||V_z-V_{z+1}||<0.01$. Repeat such procedure until the cardinality of the belief-point set reaches to 5000. The initial α-vector set for starting Algorithm 2 is $\{(-10^6,-10^6,-10^6,-10^6)\}$.

When the cardinality of B is 5000, we plot the evolving trace of $||V_z-V_{z+1}||$ in Figure 3. Figure 3 shows that, even when the size of belief points is large, the iteration process converges quickly: after 35 iterations, the distance $||V_z-V_{z+1}||$ reduces to 0.007. The time consumed for the whole procedure is 60844 seconds – around 16 hours. All computations were coded in R (version 3.2.2) on a PC with Intel Core i5-4590 CPU @3.30 GHz. After each iteration, we calculate the maximal ECD reward for each of the 5000 belief points. Then we record the minimum and maximum values of these 5000 maximal ECD rewards. To check the evolution of the decision process, we plot the 35 minimum ECD rewards and the 35 maximum ECD rewards in Figure 4. In Figure 4, after 15 iterations, the minimum ECD rewards and the maximum ECD rewards shape into two horizontal lines, implying that the value function has been stable.

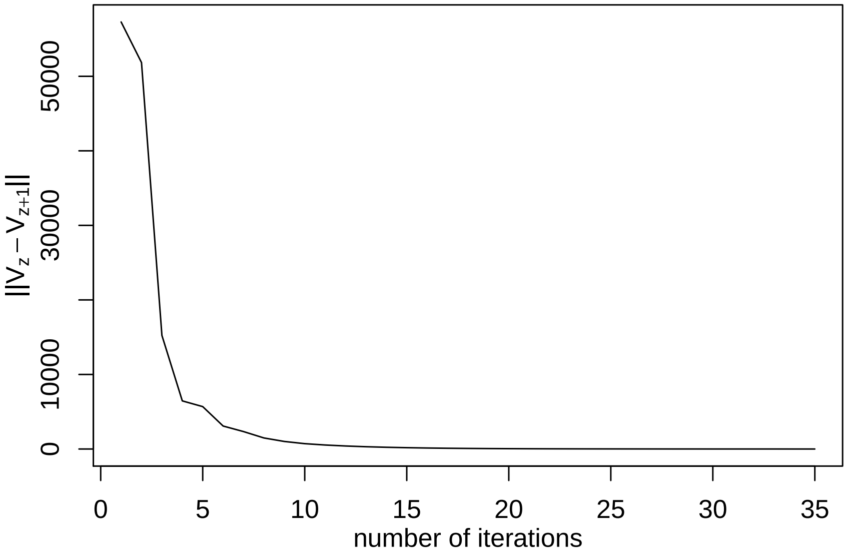

Fig. 3: The evolving trace of $||V_z-V_{z+1}||$ when the sample size of belief points is 5000.

Now the final value function should be very close to the underlying true value function. Therefore, we will use this final value function for making maintenance decisions at an arbitrary time point. For example, if the belief state is $(1,0,0,0)$, then the optimal action is "do nothing"; the corresponding ECD reward is 46357.85. If the belief state is $(0,0,0,1)$, then the optimal action is "replace"; the corresponding ECD reward is 40385.84. Table I lists some belief points from the set B (rounded to four decimal places) and the corresponding optimal actions and ECD rewards. Of course, these decision rules will change under different parameter settings.

B. Finite Planning Horizon

When w is finite, we re-encode the planning horizon with the time unit being one day. We set w to be 1095, corresponding to 3 years. As explained in Appendix C-B, we set the durations of all the actions to be integer-valued. Specifically, we set

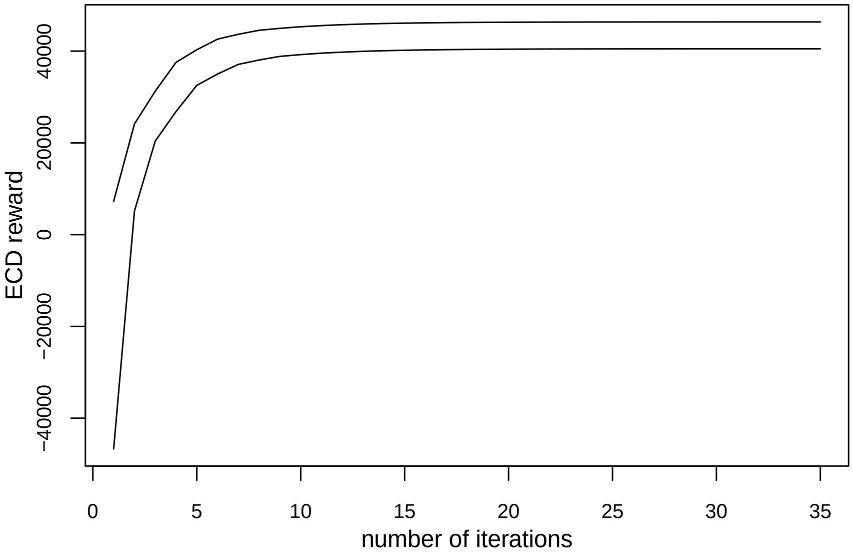

Fig. 4: The minimum and maximum ECD rewards of the 5000 belief points after each iteration.

