Reinforcement Learning with Gaussian Processes for Condition-Based Maintenance

Abstract

Condition-based maintenance strategies are effective in enhancing reliability and safety for complex engineering systems that exhibit degradation phenomena with uncertainty. Such sequential decision-making problems are often modeled as Markov decision processes (MDPs) when the underlying process has a Markov property. Recently, reinforcement learning (RL) becomes increasingly efficient to address MDP problems with large state spaces. In this paper, we model the condition-based maintenance problem as a discrete-time continuous-state MDP without discretizing the deterioration condition of the system. The Gaussian process regression is used as function approximation to model the state transition and the value functions of states in reinforcement learning. A RL algorithm is then developed to minimize the long-run average cost (instead of the commonly-used discounted reward) with iterations on the state-action value function and the state value function, respectively. We verify the capability of the proposed algorithm by simulation experiments and demonstrate its advantages in a case study on a battery maintenance decision-making problem. The proposed algorithm outperforms the discrete MDP approach by achieving lower long-run average costs.

Keywords: Condition-based maintenance; Reinforcement learning; Gaussian process regression; Markov decision process; Gaussian processes for reinforcement learning; Function approximation

1 Introduction

Condition-based maintenance (CBM) has been effectively implemented to increase system reliability, availability, and safety with reduced maintenance costs in many capital-intensive industries, e.g., energy, oil and gas, and aerospace. In engineering systems that need to be maintained preventively, degradation phenomena are often observed that possess a stochastic nature with uncertainty. Markov decision processes (MDPs) are commonly adopted to model such system dynamics and the related costs accrued in a probabilistic manner. Recently, reinforcement learning (RL) has been implemented as an effective approach for solving MDPs in maintenance problems [1–7]. However, most RL algorithms applied to maintenance decision making are restricted to discrete state and action spaces. In this research, we introduce the Gaussian process (GP) function approximation in RL algorithms to handle the continuous states for decision-making in CBM.

The early implementation of MDPs for preventive maintenance addresses a handful of states either discretized from continuous state indicators or obtained from experts' domain knowledge [8–12]. In these studies, the traditional approaches to solve the MDP models were used that can be summarized into two categories: dynamic programming (DP) and linear programming (LP) [13]. Despite the advantages of these traditional solutions to MDPs compared to exhaustive search, both DP and LP algorithms are too restrictive to be applied to many practical problems that do not commonly satisfy the assumptions: a discrete-time and finite state-action space, and the known system dynamics and reward structures. In addition, the traditional methods have a polynomial complexity in terms of the number of states and the number of actions [14]. The increase in the dimension of the state space makes traditional methods prohibitively slow, i.e., the well-known "curse of dimensionality" [15]. To overcome these limitations of original MDP models and their solutions, some extensions have been made to MDPs, such as semi-MDPs and partially observable MDPs. Both approaches have been used for maintenance decision-making [16, 17]. However, these extensions of MDPs cannot handle a large or continuous state space due to their rigorous format.

RL has recently been introduced to solve MDPs in maintenance planning problems [1–7]. Instead of assuming a strict MDP formulation, reinforcement learning treats the problem as an agent interacting with a random environment, while maximizing the rewards accrued along the process. Equipped with a set of modern approaches highlighted by techniques including temporal difference (TD) learning and function approximation, reinforcement learning is able to solve MDP problems where the underlying state-transition dynamics is unknown or the state and action spaces are extremely large. For multi-product inventory maintenance, Das et al. [1] introduced the semi-Markov average reward technique to solve the general semi-Markov process for maximizing the average reward. To prevent cascading failure and blackout in smart grids, Zarrabian et al. [2] utilized a tabular Q-learning algorithm with a Boltzmann exploration policy setting. To optimize control policies for the production and maintenance of deteriorating manufacturing systems, Xanthopoulos et al. [3] implemented a tabular state-action-reward-state-action (SARSA) algorithm with an average reward. Allen et al. implemented a Bayesian RL algorithm in a cyber preventive maintenance problem [4]. In these studies of using RL for addressing maintenance problems, the continuous degradation states are discretized to avoid the burden of a large or continuous state space, which inevitably leads to loss of accuracy in the analysis. In practical applications, however, the degradation phenomena are often described in continuous metrics, e.g., the capacity loss of rechargeable batteries [18], and the generator bearing temperature in wind turbines [19].

In this research, we explore the approach of function approximation in RL to handle the continuous states for CBM decision-making. Specifically, we propose to use the Gaussian process regression (GPR) as function approximation to model the state transition probability and the value function of states in our MDP. As a general nonparametric model, Gaussian process regression gains a reputation for its universality and good utilization of data, which is easy to implement as well [20]. Gaussian processes have been widely adopted for modeling stochastic processes in reliability and maintenance studies. To optimize preventive maintenance strategies for deteriorating assets, Gaussian processs are used as an emulator to approximate the Expected Value of Perfect Information indices for measuring

the effect of parameter uncertainty on the cost [21–23]. In wind turbine condition monitoring, Gaussian processes have been used for modeling power performance indicators to detect anomaly [24–26]. [27] used Gaussian process classifiers to the operational data to investigate fault diagnosis of wind turbines. To assess the remaining useful life of in-service components, multivariate Gaussian convolution processes are used based on condition monitoring signals [28].

The use of Gaussian process regression in RL dates back to 2003 when Engel et al. [29] used a GP to approximate state value functions that are incorporated into the TD learning (GPTD). Rasmussen & Kuss [30] then introduced an offline algorithm to maximize the discounted reward in a discrete time continuous state MDP, where the state transitions and the state value functions are both modeled by the GPR. Then the original GPTD algorithm in [29] was extended to Gaussian Process SARSA by Engel et al. [31], who imposed the GPR function approximation on the state-action function instead of the state value function. With the development of Gaussian process methods in RL, the theoretical demonstration of the sample efficiency emerged for some RL algorithms using GP as function approximation since 2014 [32]. The aforementioned RL algorithms equipped with GP focus on the discounted reward objective, which is extended to the average reward objective in this study.

