INL/CON-23-72386-Revision-0

Quantifying Uncertainty of Deep Reinforcement Learning Based Decision Making for Operations and Maintenance of Nuclear Power Plant

April 2023

Ryan Matthew Spangler, Daniel G. Cole

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-23-72386-Revision-0

Quantifying Uncertainty of Deep Reinforcement Learning Based Decision Making for Operations and Maintenance of Nuclear Power Plant

Ryan Matthew Spangler, Daniel G. Cole

April 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-NE0008909

Quantifying Uncertainty of Deep Reinforcement Learning Based Decision Making for Operations and Maintenance of Nuclear Power Plant

Ryan M. Spangler*, Daniel G. Cole

Mechanical Engineering and Materials Science, University of Pittsburgh, Pittsburgh, PA

[leave space for DOI, which will be inserted by ANS]

ABSTRACT

This paper summarizes research that integrates condition monitoring and prognostics with decision-making for nuclear power plant operations and maintenance. As part of this research, we have developed an online asset management tool to help reduce life-cycle maintenance and repair costs. Using the latest advancements in condition monitoring, supply chain analytics, and deep reinforcement learning, we have created a predictive maintenance tool that can optimize the maintenance and spare-part management of a repairable nuclear system. To demonstrate these methods, preliminary studies were conducted on a simple, representative maintenance system undergoing a stochastic degradation process that requires repairs or replacement to continue operation. Through Monte Carlo simulations, we were able to reduce maintenance spending by approximately 50% compared to optimized, time-based maintenance strategies. Not only does the decision maker reduce the average life-cycle costs, it also minimizes the chance of high cost scenarios, lowering the variance of the expected cost distributions, and reducing overall financial risk. Furthermore, this work also studies the ability of the decision maker to handle various levels of noise from observation uncertainty. By introducing uncertainty into the decision-making process, we have quantified the robustness and resiliency of the decision maker, as well as identified necessary levels of observability to demonstrate cost effectiveness.

Keywords: Machine Learning, Deep Reinforcement Learning, Operations and Maintenance, Decision Making, Nuclear

1. INTRODUCTION

Although nuclear power is one of the most reliable and available forms of energy production, it remains costly, preventing widespread adoption of the low-carbon energy source. A large part of this cost is due to high operations and maintenance (O&M) costs associated with maintaining plant structures, systems, and components. Several plants have reported O&M costs exceeding 50% of yearly operating budgets, while some have reported amounts as high as 66% [1]. For nuclear power to be a viable and competitive power generation method, O&M costs must be reduced.

Traditionally, O&M relies on overly conservative time-based schedules, historical data, and expert opinions, without considering real-time component conditions. Scheduled maintenance is known to be inefficient and can result in unexpected failures, unnecessary inspections, and overly conservative repairs, the combination of which contributes to high costs of O&M.

*rmspangler@pitt.edu

To combat these challenges, condition monitoring equipment and analytics have been developed to provide real-time estimates of component health and diagnostics. However, real-time condition monitoring does not provide decision support and still relies on expert opinion. Further complicating the decision process is the strict operating requirements and constraints that are unique to the nuclear industry. Challenges such as refueling outage schedules, costly unplanned shutdowns, supply chain uncertainties, long lead-times for high-value assets, and the inability to quickly shutdown and restart reactor operations make decision making for nuclear O&M difficult.

Handling these many risks, various inputs, and constraints is challenging for any experienced nuclear operator. What is needed is an online decision-support tool that has the ability to manage inventory and optimize maintenance, given the current conditions and various risks of the plant. Markov Decision Processes (MDP) and Partially Observable Markov Decision Process (POMDP) have been extensively used in research and industry to study decision making systems and identify optimal policies [2,3]. However, finding the optimal policy of an MDP or POMDP remains a challenging problem and is difficult for classical solvers and optimization algorithms. Policy optimization often suffers from the "curse of dimensionality," where the large state and action spaces make these problems intractable in size and difficult to solve. Performing exhaustive searches of the possible decision sequences becomes computationally expensive and, for large systems with many decisions and states, impossible. Advanced tree search algorithms have been used in attempts to alleviate these issues, but they can still take long amounts of time to solve and have limited performance with large decision and action spaces.

Further complicating this challenge is the partial observability of condition monitoring estimates. The inherent uncertainty when estimating component health makes solving for optimal solutions challenging since the probabilistic outcome can result in an almost infinite number of possible states. This means that the optimal decision sequence involves solving for the optimal solution for the entire distribution or belief of the actual state. Each possible state in that distribution must be considered the actual state, making the solution computationally expensive or intractable, even when discretized. Dynamic programming and point-based solvers have also been used in attempts to alleviate this issue, but still fall short during large branching factors, constraints, long time horizons, and limited performance when using resource-constrained computational environments.

