Adaptative Reinforcement Learning Approach for Predictive Maintenance of a Smart Building Lighting System

Anas HOSSINI
Lineact CESI, DAVID Lab
Paris Saclay University
Nanterre, France
ahossini@cesi.fr

Leïla KLOUL
DAVID Lab
Paris Saclay University
Versailles, France
leila.kloul@uvsq.fr

Maël GUIRAUD
Lineact CESI
Nanterre, France
mguiraud@cesi.fr

Benjamin COHEN BOULAKIA
EFREI Research Lab
Villejuif, France
benjamin.cohen-boulakia@efrei.fr

2024 IEEE 22nd International Conference on Industrial Informatics (INDIN) | 979-8-3315-2747-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/INDIN58382.2024.10774509

***Abstract*—Due to advancements in sensing technologies, enhanced IoT architectures, and expanded connectivity options, predictive maintenance has emerged as a compelling solution within the context of Industry 4.0 for industrial systems. However, within this landscape, such as in Smart Buildings (SBs), the lack of failure data poses a significant challenge for implementing traditional data-based approaches documented in the literature. Additionally, SBs are complex systems of systems, where failures in one subsystem can propagate and impact other interconnected systems, adding layers of complexity to maintenance decision-making processes. Lastly, predictive maintenance has traditionally been used primarily in the context of operational safety, with the primary goal of avoiding undesirable events at all costs. Only recent work focuses on predictive maintenance to optimize operating costs. In light of these challenges, this paper proposes a Reinforcement Learning approach for Predictive Maintenance tailored for the lighting system within a Smart Building. It aims to minimize costs subject to a Quality of Service constraint by leveraging a Markov Decision Process and Q-learning adaptation, which adjusts based on the life cycle of system components. The results demonstrate that, even with limited system data, meeting defined quality of service standards and minimizing maintenance costs is feasible.**

***Index Terms*—Predictive Maintenance, Smart Buildings, Markov Decision Process, Reinforcement Learning and Reliability.**

I. INTRODUCTION

Predictive maintenance generally involves monitoring conditions, diagnosing and predicting failures, and implementing maintenance strategies [1]. Current technologies can detect, identify, and isolate potential component failures in a machine, monitor and predict the progression of faults, and provide decision support for developing maintenance programs.

According to [2], three approaches can be distinguished for predictive maintenance (PdM): those based on physical models, those based on knowledge, and those based on data.
Physical models rely on a studied system's physical properties, such as its thermodynamics [3] or vibration analysis [4]. These approaches are less prevalent in the literature, as they cannot be generalized and require detailed knowledge of the physical properties of the studied systems [5].
Knowledge-based approaches use knowledge bases, usually from experts, to solve complex problems by inferring new reasoning, typically in logical assertions. Despite some limitations, such as the need for an expert to design a knowledge base and to validate the obtained solutions, as well as potential struggles with scalability and adaptability, these approaches are still explored in the literature due to their advantages, such as automating human intelligence and being easy to integrate, share, and reuse [6], [7].
Data-based approaches are currently the most commonly used for PdM, as summarized in [8], [9]. Studies conducted by [10] summarize various factors that have contributed to the development of PdM practices in data-based approaches:

- Data acquisition through IoT.
- Storage and retrieval of data through Big Data.
- Fault diagnosis and prognosis through Deep Learning.
- Decision-making through Reinforcement Learning.

Considering these advancements, several failure prediction models have been proposed for general use cases [11], [12]. In Smart Buildings (SB), most predictive maintenance approaches are data-based, mainly due to the development of supervised and unsupervised machine learning algorithms [13], [14].
However, Data-Based approaches may face challenges. Smart buildings are generally not instrumented to capture failure data due to high instrumentation costs, offsetting the benefits PdM could bring. Given the lack of failure data, stochastic discrete event systems play an increasingly important role in reliability engineering. These approaches model system degradations, often using Markov processes [15], [16]. However, designing stochastic discrete event systems presents known challenges, mainly the difficulty of validation due to several possible executions. This issue can arise due to stochastic delays, often utilized to simulate packet losses and data scheduling, a common occurrence in connected vehicle systems [17]. The generally proposed solution to address this issue is the implementation of Model Checking to algorithmically verify whether a given model, the system itself, or an abstraction of the system satisfies a specification, often formulated in terms

of temporal logic [18].

Most of the proposed solutions for PdM in SBs suggest maintenance strategies to maximize system reliability and availability based on predictive failure models (traditional machine learning approaches and heuristics in most cases [10], [19]).

Using Reinforcement Learning (RL) algorithms that leverage predictive models to devise maintenance strategies can offer substantial advantages, thereby improving the efficiency and cost-effectiveness of maintenance operations within this context. Recently, this dimension has become a hot topic in the literature. RL-based multi-agent approaches, as proposed by [20], or optimization approaches from Operations Research [21], are most commonly used in industrial applications. For instance, they optimize financial costs related to multiple machines' operations or maximize machine availability. By allowing a system to learn from its experiences autonomously, RL approaches hold substantial potential for adjusting maintenance strategies based on the actual costs and performance of the building [22].

In our previous works [23], we proposed a fault prediction methodology in a single SB lighting system. Using a Fault Tree and the manufacturer's data – the Mean Time to Failure (MTTF) metric– to build the failure probability distribution for each system component, we were able to characterize and generate statistics on the impacts of a maintenance operation on the system and its components for different intervention scenarios.

This paper aims to use this model to minimize maintenance costs using a RL-based approach for PdM applied to a SB Lighting System. The RL algorithm we propose relies on a reward assignment based on maintenance costs and integrates the QoS constraints of the predictions obtained.