We demonstrate the performance of the proposed method using a case study on the replacement threshold decision making of lithium-ion batteries with a continuous capacity loss. Battery management systems have been well developed to monitor and control the state-of-charge (SOC) and the state-of-health (SOH) during the working cycles in order to enhance the efficiency of rechargeable batteries [33, 34]. Although extensive studies have been conducted on the SOC and SOH estimation of lithium-ion batteries [35–38], there is a lack of research on the maintenance decision making for rechargeable batteries [39]. Existing studies on battery maintenance focus on time-based maintenance policies, ignoring the SOH of batteries and the stochastic nature of battery degradation [40]. To predictively maintain the health of battery systems, we introduce the RL algorithms to solve the MDP problem formulated to optimize the replacement threshold, while the continuous capacity loss of

batteries can be described using a stochastic process holding the Markovian property [40]. The results from the proposed RL algorithm using GP approximation are compared to the ones from the traditional value iteration algorithm for discrete MDPs, where the continuous state space is discretized into a finite number of states. In the proposed RL algorithm using GP as function approximation, we optimize the long-run average reward (instead of the discounted reward commonly in the literature) with iterations on both the state-action value function and the state value function. In addition, we implement a simulation experiment to compare the results with those from the exhaustive search and Monte Carlo estimation. The proposed research is expected to enhance the maintenance decision-making capability for complex systems, especially those concerning continuous states and discrete time. This enables us to solve the problem without discretizing the degradation state and leads to a more accurate result with a lower long time average cost compared with the discretized MDP formulation.

The remainder of the paper is organized as follows. Section 2 describes the general CBM problem under the MDP framework. In Section 3, we construct the RL algorithm with the GP as function approximation for CBM. Section 4 presents the numerical experiments of the proposed algorithm for the maintenance decision making of lithium-ion batteries, including both simulation studies and case studies. Finally, we provide a summary and discussion in Section 5.

2 System Description

condition-based maintenance is an effective maintenance approach that takes into account the real-time conditions of a system (e.g., degradation, capacity loss) observed through continuous monitoring or periodic inspection [41, 42]. The conditions of a system are either descriptive defined by domain experts [8–10] or extracted from sensor data commonly in recent applications [2, 3, 11]. For a non-repairable system, we study a CBM approach that considers actions of corrective maintenance and preventive maintenance. The objective is to

optimize the threshold for preventive maintenance that can minimize the total maintenance-related cost, where the maintenance actions take place according to the detected condition of the system at each decision epoch. The additional assumptions for the system are listed as follows.

1. A non-repairable system is continuously monitored or periodically inspected before each decision epoch.
2. The condition of the system is a continuous random variable, denoted as $S(t)$, which satisfies the Markovian property.
3. The system fails when $S(t)$ reaches a pre-determined end-of-life threshold, H.
4. Maintenance decisions are made at equally-spaced decision epochs based on the observed condition of the system:
   (a) If the system condition is observed to be beyond the end-of-life threshold, $S(t) > H$, then a *corrective maintenance* (CM) is performed, incurring a combined cost of the replacement cost, C_R, and a penalty cost due to downtime, C_D.
   (b) If the system condition exceeds a threshold for replacement, H_p, where $H_p < S(t) < H$, then a *preventive maintenance* (PM) is implemented even though the system is still functioning, and only the replacement cost, C_R, is incurred.
   (c) If the system condition is less than H_p, *no action* (N) is needed.
5. The time horizon is approximately infinite.

To evaluate the performance of the maintenance policy, we use a long-term average maintenance cost rate model, in which the optimal threshold for preventive replacement, H_p^*, is the decision variable:

$$C = \lim_{t\to\infty} \left(C_C \times N_C(t) + C_R \times N_P(t)\right)/t,$$

where C is the long-term average maintenance cost per unit of time, $C_C = C_R + C_D$ is the cost induced by a corrective maintenance action. $N_C(t)$ and $N_P(t)$ are the numbers of corrective maintenance and preventive maintenance actions by time t, respectively.

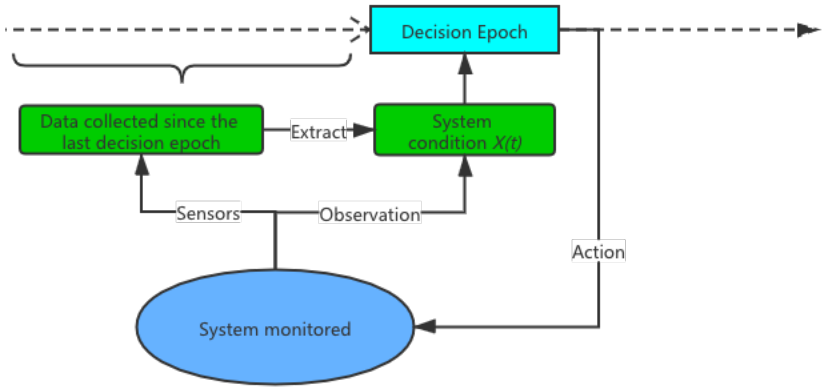

Figure 1: The Framework of MDP for CBM

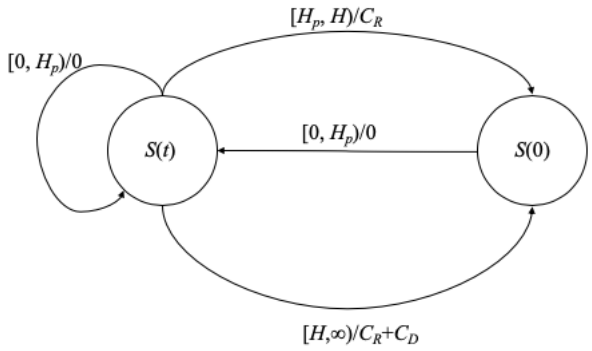

Figure 2: The State-Machine of MDP for CBM

2.1 Markov Decision Process for Condition-Based Maintenance

The overall framework of the MDP for the condition-based maintenance is shown in Figure 1. The system under maintenance is monitored to obtain its state at each decision epoch. The state of the system can be directly represented by its sensor data, which commonly leads to a large state space. To avoid the computational burden, the system states can be extracted from the sensor data or determined by field experts' observations. Based on the system state at each decision epoch, an action is taken on the system maintained.