The latest advancements in machine learning and artificial intelligence have proven successful in these difficult environments. The use of neural networks and reinforcement learning combine to create and train nonlinear function approximations for optimal decision-making. Most notably, the method of reinforcement learning and neural networks, known as Deep Reinforcement Learning (DRL), have been successfully paired together in a number of competitive games, such as several Atari games, Go, Starcraft, and Dota 2 [4–6]. Neural networks have overcome several challenges that hindered classic artificial intelligence (AI) algorithms such as the curse of dimensionality, large decision spaces, uncertainty, and long planning horizons. These challenges also plague decision making in nuclear plant asset management, making DRL a suitable algorithm for policy optimization and deployment.

More closely related applications of reinforcement learning applied to inspection and maintenance planning in various engineering fields have lowered overall lifetime costs by balancing inspections, maintenance, and risk [7–10]. However, the unique constraints of the nuclear industry are missing. Some of these constraints involve limited maintenance access during operation, high costs of unplanned downtime, long lead times, and uncertain supply chain delays. Furthermore, sensitivity analysis of model uncertainty has not been directly studied using a trained DRL agent in this environment. Different levels of observability have been studied in DRL literature, but its affect on nuclear O&M remains unknown. Understanding the advantages and disadvantages for these methods will be especially helpful for proving the viability of AI and machine-learning methods for the nuclear industry.

2. METHODS

This analysis has two main components: modeling and optimization. The decision-making process is first modeled with stochastic models and includes states, actions, and costs. The decision-making policy is then optimized using reinforcement learning to minimize long-term costs. Using the optimized policy, we

can then compare the result to current maintenance strategies and identify uncertainty through sensitivity analysis.

2.1. Partially Observable Markov Decision Process

To integrate and model the decision-making process, we will use a generalized version of MDP, known as a POMDP. MDPs and POMDPs have been used broadly in research applications as a method for modeling and optimizing discrete stochastic systems that include decision-making effects. The difference between MDPs and POMDPs is the observability of hidden system variables. MDPs make decisions based on the known state of the system, whereas POMDPs make decisions based on estimates of the hidden state, or observations. These estimates are often referred to as *beliefs*, which are probabilistic distributions over all states of the variable inferred from noisy observations, such as sensors or inspections. This partial observability creates a challenge when solving for the optimal decision sequence since the actual state of the system could be any value in the distribution.

A POMDP is a 7-tuple $(S, A, T, R, \Omega, O, \gamma)$ containing: states S, actions A, transition probabilities such that $T(s, a, s') = P(s'|s, a)$, rewards $R(s, a)$, observations Ω, conditional observation probabilities O, and a discount factor γ. At each timestep, the agent takes an action, $a \in A$, which then transitions the state $s \to s'$ according to the transition probability $T(s'|s, a)$. Once this occurs, the agent then receives a reward for the action $R(s, a)$. In this process, a sequence of actions is considered a policy, denoted as π. At each timestep, the policy specifies an action to execute based upon the state of the system, denoted as $\pi(s)$. The goal of solving the POMDP is to identify the optimal policy, π^*, that maps the current observation to the action that maximizes the long-term reward. The optimal policy will also take into account future rewards so that the current action maximizes the total reward when a terminal state is reached. In the context of asset management, the objective is to minimize costs over the lifetime of the plant.

Combining the methods of degradation modeling with the mathematical decision frameworks established by POMDPs, we get improved modeling of hidden states, imperfect observations, and uncertain beliefs. POMDPs have provided researchers with the ability to model and solve decision-making processes in discrete, stochastic systems. A representative POMDP can be seen in Figure 1. The issue with the current

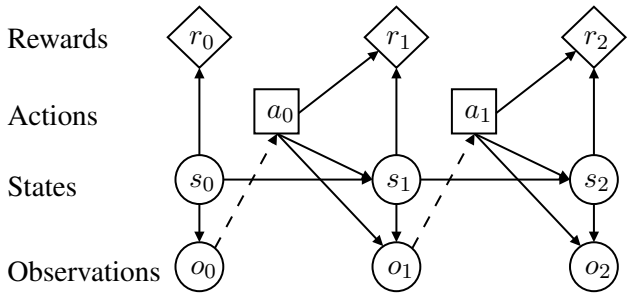

Figure 1: A POMDP with rewards, actions, states, and observation will be used to model the environment.

state of the art for POMDP solvers is that they must be solved every time the state changes. This creates a slow decision-making algorithm that must be constantly resolved when used with online condition monitoring techniques. What is needed is a once-solved optimization algorithm that can provide a function approximation to map the current state to the actions, maximizing the finite or possibly infinite horizon rewards. This solution will provide an optimal online decision-making tool that can make quick decisions based upon rapidly changing states.

