ARTICLE IN PRESS

Nuclear Engineering and Technology xxx (xxxx) xxx

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Flexible operation and maintenance optimization of aging cyber-physical energy systems by deep reinforcement learning

Zhaojun Hao [a], Francesco Di Maio [a,*], Enrico Zio [a,b]

[a] Energy Department, Politecnico di Milano, Milan, Italy
[b] Mines Paris, PSL Research University, CRC, Sophia Antipolis, France

ARTICLE INFO

Keywords:
Cyber-physical energy system (CPES)
Operation & maintenance (O&M)
Cyber aging
Deep reinforcement learning (DRL)
Nuclear power plant (NPP)
Optimization
Advanced lead-cooled fast reactor european demonstrator (ALFRED)

ABSTRACT

Cyber-Physical Energy Systems (CPESs) integrate cyber and hardware components to ensure a reliable and safe physical power production and supply. Renewable Energy Sources (RESs) add uncertainty to energy demand that can be dealt with flexible operation (e.g., load-following) of CPES; at the same time, scenarios that could result in severe consequences due to both component stochastic failures and aging of the cyber system of CPES (commonly overlooked) must be accounted for Operation & Maintenance (O&M) planning. In this paper, we make use of Deep Reinforcement Learning (DRL) to search for the optimal O&M strategy that, not only considers the actual system hardware components health conditions and their Remaining Useful Life (RUL), but also the possible accident scenarios caused by the failures and the aging of the hardware and the cyber components, respectively. The novelty of the work lies in embedding the cyber aging model into the CPES model of production planning and failure process; this model is used to help the RL agent, trained with Proximal Policy Optimization (PPO) and Imitation Learning (IL), finding the proper rejuvenation timing for the cyber system accounting for the uncertainty of the cyber system aging process. An application is provided, with regards to the Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED).

1. Introduction

Once in operation, the productivity and safety of Cyber-Physical Energy Systems (CPESs) are accomplished by proper Operation & Maintenance (O&M) strategies aiming to increase profits, prevent unexpected failures and lower risk [1–3].

Collecting and using condition monitoring data, along with estimating component health states and predicting their Remaining Useful Life (RUL) [4–6], has significantly aided in diagnosing component faults [7–9]. Moreover, it has enabled the adoption of the Predictive Maintenance (PdM) paradigm, facilitating just-in-time maintenance interventions to maximize system availability and minimize O&M costs [10–12]. PdM has proven to outperform the traditional Scheduled Maintenance (SM) strategy, which relies on pre-defined inspection intervals [13,14].

The penetration of Renewable Energy Sources (RESs) onto the power grid, with high degree of variability in power generation, challenges O&M to guarantee flexibility of operation (e.g., load-following [15]) for dealing with sudden imbalances between demand and production [16,17]. Thus, to safely provide flexible operation, O&M strategies should not only take into account the components health status and their RUL [10,18], but also the fluctuation of power consumption and generation over long-time horizons [19]. However, actual O&M strategies, even if considering the hardware component stochastic failures [14,20], overlook the deterioration and aging of the cyber system and their effect on the flexible & safe energy supply. Cyber system aging (also known as software aging [21,22]) is, indeed, a commonly neglected phenomenon occurring in long-running cyber-physical systems, that can lead to performance degradation and catastrophic failures [21,22]. The cause of cyber system aging is the trigger of internal aging-related bugs which exhaust the operating system resources (e.g., memory leaking), corrupt data and accumulate numerical errors [21]. Since the cyber system is the sensitive control part of a CPES, aging and performance degradation significantly affect the control of the system [23,24]. Proactive measures, known as "rejuvenation" [25,26], are, therefore necessary to clear the cyber components from such aging level that might lead the CPES to catastrophic failures.