The MDP model can be fully described by the quintuple $\{\mathcal{T}, \mathcal{S}, \mathcal{A}, p(s'|s,a), r(s,a)\}$ [43], where $\mathcal{T}$ stands for the countable set of decision epochs. In our MDP model for condition-based maintenance, $\mathcal{S}$ is the set of all possible values of the continuous system condition, instead of the discrete states [8–10] or the discretized sensor data [3] in most of the literature.

Although the system is constantly monitored, we take only its conditions at the decision epochs for decision making with the Markovian property assumption. At each decision epoch, if the system is in state $s \in \mathcal{S}$, an action $a \in \mathcal{A}$ is made that incurs an expected reward of $r(s, a)$. In our CBM model, $A = \{N, PM, CM\}$ denoting three different actions: doing nothing, preventive maintenance, and corrective maintenance. The reward function r is evaluated based on the total costs related to maintenance: $r = 0, -C_R$, or $-(C_R + C_D)$ for $a = N, PM$, or CM, respectively. The state transition probability distribution is represented by $p(s'|s, a)$. When $a = PM$ or $a = CM$, $p(0|s, a) = 1$ for all s. When $a = N$, the state transition probability, $p(s'|s, a)$, is estimated from existing degradation paths. When $\mathcal{S}$ is a countable set, $p(s'|s, a)$ provides a value for each $s, s' \in \mathcal{S}, a \in \mathcal{A}$. Otherwise, $p(s'|s, a)$ is assumed to be a probability density function.

Figure 2 depicts the state machine of the MDP model for the condition-based maintenance. The system starts with the as-good-as-new state $S(0) = 0$. As long as the system state is below the preventive maintenance threshold, $S(t) < Hp$, no action is taken with no cost accrued. Once $S(t)$ goes beyond H_p ($H_p < S(t) < H$), a preventive maintenance action is taken that brings the system to the as-good-as-new state $S(0) = 0$, incurring a preventive maintenance cost C_R. When $S(t)$ goes beyond H, a corrective maintenance action is performed to restore the system to the as-good-as-new state, with the maintenance cost C_R and the downtime cost C_D accrued to the total cost.

To learn from existing samples, we consider deterministic policies π that are functions, $\pi : \mathcal{S} \to \mathcal{A}$, which assign a single action to each range of state:

$$\pi(s) = \begin{cases} N & s \in [0, H_p) \\ PM & s \in [H_p, H) \\ CM & s \in [H, \infty) \end{cases}$$

The optimal policy, namely, H_p^*, can be obtained by using the RL approach to minimize the long-term average maintenance cost, denoted by C^*.

3 Methodology Description

3.1 Reinforcement Learning

In reinforcement learning, an agent interacts with the random environment at discrete times. At specified times called decision epochs, the agent observes the state of the system s and takes an action a accordingly, leading to an instant state s' and an instant reward r. The rule followed by the agent in selecting an action given the current state is called the policy of the agent, denoted by a function $\pi : \mathcal{S} \to \mathcal{A}$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and the action spaces, respectively. As aforementioned, reinforcement learning has been introduced to solve CBM problems, where the degradation state at each decision epoch constitutes the state transition samples of the underlying MDP. The main goal of RL in condition-based maintenance is to achieve an optimal policy π^* that minimizes the long-term average maintenance cost. Such an optimal policy satisfies the well-known Bellman optimality equation [44]:

$$v_*(s) + \rho^* = \max_a \sum_{s',r} p(s', r|s, a)[r + v(s')], \tag{1}$$

where the scalar $\rho^* = -C^*$ is the maximum long-term reward under policy π^*, and $v_*(s)$ denotes the value function of state s under policy π^*. For an average-reward MDP, the optimal policy is equivalent to the solution to its Bellman equation.

The state-of-the-art RL approach is characterized by the TD and function approximation. With the introduction of TD learning, RL algorithms no longer require exact transition models and reward structures to be known in advance [45]. A typical update in TD prediction takes the format

$$v_\pi(s) \leftarrow v_\pi(s) + \alpha[r + v_\pi(s') - v_\pi(s)],$$

where α is a constant called the learning rate. The Q-learning algorithm [46], the SARSA algorithm [47], the n-step $Q(\sigma)$ algorithm [48] and Dyna-Q [49] are important variants of the original TD algorithm. An effective technique to model the continuous state space in

RL is to use function approximation, which approximates the value function by estimating a function mapping the state-action pairs to values, instead of maintaining a table for the value function of each state-action pair [50]. In this approach, we update an approximate value function $\tilde{v}$ instead of precisely updating the table storing its value at every point. We carry out the value iteration on sample points $\{s_i\}_{i=1}^{n_s}$, rather than every state action pair. The Bellman equation in Eq. (1) changes to the maximization over $\tilde{v}$. Hence, the update in the value iteration becomes [51]:

$$v_{s_i} = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p(s'|s_i, a)\tilde{v}_*(s') + r(s_i, a), \quad for\ i = 1, \ldots, n_s.$$

Free of an exact representation of all possible values in value functions, a function approximation makes it possible for RL algorithms to estimate the value function over a large or continuous state-action space with a moderate number of available samples. As the value-function model has no restriction on the form of mappings, it can utilize modern modeling techniques, such as artificial neural network models [52] and the gradient descent improvement [53].

3.2 Gaussian Process in Reinforcement Learning

As aforementioned, it is challenging for a discrete MDP to provide an accurate estimation of transition probabilities and reward structures, while maintaining the Markov property. In order to handle a large or continuous state space that cannot be addressed by the tabular method, we turn to the function approximation to model the state transitions of the system and the value functions (both state value functions and state-action value functions) for continuous states. When there is not enough information on the possible function to approximate, a general approximator is preferred. Although neural networks are capable to model various relationships, they usually require a large amount of data. Therefore, we turn to the GPR that can fit small datasets without loss of generality.