2.2. Deep Reinforcement Learning for Optimal Policy Approximation

For optimal O&M decisions, we must be able to create a policy that maps the current state of the plant to the most effective action. The optimal policy, π^*, maps the current state to the action that maximizes the expected long-term reward, or in the case of O&M, minimizes the expected long-term cost. To solve

for the optimal O&M policy, we will use DRL that combines deep neural networks with reinforcement learning. Using this optimal policy, operators will be assisted in making O&M decisions that take into account complicated system interactions, uncertainty, and financial risk.

2.2.1. Reinforcement learning algorithm

Reinforcement learning is a machine-learning method used to train automated intelligent agents to take actions in their environment that will maximize the expected reward. The agent is the decision maker and the environment is an interactive model of the system. In the case of inspection and maintenance planning, we train an agent that can use the current and predicted health of a component to select optimal decisions that minimize expected lifetime maintenance costs.

Creating and training a DRL agent is an iterative process of four basic steps:

1. The neural network structure is created and initialized.
2. The agent uses the observation to take an action.
3. The environment transitions based on the action and the agent receives a reward.
4. The network is trained to match the new reward values.

Steps 2 and 3 are repeated until a terminal state is reached. Once a terminal state is reached, the environment is reset and the agent continues taking actions. While the training is taking place, the network continues to learn the approximate policy that maximizes the total reward. This iterative process continues until a terminal state is reached in the environment, where the final episode reward is summed and the simulation environment is reset for the next training episode. The network continues training until the agent achieves an acceptable reward or the training has exceeded the computational budget. A depiction of the reinforcement learning process can be seen in Figure 2.

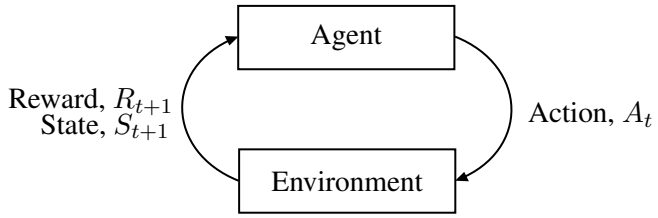

Figure 2: A graphical representation of reinforcement learning where an agent iteratively takes actions in the environment, receiving state information and a reward.

2.2.2. Deep reinforcement learning for nuclear O&M

Deep reinforcement learning provides several advantages over classical optimization solvers. One such advantage is that the learning of the network happens once during training. This allows for the trained network to act as on online agent that will continuously monitor and adjust actions as needed, whereas classical algorithms must be solved for every change in state. Another advantage is that DRL has been shown to overcome challenges associated with solving partially observable systems and systems with uncertainty. This particular advantage will be especially helpful with maintenance decision making with uncertain component states. Deep reinforcement learning has been shown to handle the "curse of dimensionality" better than classical methods by creating a generalized approximation of the optimal policy. The deep neural networks used to approximate the optimal policy can handle large decision and action spaces and long forecast horizons, the combination of which normally results in computationally intractable solving requirements. The last advantage is the ability to handle non-linear dynamics. Much of the dynamics and resulting cost equations are non-linear and thus provide challenges for linear optimization methods.

To train a DRL agent for nuclear O&M, we have created a decision process environment that contains the following sub-models: component reliability, inspection and maintenance, resource availability, and business and costs. These models are integrated as a POMDP that can be simulated with decisions for cost analysis, creating the interactive environment for the DRL agent. The output of the environment is a vector containing the current observations. This observation vector is the input to the DRL agent, or in other words, the neural network. The output of the neural network is a probability assigned to each action that corresponds to its likelihood of it being the optimal action. The action with the highest likelihood is chosen as the optimal maintenance action. The chosen action is then implemented in the environment, where the process starts again. A representation of this process can be seen in Figure 3.