TABLE I: A sample of belief points, their corresponding optimal actions and ECD rewards.

belief state	optimal action	corresponding ECD reward
(0.9972, 0.0028, 0.0000, 0.0000)	1	46316.40
(0.9965, 0.0035, 0.0000, 0.0000)	1	46306.04
(0.8714, 0.1286, 0.0000, 0.0000)	2	44454.43
(0.8160, 0.1840, 0.0000, 0.0000)	2	43634.51
(0.0031, 0.6803, 0.3165, 0.0001)	3	41215.31
(0.0001, 0.0390, 0.9457, 0.0152)	3	40574.81
(0.0000, 0.0003, 0.8488, 0.1509)	4	40498.43
(0.0000, 0.0000, 0.0000, 1.0000)	4	40385.84

$u_1^* = 78$, $u_2^* = 85$ and $u_3^* = 3$. The transition matrices of actions 1 and 2 hence change slightly:

$$P(a=1) = \begin{bmatrix} 0.1111 & 0.7411 & 0.1430 & 0.0048 \\ 0 & 0.1111 & 0.7411 & 0.1478 \\ 0 & 0 & 0.1111 & 0.8889 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$P(a=2) = \begin{bmatrix} 0.1068 & 0.7413 & 0.1469 & 0.0050 \\ 0 & 0.1068 & 0.7413 & 0.1519 \\ 0 & 0 & 0.1068 & 0.8932 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

u_4^* is a discrete random variable, taking values from $\{8,9,10,11,12\}$ respectively with probabilities $\{0.1,0.23,0.34,0.23,0.1\}$. All the other parameter values are identical with what are given in the above section.

When w is finite, the size of the belief-point set depends on the value of w. In our example, it is not recommended to simulate 5000 belief points as what we did for infinite w. If the belief-point set B contains 5000 elements, the whole procedure will take about 700 hours. We now randomly generate 1000 belief points by using Algorithm 1. We then in turn calculate $V_{1095}(\cdot), V_{1094}(\cdot), V_{1093}(\cdot), \ldots$, until we arrive at $V_0(\cdot)$. Apparently, the computational load is heavier than that when w is infinite: the computational time consumed is 166178 seconds – around 46 hours. At time 1095, the α-vector set contains only one element: $\{(0,0,0,0)\}$, and the optimal action for every belief point is “do nothing”. Similarly, since the shortest duration of all the actions is 3 days, the optimal actions for time points 1094, 1093 and 1092 are all “do nothing”, which is in tune with practice. Starting from time 1091 (backward), other maintenance actions are likely to be optimal for certain belief points. For example, under the above parameter setting, at time 1091 if the belief point is $(0,0.0272,0.9693,0.0035)$, then the optimal action is “dose chemicals”, and the corresponding ECD reward is 70.5906.

The iterative procedure stops when z goes back to 0, and then we approximate the underlying true value function by the set

$\{\alpha_0^k\}_k$. Each α vector in $\{\alpha_0^k\}_k$ is associated with an optimal action. Now at time 0, if the decision maker's belief point is $\boldsymbol{b}$, we chose from $\{\alpha_0^k\}_k$ the one that maximizes $< \alpha_0^k,\ \boldsymbol{b} >$: $\alpha_0^* = \arg \max_{\alpha_0^k \in \{\alpha_0^k\}_k} < \alpha_0^k,\ \boldsymbol{b} >$. Then the optimal action to take at time 0 is the action associated with α_0^*, and the ECD reward by taking this action is approximately $< \alpha_0^*,\ \boldsymbol{b} >$. Likewise, if the parameter setting does not change, then at any decision-making time point z ($1 \leq z \leq 1095$), we all use $\{\alpha_z^k\}_k$ to approximate the underlying true value function. With z approaching to 1095 (and hence the planning horizon w is not large), we can refine the α vector set by expanding the belief-point set B.

To check the evolution of the approximating value function, at each time point, we calculate the maximal ECD reward for each of the 1000 belief points. Then we record the minimum and maximum values of these 1000 maximal ECD rewards. We plot the 1096 minimum ECD rewards and the 1096 maximum ECD rewards in Figure 5. It is observed that the minimum ECD reward and maximum ECD reward increase rapidly at first (for example, during the time period [800, 1095]). If both the minimum ECD reward and the maximum ECD reward do not increase, then we can infer that the approximating value function has become stable. Yet, in Figure 5, from time 400 backward to time 0, the minimum ECD reward and maximum ECD reward are still increasing. To see this, we plot the minimum ECD rewards and the maximum ECD rewards over the time period [0, 400] in Figure 6. Figures 5 and 6 demonstrate that with z evolving from 1095 backward to 0, the elasticity of the minimum ECD reward (as well as the elasticity of the maximum ECD reward) with respect to z decreases.

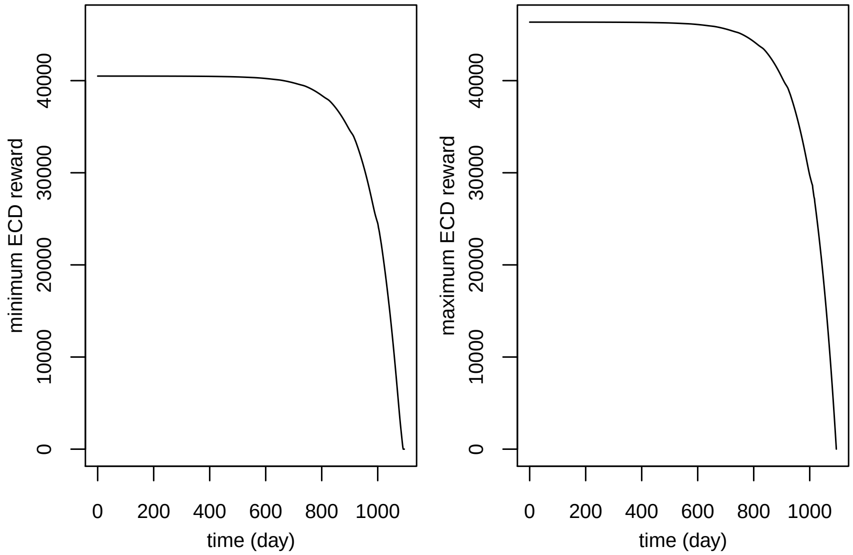

Fig. 5: The minimum and maximum ECD rewards of the 1000 belief points for the time period [0, 1095].