In this paper, we formalize the problem of O&M optimization considering the cyber aging as a Sequential Decision Problem (SDP): we search for the optimal arrangement of maintenance of hardware

* Corresponding author.
E-mail addresses: zhaojun.hao@polimi.it (Z. Hao), francesco.dimaio@polimi.it (F. Di Maio), enrico.zio@polimi.it, enrico.zio@mines-paristech.fr (E. Zio).

https://doi.org/10.1016/j.net.2023.11.052
Received 10 February 2023; Received in revised form 5 October 2023; Accepted 30 November 2023
Available online 4 December 2023

components and rejuvenation of cyber components that maximize productivity and safety, while providing flexible supply (load-following). Tabular Reinforcement Learning (RL) has been widely used to solve such SDP [27]. However, the computation cost of tabular RL is not compatible with the application to complex CPESs, whose state and action spaces are large due to the numerous components [27]. Thus, as proposed in Refs. [2,15], we resort to Deep RL [27], a feasible extension of Reinforcement Learning (RL), by originally integrating the Proximal Policy Optimization (PPO) algorithm [28], Imitation Learning (IL) [29] and a CPES model [30] that embeds the hardware components RULs estimator, the hardware components failure process model and the cyber aging model [31,32]. The Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED) case study is presented [33]. This advanced Nuclear Power Plant (NPP) is specifically designed to offer flexible operation by providing the possible daily changing power output between full (100 %) power and 20 % power levels. The main hardware components of ALFRED, i.e., water pump, sensors, turbine admission valve and control rods, are considered equipped with RUL estimation capabilities, while the cyber component (controller) is considered able to access current available memory and operating time. For the system failure process, an available Goal Tree Success Tree-Master Logic Diagram (GTST-MLD) reliability model [30] is considered.

The remainder of the paper is organized as follows: in Section 2, the cyber component aging and rejuvenation is presented; Section 3 states the problem and formulates it as a SDP; in Section 4, details about the RL algorithm developed in this work are provided; Section 5 describes the case study; in Section 6, the results are discussed; conclusions are drawn in Section 7.

2. The cyber system aging and rejuvenation

2.1. Aging

For modeling the cyber aging and the rejuvenation, the multi-state model of the aging process of a CPES presented in Ref. [32], is considered.

In brief, cyber aging caused by aging-related bugs (such as memory leakage) generate errors that propagate inside the system [21]. Memory leakage may lead to data-jamming and prevent data processing or tasks delivering in due time, ultimately resulting in data queueing and memory request increase that blocks the system [25], reduces the controllability and stability of the controlled physical system, and leads to system failure [34]. In this work, we model the memory leakage and data-jamming by a Continuous-Time Markov Chain (CTMC), as proposed in Ref. [32]. Combining the resulting available memory at time t, $M(t)$, and the data-jamming probability $P_{jam}(i,j)$ of j data jammed when the cyber system is in aging state i, the cyber system blocking probability $P_{blocking}(t)$ can be calculated and used in the RL, as we shall see in Section 5.

2.2. Rejuvenation

Rejuvenation consists in cleaning up the in-memory data structures to prevent cyber system degradation or crashes [26,35]. Two types of rejuvenation policies have been recently used [26]: a periodic policy (i.e., the rejuvenation is performed each pre-defined deterministic interval); and a prediction-based policy (i.e., rejuvenation is performed when suggested by the collected cyber system condition monitoring data and their statistical analysis). In this work, we assume that the cyber components are continuously monitored and, therefore, a prediction-based policy can be adopted.

3. Problem formulation

The CPES load-following power production plan $P(t)$ can range from full (100 %) power (typically produced in base-load-regime) to 20 % (i.e., the minimum assumed in the daily cycles of load-following): this allows dealing with the RES fluctuation at each time $t = 1, 2, \ldots, T_M$ (the mission time). Both base-load and load-following operations create revenue, which are denoted as K_{base} and K_{load}, respectively.

The CPES consists of L hardware components and one cyber controlling system. The generic l-th hardware component, $l \in \Lambda = \{1, \ldots, L\}$, is assumed to be equipped with PHM capability, which allow estimating its RUL. Given the ground truth failure time T_l^* of the l-th hardware component, the RUL is:

$$R_l^* = T_l^* - t \tag{2}$$

whose estimation provided by the PHM tool is:

$$R_l = R_l^* + \epsilon_R \tag{3}$$

where $\epsilon_R \sim N(0, \sigma_R)$ is a Gaussian noise representing the error of the RUL estimation [2]. The number of maintenance crews is assumed equal to the number of hardware components in need of repair, and the maintenance assumed as good as new (AGAN). The generic l-th hardware component will undergo *i*) Preventive Maintenance (PM), if the component is not failed, i.e., $R_l^* > 0$, or *ii*) Corrective Maintenance (CM), if the component is failed, i.e., $R_l^* = 0$. The downtimes caused by PM and CM, Π_{PM} and Π_{CM} (typically $\Pi_{PM} < \Pi_{CM}$) are regarded as a deterministic time period [36,37], with the resulting cost of downtimes U_{PM} and U_{CM}, respectively.