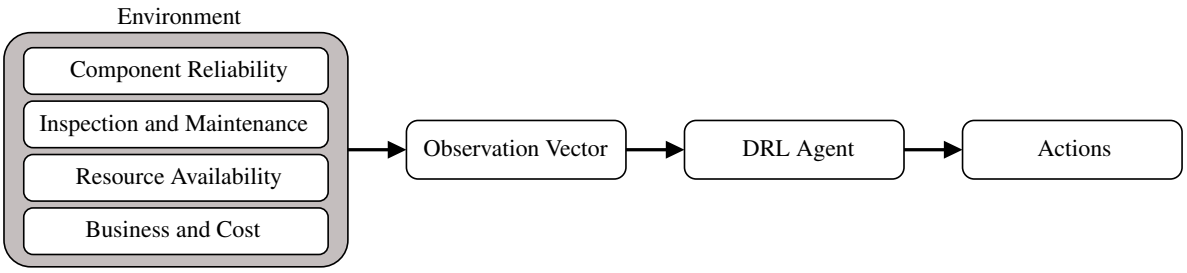

Figure 3: As part of this research, we will model the nuclear O&M decision making environment that includes several sub-models to train a DRL agent to take optimal actions given the current observations.

3. RESULTS

To demonstrate these methods, preliminary studies were conducted on a simple, representative maintenance system. In this study, we simulated the stochastic degradation of a component to test the ability of the decision-making tool to make optimal decisions and ensure a low lifetime cost.

3.1. Environment

For this preliminary study, the results present a single component system with inventory management. The degradation of this component simulates a stochastic degradation process, similar to many commonly studied wear-out degradation profiles [11,12]. The component's health is considered an observable feature for this component, subject to estimation uncertainty, and can be related to its remaining useful life. The component's life begins with full health near 1 and degrades randomly with process noise until the health reaches 0, where the component fails. Realizations of the component's stochastic degradation profile can be seen in Figure 4.

The component's repair and replacement costs are subject to the plant's current operating condition, which can be normal operation or planned shutdown. The periods of operation and shutdown correspond to an 18-month refueling cycle that is typical for a nuclear power plant. During operation, if the pump fails or needs maintenance, the plant must derate or reduce a fraction of the operating capacity, accruing a negative cost. The component may be repaired or replaced to bring the component back online. If the component is repaired, a fraction of the component is restored and the component can continue operation. If the component is replaced, the component is restored to full health and the component can continue operation. However, the component may only be replaced if there is available inventory. The inventory decisions (i.e. order replacement or do nothing) are also available to the agent during every decision step. When the agent orders a component, a fixed, 5-month lead time predates the arrival of the new component. When the spare component arrives into inventory, there is a storage cost for each month the component remains in storage. Having an extended lead time and storage costs requires the agent to perform just-in-time inventory management to minimize overall costs.

Once completed, the component, actions, and costs were then encoded into a MATLAB reinforcement learning environment, where they could be simulated with nuclear maintenance schedules and constraints.

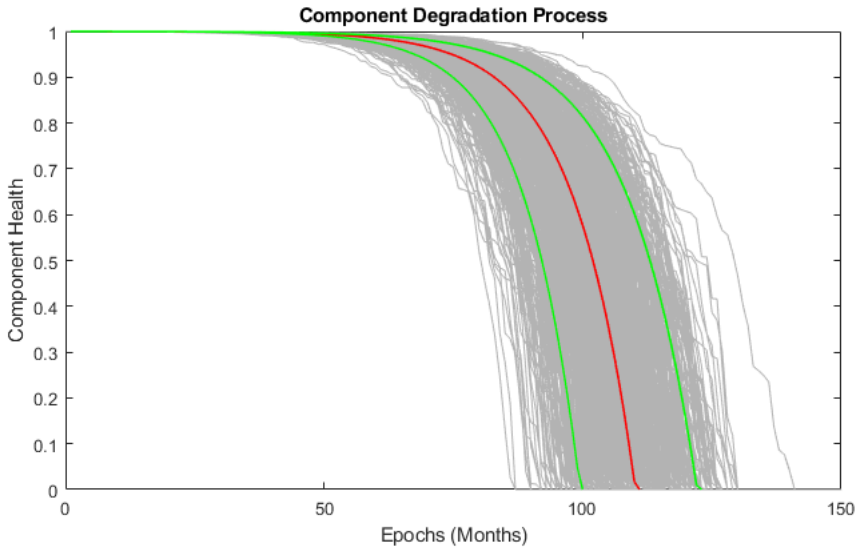

Figure 4: Realizations of the component degradation process, plotted with the mean and ± 1 standard deviation.