3.2.1 Inference and Prediction of Gaussian Process Regression

The Gaussian Process for reinforcement learning (GPRL) method first models the state transitions as a GP, then uses a set of support points for the value iteration of the value function that is modeled by another GP. A GP $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j)))$ is a collection of random variables, any finite number of which have a joint Gaussian distribution. A Gaussian process can be fully characterized by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ [54]. In our CBM model, the independent variable $\mathbf{x}$ and the dependent variable $f(\mathbf{x})$ of the GP are replaced by s and s', respectively, i.e., the state of the system under maintenance at the current and the next decision epochs, both of which are one-dimensional signals. As described in Section 2, when the action $a = PM$ or CM, the state of the system returns to the as-good-as-new state. Therefore, the following Gaussian process, $f(s)$, is utilized to model the transition dynamics when $a = N$:

$$\begin{aligned} m(s) &= \mathbb{E}(f(s)) \\ k(s_i, s_j) &= \mathbb{E}[(f(s_i) - m(s_i))(f(s_j) - m(s_j))]. \end{aligned}$$

In a Gaussian process, the noise in the observed data over time is commonly modeled as a normal random variable that is assumed to be independent of the process [54]:

$$s' = f(s) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{2}$$

where s' is the dependent variable and ε is the residual.

For one-dimensional data in this research, we consider a kernel function that is specified as the squared exponential function over states [54]:

$$k(s_i, s_j) = v^2 e^{\frac{-(s_i - s_j)^2}{2l^2}} \tag{3}$$

where v and l are the parameters of interest.

In addition, we consider the Matern kernel function that is given as [54]:

$$k_{matern}(s_i, s_j) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\sigma_l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r}{\sigma_l} \right), \tag{4}$$

where $\Gamma(\cdot)$ is the gamma function and $K_\nu(\cdot)$ is the modified Bessel function of the second kind, and $r = \sqrt{(s_i - s_j)^2}$. σ_l, and σ_f are the model parameters to be estimated and the parameter ν needs to be manually specified.

In order to catch the non-zero mean function, we can use a basis function $\mathbf{h}(s)$ whose components constitute the mean function of the GP with coefficients $\boldsymbol{\beta}$. The GP $f(s)$ in Eq. (2) can then be represented by a zero-mean GP, $g(s)$, and the basis function whose closed-form likelihood function can be obtained from the literature [54]:

$$f(s) = g(s) + \mathbf{h}(s)^\top \boldsymbol{\beta}, \quad where\ g(s) \sim \mathcal{GP}(0, k(s_i, s_j)). \tag{5}$$

With a sample transition $(\mathbf{s}, \mathbf{s}')$, the parameter estimates can be obtained by maximizing the likelihood function of the state transition probability function [54]:

$$\log p(\mathbf{s}'|\mathbf{s}, \mathbf{b}) = -\frac{1}{2}(H^\top \mathbf{b} - \mathbf{s}')^\top K_{s'}^{-1}(H^\top \mathbf{b} - \mathbf{s}') - \frac{1}{2}\log |K_{s'}| - \frac{n}{2}\log 2\pi, \tag{6}$$

where $H = (\mathbf{h}(s_1), ..., \mathbf{h}(s_n))$, $K_{s'} = K + \sigma_n^2 I$ with K satisfying $K_{ij} = k(s_i, s_j)$ by taking the observation noise into consideration.

With the estimation of parameters, the upcoming state transitions are predicted by integrating the new point s_* into the current GP [20] :

$$\begin{bmatrix} \mathbf{s}' \\ s'_* \end{bmatrix} \sim \mathcal{N} \left([H, \mathbf{h}(s_*)]^\top \boldsymbol{\beta}, \begin{bmatrix} K & K_*^\top \\ K_* & K_{**} \end{bmatrix} \right)$$

where

$$K_* = [k(s_*, s_1)\ ...\ k(s_*, s_n)], \quad K_{**} = k(s_*, s_*).$$

Then the distribution of the updated state transition can be represented by [20]

$$s'_*|\mathbf{s}' \sim N\left(K_* K^{-1}\mathbf{s}' + \mathbf{h}(s_*)^\top \boldsymbol{\beta}, K_{**} K^{-1} K_*^\top\right). \tag{7}$$

Given s_*, the mean term $K_* K^{-1}\mathbf{s}' + \mathbf{h}(s_*)^\top \boldsymbol{\beta}$ in Eq. (7) can be regarded as the point estimation of s'_*. The confidence interval on s'_* can be obtained using the estimated variance of s'_* given by $K_{**} K^{-1} K_*^\top$.

3.2.2 Gaussian Process in Reinforcement Learning over Long Term Average Reward

The commonly used RL algorithms in the literature are mostly for optimizing the expected discounted reward, in which a discount factor is used to control the impact of the future reward on the current state [29, 32]. In this research, however, we need to optimize the long-term average reward, i.e., to minimize the long-term average maintenance cost of the system. Therefore, the algorithms for optimizing the expected discounted reward cannot be readily utilized. With the absence of the discount factor, the accumulated reward for every state of this ergodic process is infinite. Inspired by the idea of policy iteration for the long-term average reward in the discrete Markov decision model [13], we can assume that the change in reward after each step for every state stays the same in the long run under a fixed policy. In this way, we can change the stabilization condition accordingly to optimize over the long-term average cost.

The detailed steps are described in Algorithm 1, or GPRL algorithm, in which the increment of value function at each support point is evaluated as its Euclidean diameter and calculated after every iteration. Compared to the RL algorithms for optimizing the expected discounted reward, we change the condition for convergence to achieve the optimal long-term average reward: it converges when the diameter of the increment becomes smaller than a specified threshold value. We also store $p(s'|s,a)$ to save the time of numerical integration. Another version of Algorithm 1 is developed and implemented for the state-action value iteration, in which the state-action value function is modeled by another GP, and the update is changed accordingly.