The cyber system is assumed to be continuously monitored and supposed to undergo rejuvenation to clear the software aging level if the aging level is too high (i.e., low available memory M_t), with a downtime of rejuvenation assumed to be the same as PM, $\Pi_{rej} = \Pi_{PM}$ and the cost $U_{rej} = U_{pm}$.

When either a hardware component or the cyber controller fails, the CPES may undergo safe shutdown or severe (damaged) shutdown, whose costs per unit of time are U_{safe} and U_{severe}, respectively.

For simplicity's sake, but without loss of generality, we i) neglect backup components or safety-related protection systems, ii) assume that load-following operation can be implemented only when there are no components failed or under maintenance.

In this setting, the O&M problem is formulated as a SDP defined by the set $\mathscr{S}, \mathscr{A}, \mathscr{P}, \mathscr{R}, \gamma$, (described in Table I and Sections 3.1, 3.2 and 3.3).

Solving the SDP means defining the optimal O&M policy $\pi^*(a|s)$ (i.e., the actions sequence a to be adopted at each decision time t, with regards to environment state s) that maximizes the system profit over the mission time T_M.

3.1. State space $\mathscr{S}$

At each decision time t, the state space $\mathscr{S}$ is defined by the vector $\vec{s}_t = [\vec{R}_t, \vec{Comp}_t, \vec{MT}_t, \vec{P}_t, \vec{M}_t, T_C, Con_t, Sys_t, t] \in \mathbb{R}^{3L+J+2}$, obtained appending the vectors of RUL estimations $\vec{R}_t = [R_1, R_2, \ldots, R_L]$, the component state vector (operating, failed, CM and PM) $\vec{Comp}_t = [Comp_1, Comp_2, \ldots, Comp_L]$, the vector of the times needed to complete the

Table 1
SDP formulation..

Symbol	Meaning
$\mathscr{S}$	State space
$\mathscr{A}$	Action space
$\mathscr{P}$	Transition probability $\mathscr{P}(s'\|s,a)$
$\mathscr{R}$	Reward function $\mathscr{R}(s'\|s,a)$
γ	Discount factor $[0,1]$

current maintenance $\vec{MT}_t = [MT_1, MT_2, \ldots, MT_L]$, the previous I days cyber controller available memory from day $t-I$ to day t ($I = 1, 2, \ldots, I-1$) $\vec{M}_t = [M_{-I}, M_{-I+1}, \ldots, M_{-1}, M_0]$, the production plan vector for consecutive J days from day t to day $t+J-1$ ($J = 1, 2, \ldots, T_M - t + 1$) $\vec{P}_t = [P_0, P_1, \ldots, P_{J-1}]$, the cyber controller continuous operation time since last rejuvenation T_C, the cyber controller state (operating and rejuvenation) Con_t and the system state (operating, PM, shutdown and failure). Typically, the state space $\mathscr{S}$ cannot be explored by forcing a real system to experience all the possible states for economic, safety and time issues. Therefore, a model (typically white-box) is used as surrogate (see Section 3.4).

3.2. Action space $\mathscr{A}$

At each decision time t, the maintenance actions space $\mathscr{A}$ is defined by the vector $\vec{a}_t = [a_1, \ldots, a_l, \ldots, a_L, a_C]$: if a decision is taken to maintain the l-th component, the corresponding a_l is set to 1, resulting in $\vec{a}_t = [0, \ldots, 0, a_l = 1, 0, \ldots, 0]$; if a decision is taken to rejuvenate the cyber controlling system, a_C is set to 1, resulting in $\vec{a}_t = [0, \ldots, 0, a_C = 1]$; $\vec{a}_t = [0, \ldots, 0]$ for no maintenance or rejuvenation actions.