3.2. Agent

An actor-critic network was then developed as the agent. Each actor and critic network consisted of two fully connected hidden layers. Each layer is fully connected to the previous layer, where each neuron is represented as a Rectified Linear Unit (ReLU) activation function. The input to the agent was the state vector and the output was the probability distribution over available actions and the value approximation.

The input to the network is the observation or state of the system, organized as a normalized vector, known as the state vector. The state vector is comprised of five inputs to the network: the health of the component, the months until the outage, the amount of inventory, an inventory availability flag (1 or 0), and the remaining lead-time of an ordered component.

The output of the network is the decision for that time step. The possible maintenance decisions were to do nothing, repair, or replace, while the inventory management decisions were to order or not order a new component. For the agent to be able to make a simultaneous maintenance decision and inventory management decision, six decision vectors were created that contained all possible decision combinations. Therefore, the output of the network contained six nodes corresponding to the available decision combinations.

3.3. Training

Several DRL algorithms were tested, but the best performing agent was found when using the Advantage Actor Critic (A2C) algorithm [13]. The agent was trained numerous times with different parameter combinations to identify the best agent. During each training run, the network was trained for over 50,000 training episodes, each consisting of 200 decision steps (months). The best agent of each training run was then tested for its ability to make optimal decisions and minimize long-term costs.

The best agent was found when using a discount factor of $\gamma = 0.999$ and a learning rate of 10^{-5}. A high discount rate was used so that future rewards were given more importance. An ϵ-greedy exploration strategy with a value of $\epsilon = 0.001$ was used to allow the agent to sufficiently explore the state-action space and to ensure local optimums could be avoided. The best agent was able to be converge to a near-optimal solution that minimized the expected lifetime cost for the component. Results from the training episodes can be seen in Figure 5.

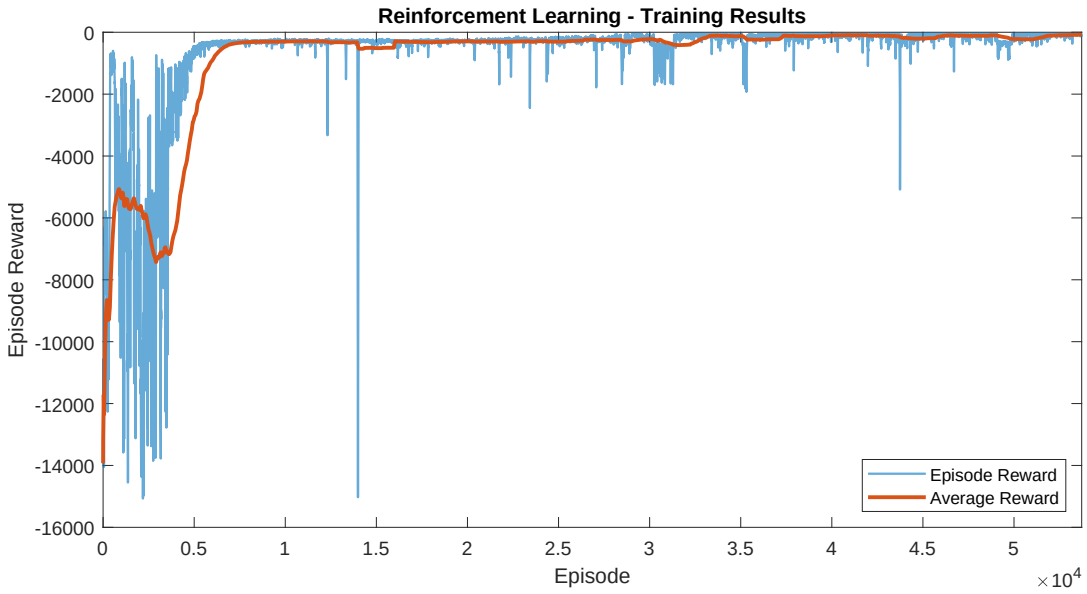

Figure 5: Deep reinforcement learning training results showing policy convergence during the best training run that lasted over 50,000 episodes.

3.4. Deployment Results

Using the trained agent, we were able to simulate the system and the decisions for 200 decision steps (months). This length of simulation corresponds to over 16 years of operation and includes planned refueling periods that occur every 18 months. The typical life of a nuclear plant can extend to 40 or even 60 years with license extensions; however, 200 months was used to best visualize the results of the decision making. The results of the trained agent in a simulation can be seen in Figure 6.