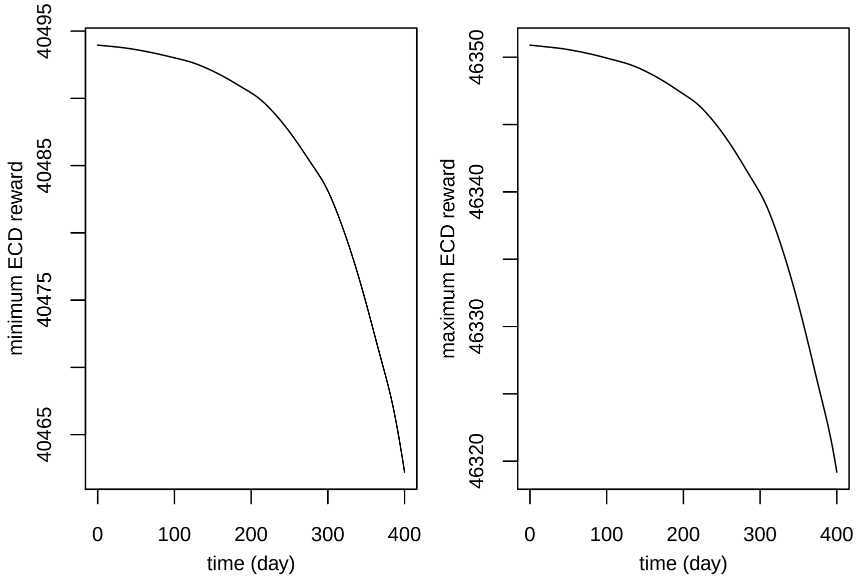

Fig. 6: The minimum and maximum ECD rewards of the 1000 belief points for the time period [0, 400] when $w = 1095$.

We now increase w to be 1825, corresponding to five years. The value functions for the time period [730, 1825] will be identical with what we have obtained. Hence we start from time 729 backward to time 0. The iterative procedure stops when z goes back to 0, and then we approximate the underlying true value function by the set $\{\alpha_0^k\}_k$. The computational time consumed is 114425 seconds – around 31 hours. Likewise, we plot the minimum and maximum ECD rewards of the 1000 belief points for the time period [0, 729] in Figure 7. Figure 7 shows that over the time period [0, 729], both the minimum ECD reward and the maximum ECD reward almost do not change. We can claim that the approximating value function has become stable. Therefore, if we want to study a planning horizon of, say, ten years, we might just use $\{\alpha_0^k\}_k$ to approximate all the true value functions for the first five years. There is no need to calculate the α vector set from time 3650 backward to time 0.

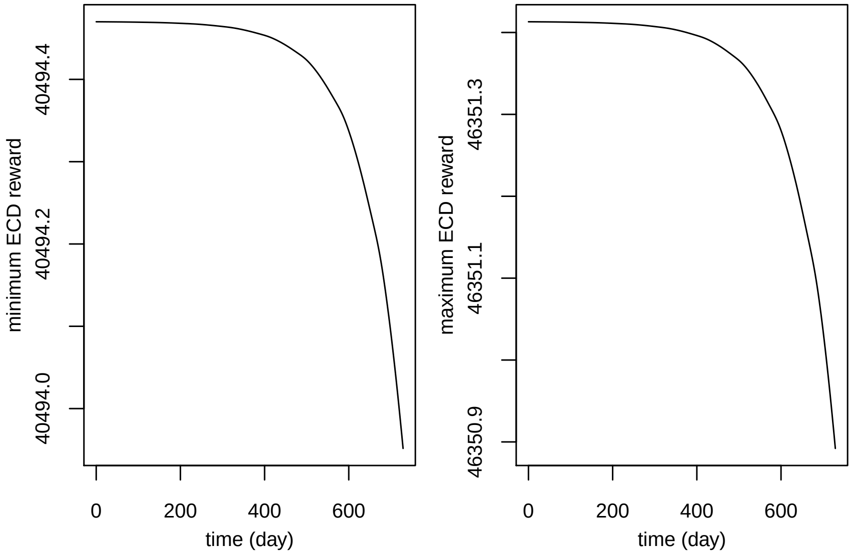

Fig. 7: The minimum and maximum ECD rewards of the 1000 belief points for the time period [0, 729] when $w = 1825$.

VI. Conclusions

In this paper, we studied the continuous-observation POSMDP which is a natural tool for the maintenance optimization problem of discrete-state systems. We reasoned, via Properties 1 and 2, why the point-based value iteration algorithm is an efficient method for approximating the true value function. We also addressed several practical issues on incorporating different maintenance actions into the POSMDP model. For practical implementation, we studied both finite planning horizon and infinite planning horizon. While the general framework introduced may face computational challenges, we have investigated cases of practical interest in which it remains quite tractable. The numerical study showed that, when the planning horizon is infinite, the value iteration procedure converges quickly; when the planning horizon is finite, even if we consider a long planning horizon, the computational load is still acceptable. Hence, we say that the continuous-observation POSMDP model is of application value in the area of machine maintenance. The work developed in this paper can also be applied to problems in the area of reinforcement learning, where Markov decision process is a topic of great interest.

This work can be enriched in several ways. We think two challenging avenues of research are particularly promising.

- In this work, we only study one system. Yet, in some cases, maintenance decision making may concern several interactive systems. Take the water utility for example. A water treatment works typically has several RGFs. Though these RGFs are independent of each other, if one of the RGFs fails, the other RGFs have to bear the additional working load. Consequently, the deteriorating behavior of these working RGFs will change. One solution to this problem is to add another action, called “add workload”. The time to take this action depends on the time at which an RGF fails.
- Another common action in machine maintenance is inspection, which will reveal the true state of the maintained machine. Recall that one distinctive feature of POSMDP is the partial observability. Yet, via an inspection, the decision process changes from POSMDP to SMDP, and, whenever the action is not inspection, the decision process changes back to POSMDP. Incorporating inspection into POSMDP is an open problem of interest.