To make the best maintenance decisions based on the system's operating state, we use the events (Basic and Intermediary) of the Fault Tree to define RL agents. Each agent makes maintenance decisions related to a specific system component. Agent decisions are determined by a Markov Decision Process (MDP) when a system component is in its useful life phase (constant failure rate). Otherwise, agents make decisions without knowledge of the environment, i.e., using an exploration and exploitation approach ($\epsilon - greedy$ algorithm). This approach enables the determination of the most cost-effective maintenance actions to be carried out over a given simulation period, considering compliance with a user-defined QoS and the manufacturer's maintenance costs for each system's component.
The structure of this paper entails a presentation of the SB Lighting system in Section 2. The PdM framework is detailed in Section 3. The numerical results of the proposed PdM approach applied to the lighting system are covered in Section 4, and a conclusion is given in Section 5.

II. Case Study: Nobel Room Lighting System

The Nanterre 3 (NR3) Smart Building is a two-storey structure affiliated with CESI-Nanterre, an Engineering School in Paris, France [24]. This smart building features an *electrical room* housing the centralized electrical system, a *server room* centralizing the computer system, four multipurpose rooms and one office. Each room has a lighting system and an HVAC (Heating, Ventilation, and Air Conditioning) system. Both systems consist of physical and logical components, incorporating multiple sensors capable of interacting with each other and with sensors in other rooms.
This paper examines the lighting system within a specific room (the Nobel Room). Figure 1 provides an overview of the general architecture of this room, where various interactions are highlighted. The Nobel Room's lighting system comprises:

- **Light fixtures**: There are 6 of them, each representing a block composed of 12 bulbs, a temperature sensor (°C), an infrared sensor (IR), and a Lux meter sensor (Lux)
- **Internal switch (Cisco 3650CX)**: It interacts with all the sensors in the room, providing Power Over Ethernet energy to the lighting fixtures.

In the NR3 SB, the network system manages the lighting system. We integrate the network system components that affect the lighting system into the lighting system modeling. These components are:

- **Floor switch (Cisco 2950X)**: It communicates with the room's internal switch, the server, and the electrical system.
- **Cisco's MicroGrid LT**: represents the set of power cables connecting the Cisco 3650CX and Cisco 2950X.
- **MicroGrid for low voltage (MicroGrid LT):** There are 6 of them, each linked to a Light Fixture to ensure Power over Ethernet supply.

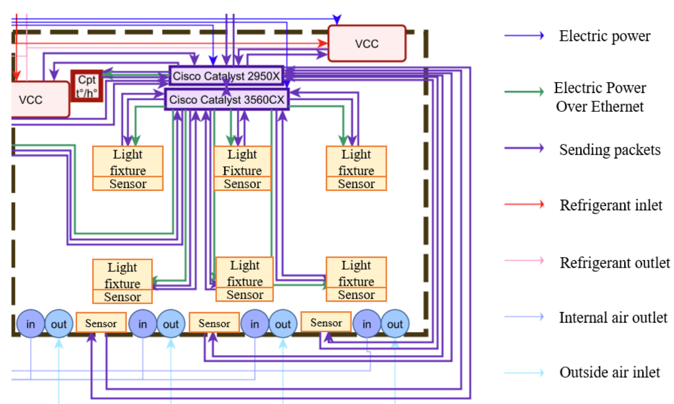


Fig. 1. Architecture of the Nobel Room

III. Predictive Maintenance Framework

We introduce a PdM model for a given system. This model requires the knowledge of:

- The operating status of each room's lighting system component.
- The impact of each component on the operation of the room.

- The room operating status after a maintenance operation has occurred.

The following subsections present the techniques that will help us meet the above requirements.

A. Fault Tree and Weibull Distribution

The Fault Tree (FT) Analysis is a technique that employs a hierarchical tree structure to depict elementary events (root causes of failures) and their interconnections, culminating in the manifestation of a dreaded event, such as the failure of the entire system. These combinations of events are established using logical gates. This method offers the advantage of quantifying the probability of the occurrence of an undesirable event — in our case, the failure of the lighting system. Additionally, it helps pinpoint critical paths, which are the shortest routes leading to the occurrence of this event. The Fault Tree diagram for the Nobel room lighting system is shown in Figure 2. The critical event is: *the lighting system does not work (the bulbs do not light up)* and it can be due to one of the following failures:

- The failure of all the bulbs in the room.
- The failure of all the IR sensors.
- The failure of all the Lux sensors.
- The failure of the Cisco 3650 CX.
- The failure of the Cisco 2950X.
- The failure of the Cisco's MicroGrid LT.
- The failure of all the MicroGrid LT.

The last three points refer to component failures within the network system that affect the lighting system. These components are basic events depicted in dashed lines in Figure 2. Note that in the network system, the operational state of these components solely has an impact on the lighting system.

In this paper, we suppose that the failure of each system component follows the Weibull distribution (α, β). Indeed, several operational safety studies outline the lifecycle of a component by delineated it into three phases according to the parameter β of the Weibull distribution (α, β). Consequently, when an intervention is conducted on a component, its operational status improves. We assume this enhancement can be represented as a decrease in the β value.

B. Reinforcement Learning model

In the RL approach adopted, we distinguish between two types of agents (see Figure 2):

- **Basic Agent (BA):** is an agent that represents a basic event (elementary event) in the FT.
- **Intermediate Agent (ItA):** is an agent that represents an intermediate event in the FT that can be *maintainable*. An intermediate event in the FT is said to be *maintainable* if it results from a logical combination that includes basic events **and** if it is possible to apply a maintenance action to it.
  - A Bulb Block Failure is an intermediate event that is *not-maintainable* because we can't apply a maintenance action on the Bulb Block, so an Intermediate Agent does not represent it.
  - A Light Fixture Failure is an intermediate event *maintainable* because we can apply a maintenance action to the Light Fixture, so an Intermediate Agent represents it.