3.3. Reward function

At each decision time t, a reward r_t is calculated on the basis of $\vec{s}_t$ and $\vec{a}_t$ as follows:

$$r_t = G_t - W_t - X_t \tag{4}$$

where G_t is the revenue (see Eq. (5) below), W_t is the cost when the system is under safe shutdown or severe shutdown (see Eq. (6) below) and X_t is the maintenance intervention cost (see Eq. (7) below).

G_t can be calculated as follows:

$$G_t = I_{base} \bullet K_{base} + I_{load} \bullet K_{load} \tag{5}$$

where I_{base} and I_{load} are Boolean variables equal to 1 and 0, respectively, when the system operates in base-load regime, $P(t) = 0$, or 0 and 1, respectively, when the system operates in load-following regime, $P(t) = 1$.

W_t can be calculated as follows:

$$W_t = I_{safe} \bullet U_{safe} + I_{severe} \bullet U_{severe} \tag{6}$$

where I_{safe} and I_{severe} are Boolean variables equal to 1 when the system, at time t, is unavailable due to safe shutdown or severe shutdown.

X_t can be calculated as follows:

$$X_t = \sum_{l=1}^{L} I_l^{RUL>0} \bullet U_{PM} + I_l^{RUL=0} \bullet U_{CM} \tag{7}$$

where $I_l^{RUL=0}$ and $I_l^{RUL>0}$ are Boolean variables that indicate whether the component has (not) failed at time t and, therefore, should undergo corrective (preventive) maintenance.

3.4. The environment model

Although the agent could theoretically discover the optimal O&M policy through direct interactions with the real-world system, this has been proven to be impractical in the case of complex CPES due to economic, safety and time issues: a white-box environment model that must be ensured to reproduce the real system behavior with fidelity is therefore often used to train the learning agent [2].

4. Reinforcement learning algorithms

Fig. 1 sketches the RL procedure applied in this paper. The agent is identified as the decision maker, and the environment is the system with which it interacts. They continuously interact until the agent selects the action and the environment responds to this with a reward that the agent aims at maximizing over time [27]. Specifically, at each decision time t, the agent receives a representation of the environment state $\vec{s}_t$ (here including the components RULs $\vec{R}_t$, the components state $\vec{Comp}_t$, the maintenance remaining times $\vec{MT}_t$, the production plan $\vec{P}_t$, the cyber system previous available memory $\vec{M}_t$, the cyber system working time T_C, the cyber system state Con_t and the system state Sys_t); based on this, it selects an action $\vec{a}_t$ to provide the optimal order of maintenance actions for the current situations. The environment system model simulates the system response to the selected action $\vec{a}_t$, moves to the new state $\vec{s}_{t+1}$ resulting from such action and returns the corresponding numerical reward r_t to the agent. By a trial-and-error iterative procedure, the agent reaches the optimal policy $\pi^*(a|s)$, which maps the possible environment states s into the optimal actions a maximizing the expected cumulative sum of rewards over the time horizon $E[\sum_{t=0}^{T_M} \gamma^t \bullet r_t(\vec{a}_t, \vec{s}_t, \vec{s}_{t+1})]$, where γ is the discount parameter of future rewards.

In this work, we adopt Proximal Policy Optimization (PPO) [28] algorithm to optimize the O&M strategy because PPO recently shown on several applications [2,38] to be not only relatively easy to implement and tune, but also outperforming many state-of-the-art approaches. However, given the size of the state space, the agent might still be challenged in efficiently choosing the optimal policy $\pi^*(a|s)$: therefore, Imitation Learning (IL) [29], specifically Behavioral Cloning [39], is here used as in Refs. [2,40] to first heuristically generate trajectories that are used as training data for the policy neural network that learns the pairs of state $\vec{s}_t$ and action $\vec{a}_t$, and then, to fine-tune the agent, using RL to explore new policies and discover the optimal one. The interested reader may refer to Refs. [2,40] for a detailed description of the IL implementation and the proof that IL can ensure effectiveness of the RL.