Appendix A
Proof of the Continuity of the Set $\{\alpha_z^k\}_k$

We now prove that $\{\alpha_z^k\}_k$ is a continuous set by utilizing an idea proposed by [32]. If $\{\alpha_{z+1}^k\}_k$ is a finite set, then the belief space Δ can be divided into a finite set of convex regions separated by linear hyperplanes such that $V_{z+1}(\boldsymbol{b}) =< \alpha_{z+1}^k, \boldsymbol{b} >$

within a region for a single index k. We let $\alpha_{z+1}^k(\Delta)$ denote the region corresponding to the vector α_{z+1}^k; that is, the maximizing vector for all the belief points within $\alpha_{z+1}^k(\Delta)$ is α_{z+1}^k. With $\ddot{\boldsymbol{b}}_z$, a and u being fixed, all observations that lead to belief states $\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o)$ falling within $\alpha_{z+1}^k(\Delta)$ can be aggregated into one meta observation $O_k(\ddot{\boldsymbol{b}}_z,a,u)$:

$$O_k(\ddot{\boldsymbol{b}}_z,a,u) = \{o \in \mathcal{O} | \alpha_{z+1}^k = \arg\max_{\{\alpha_{z+1}^k\}_k} < \alpha_{z+1}^k,\ \boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o) >\}.$$

Then we have

$$\begin{aligned}
&\int\limits_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o | \ddot{\boldsymbol{b}}_z, \ddot{A}_z = a) \exp(-\theta u) V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o)) do \\
&= \exp(-\theta u) \sum_k \int_{O_k(\ddot{\boldsymbol{b}}_z,a,u)} \Pr(U_z = u, \ddot{O}_{z+1} = o | \ddot{\boldsymbol{b}}_z, \ddot{A}_z = a) < \alpha_{z+1}^k,\ \boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o) > do \\
&= \exp(-\theta u) \sum_k \int_{O_k(\ddot{\boldsymbol{b}}_z,a,u)} \sum_{i=1}^n \alpha_{z+1}^k(i) \sum_{j=1}^n \ddot{b}_z^j f_{ji}(u;a) g_i(o;a) p_{ji}(a) do \\
&= \exp(-\theta u) \sum_{j=1}^n \left[\sum_k \sum_{i=1}^n \alpha_{z+1}^k(i) f_{ji}(u;a) g_i(O_k(\ddot{\boldsymbol{b}}_z,a,u);a) p_{ji}(a) \right] \ddot{b}_z^j,
\end{aligned}$$

where

$$g_i(O_k(\ddot{\boldsymbol{b}}_z,a,u);a) = \int_{O_k(\ddot{\boldsymbol{b}}_z,a,u)} g_i(o;a) do.$$

The value function $V_z(\ddot{\boldsymbol{b}}_z)$ can be written as

$$V_z(\ddot{\boldsymbol{b}}_z) = \max_{a \in \mathcal{A}} \left\{ \sum_{j=1}^n \ddot{b}_z^j \left[\bar{R}_z^a(j) + \int\limits_0^{w-t_z} \exp(-\theta u) \sum_k \sum_{i=1}^n \alpha_{z+1}^k(i) f_{ji}(u;a) g_i(O_k(\ddot{\boldsymbol{b}}_z,a,u);a) p_{ji}(a) du \right] \right\}. \qquad (12)$$

To prove that $\{\alpha_z^k\}_k$ is a continuous set, we only need to prove that the bracketed quantity changes continuously with $\ddot{\boldsymbol{b}}_z$. Recall that, the region $O_k(\ddot{\boldsymbol{b}}_z,a,u)$ is defined as follows:

$$O_k(\ddot{\boldsymbol{b}}_z,a,u) = \{o \in \mathcal{O} | \boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o) \in \alpha_{z+1}^k(\Delta)\}.$$

From Equation (2) we know that, with a and u being fixed, $\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o)$ is a continuous function of both $\ddot{\boldsymbol{b}}_z$ and o. Therefore, the boundary of the region $O_k(\ddot{\boldsymbol{b}}_z,a,u)$ changes continuously with $\ddot{\boldsymbol{b}}_z$. Then it follows that $g_i(O_k(\ddot{\boldsymbol{b}}_z,a,u);a)$ (and, consequently, the bracketed quantity) is a continuous function of $\ddot{\boldsymbol{b}}_z$.

Appendix B
Proof of Proposition 2

The Bellman backup operator H can be re-written as $HV_{z+1}(\ddot{\boldsymbol{b}}_z) = \max_{a \in \mathcal{A}} H^a V_{z+1}(\ddot{\boldsymbol{b}}_z)$ with

$$H^a V_{z+1}(\ddot{\boldsymbol{b}}_z) = < \bar{R}_z^a, \ddot{\boldsymbol{b}}_z > + \int\limits_0^{+\infty} \int\limits_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o | \ddot{\boldsymbol{b}}_z, \ddot{A}_z = a) \exp(-\theta u) V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o)) dodu$$

Assume that $|V_z - U_z|$ is maximized at point $\boldsymbol{b}$:

$$||V_z - U_z|| = |V_z(\boldsymbol{b}) - U_z(\boldsymbol{b})| = |HV_{z+1}(\boldsymbol{b}) - HU_{z+1}(\boldsymbol{b})|.$$

Denote as $\hat{a}$ the optimal action for HV_{z+1} at $\boldsymbol{b}$, and $\check{a}$ the optimal action for HU_{z+1} at $\boldsymbol{b}$. Assuming $HV_{z+1}(\boldsymbol{b}) \geq HU_{z+1}(\boldsymbol{b})$, then it holds that

$$|HV_{z+1}(\boldsymbol{b}) - HU_{z+1}(\boldsymbol{b})| = H^{\hat{a}} V_{z+1}(\boldsymbol{b}) - H^{\check{a}} U_{z+1}(\boldsymbol{b}).$$