Besides the agent's formulation, we define:

- **Local Minimum Cut Set** ($LMCS^{ita}$) **of an Intermediate Agent** ita: it is the set of events (basic or intermediate) whose failure implies the failure of the intermediate event represented by the Intermediate Agent ita. For example, the $LMCS^{LF1}$ ($LF1$ refers to *Light Fixture Failure 1* in Figure 2) are:
  - MicroGrid LT Failure 1
  - IR Sensor Failure 1
  - Lux Sensor Failure 1
  - (Bulb Failure 1,..., Bulb Failure 12): Simultaneous failure of all its bulbs
- **A Basic Agent Parents** ($Parent^{ba}$): refers to the set of maintainable parents of a BA. For example, the BA Bulb 1 set of Parents is {Bulb Block Failure 1, Light Fixture Failure 1}, but the Bulb Block Failure 1 is not maintainable. Hence, $Parent^{Bulb1}$ = {Light Fixture Failure 1}.

1) <u>***Markov Decision Process***</u>:

An MDP is used to model the sequential decision-making of basic and intermediate agents. Since failures can occur randomly over a given time interval, and the maintenance costs for each component can change over time, we adopt a Continuous-Time Markov Chain (CTMC).

We present the states, actions, transitions, and rewards for basic agents.

- <u>**BA States:**</u> The set of states of a BA is represented by $S^{ba} = \{Functional(F), Faulty(\overline{F})\}$.

- <u>**BA Actions:**</u> represent the types of maintenance actions to be performed when the agent is in state $s \in S^{ba}$. The agent can choose one of the following actions:
  Local Replacement: When the BA reaches a particular state, it performs a replacement. In other words, the decision to perform a replacement action is solely based on his decision.
  Global Replacement: The transitions depend on the maintenance action decided by the BA's parents.
  The set of actions is represented by a set of vectors A^{ba} such that $\overrightarrow{a} \in A^{ba}$ is defined by $\overrightarrow{a} = (a_i)_{i \in \{0,1\}}$, where:
  - a_0 = Local Replacement (LR).
  - a_1 = Global Replacement (GR).

- <u>**BA transitions:**</u> Figure 3 shows the possible transitions for a BA. The transition from F to $\overline{F}$ is due to :
  - A **failure** with a rate h^{ba}.