Algorithm 1: Value Iteration for Average Reward in GPRL

Step 1: Input n observations of state transition in a fixed interval Δt of the form (s, a, s') and one-step reward function $r(s, a)$.
Step 2: Model the state transition with GPR, and get the transition probability as a normal distribution $p(s'|s, a) = \mathcal{N}(\mu_{s'}(s), \Sigma_{s'}(s))$.
Step 3: Choose a set of m supporting points $S = \{s_1, ..., s_m\}$ and initialize their value function as $V_i = R_i = r(s_i, a_0)$ where a_0 is assigned arbitrarily. Then fit the GPR for value function.
Step 4: set $\Delta = 1$, $\epsilon = 0.00001$
while $\Delta \geq \epsilon$, **do**
  for $i = 1, ..., m$, **do**
    Choose the optimal policy based on current values for supporting points:
    $a_i = \arg\max_a \int p(s'|s_i, a)[r(s_i, a) + V(s')]ds'$.
    Compute new values for supporting points: $V(s_i)^{new} = \int p(s'|s_i, a_i)[r(s_i, a_i) + V(s')]ds'$.
    Update the reward function for supporting points: $R_i = r(s_i, a_i)$.
  end
  Compute the maximal and minimal differences and test the stationary property:
  $\delta_{max} = \max(V^{new} - V)$, $\delta_{min} = \min(V^{new} - V)$, $\Delta = \delta_{max} - \delta_{min}$.
  Update the GPR model for value function with new R_is.
end

4 Numerical Experiments

We demonstrate the performance of the proposed RL algorithm with GP approximation (GPRL algorithm) using simulation experiments and case studies on a CBM decision-making problem for lithium-ion batteries, while the continuous capacity loss of batteries can be described using a stochastic process holding the Markovian property. The following model parameters are shared among simulation experiments and case studies with real data. The preventive maintenance cost is the cost of replacing a battery, i.e., the cost of a new battery and a negligible labor cost, which is estimated as \$10,000 for a 100kWh battery set. The downtime cost when a battery ceases to function is estimated to be \$3,000. To normalize the cost parameters, we assign $C_R = -1.0$ for the preventive maintenance cost and $C_D = -0.3$ for the downtime cost, respectively. The system state, $\mathcal{S} = [0, 1]$, indicates the percentage of capacity loss of a battery. Our objective is to minimize the average cost per decision epoch in the long term.

4.1 Simulation Study

We start with a simulation experiment to compare the results of the GPRL algorithm with those from the exhaustive search using Monte Carlo simulation, in terms of the accuracy in the optimal policy and cost. In this experiment, the degradation paths for the continuous capacity loss of batteries are sampled from a Wiener process with a drift of 0.0002 and a variance of 0.001^2, where the drift value is chosen to obtain an expected cycle life of 1,000 when $H = 0.2$.

For the GPRL algorithm, we have 100 paths in the training set and 100,000 paths in the test set to validate the performance. The numerical integration and policy evaluation are conducted at an interval of 0.001 for $\mathcal{S} = [0, 1]$. In the GPRL algorithm, the mean function is set to be the linear base function $\mathbf{h}(s)$ in Eq. (5) and the covariance function is the squared exponential function in Eq. (3). The GPRL algorithm achieved the optimal policy of $H_p^* = 0.183$ with the average cost per 50 cycles of \$531.71.

On the other hand, the result of the Monte Carlo simulation at the same interval length of 0.001 is shown in Figure 3, where the average cost per 50 cycles from the exhaustive search is \$533.69 with the same optimal policy. The agreement of the results from the GPRL algorithm and the Monte Carlo simulation indicates that the proposed GPRL method is capable of reaching the optimal policy without the degradation model to be known. In addition, the proposed GPRL algorithm is more computationally effective for complicated, unknown degradation models.

4.2 Case Study

4.2.1 Data Description and Model Details

Case studies are implemented to demonstrate the performance of the GPRL algorithm on the randomized battery usage data from NASA Ames Research Center [56]. We used the data from six different experiments carried out under different current and temperature conditions, each containing four trajectories of capacity loss obtained under the same exper-

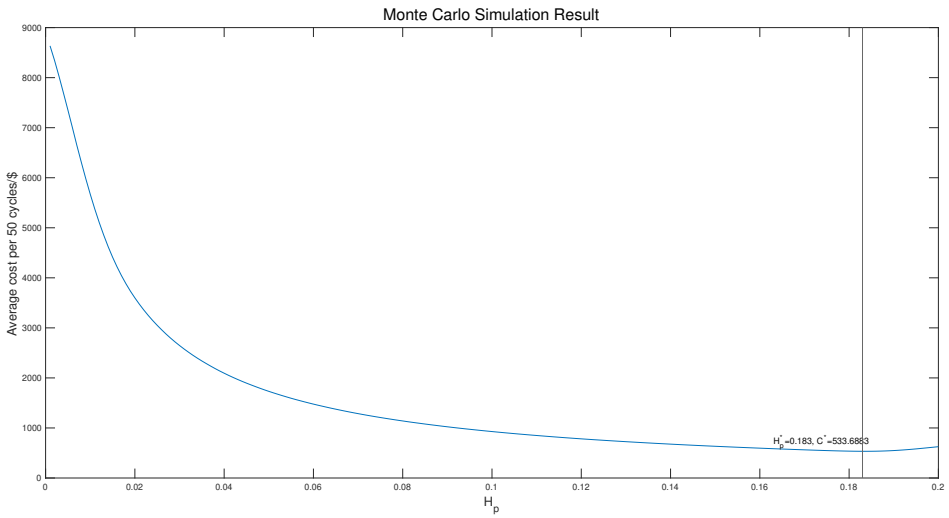

Figure 3: Monte Carlo Simulation Results When $H = 0.2$

imental environment. For each of the 24 trajectories, we calculated the capacity loss from the recorded voltage and current using reference periods of every 50 charging-discharging cycles. Figure 4 presents the capacity loss in percentage over time for the four trajectories in one of the six experiments.

Before we implement the proposed GPRL algorithm, the discrete MDP approach using the value-iteration algorithm is applied to this dataset of 24 trajectories, where the continuous state space is discretized into a finite number of states. The results are then compared in terms of the optimal objective function and the computational burden. According to industry standards [18, 33, 36], the battery lifetime is determined by using 20%, 30%, and 40% capacity loss as the end-of-life threshold of lithium batteries, i.e., $H = 0.2, 0.3$, and 0.4, respectively.