5. Case study: The Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED)

ALFRED is a perfect candidate among NPPs for handling the fluctuation of RESs in a load-following schedule [41]. The control of ALFRED is implemented by four feedback control loops (see Fig. 2) [33], that keep four variables $\vec{y}$ (cold leg lead temperature $T_{L,cold}$, steam temperature T_{steam}, thermal power P_{Th} and Steam Generator (SG) pressure p_{SG}) controlled within the safety thresholds in any operational condition. The ALFRED control system is here simplified as composed of $L = 7$ hardware

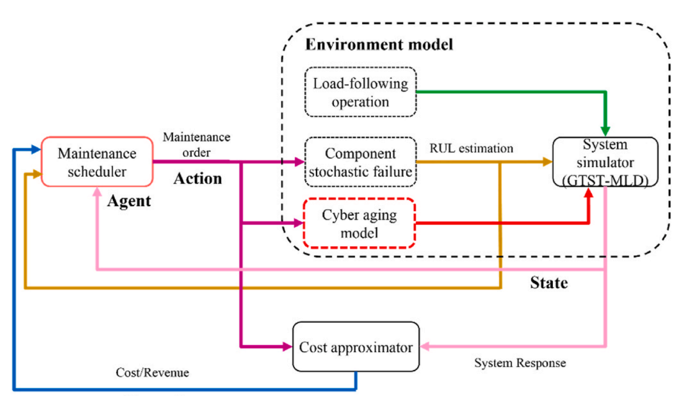

Fig. 1. Schematic representation of RL procedure.

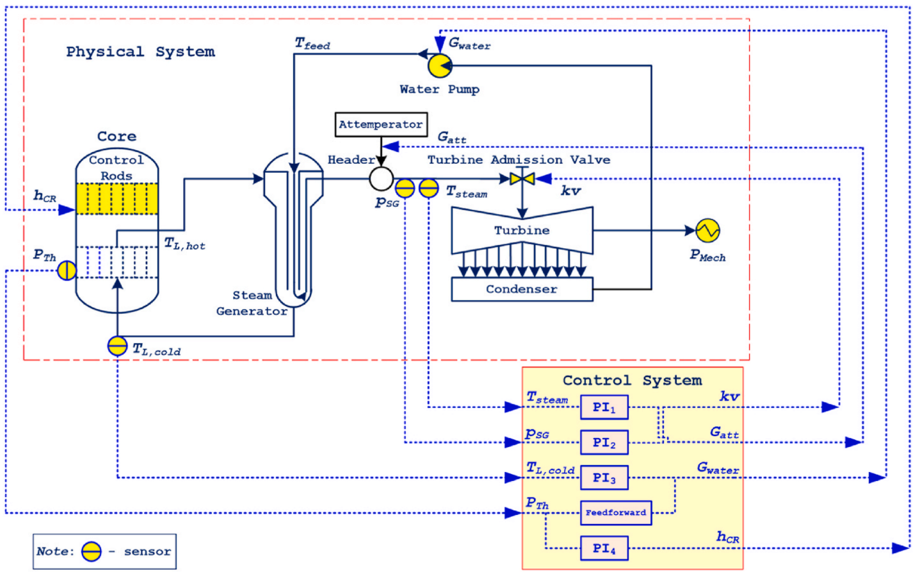

Fig. 2. The control system of ALFRED [23].

components (4 sensors for the variables T_{steam}, p_{SG}, $T_{L,cold}$ and P_{Th}, and 3 actuators for the water pump (G_{water}), the turbine admission valve (kv) and the control rods (CR)) and one cyber controlling system. All hardware components are subjected to stochastic failures over a mission time T_M of 5 years (1825 days) and are equipped with PHM capabilities for estimating their RULs, with a zero-mean Gaussian error whose standard deviation is σ_R= 10 days (see Eqs. (1) and (2)). The failure rates for the hardware components are listed in Table II [24]. The cyber system available memory curve (with 95 % confidence interval) is shown in Fig. 3.