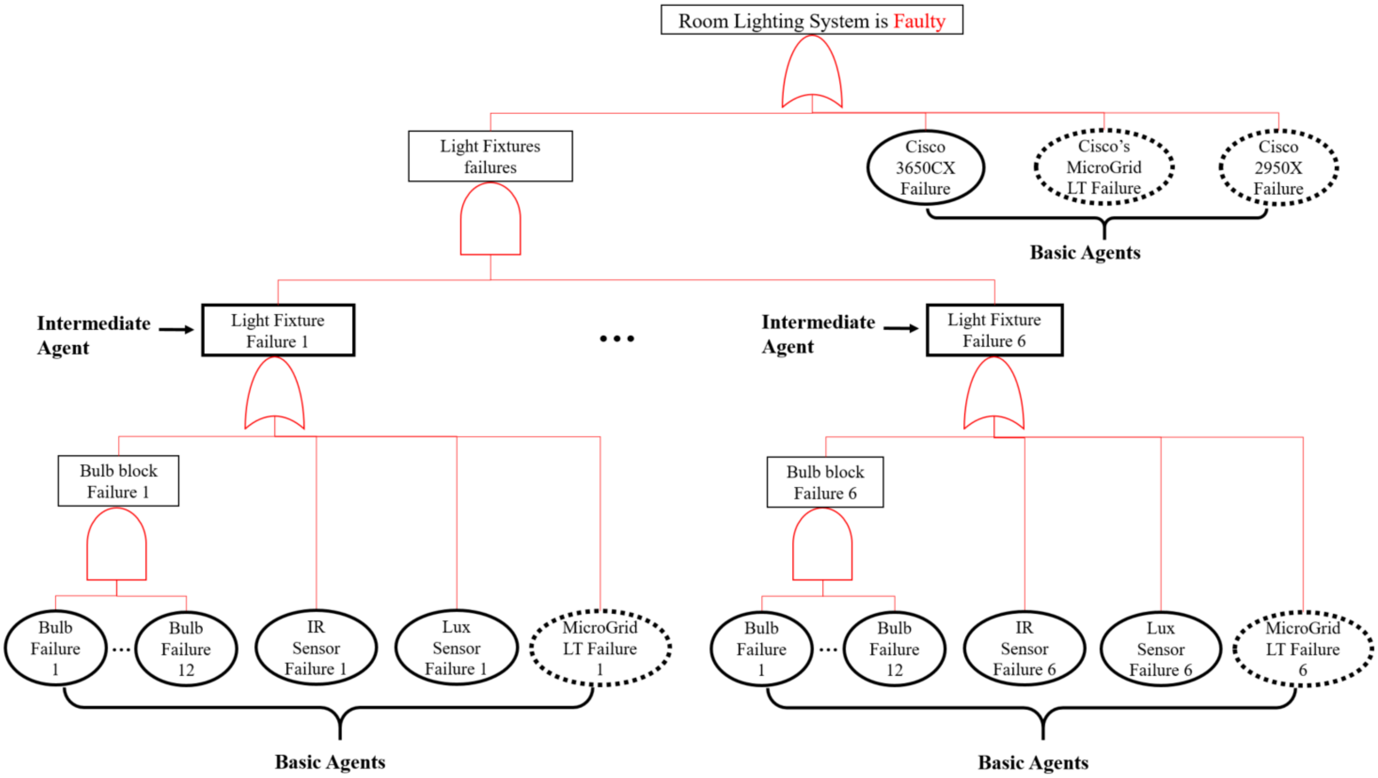


Fig. 2. Nobel Room Lighting System's Fault Tree

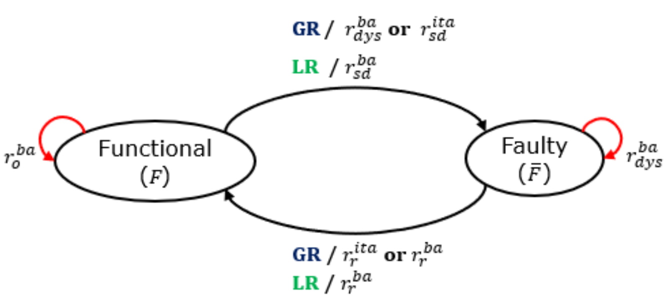


Fig. 3. Basic Agent CTMC

- A **shutdown** with a rate μ_{sd}^{ba} if the BA action is a_0 (LR) and the BA parent action is a_1.
- A **shutdown** with a rate $\mu_{sd}^{p,ba}$ if the BA action is a_1 (GR) and the BA parent action is a_0.

The transition from $\overline{F}$ to F is due to :

- A **replacement** with a rate μ_{rep}^{ita} if the BA parent action is a_0 (GR due to the parent action).
- A **replacement** with a rate μ_{rep}^{ba} otherwise (LR of the component).

• **BA Reward:** Recall that we associate a reward with each transition, modeling the gain or loss it induces in terms of maintenance costs. Figure 3 shows the reward received by the agent when a transition occurs associated with a maintenance cost, such that:

- ☐ $r_o^{ba} = c_o^{ba}$: BA's operating cost.
- ☐ $r_{dys}^{ba} = -c_{dys}^{ba}$: BA's failure cost.
- ☐ $r_{sd}^{ba} = -c_{sd}^{ba}$: BA's shutdown cost ($r_{dys}^{ba} < r_{sd}^{ba} < 0$).
- ☐ $r_{sd}^{ita} = -c_{sd}^{ita}$: ItA's shutdown cost.
- ☐ $r_r^{ba} = c_r^{ba}$: BA's return to service cost.
- ☐ $r_r^{ita} = c_r^{ita}$: ItA's return to service cost.

We present the states, actions, transitions, and rewards for intermediate agents.

• **ItA States:** The set of states of an ItA is noted $S^{ita} = \{Functional(F), Degradation(Deg), Faulty(\overline{F})\}$, where.

- $Functional$: The intermediate event represented by the ItA is functional (all its components work).
- $Degradation$: The intermediate event represented by the ItA does not function perfectly; this is the state in which one or more of its non-critical children fail.
  Let $Child^{ita}$ be the set of all the children of intermediate event ita in the Fault Tree. ItA is in the degradation state if: $\exists x \in Child^{ita} \backslash \{LMCS^{ita}\} : x$ is faulty.
- $Faulty$: The intermediate event represented by ItA is faulty, i.e., $\exists x \in \{LMCS^{ita}\}/x$ is faulty.

- **ItA Actions:** represent the types of maintenance actions to be performed when the agent is in state $s \in S^{ita}$. The agent can choose one of the following actions: ***Local Replacement*** or ***Global Replacement***.
  The ***Local Replacement*** action is solely based on the ItA decision (similar to the BA case).
  The decisions that improve the state of the ItA when the ***Global Replacement*** action is chosen depend either on the ItA's children or the maintenance actions of the maintainable ItA's parent maintenance actions. As a result, the ItA agent can either perform maintenance or wait for one or more global changes to improve its state.
  The set of actions is represented by a set of vectors A^{ita} such that $\overrightarrow{a} \in A^{ita}$ is defined by $\overrightarrow{a} = (a_i)_{i\in\{0,1\}}$ where:
  - a_0 = Local Replacement (LR).
  - a_1 = Global Replacement (GR).

- **ItA transitions:** Figure 4 shows the possible transitions and the reward obtained for an ItA according to each action.

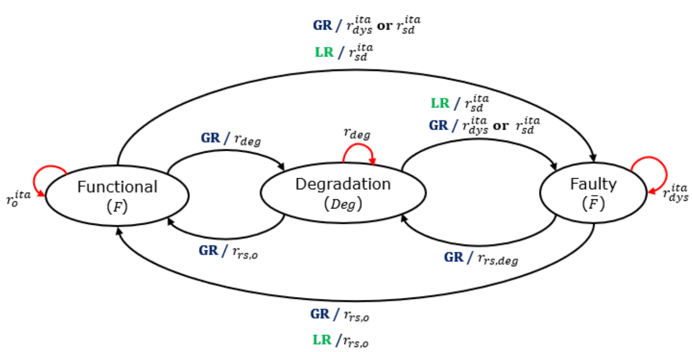


Fig. 4. Intermediate Agent CTMC for both actions

As the FT (see Figure 2) includes only OR and AND gates, the failure probability of the ItA involves a combination of union and intersection operations of failure events of its children. We assume that all basic events are independent, meaning the failure of one component is not dependent on another. We can use Poincaré's formula [25] to express the hazard function of the Union of events A_i: see Eq.1 and the reliability and the hazard function of the intersection of events A_i: see Eq.2, 3 such that:

$$h_{\bigcup_{i=1}^{n} A_i}(t) = \sum_{i=1}^{n} h_{A_i}(t) \tag{1}$$

$$R_{\bigcap_{i=1}^{n} A_i}(t) = 1 - \prod_{i=1}^{n}(1 - R_{A_i}(t)) \tag{2}$$

$$h_{\bigcap_{i=1}^{n} A_i}(t) = -\frac{\frac{\partial R_{\bigcap_{i=1}^{n} A_i(t)}}{\partial t}}{R_{\bigcap_{i=1}^{n} A_i}(t)} \tag{3}$$

(Due to page limitations, the details of Eq.1,2 and 3 have been omitted).