Since $H^{\hat{a}} U_{z+1}(\boldsymbol{b}) \leq H^{\check{a}} U_{z+1}(\boldsymbol{b})$, we have

$$||V_z - U_z|| = H^{\hat{a}} V_{z+1}(\boldsymbol{b}) - H^{\check{a}} U_{z+1}(\boldsymbol{b}) \leq H^{\hat{a}} V_{z+1}(\boldsymbol{b}) - H^{\hat{a}} U_{z+1}(\boldsymbol{b}),$$

in which

$$\begin{aligned}
&H^{\hat{a}}V_{z+1}(\boldsymbol{b}) - H^{\hat{a}}U_{z+1}(\boldsymbol{b}) \\
&= \int_{0}^{+\infty}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z = \boldsymbol{b}, \ddot{A}_z = \hat{a})\exp(-\theta u)[V_{z+1}(\boldsymbol{l}_{\hat{a}}(\boldsymbol{b},u,o)) - U_{z+1}(\boldsymbol{l}_{\hat{a}}(\boldsymbol{b},u,o))]dodu \\
&\leq \int_{0}^{+\infty}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z = \boldsymbol{b}, \ddot{A}_z = \hat{a})\exp(-\theta u)||V_{z+1} - U_{z+1}||dodu \\
&= \left[\int_{0}^{+\infty} \Pr(U_z = u|\ddot{\boldsymbol{b}}_z = \boldsymbol{b}, \ddot{A}_z = \hat{a})\exp(-\theta u)du\right]||V_{z+1} - U_{z+1}||.
\end{aligned}$$

Since the bracketed quantity is strictly smaller than one, we have $||V_z - U_z|| < ||V_{z+1} - U_{z+1}||$.

Now assume that $\boldsymbol{b}$ is an arbitrary point. $\hat{a}$ is the optimal action for HV_{z+1} at $\boldsymbol{b}$, and $\check{a}$ is the optimal action for HU_{z+1} at $\boldsymbol{b}$. If $V_{z+1} \geq U_{z+1}$, then, $\forall$ u and o, $V_{z+1}(\boldsymbol{l}_{\check{a}}(\boldsymbol{b},u,o)) \geq U_{z+1}(\boldsymbol{l}_{\check{a}}(\boldsymbol{b},u,o))$. By taking integration we have

$$\begin{aligned}
&\int_{0}^{+\infty}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z = \boldsymbol{b}, \ddot{A}_z = \check{a})\exp(-\theta u)V_{z+1}(\boldsymbol{l}_{\check{a}}(\boldsymbol{b},u,o))dodu \\
\geq &\int_{0}^{+\infty}\int_{\mathcal{O}} \Pr(U_z = u, \ddot{O}_{z+1} = o|\ddot{\boldsymbol{b}}_z = \boldsymbol{b}, \ddot{A}_z = \check{a})\exp(-\theta u)U_{z+1}(\boldsymbol{l}_{\check{a}}(\boldsymbol{b},u,o))dodu.
\end{aligned}$$

Consequently,

$$V_z(\boldsymbol{b}) = HV_{z+1}(\boldsymbol{b}) = H^{\hat{a}}V_{z+1}(\boldsymbol{b}) \geq H^{\check{a}}V_{z+1}(\boldsymbol{b}) \geq H^{\check{a}}U_{z+1}(\boldsymbol{b}) = HU_{z+1}(\boldsymbol{b}) = U_z(\boldsymbol{b}).$$

Since $\boldsymbol{b}$ is arbitrary, we have $V_z \geq U_z$.

Appendix C
Belief Point Backup

By exemplifying the maintenance of an RGF, we give below the details for backuping a belief point. Since the sojourn time distribution in our example does not depend on the current state or the following state, we might write $f(u;a)$ for $f_{ji}(u;a)$.

A. Infinite Planning Horizon

If $w = +\infty$, the double integral in Equation (9) can be simplified:

$$\begin{aligned}
&\int_{0}^{+\infty}\int_{0}^{1} \exp(-\theta u) \max_{\{\alpha_{z+1}^{k}\}_k} \left\{\sum_{j=1}^{n}\left[\sum_{i=1}^{n}\alpha_{z+1}^{k}(i)f_{ji}(u;a)g_i(o;a)p_{ji}(a)\right]\ddot{b}_z^j\right\}dodu \\
&= \int_{0}^{+\infty} \exp(-\theta u)f(u;a)du \int_{0}^{1} \max_{\{\alpha_{z+1}^{k}\}_k} \{\sum_{j=1}^{n}[\sum_{i=1}^{n}\alpha_{z+1}^{k}(i)g_i(o;a)p_{ji}(a)]\ddot{b}_z^j\}do.
\end{aligned}$$

Here, we have replaced the supremum with the maximum, since the α-vector set in Algorithm 3 is finite. The first integral w.r.t. u can be readily calculated; the second integral can be re-written into

$$\int_{0}^{1} \max_{\{\alpha_{z+1}^{k}\}_k} < \delta_k^a(o), \ddot{\boldsymbol{b}}_z > do,$$

where $\delta_k^a(o) = (\delta_{k1}^a(o), \cdots, \delta_{kn}^a(o))$ and

$$\delta_{kj}^a(o) = \sum_{i=1}^{n}\alpha_{z+1}^{k}(i)g_i(o;a)p_{ji}(a), \quad \text{for} \quad 1 \leq j \leq n.$$

Define

$$O_k(\ddot{\boldsymbol{b}}_z, a) = \{o \in [0,1]|\alpha_{z+1}^{k} = \arg\max_{\{\alpha_{z+1}^{k}\}_k} < \delta_k^a(o), \ddot{\boldsymbol{b}}_z >\}.$$