We assume that *i)* the available memory for I= 2 previous days is known, i.e., $\vec{M_t} = [M_{-2}, M_{-1}, M_0]$, *ii)* the production plan $\vec{P_t}$ (base-load or load-following with respect to the probabilities listed in Table III) for J= 2 successive days is known, i.e., $\vec{P_t} = [P_0, P_1, P_2]$, *iii)* the maintenance/rejuvenation durations Π_{PM}, Π_{CM} and Π_{rej} are considered as deterministic time periods $\Pi_{PM} = \Pi_{rej}$= 1.25 days [43] and Π_{CM} = 3.37 days [44], respectively, *iv)* the daily revenues and maintenance costs of PM and CM are those listed in Table IV.

The ALFRED system model we use in the RL environment is the GTST-MLD shown in Fig. 4, proven to be accurate enough to reproduce the system behavior [30,42]. The cyber controlling system aging is modeled with the Influencing Factor (IF) D_{aging} (see Fig. 4) that can cause the failure of the controller software (PI gains of each controlled variables) and communication (sensors) with a controller blocking probability $P_{blocking}$. Therefore, the probability that the controller fails due to blocking during load-following operations is:

$$P_{D_{aging}} = P_{blocking} \bullet P_{load} \tag{8}$$

where P_{load} is the probability of load-following occurrence. After initializing the components state, sampling the influencing factor occurrence and propagating the corresponding hardware component/aging caused cyber parts failure through the GTST-MLD (the interested reader may refer to Ref. [42] for implementation details), the GTST-MLD reliability model evaluates the system response with respect to whether the component/cyber parts failure leads the four controlled variables (T_{steam}, p_{SG}, $T_{L,cold}$ and P_{Th}) out of the safety thresholds, i.e., the system fails leading to severe consequences. In other words, by hierarchically decomposing the ALFRED structure and functionality into a GTST-MLD white-box model, we can guarantee that we are mimicking the ALFRED real-world behavior across the widest range of configurations required

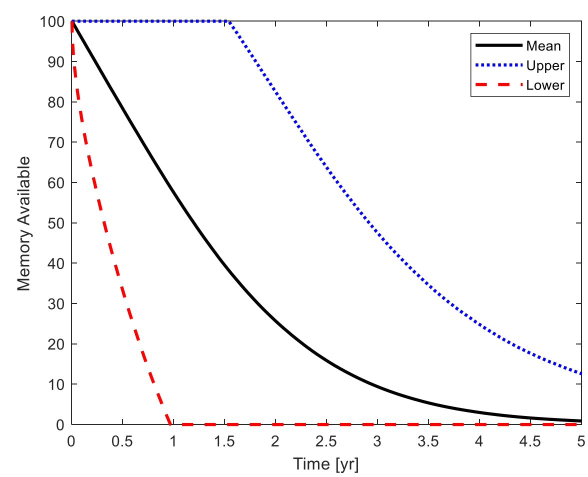

Fig. 3. Cyber system available memory decreasing curve.

Table 3
NPP load-following cycles [32,45]..

Load Cycle	Number of Load Cycles in 70 years lifetime	Probability per day
100-90-100	100,000	0.163
100-80-100	100,000	0.163
100-60-100	15,000	0.0245
100-40-100	12,000	0.0196
100-20-100	100	1.65E-4
Load-following	–	0.3703
Base-load	–	0.6297

Table 2
Component failure rate [42]..

Failure rate/occurrence probability	Value
λ_{sensor}	6.20E-3/Year
λ_{kv}	6.57E-4/Year
λ_{water}	1.14E-2/Year
λ_{CR}	5.30E-3/Year

Table 4
Daily revenues and maintenance costs [43,46,47]..

Revenue/Cost	Value [KEuros per day]
Normal operation revenue K_{base}	720
Flexible operation revenue K_{load}	900
Shutdown cost $U_{shutdown}$	720
Failure cost $U_{failure}$	1200
PM cost U_{PM}	1.5
CM cost U_{CM}	6.2

by the RL environment model when interacting with the agent, that would have been instead impractical for economic, reproducibility, safety and time issues by forcing the real ALFRED to undergo such a multitude of scenarios.