4.2.2 Results from Discrete MDP

We first formulate the degradation of capacity loss as a discrete-time discrete-state MDP to serve as a benchmark. The continuous capacity loss data is discretized into bins that are small enough to maintain the Markov property. It can be verified using a chi-square homogeneity test on the transition probability from the current state to the next state

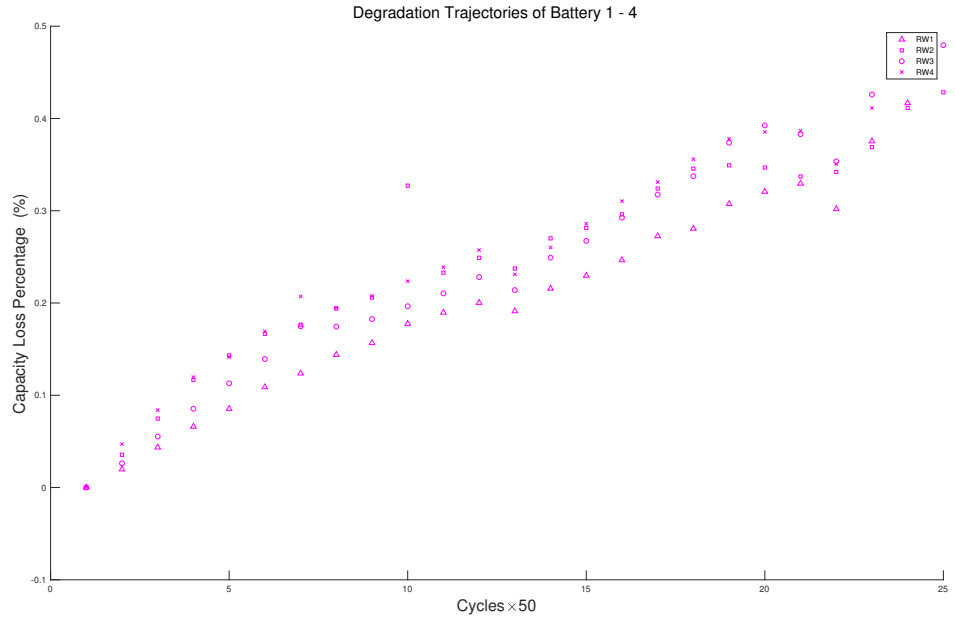

Figure 4: Capacity Loss Trajectories of Four Batteries in NASA Randomized Usage Test [56]

conditioning on the previous state. For the NASA randomized battery usage data, the capacity loss in percentage is discretized into 21 states from 0% to 40%. The test is conducted for each state and the P-value is larger than 0.05 in 18 out of 21 states, indicating that the Markovian property holds for these states. With the determined bin size, we can estimate the transition probability $p(s'|s,a)$ and the expected reward $r(s,a)$ from the sample trajectories. The transition matrix of states is sparse as only a few neighboring states can be reached in each decision epoch. The results are obtained after solving the MDP using the value iteration algorithm, and are presented in Table 1 to serve as a benchmark.

4.2.3 Results from GPRL Algorithm

In this case study, we implement the GPRL value iteration algorithms with the state value function and the state-action value function, respectively, to compare their solutions and computation time. In both algorithms, the state transition is modeled by a GP. To determine the mean and kernel functions in the GPRL iteration algorithms, we used another dataset generated by Severson et al [18] that has the capacity loss in percentage recorded

after each charging-discharging cycle (rather than every 50 cycles).

To determine the mean function in Eq. (5) among the choices of 0, constant and linear ones, we estimate the state transition model from the initial samples of a selected capacity loss trajectory in the dataset [18]. The extrapolation capability of the model is tested using the mean squared error (MSE) on the remaining state transition samples of the same trajectory. Different numbers of initial state transition samples are used to evaluate their MSE in estimating the system dynamics. The results show that the GPR with the linear mean function outperforms the other choices with a minimum out-of-sample MSE by using the first 200 cycles. The Gaussian process regression helps to secure a reasonably accurate state transition model using only samples at the early stage of the degradation. With the chosen linear mean function, we explore the kernel function in the algorithm among the choices of the squared exponential kernel in Eq. (3), and the Matern kernel in Eq. (4) with the parameter ν of 3/2 and 5/2, respectively. Since the resulted out-of-sample MSE is similar among different kernels, we decide to choose the squared exponential kernel for computational simplicity and the advantage of the squared exponential kernel demonstrated in [26].

With the chosen mean and kernel functions, the GPR model for the state transitions of the capacity loss trajectory is obtained by maximizing the loglikelihood function in Eq. (6), which is approximately linear as shown in Figure 5. In the GPRL algorithm, we apply the value iteration over the state value function and the state-action value function to the NASA dataset, respectively. As given in Table 1, similar results are obtained from the two algorithms for the different values of the end-of-life threshold, H, respectively. The algorithm conducting the value iteration over the state value function runs significantly faster, as the value function approximation is more straightforward than that of the state-action value function. The difference in computation time increases when the action set becomes larger.

To compare with the discrete MDP approach, we use the results from the GPRL algorithm with state value iteration, due to its shorter computation time without much compromise on the optimal policies. When $H = 0.2$, 0.3, and 0.4, the minimal average cost per 50 cycles is

$1330, $839, and $579, respectively, which saves about 2.3%, 7.3%, and 11.9% in the average cost compared to the MDP results. The improvement in the average cost is explained by the more accurate policy the GPRL algorithm can provide, while H_p^* is restricted to one of its bin edges in the discrete MDP. When the end-of-life threshold, H is larger, the maintenance actions are less frequently conducted and thus the error in H_p^* leads to a greater proportion of the additional costs in C^* provided by the discrete MDP.