From the figure, we can see that the agent chose to do full two full repairs to the components during months 36 and 148. The agent chose to repair the component when the health was getting to levels that were deemed too risky to wait longer. The inventory management decisions also worked as expected, only ordering parts as needed and just before the outage replacement. The costs incurred by the agent can be seen in the bottom of Figure 6. These costs were the result of ordering, storing, and then the cost to install the new component. The total lifetime cost of that component is the sum of each month's costs.

3.5. Baseline Performance Comparison

To evaluate the success of the agent, we have established a performance metric by which we can measure success. To do this, we have created several baseline maintenance strategies to evaluate the performance of the agent through comparison of current maintenance strategies. We have developed a preventative maintenance strategy that emulates time-based maintenance programs that repair components periodically, regardless of the component's condition. This type of maintenance program schedules inspections and repairs at predefined intervals before the components become unreliable in an effort to prevent unexpected failures. These intervals are often determined using historical, component-specific data from the plant and broader nuclear industry. Due to the lack of real-time component health data, the time between maintenance intervals are usually conservative to minimize the risk of unplanned shutdowns.

For this analysis, we have developed five time-based maintenance strategies. Each of the strategies are different maintenance intervals, replacing the component every n outages. The selected intervals were three, four, five, six, and seven outages. For component replacements, the inventory management ordered a new component $t = 5$ months before the beginning of the scheduled outage. The component was also ordered well before the replacement interval to ensure the component arrived and was received into inventory before

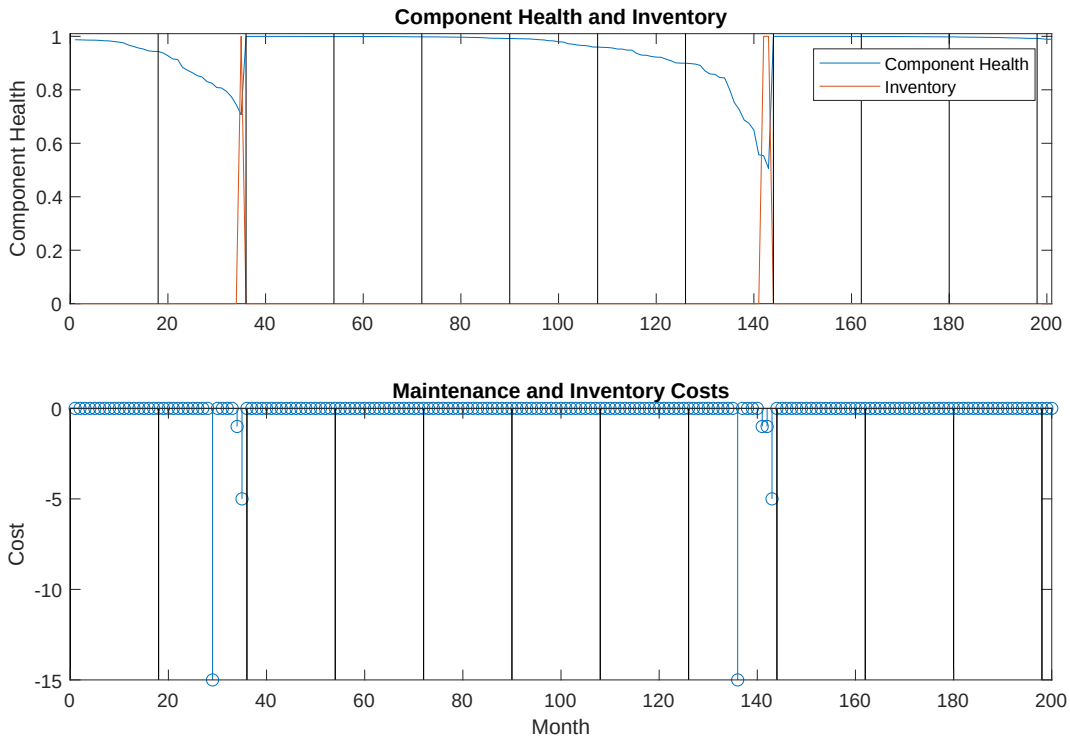

Figure 6: After training, we can simulate an instance of a component degradation process with effects of actions taken by agent. The top plot shows the component degradation and the bottom plot shows costs.

the replacement procedure was scheduled to begin. Once the strategies were created, they were simulated thousands of times for a period of 60 years. The DRL agent was able to reduce the expected lifetime cost when compared to the best time-based maintenance strategy by 53%. The results of the simulations can be seen in Figures 7 and 8.