We define:
- Z^{ita} = $Child^{ita} \backslash \{LMCS^{ita}\}$
- K_i = The event that describes that element i is faulty

Details of the transitions in Figure 4 are as follows:
- ***Functional* to *Degradation*:** this transition occurs if one or many non-critical ItA's children are faulty, and the ItA's parent does not choose to perform a Local Replacement with a rate $h_{\bigcup_{i=1}^{|Z^{ita}|} K_i}$.
- ***Functional or Degradation* to *Faulty*:** if the ItA's parent chooses to perform a Local Replacement, the ItA is shutdown with a rate μ_{sd}^{ita}. Otherwise, this transition occurs if one or many critical ItA's children are faulty with a rate $h_{\bigcup_{i=1}^{|LMCS^{ita}|} K_i}$.
- ***Faulty* to *Degradation*:** this transition occurs according to the replacement rate of the $LMCS^{ita}$ and only if the action chosen by the ItA is a_1 with a rate $\max\limits_{i \in LMCS^{ita}} (\mu_{rep}^{i})$.
- ***Faulty* to *Functional*:** if the action chosen by the ItA is a_1, this transition occurs according to either the replacement rate of the $Child^{ita}$ with a rate $\max\limits_{i \in Child^{ita}} (\mu_{rep}^{i})$ or the replacement rate of the $Parent^{ita}$ with a rate $\mu_{rep}^{P^{ita}}$.
  If the action is a_0, this transition occurs according to the replacement rate of the ItA with a rate μ_{rep}^{ita}.

- **ItA Reward:** Figure 4 shows the reward received by the agent when a transition occurs, such that:
  - ☐ $r_o^{ita} = c_o^{ita}$: ItA's operating cost.
  - ☐ $r_{deg} = -c_{deg}$: ItA's degradation cost.
  - ☐ $r_{dys}^{ita} = -c_{dys}^{ita}$: ItA's failure cost ($r_{dys}^{ita} < r_{deg} < 0$).
  - ☐ $r_{rs,o} = c_{rs,o}$: ItA's return to functional service cost.
  - ☐ $r_{rs,deg} = c_{rs,deg}$: ItA's return to degraded service cost.

2) *Q-learning and MDP adaptation*:

MDP is a framework that can be employed in both model-free and model-based RL. Model-free methods directly learn policies or value functions, while model-based methods build a model of the environment to facilitate planning and decision-making.
This paper uses MDP as a model-based approach in RL mainly because when the agent is in the useful life, MDPs provide a clear and explicit representation of the transition dynamics between states, capturing how the system evolves in response to actions. This enables a better understanding of the environment.
Q-learning is an RL algorithm to find the optimal action-selection policy for a finite MDP. As a model-free algorithm, the agent has no pre-existing information about states, actions, state transitions, or associated rewards from the MDP. The

agent must explore the environment iteratively and learn through trial and error until it has explored enough to converge toward a Q-function that accurately reflects the values of state-action pairs in the given environment; exploration may be adapted to favor more exploitation of acquired knowledge.

Since a Markov process satisfies the Markov property, we apply an MDP method when the failure rate is constant, i.e., when the agent is in its useful life phase ($\beta \approx 1, h(t) = \frac{1}{\alpha}$) (this also preserves the Markov Chain homogeneity). Based on this, we define a $threshold$ above which the agent no longer remains in its useful life but begins to age and enters its wear-out phase. In other words, once the $threshold$ is exceeded, the β value of the Weibull distribution is increased. Hence, the component enters the wear-out phase.

When the basic agents are in the useful life phase, the MDPs can be formulated for each agent, and in this case, the optimal policy is obtained thanks to the MDP model. On the other hand, if only one BA is in its wear-out phase, the agent no longer has any knowledge of the environment (the MDP formulation is not applicable). In this case, a Q-Learning algorithm is applied to obtain the optimal policy.

Figure 5 illustrates two cases where the MDP method and Q-learning algorithm define maintenance actions.

- In case 1, the maintenance action is obtained by the MDP method since the agent did not reach the *threshold* ($t_1 < threshold$).
- In case 2, no maintenance action was decided in the Useful Life. Once the *threshold* is exceeded, the β value increases and maintenance actions are obtained using the Q learning algorithm.

IV. Numerical Results

As presented in Section I, to our knowledge, the research on predictive maintenance for SB, which seeks to determine the optimal maintenance actions in terms of costs under QoS constraints, has yet to be thoroughly explored. Therefore, in the absence of experimental data or theoretical convergence results of RL models, evaluating the performance of our algorithm in terms of costs is a full-fledged research work that will be addressed in future work. Thus, we focus this experimental study on the algorithm's ability to maintain the system in an acceptable operational condition given QoS constraints.

As we assume that the failure of each component of the system is given by the Weibull distribution (α, β), Table I shows the manufacturer's $MTTF_0$ values of each component and its corresponding Weibull parameters obtained by simulation [23].