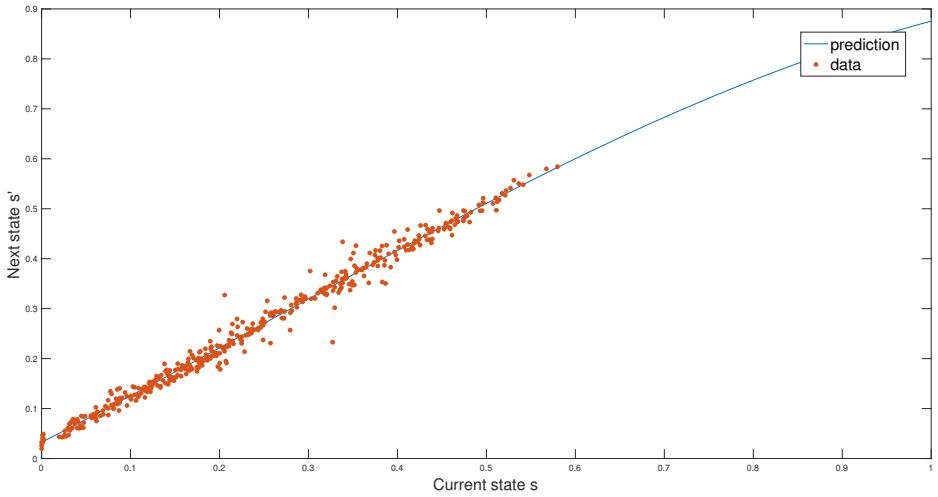

Figure 5: State Transition Gaussian Process

Table 1: Results of GPRL and Discrete MDP

H	Algorithm	C^* ($1000)	H_p^*	Running time(s)
0.2	State-action value iteration	1.323	0.174	156.11
	State value iteration	1.330	0.175	90.11
	Discrete MDP	1.362	0.160	-
0.3	State-action value iteration	0.835	0.271	394.05
	State value iteration	0.839	0.271	213.31
	Discrete MDP	0.905	0.260	-
0.4	State-action value iteration	0.576	0.371	936.97
	State value iteration	0.579	0.371	537.46
	Discrete MDP	0.657	0.340	-

5 Conclusions

Although reinforcement learning has been implemented as an effective approach for solving MDPs in maintenance problems, most RL algorithms applied are restricted to discrete

state and action spaces. In this research, we develop an algorithm to find the optimal policy for discrete-time continuous-state MDPs for CBM decision-making. We use the GPR as function approximation to model the state transition and the value functions of states in RL. In addition, in the proposed GPRL algorithm, the long-run average reward (instead of the discounted reward commonly in the literature) is optimized with iterations on the state-action value function and the state value function, respectively.

Specifically, to demonstrate the proposed method, we model the battery maintenance decision-making problem by an MDP, where the Gaussian process regression is introduced to describe the system dynamics and value functions. Using NASA battery randomized usage data, we implement the proposed GPRL algorithm over the state value iteration and the state-action value iteration, respectively. The results from the two versions of GPRL algorithms are compared with the ones from the discrete MDP approach, which verifies the capability of our proposed algorithm in achieving accurate results. The advantage of using our approach in RL is evident by keeping the continuous states of the degradation process. The methodology we investigated in this research can be readily applied to maintenance decision-making for various discrete-time continuous-state systems, such as equipment or facilities under daily or weekly inspection.

Further studies can be implemented to the applications of RL using the Gaussian process to other engineering systems. We can also explore different ways of modeling continuous states in a decision process, such as a specific parametric model (instead of a universal model of GPR) that can capture the physical properties of a system. In addition, the proposed work considers a relatively simple situation where a single variable deterioration state is studied. When multiple degradation processes exhibit in a system, the proposed GPRL method can be readily extended to the maintenance of such a system, as the Gaussian process regression can be naturally generalized to the vector state and the RL can well handle the delayed reward.

References

[1] Tapas K. Das, Abhijit Gosavi, Sridhar Mahadevan, and Nicholas Marchalleck. Solving semi-markov decision problems using average reward reinforcement learning. *Management Science*, 45(4):560–574, 1999.

[2] Sina Zarrabian, Rabie Belkacemi, and Adeniyi A. Babalola. Reinforcement learning approach for congestion management and cascading failure prevention with experimental application. *Electric Power Systems Research*, 141:179 – 190, 2016.

[3] A. S. Xanthopoulos, A. Kiatipis, D. E. Koulouriotis, and S. Stieger. Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. *IEEE Access*, 6:576–588, 2018.

[4] Theodore T. Allen, Sayak Roychowdhury, and Enhao Liu. Reward-based monte carlo-bayesian reinforcement learning for cyber preventive maintenance. *Computers & Industrial Engineering*, 126:578 – 594, 2018.

[5] Michele Compare, Luca Bellani, Enrico Cobelli, Enrico Zio, Francesco Annunziata, Fausto Carlevaro, and Marzia Sepe. A reinforcement learning approach to optimal part flow management for gas turbine maintenance. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 234(1):52–62, 2020.

[6] X. Xu and X. Wang. Aircraft engine maintenance based on reinforcement learning. In *2020 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pages 958–960, 2020.

[7] Yu Liu, Yiming Chen, and Tao Jiang. Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research*, 283(1):166 – 181, 2020.

[8] Kamal Golabi, Ram B. Kulkarni, and George B. Way. A statewide pavement management system. *INFORMS Journal on Applied Analytics*, 12(6):5–21, 1982.

[9] John Rust. Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica*, 55(5):999–1033, 1987.

[10] William T. Scherer and Douglas M. Glagola. Markovian models for bridge maintenance management. *Journal of Transportation Engineering*, 120(1):37–51, 1994.

[11] Y. Wu and H. Zhao. Optimization maintenance of wind turbines using markov decision processes. In *2010 International Conference on Power System Technology*, pages 1–6, Oct 2010.

[12] Yinhui Ao, Huiping Zhang, and Cuifen Wang. Research of an integrated decision model for production scheduling and maintenance planning with economic objective. *Computers & Industrial Engineering*, 137:106092, 2019.

[13] Henk C.Tijms. *A First Course in Stochastic Models*. John Wiley & Sons, Ltd, 2004.

[14] Michael L Littman, Thomas L Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh conference on*

Uncertainty in artificial intelligence, pages 394–402. Morgan Kaufmann Publishers Inc., 1995.

[15] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[16] Dongyan Chen and Kishor S. Trivedi. Optimization for condition-based maintenance with semi-markov decision process. *Reliability Engineering & System Safety*, 90(1):25 – 29, 2005.

[17] E. Byon, L. Ntaimo, and Y. Ding. Optimal maintenance strategies for wind turbine systems under stochastic weather conditions. *IEEE Transactions on Reliability*, 59(2):393–404, June 2010.