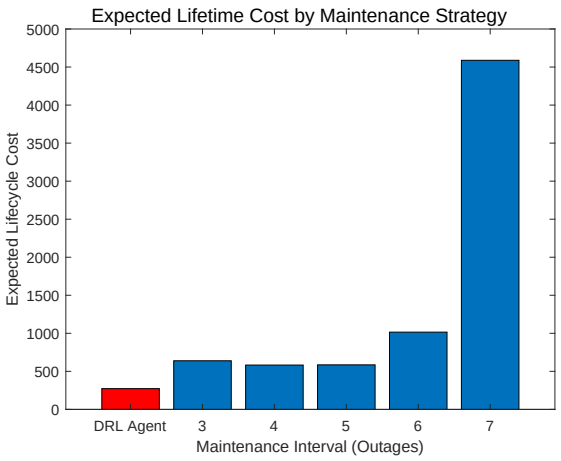

Figure 7: Using the trained DRL agent, we can simulate a realization of a component degradation process with effects of actions taken by the agent.

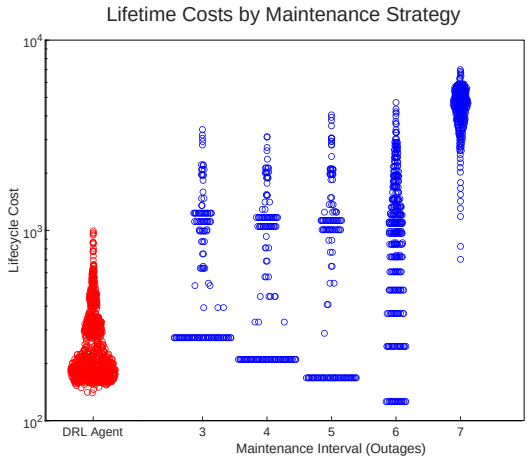

Figure 8: Swarm plot (semi-log) showing costs during Monte Carlo simulations for each maintenance strategy. Each simulation had a 60-year duration.

3.6. Uncertainty Analysis

When undergoing modeling and control, it is well known that the uncertainty of the modeling parameters and observability must be studied to understand their effects. This statement remains of the utmost importance in the nuclear industry due to strict operating requirements and the need for high assurance of critical infrastructure. Studying the effect of observability uncertainty will help quantify how the reinforcement learning policy will behave under different conditions than were experienced during training.

The observability of this system was chosen as the parameter of study to simulate the effects of an uncertain sensor observation, a common point of uncertainty in condition monitoring and prognostics. To study the observability of this system, we added zero-mean Gaussian noise to each observation, changing the magnitude of added noise by increasing or decreasing the amount of variance in the noise. Since the noise is zero-mean and only added to the observation, the degradation process and the cost of the optimal decision path remains unchanged. The magnitude of the variance was studied from a range of 10^{-6} to 10^{-1} and compared to perfect observability where the variance had a magnitude of 0. Large amounts of uncertainty that would be considered unrealistic for condition monitoring systems will not be studied. Therefore, the maximum amount of variance studied is 10^{-1}. The results from the uncertainty analysis can be seen in Figure 9.

When compared to perfect observability, the added observation noise does not provide significant increases to expected lifetime costs when equal to or less than a magnitude of 10^{-2}. As long as noise remains below that threshold, the expected lifetime costs do not change by more than 5% when compared to perfect observability. If the noise variance reaches a magnitude of 10^{-1}, the expected lifetime costs increase by almost 35%. This significant increase in cost is due to the large amount of uncertainty and estimation error of the current health of the component when in the presence of noise variance with large magnitudes. This large amount of estimation error and uncertainty leads to the DRL agent making overly conservative and sub-optimal decisions, resulting in an increase in costs. Although the expected costs of using the DRL agent increase with noise, the costs were still significantly lower when compared to the time-based maintenance strategies. Even with significantly large levels of estimation uncertainty, the DRL agent performs significantly better than the time-based maintenance strategies.