Then the second integral can be further simplified:

$$\begin{aligned}
\int_{0}^{1} \max_{\{\alpha_{z+1}^{k}\}_k} \left\{\sum_{j=1}^{n}\left[\sum_{i=1}^{n}\alpha_{z+1}^{k}(i)g_i(o;a)p_{ji}(a)\right]\ddot{b}_z^j\right\}do &= \sum_{k}\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{z+1}^{k}(i)g_i(O_k(\ddot{\boldsymbol{b}}_z,a);a)p_{ji}(a)\ddot{b}_z^j \\
&= \sum_{j=1}^{n}\sum_{k}\sum_{i=1}^{n}\alpha_{z+1}^{k}(i)g_i(O_k(\ddot{\boldsymbol{b}}_z,a);a)p_{ji}(a)\ddot{b}_z^j \\
&=< \delta(\ddot{\boldsymbol{b}}_z, a), \ddot{\boldsymbol{b}}_z >,
\end{aligned}$$

where $\delta(\ddot{\boldsymbol{b}}_z,a)=(\delta_1(\ddot{\boldsymbol{b}}_z,a),\cdots,\delta_n(\ddot{\boldsymbol{b}}_z,a))$ and

$$\delta_j(\ddot{\boldsymbol{b}}_z,a)=\sum_k\sum_{i=1}^n\alpha^k_{z+1}(i)g_i(O_k(\ddot{\boldsymbol{b}}_z,a);a)p_{ji}(a),\quad\text{for}\quad 1\le j\le n.$$

Consequently, we have

$$V_z(\ddot{\boldsymbol{b}}_z)=\max_{a\in\mathcal{A}}<\bar{R}^a_z+\delta(\ddot{\boldsymbol{b}}_z,a)\int_0^{+\infty}\exp(-\theta u)f(u;a)du,\ \ddot{\boldsymbol{b}}_z>,\tag{13}$$

in which $\bar{R}^a_z(j)$, $1\le j\le n$, also can be simplified:

$$\begin{aligned}\bar{R}^a_z(j)&=R_1(j,a)+\frac{1}{\theta}\left\{1-E\left[\exp(-\theta U_z)\middle|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a\right]\right\}R_2(j,a)\\&=R_1(j,a)+\frac{1}{\theta}\left\{1-\int_0^{+\infty}\exp(-\theta u)f(u;\ a)du\right\}R_2(j,a).\end{aligned}$$

To backup the belief point $\ddot{\boldsymbol{b}}_z$, the most challenging thing is the calculation of $\delta(\ddot{\boldsymbol{b}}_z,a)$. A traditional approach to backup $\ddot{\boldsymbol{b}}_z$ is to identify, for every vector α^k_{z+1}, the mega observation $O_k(\ddot{\boldsymbol{b}}_z,a)$. Note that $O_k(\ddot{\boldsymbol{b}}_z,a)$ may be composed of several separated intervals. In the current work, instead of identifying the mega observation for each α vector, we indeed determine the optimal α vector for each observation, which is accomplished by discretizing the [0, 1] interval. Specifically, for a large enough positive integer r, define $o_v=\frac{2v-1}{2r}$ for $1\le v\le r$. Let α_v denote the maximizing α vector for observation o_v:

$$\alpha_v=\arg\max_{\{\alpha^k_{z+1}\}_k}\left\{\sum_{j=1}^n\left[\sum_{i=1}^n\alpha^k_{z+1}(i)g_i(o_v;a)p_{ji}(a)\right]\ddot{b}^j_z\right\}.$$

Then, the second integral can be numerically approximated:

$$\begin{aligned}\int_0^1\max_{\{\alpha^k_{z+1}\}_k}\left\{\sum_{j=1}^n\left[\sum_{i=1}^n\alpha^k_{z+1}(i)g_i(o;a)p_{ji}(a)\right]\ddot{b}^j_z\right\}do&=\frac{1}{r}\sum_{v=1}^r\max_{\{\alpha^k_{z+1}\}_k}\left\{\sum_{j=1}^n\left[\sum_{i=1}^n\alpha^k_{z+1}(i)g_i(o_v;a)p_{ji}(a)\right]\ddot{b}^j_z\right\}\\&=\frac{1}{r}\sum_{v=1}^r\sum_{j=1}^n\left[\sum_{i=1}^n\alpha_v(i)g_i(o_v;a)p_{ji}(a)\right]\ddot{b}^j_z\\&=\sum_{j=1}^n\left[\frac{1}{r}\sum_{v=1}^r\sum_{i=1}^n\alpha_v(i)g_i(o_v;a)p_{ji}(a)\right]\ddot{b}^j_z.\end{aligned}$$

Consequently, we have

$$\delta_j(\ddot{\boldsymbol{b}}_z,a)=\frac{1}{r}\sum_{v=1}^r\sum_{i=1}^n\alpha_v(i)g_i(o_v;a)p_{ji}(a),\quad\text{for}\quad 1\le j\le n.$$

B. Finite Planning Horizon

We re-write Equation (5) here:

$$V_z(\ddot{\boldsymbol{b}}_z)=\max_{a\in\mathcal{A}}\left\{\sum_{i=1}^n\bar{R}^a_z(i)\ddot{b}^i_z+\int\limits_0^{w-t_z}\int\limits_{\mathcal{O}}\Pr(U_z=u,\ddot{O}_{z+1}=o|\ddot{\boldsymbol{b}}_z,\ddot{A}_z=a)\exp(-\theta u)V_{z+1}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_z,u,o))dodu\right\}.$$

Note that, if $w=+\infty$, then theoretically all the value functions $V_z(\cdot)$ $(z=0,1,2,\cdots)$ are identical, and Equation (5) is indeed Equation (8). However, if w is finite, the value function $V_{z+1}(\cdot)$ changes with t_{z+1} $(=t_z+u_z)$. Therefore, if w is finite, the U_z's must be discrete random variables. Otherwise, the calculation of $V_z(\cdot)$ will demand infinitely many different $V_{z+1}(\cdot)$'s. In machine maintenance, we can let the planning horizon be encoded by a relatively small time unit (e.g., one hour); then it is quite practical to assume that the U_z's are integer-valued.