[18] Kristen A. Severson, Peter M. Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H. Chen, Muratahan Aykol, Patrick K. Herring, Dimitrios Fraggedakis, Martin Z. Bazant, Stephen J. Harris, William C. Chueh, and Richard D. Braatz. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391, 2019.

[19] Andrew Kusiak and Anoop Verma. Analyzing bearing faults in wind turbines: A data-mining approach. *Renewable Energy*, 48:110 – 116, 2012.

[20] Mark Ebden et al. Gaussian processes for regression: A quick introduction. *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 91:424–436, 2008.

[21] A. Zitrou, T. Bedford, and A. Daneshkhah. Robustness of maintenance decisions: Uncertainty modelling and value of information. *Reliability Engineering & System Safety*, 120:60 – 71, 2013.

[22] A. Daneshkhah, N.G. Stocks, and P. Jeffrey. Probabilistic sensitivity analysis of optimised preventive maintenance strategies for deteriorating infrastructure assets. *Reliability Engineering & System Safety*, 163:33 – 45, 2017.

[23] Alireza Daneshkhah, Amin Hosseinian-Far, and Omid Chatrabgoun. Sustainable maintenance strategy under uncertainty in the lifetime distribution of deteriorating assets. In *Strategic Engineering for Cloud Computing and Big Data Analytics*, pages 29–50. Springer, 2017.

[24] Ravi Kumar Pandit and David Infield. Scada-based wind turbine anomaly detection using gaussian process models for wind turbine condition monitoring purposes. *IET Renewable Power Generation*, 12(11):1249–1255, 2018.

[25] Ravi Kumar Pandit and David Infield. Comparative analysis of binning and gaussian process based blade pitch angle curve of a wind turbine for the purpose of condition monitoring. *Journal of Physics: Conference Series*, 1102:012037, oct 2018.

[26] Ravi Kumar Pandit and David Infield. Comparative analysis of gaussian process power curve models based on different stationary covariance functions for the purpose of improving model accuracy. *Renewable Energy*, 140:190 – 202, 2019.

[27] Yanting Li, Shujun Liu, and Lianjie Shu. Wind turbine fault diagnosis based on gaussian process classifiers applied to operational data. *Renewable Energy*, 134:357 – 366, 2019.

[28] Raed Kontar, Shiyu Zhou, Chaitanya Sankavaram, Xinyu Du, and Yilu Zhang. Nonparametric modeling and prognosis of condition monitoring signals using multivariate gaussian convolution processes. *Technometrics*, 60(4):484–496, 2018.

[29] Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets Bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 154–161, 2003.

[30] Carl Edward Rasmussen and Malte Kuss. *Gaussian Processes in Reinforcement Learning*. 2004.

[31] Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. pages 201–208, 2005.

[32] Robert Grande, Thomas Walsh, and Jonathan How. Sample efficient reinforcement learning with gaussian processes. 32(2):1332–1340, 22–24 Jun 2014.

[33] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M. Chow. Battery management system: An overview of its application in the smart grid and electric vehicles. *IEEE Industrial Electronics Magazine*, 7(2):4–16, June 2013.

[34] L. Liu, L. Y. Wang, Z. Chen, C. Wang, F. Lin, and H. Wang. Integrated system identification and state-of-charge estimation of battery systems. *IEEE Transactions on Energy Conversion*, 28(1):12–23, March 2013.

[35] Eberhard Meissner and Gerolf Richter. The challenge to the automotive battery industry: the battery has to become an increasingly integrated component within the vehicle electric power system. *Journal of Power Sources*, 144(2):438 – 460, 2005. Selected papers from the Ninth European Lead Battery Conference.

[36] Anthony Barré, Benjamin Deguilhem, Sébastien Grolleau, Mathias Gérard, Frédéric Suard, and Delphine Riu. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 241:680 – 689, 2013.

[37] Dave Andre, Christian Appel, Thomas Soczka-Guth, and Dirk Uwe Sauer. Advanced mathematical methods of soc and soh estimation for lithium-ion batteries. *Journal of Power Sources*, 224:20 – 27, 2013.

[38] Marcus Johnen, Christian Schmitz, Maria Kateri, and Udo Kamps. Fitting lifetime distributions to interval censored cyclic-aging data of lithium-ion batteries. *Computers & Industrial Engineering*, 143:106418, 2020.

[39] Lew Fulton, J Ward, P Taylor, and T Kerr. *Technology roadmap: Electric and plug-in hybrid electric vehicles*. OECD/IEA, 2009.

[40] Zhi-Sheng Ye and Min Xie. Stochastic modelling and analysis of degradation for highly reliable products. *Applied Stochastic Models in Business and Industry*, 31(1):16–32, 2015.

[41] Ernnie Illyani Basri, Izatul Hamimi Abdul Razak, Hasnida Ab-Samat, and Shahrul Kamaruddin. Preventive maintenance (pm) planning: a review. *Journal of Quality in Maintenance Engineering*, 2017.

[42] Ciriaco Valdez-Flores and Richard M Feldman. A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics (NRL)*, 36(4):419–446, 1989.

[43] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[44] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[45] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, Aug 1988.

[46] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, 1989.

[47] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[48] Kristopher De Asis, J Fernando Hernandez-Garcia, G Zacharias Holland, and Richard S Sutton. Multi-step reinforcement learning: A unifying algorithm. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[49] Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163, July 1991.

[50] Dimitri P. Bertsekas. *Neuro-Dynamic Programming*, pages 2555–2560. Springer US, Boston, MA, 2009.

[51] Sridhar Mahadevan. Average reward reinforcement learning: foundations, algorithms, and empirical results. *Machine learning*, 22(1-3):159–195, 1996.

[52] Andrew G Barto, Charles W Anderson, and Richard S Sutton. Synthesis of nonlinear control surfaces by a layered associative search network. *Biological Cybernetics*, 43(3):175–185, 1982.

[53] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[54] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[55] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[56] Brian Bole, Chetan S Kulkarni, and Matthew Daigle. Adaptation of an electrochemistry-based li-ion battery model to account for deterioration observed under randomized use. Technical report, SGT, Inc. Moffett Field United States, 2014.