The proposed model was tested on an NR3 Nobel room lighting system. In a simulation interval $[0, T_{sim}]$ with T_{sim} set by the user, we calculate and display the reliability $R(t)$. The parameters used in the simulations are summarized in Table II:

The exploration probability (ϵ) is the probability of choosing a random action over the action with the highest estimated reward. The number of episodes (N_{eps}) in Q-learning is chosen

TABLE I
Lighting System component's manufacturer values.

Component	$MTTF_0$ (days)	α_0	β_0
Bulb	3650	3650	1
IR Sensor	1825	1825	1
Lux Sensor	730	730	1
Cisco 3650CX Switch	10 950	10 950	1
Cisco 2950X Switch	12 000	12 000	1
MicroGrid LT	7300	7300	1
Cisco's MicroGrid LT	9125	9125	1

TABLE II
Simulation parameters

Parameters	Values
Simulation interval T_{sim}	3650 days
Exploration probability ($\epsilon - greedy$)	0.2
Number of episodes N_{eps}	20
Q-learning Learning rate η	$\frac{0.1}{1+0.3.N_{eps}}$
Q-learning iterations	100
Discount factor γ (Future rewards importance)	0.9

to ensure sufficient exploration of the state-action space. The algorithm uses a decay learning rate (η) to balance the weight given to new information versus existing knowledge. The discount factor (γ) is chosen to balance the importance of immediate and future rewards.

Figure 6 illustrates the evolution of the Lighting system and the Network system's components reliability $R(t)$ using a fixed reliability QoS constraint (75%) and different thresholds. The threshold in Figure 6.a for each component i is $0.9MTTF^i$ and $3MTTF^i$ in Figure 6.b.

According to the results obtained, the network system strongly influences the lighting system, ensuring QoS relies on enhancing the reliability of at least one component of the network system. For example, examining the Minimal Cut Sets (MCS) of the lighting system, we observe both single-component sets like {Cisco 2950X} and {Cisco's MicroGrid LT}, as well as larger sets like {MicroGrid LT 1, ..., MicroGrid LT 6}. Consequently, QoS adherence in the lighting system is notably contingent on meeting QoS standards for these smaller CutSets components, regardless of their associated costs.

As we adjust the threshold, replacement decisions are made at different intervals, consequently enhancing the reliability of the lighting system. Given that the MTTF of all network system components exceeds 20 years, adjusting the threshold has minimal impact on maintenance decisions for MCS of size 1 ({Cisco 2950X} and {Cisco's MicroGrid LT}), as maintenance actions are consistently determined when these components are within their useful life. Any observed differences stem from maintenance decisions for MCS of size 6 ({MicroGrid LT 1, ..., MicroGrid LT 6}), where maintenance actions are decided by their associated ItA. Thus, threshold adjustment exerts a more pronounced impact on light fixtures, as depicted in Figure 7. This discrepancy arises because the light fixtures' components possess significantly shorter MTTF than the network system's. Consequently, the transition from the

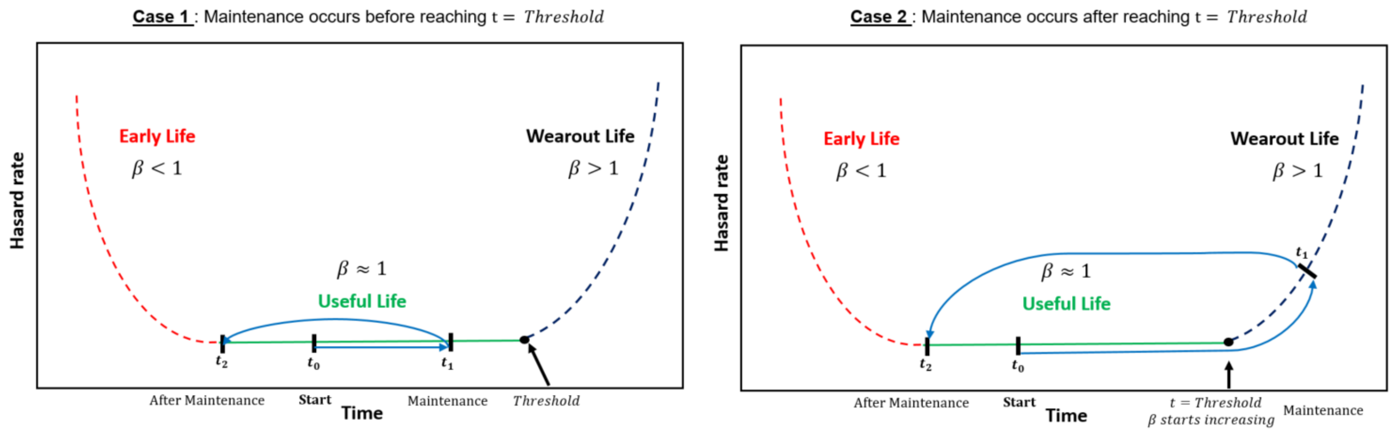


Fig. 5. Q-learning and MDP adaptation

useful life phase to the wear-out phase of a component occurs more rapidly, resulting in varied maintenance frequencies.

Furthermore, two types of reliability improvement in Light Fixtures are observed. When reliability reaches 100 %, it indicates that maintenance has been conducted on the entire Light Fixture, implying that ItA made the maintenance decision. Conversely, situations where light fixture reliability only partially improves, such as in the case of Light Fixture 1 in Figure 7.a at time 1900 days, occur due to the maintenance of one or more of its components (children), with the maintenance decision made by basic agents.