As RL agent, based on the settings in Refs. [2,15], we use a DNN with two hidden layers of 64 neurons. The IL step is performed by generating 500 PdM trajectories, which list the state-action pairs following the PdM policy that are used to pre-train the agent for 50 epochs to reproduce the PdM behavior. Finally, the PPO RL is implemented. The discount factor γ is set equal to 0.99 by grid searching around the empirical value [2].

6. Results

For a fair comparison of the PPO RL that considers cyber aging with state-of-practice strategies, we have considered (in increasing order of complexity) *i*) a CM strategy, *ii*) a SM strategy, *iii*) a PdM strategy (i.e., the same policy of the IL step used to pre-train the agent in Section 5) and *iv*) a PPO RL strategy that neglects cyber aging. All strategies are tested on a set of 100 test sequences of O&M and the corresponding profits and losses within the mission time T_M of 5 years are compared. The SM and PdM are performed with 173 days of SM interval and 35 days of PdM RUL threshold (found by grid search) for hardware components, respectively, and 730 days of rejuvenation interval for cyber controller [24,32].

Conditional Value at Risk (CVaR) is used to evaluate the strategies performance, while Value at Risk (VaR) quantifies the extent of possible financial losses (e.g., if the CPES operation profit within the mission time has a 95 % VaR of 7 million euros, the CPES profit has a 5 % probability of losing its value by 7 million euros after the operation of the mission

Table 5
Performance of the tested strategies in terms of average profit, 95 % CVaR, average number of CM and PM actions over 100 test sequences.

Maintenance strategy	Average profit [10^9 euro] (Ranking)	95 % CVaR [10^9 euro] (Ranking)	Average number of CM (Ranking)	Average number of PM (Ranking)
Corrective	0.05 ± 0.18 (5)	1.43 ± 0.76 (5)	38.75 ± 5.32 (5)	–
Scheduled	0.42 ± 0.14 (4)	1.03 ± 0.47 (4)	26.43 ± 2.17 (4)	62.56 ± 7.21 (4)
Predictive	1.15 ± 0.05 (2)	0.45 ± 0.25 (2)	$\mathbf{0.04 \pm 0.01}$ **(1)**	$\mathbf{45.35 \pm 4.28}$ **(1)**
PPO	1.02 ± 0.07 (3)	0.53 ± 0.29 (3)	0.05 ± 0.02 (2)	43.28 ± 3.01 (3)
PPO-aging	$\mathbf{1.41 \pm 0.03}$ **(1)**	$\mathbf{0.02 \pm 0.01}$ **(1)**	0.05 ± 0.02 (2)	43.65 ± 3.24 (2)

*In bold the best performance.

time). CVaR estimates the expected loss if the losses go beyond the VaR cut-off (e.g., the CPES operation profit having a 95 % CVaR of 5 million euros means that the average of losses that are larger than the 95 % VaR cut-off threshold (e.g., 3 million euros losses) is 5 million euros within the mission time) [48]. The obtained comparison results are listed in Table V, with the ranking of the alternative strategies with respect to average profit, 95 % CVaR and average number of CM, PM and rejuvenation actions needed in the sequence mission time.

From Tables V and it can be noticed that (for hardware maintenance) CM and SM policies, which are the most commonly adopted strategies, cause a large number of hardware components failures, leading to an average of 38.75 and 26.43 times of NPP system dysfunction (safe shutdown and severe shutdown) during the 5 years mission time, respectively (which is equal to the number of CM actions consequently performed). Due to the exploitation of the components health information, PdM, PPO and PPO-aging policies arrange just-in-time PM actions (45.35, 43.28 and 43.65 on average, respectively) and perform better than CM and SM in profits, CVaR and system dysfunction. It is necessary to point out that the number of PM actions of PPO (43.28) and PPO-aging (43.65) is slightly smaller than PdM (45.35), due to the smaller average RUL thresholds (35 days for PdM policy, 31.3 days for PPO policy and 31.5 days for PPO-aging policy on average) shown in