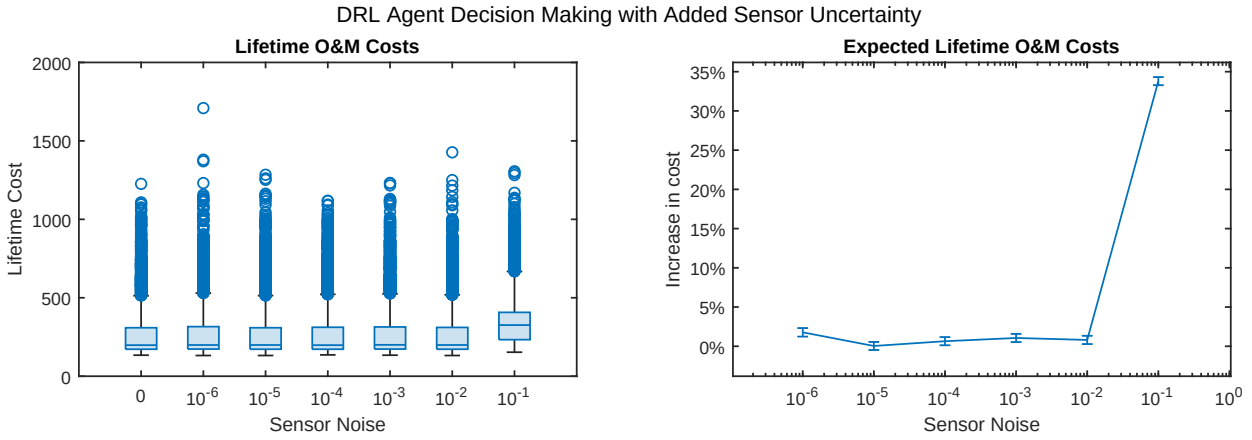

Figure 9: The addition of sensor uncertainty can impact the DRL agent's ability to make decisions. The plot on the left shows the results of the Monte Carlo simulations and the distributions of cost, as a result of increasing sensor noise amplitude. The plot on the right shows the increase in expected lifetime costs, with standard square error of mean and normalized to zero sensor noise, as a result of increasing sensor noise amplitude.

4. CONCLUSION

In this paper, we have discussed and analyzed novel methods that can be used for decision making in nuclear O&M. Through baseline performance comparisons we have shown that deep reinforcement learning is

a good alternative to time-based maintenance strategies, and that it provides significant robustness from uncertain observations. Although observation uncertainty does increase the expected costs, the DRL agent was able to make near-optimal decisions in the presence of high amounts of noise and outperformed time-base maintenance strategies. These results provided significant assurance that cost reductions can be achieved by using deep reinforcement learning for condition- and risk-based decision making.

ACKNOWLEDGEMENTS

This work was supported in part by the following projects: NRC grant number 31310018M0048, University of Pittsburgh Nuclear Engineering Graduate Fellowship Program; and U.S. DOE Office of Nuclear Energy's Nuclear Energy University Program DE-NE0008909 under the Nuclear Energy Enabling Technologies Advanced Sensors and Instrumentation Program.

REFERENCES

[1] WNA. "Nuclear Power Economics — Nuclear Energy Costs - World Nuclear Association." (2020).

[2] P. G. Morato, K. Papakonstantinou, C. Andriotis, J. S. Nielsen, and P. Rigo. "Optimal inspection and maintenance planning for deteriorating structural components through dynamic Bayesian networks and Markov decision processes." *Structural Safety*, **volume 94**, p. 102140 (2022).

[3] P. G. Morato Dominguez, J. S. Nielsen, A. Q. Mai, and P. Rigo. "POMDP based maintenance optimization of offshore wind substructures including monitoring." In *13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13)* (2019).

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. "Playing atari with deep reinforcement learning." *arXiv preprint arXiv:13125602* (2013).

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature*, **volume 529**(7587), pp. 484–489 (2016).

[6] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. "Dota 2 with large scale deep reinforcement learning." *arXiv preprint arXiv:191206680* (2019).

[7] C. P. Andriotis and K. Papakonstantinou. "Managing engineering systems with large state and action spaces through deep reinforcement learning." *Reliability Engineering & System Safety*, **volume 191**, p. 106483 (2019).

[8] C. P. Andriotis and K. G. Papakonstantinou. "Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints." *Reliability Engineering & System Safety*, **volume 212**, p. 107551 (2021).

[9] R. Rocchetta, L. Bellani, M. Compare, E. Zio, and E. Patelli. "A reinforcement learning framework for optimal operation and maintenance of power grids." *Applied energy*, **volume 241**, pp. 291–301 (2019).

[10] S. Wei, Y. Bao, and H. Li. "Optimal policy for structure maintenance: A deep reinforcement learning framework." *Structural Safety*, **volume 83**, p. 101906 (2020).

[11] C. Letot and P. Dehombreux. "Reliability assessment from degradation models." (2010).

[12] C. Letot and P. Dehombreux. "Dynamic reliability degradation based models and maintenance optimization." In *Proc. 9th National Congress on Theoretical and Applied Mechanics (NCTAM), Brussels, Belgium*, pp. 1–9. Citeseer (2012).

[13] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning." (2016).