Thus, maintaining the MCS of size one from the network system {Cisco 2950X} and {Cisco's MicroGrid LT}, along with the critical component {Cisco 3650CX} from the lighting system, is essential for ensuring QoS. Moreover, since the other MCSs are of size six or larger, they have a less pronounced impact on the system's reliability. For these MCSs, maintenance decisions are linked to the initially defined threshold. Indeed, as a component ages faster, the choice of maintenance is decided by the BA or the ItA at different intervals for each threshold. This decision is based on minimizing maintenance costs. Taking the example of a Light Fixture, if components age rapidly (low threshold), we encounter a scenario where the Lux Sensor enters the wear-out phase while all other remaining components of the Light Fixture are still within their useful life phase because their MTTF is greater than that of the Lux Sensor. This scenario prompts a maintenance decision by the BA modeling the Lux Sensor. Similarly, if multiple components of the Light Fixture are in the wear-out phase, the ItA makes the maintenance decision in this case. Another advantage is that since the MicroGrid LT is a component for each light fixture, the maintenance decision of the light fixture chosen by the ItA helps manage a part of the network system based on the reliability of the lighting system.

V. Conclusion

In this paper, an RL-based approach model for PdM in a SB has been proposed and applied to a SB Lighting system. First, a FT models this system's components interactions. Then, a RL model is developed based on the FT architecture and the life cycle representation. When a component is in its useful life phase, agent decisions are modeled by MDPs. However, once the component has entered its wear-out phase, the MDP formulation with a transition matrix is no longer applicable. In this case, the Q learning algorithm obtains the agents' decisions. The results show that using a few data points, it is possible to obtain maintenance decisions that optimize a reward and satisfy a defined QoS. In addition, the model can be integrated into a real SB system case, such that the component operating data will be used to define the exact parameters of the Weibull distribution, and the maintenance cost data will be integrated into the agent rewards matrix.

In our future works, we aim to design maintenance strategies for the entire system based on maintenance actions obtained from the model presented in this paper. Subsequently, we will develop a general model for the maintenance of the entire SB.

References

[1] J. Wang, L. Zhang, L. Duan, and R. X. Gao. A new paradigm of cloud based predictive maintenance for intelligent manufacturing. *Journal of Intelligent Manufacturing, vol. 28, no. 5, pp. 1125–1137*, 2017.

[2] Zonta, Costa, Righi, de Lima, Trindade, and Li. Predictive maintenance in the industry 4.0: A systematic literature review. *Comput. Ind. Eng.*, 150, 106889, 2020.

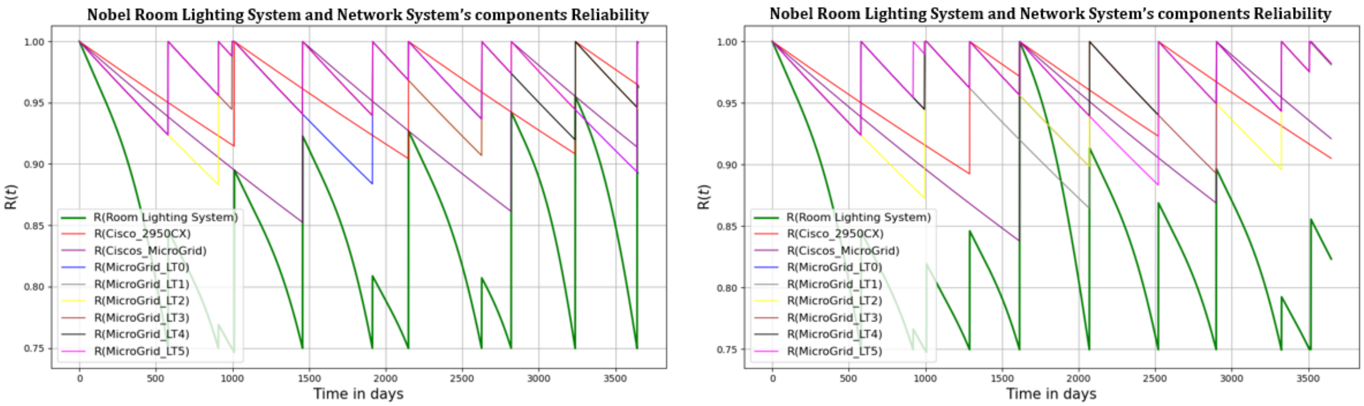
[3] Vinicius G. M. Cruz, Andre L. H. Costa, and Marcio L. L. Paredes. Development and evaluation of a new dga diagnostic method based on thermodynamics fundamentals. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2015.

[4] John T. Renwick and Paul E. Babson. Vibration analysis—a proven technique as a predictive maintenance tool. *IEEE Transactions on Industry Applications*, 1985.

[5] Tiedo Tinga and Richard Loendersloot. Physical model-based prognostics and health monitoring to enable predictive maintenance: Advanced methods, decision support tools and real-world applications. *Predictive Maintenance in Dynamic Systems, ELSEVIER pp 313–353*, 2019.