We let integer m indicate time: $\{m=0,1,2,\cdots,w\}$; the belief state and value function at time m are indexed by m: $\ddot{\boldsymbol{b}}_m$ and $V_m(\cdot)$. Equation (5) can be re-written into

$$\begin{aligned}&V_m(\ddot{\boldsymbol{b}}_m)\\&=\max_{a\in\mathcal{A}}\left\{\sum_{j=1}^n\bar{R}^a_m(j)\ddot{b}^j_m+\sum_{u=1}^{w-m}\int\limits_{\mathcal{O}}\Pr(U_m=u,\ddot{O}_{m+u}=o|\ddot{\boldsymbol{b}}_m,\ddot{A}_m=a)\exp(-\theta u)V_{m+u}(\boldsymbol{l}_a(\ddot{\boldsymbol{b}}_m,u,o))do\right\}.\end{aligned}\tag{14}$$

Let $\{\alpha_m^k\}_k$ denote the set of α vectors for value function $V_m(\cdot)$, $m = 0,1,2,\cdots,w$. It is easy to prove that

$$\begin{aligned} V_m(\ddot{\boldsymbol{b}}_m) &= \max_{a\in\mathcal{A}}\{\sum_{j=1}^{n}\bar{R}_m^a(j)\ddot{b}_m^j \\ &+ \sum_{u=1}^{w-m}\Pr(U_m = u|\ddot{A}_m = a)\exp(-\theta u)\sum_k\sum_{i=1}^{n}\alpha_{m+u}^k(i)\sum_{j=1}^{n}\ddot{b}_m^j g_i(O_k(\ddot{\boldsymbol{b}}_m,a,u);a)p_{ji}(a)\} \\ &= \max_{a\in\mathcal{A}} < \delta(\ddot{\boldsymbol{b}}_m,a),\ddot{\boldsymbol{b}}_m >, \end{aligned}$$

where $\delta(\ddot{\boldsymbol{b}}_m,a) = (\delta_1(\ddot{\boldsymbol{b}}_m,a),\cdots,\delta_n(\ddot{\boldsymbol{b}}_m,a))$ and

$$\delta_j(\ddot{\boldsymbol{b}}_m,a) = \bar{R}_m^a(j) + \sum_{u=1}^{w-m}\Pr(U_m = u|\ddot{A}_m = a)\exp(-\theta u)\sum_k\sum_{i=1}^{n}\alpha_{m+u}^k(i)g_i(O_k(\ddot{\boldsymbol{b}}_m,a,u);a)p_{ji}(a).$$

The mega observation $O_k(\ddot{\boldsymbol{b}}_m,a,u)$ is defined as

$$O_k(\ddot{\boldsymbol{b}}_m,a,u) = \{o\in[0,1]|\alpha_{m+u}^k = \arg\max_{\{\alpha_{m+u}^k\}_k}\sum_{i=1}^{n}\alpha_{m+u}^k(i)\sum_{j=1}^{n}\ddot{b}_m^j g_i(o;a)p_{ji}(a)\},$$

and $\bar{R}_m^a(j)$ has expression

$$\bar{R}_m^a(j) = R_1(j,a) + \frac{1}{\theta}[1 - \sum_{u=1}^{w-m}\exp(-\theta u)\Pr(U_m = u|\ddot{A}_m = a)]R_2(j,a), \qquad 1\le j\le n.$$

Likewise, for a large enough positive integer r, define $o_v = \frac{2v-1}{2r}$ for $1\le v\le r$. Let α_{m+u}^v denote the maximizing α vector for observation o_v:

$$\alpha_{m+u}^v = \arg\max_{\{\alpha_{m+u}^k\}_k}\sum_{i=1}^{n}\alpha_{m+u}^k(i)\sum_{j=1}^{n}\ddot{b}_m^j g_i(o_v;a)p_{ji}(a).$$

Then we have

$$\begin{aligned} \int_{\mathcal{O}}\max_{\{\alpha_{m+u}^k\}_k}\sum_{i=1}^{n}\alpha_{m+u}^k(i)\sum_{j=1}^{n}\ddot{b}_m^j g_i(o;a)p_{ji}(a)do &= \frac{1}{r}\sum_{v=1}^{r}\max_{\{\alpha_{m+u}^k\}_k}\sum_{i=1}^{n}\alpha_{m+u}^k(i)\sum_{j=1}^{n}\ddot{b}_m^j g_i(o_v;a)p_{ji}(a) \\ &= \frac{1}{r}\sum_{v=1}^{r}\sum_{i=1}^{n}\alpha_{m+u}^v(i)\sum_{j=1}^{n}\ddot{b}_m^j g_i(o_v;a)p_{ji}(a), \end{aligned}$$

and finally

$$V_m(\ddot{\boldsymbol{b}}_m) = \max_{a\in\mathcal{A}}\sum_{j=1}^{n}\ddot{b}_m^j\left[\bar{R}_m^a(j) + \sum_{u=1}^{w-m}\Pr(U_m = u|\ddot{A}_m = a)\exp(-\theta u)\frac{1}{r}\sum_{v=1}^{r}\sum_{i=1}^{n}\alpha_{m+u}^v(i)g_i(o_v;a)p_{ji}(a)\right].$$

References

[1] M. S. Wulfsohn and A. A. Tsiatis, "A joint model for survival and longitudinal data measured with error," *Biometrics*, vol. 53, no. 1, pp. 330–339, 1997.