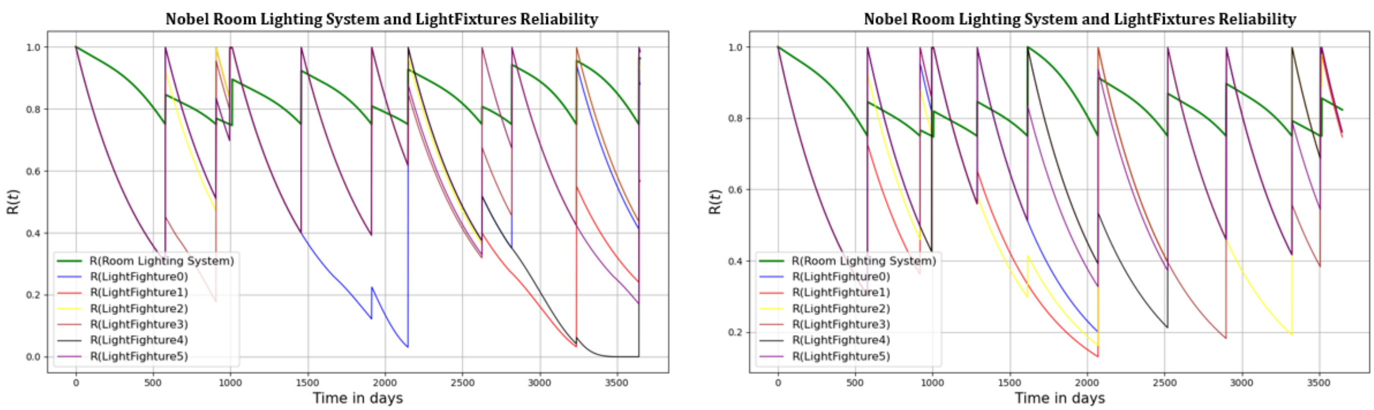
[6] Qiushi Cao, Cecilia Zanni-Merk, Ahmed Samet, Christoph Reich, François de Bertrand de Beuvron, Arnold Beckmann, and Cinzia Giannetti. Kspmi: A knowledge-based system for predictive maintenance in industry 4.0. *Robotics and Computer-Integrated Manufacturing, ELSEVIER*, 2022.

[7] Jeffrey Schein, Steven T. Bushby, Natascha S. Castro, and John M. House. A rule-based fault detection method for air handling units. *Energy Build 38(12):1485-92, ELSEVIER*, 2006.



a). Threshold = 0.9 MTTF b). Threshold = 3 MTTF

Fig. 6. Lighting System and Network System's Reliability for different threshold using a fixed Reliability QoS 75 %



a). Threshold = 0.9 MTTF b). Threshold = 3 MTTF

Fig. 7. Lighting System and Light Fixtures Reliability for different threshold using a fixed Reliability QoS 75 %

[8] W Zhang, D Yang, and H Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 2019.

[9] Yuxin Wen, Md. Fashiar Rahman, Honglun Xu, and Tzu-Liang Bill Tseng. Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective. *Measurement, ELSEVIER(187)*, 2022.

[10] Yongyi Ran, Xin Zhou, Pengfeng Lin, Yonggang Wen, and Ruilong Deng. A survey of predictive maintenance: Systems, purposes and approaches. *IEEE Communication Survey Tutorials*, 2019.

[11] Muthumani. A survey on failure prediction methods. *ACM Computing Surveys*, 42(3), 2010.

[12] Davide Pagano. A predictive maintenance model using long short-term memory neural networks and bayesian inference. *Decision Analytics Journal, ELSEVIER*, 2023.

[13] Susto, Schirru, Pampuri, McLoone, and Beghi. Machine learning for predictive maintenance: A multiple classifiers approach. *IEEE Transactions on Industrial Informatics*, 11, 2015.

[14] Bouabdallaoui, Lahfaj, Yim, Ducoulombier, and Bennadji. Predictive maintenance in building facilities: A machine learning-based approach. *Sensors, ISSN*, 1424-8220, 2021.

[15] Hessam Mohseni, Sujeeva Setunge, Guomin Zhang, and Ron Wakefield. Markov process for deterioration modeling and asset management of community buildings. *Journal of Construction Engineering and Management Volume 143, Issue 6*, 2017.

[16] Cauchi, Hoque, Stoelinga, and Abate. Maintenance of smart buildings using fault trees. *ACM Transactions on Sensor Network*, 1, 2018.

[17] Sijia Kong. Prediction-based control of dynamical systems with stochastic input delay. *Automatic Control Engineering. Université Paris sciences et lettre, ParisTech*, 2022.

[18] Michel Batteux, Tatiana Prosvirnova, and Antoine Rauzy. Abstract executions of stochastic discrete event systems. *International Journal of Critical Computer-Based Systems, Vol 10, No.3*, 2022.

[19] Kari Alanne and Seppo Sierla. An overview of machine learning applications for smart buildings. *ELSEVIER*, January 2022.

[20] Marcelo Luis Ruiz Rodríguez, Sylvain Kubler, Andrea de Giorgio, Maxime Cordy, Jérémy Robert, and Yves Le Traon. Multi-agent deep reinforcement learning based predictive maintenance on parallel machines. *Robotics and Computer–Integrated Manufacturing, ELSEVIER*, 2022.

[21] Ornella Pisacane, Domenico Potena, Sara Antomarioni, Maurizio Bevilacqua, Filippo Emanuele Ciarapica, and Claudia Diamantini. Data-driven predictive maintenance policy based on multi-objective optimization approaches for the component repairing problem. *Engineering Optimization Volume 53,- Issue 10*, 2021.

[22] Dimitrios Vamvakas, Panagiotis Michailidis, Christos Korkas, and Elias Kosmatopoulos. Review and evaluation of reinforcement learning frameworks on smart grid applications. *Energies 2023, 16(14), 5326*, 2023.

[23] Anas Hossini, Leïla Kloul, and Benjamin Cohen Boulakia. Fault prediction in a smart building lighting system. *European Safety and Reliability Conference(ESREL)*, 2023.

[24] CESI's Smart Building. https://www.cesi.fr/en/research-innovation/cesi-demonstrators/.

[25] Jozsef Dombia and Tamas Jonas. An elementary proof of the general poincare formula for λ -additive measures. *Acta Cybernetica 24 pages: 173–185.*, 2019.