[2] Q. Zhou, J. Son, S. Zhou, X. Mao, and M. Salman, "Remaining useful life prediction of individual units subject to hard failure," *IIE Transactions*, vol. 46, no. 10, pp. 1017–1030, 2014.

[3] Q. Zhang, C. Hua, and G. Xu, "A mixture weibull proportional hazard model for mechanical system failure prediction utilising lifetime and monitoring data," *Mechanical Systems and Signal Processing*, vol. 43, no. 12, pp. 103 – 112, 2014.

[4] M. H. Ling, K. L. Tsui, and N. Balakrishnan, "Accelerated degradation analysis for the quality of a system based on the gamma process," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 463–472, 2015.

[5] M. Zhang, O. Gaudoin, and M. Xie, "Degradation-based maintenance decision using stochastic filtering for systems under imperfect maintenance," *European Journal of Operational Research*, vol. 245, no. 2, pp. 531 – 541, 2015.

[6] Y. Zhang and H. Liao, "Analysis of destructive degradation tests for a product with random degradation initiation time," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 516–527, 2015.

[7] X. Zhang and H. Gao, "Road maintenance optimization through a discrete-time semi-markov decision process," *Reliability Engineering & System Safety*, vol. 103, pp. 110 – 119, 2012.

[8] S. Kahrobaee and S. Asgarpoor, "A hybrid analytical-simulation approach for maintenance optimization of deteriorating equipment: Case study of wind turbines," *Electric Power Systems Research*, vol. 104, pp. 80 – 86, 2013.

[9] H. Rivera-Gomez, A. Gharbi, and J. Kenne, "Joint control of production, overhaul, and preventive maintenance for a production system subject to quality and reliability deteriorations," *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 9-12, pp. 2111–2130, 2013.

[10] J. A. Flory, J. P. Kharoufeh, and D. T. Abdul-Malak, "Optimal replacement of continuously degrading systems in partially observed environments," *Naval Research Logistics*, vol. 62, no. 5, pp. 395–415, 2015.

[11] E. Byon, L. Ntaimo, and Y. Ding, "Optimal maintenance strategies for wind turbine systems under stochastic weather conditions," *IEEE Transactions on Reliability*, vol. 59, no. 2, pp. 393–404, 2010.

[12] M. Chen, H. Fan, C. Hu, and D. Zhou, "Maintaining partially observed systems with imperfect observation and resource constraint," *IEEE Transactions on Reliability*, vol. 63, no. 4, pp. 881–890, 2014.

[13] R. Srinivasan and A. Parlikad, “Semi-markov decision process with partial information for maintenance decisions,” *IEEE Transactions on Reliability*, vol. 63, no. 4, pp. 891–898, 2014.

[14] K. Papakonstantinou and M. Shinozuka, “Planning structural inspection and maintenance policies via dynamic programming and markov processes. Part II: POMDP implementation,” *Reliability Engineering & System Safety*, vol. 130, pp. 214 – 224, 2014.

[15] C. C. White, “Procedures for the solution of a finite-horizon, partially observed, semi-markov optimization problem.” *Operations Research*, vol. 24, no. 2, pp. 348–358, 1976.

[16] Z. Lim, L. Sun, and D. J. Hsu, “Monte carlo value iteration with macro-actions,” in *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., 2011, pp. 1287–1295.

[17] N. A. Vien, H. Ngo, S. Lee, and T. Chung, “Approximate planning for bayesian hierarchical reinforcement learning,” *Applied Intelligence*, vol. 41, no. 3, pp. 808–819, 2014.

[18] S. Omidshafiei, A. A. Agha-Mohammadi, C. Amato, and J. How, “Decentralized control of partially observable markov decision processes using belief space macro-actions,” vol. 2015-June, 2015, pp. 5962–5969.

[19] D. Tang, V. Makis, L. Jafari, and J. Yu, “Optimal maintenance policy and residual life estimation for a slowly degrading system subject to condition monitoring,” *Reliability Engineering & System Safety*, vol. 134, pp. 198 – 207, 2015.

[20] D. Tang, J. Yu, X. Chen, and V. Makis, “An optimal condition-based maintenance policy for a degrading system subject to the competing risks of soft and hard failure,” *Computers & Industrial Engineering*, vol. 83, pp. 100 – 110, 2015.

[21] F. Naderkhani ZG and V. Makis, “Optimal condition-based maintenance policy for a partially observable system with two sampling intervals,” *The International Journal of Advanced Manufacturing Technology*, vol. 78, no. 5-8, pp. 795–805, 2015.

[22] Y. Zhou, L. Ma, J. Mathew, Y. Sun, and R. Wolff, “Maintenance strategy optimization using a continuous-state partially observable semi-markov decision process,” *Microelectronics Reliability*, vol. 51, no. 2, pp. 300 – 309, 2011.

[23] R. S. Sutton, D. Precup, and S. Singh, “Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning,” *Artificial Intelligence*, vol. 112, no. 12, pp. 181 – 211, 1999.

[24] E. J. Sondik, “The optimal control of partially observable markov processes over the infinite horizon: Discounted costs,” *Operations Research*, vol. 26, no. 2, pp. 282–304, 1978.

[25] R. D. Smallwood and E. J. Sondik, “The optimal control of partially observable markov processes over a finite horizon,” *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[26] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains,” *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[27] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[28] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[29] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for pomdps,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, August 2003, pp. 1025 – 1032.

[30] G. Shani, J. Pineau, and R. Kaplow, “A survey of point-based pomdp solvers,” *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, 2013.

[31] M. Spaan and N. Vlassis, “Perseus: Randomized point-based value iteration for pomdps,” *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[32] J. Hoey and P. Poupart, “Solving pomdps with continuous or large discrete observation spaces,” in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 1332–1338.