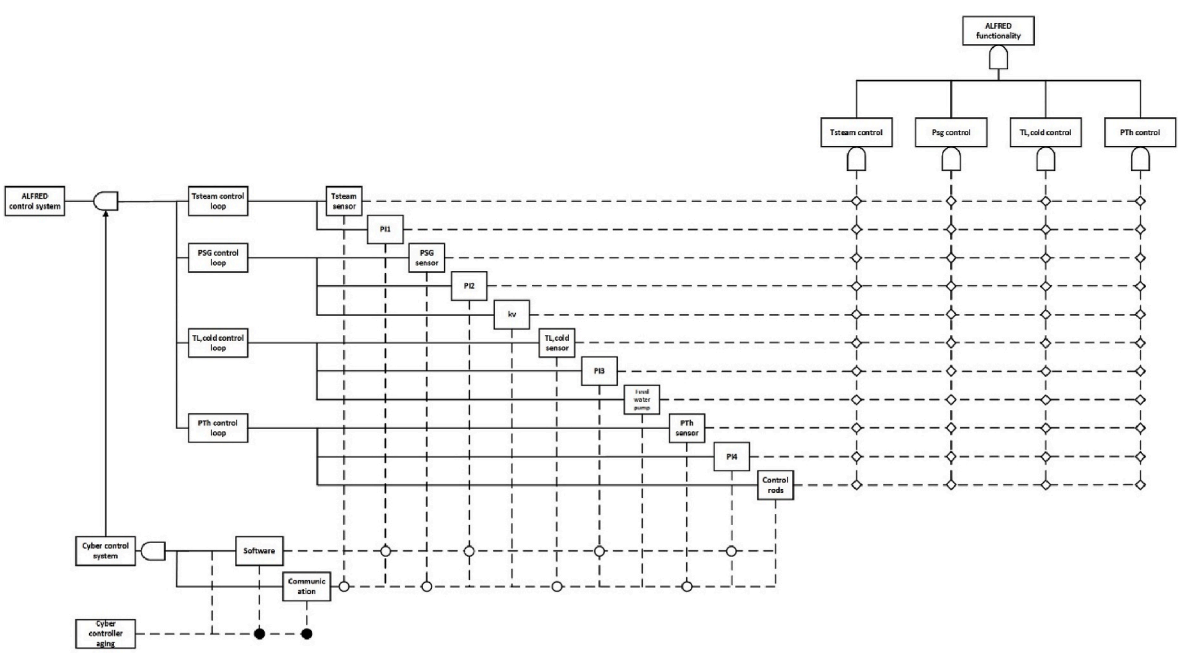

Fig. 4. GTST-MLD of ALFRED.

Table 6
Components RUL thresholds of maintenance interventions and corresponding GTST-MLD weights..

Components	RUL threshold of PPO policy [days]	RUL threshold of PPO-aging policy [days]	GTST-MLD weights			
			T_{steam} control	p_{SG} control	$T_{L,cold}$ control	P_{Th} control
Sensor T_{steam}	27.7	27.8	0	0	0	0
Sensor p_{SG}	45.5	45.3	0.35	0.69	1.54E-5	0.12
Sensor $T_{L,cold}$	27.8	27.9	0	0.09	0	0
Sensor P_{Th}	51.9	52.1	0.11	0.72	0	0.98
Turbine admission valve (kv)	28.2	27.7	0	0	0	0
Water pump (G_{water})	28.7	29.1	0	0	0	2.50E-3
Control rods (CR)	43.1	43.7	0.06	0.58	0	0.05
Average RUL threshold	31.3	31.5	–			

Table 7
Performance of the tested strategies in terms of average number of safe/severe shutdowns caused by hardware components failure in 100 test sequences.

Maintenance strategy	Average number of safe shutdowns (Ranking)	Average number of severe shutdowns (Ranking)
Predictive	**0.01 ± 0.01 (1)**	0.03 ± 0.02 (2)
PPO	0.04 ± 0.01 (2)	**0.01 ± 0.01 (1)**
PPO-aging	0.04 ± 0.01 (2)	**0.01 ± 0.01 (1)**

Table 8
Performance of the tested strategies in terms of average profit, 95 % CVaR, average number of cyber aging caused failures and rejuvenation actions over 100 test sequences.

Maintenance strategy	Average profit [10^9 euro] (Ranking)	95 % CVaR [10^9 euro] (Ranking)	Average number of cyber aging caused failures Ranking)	Average number of rejuvenations (Ranking)
Corrective	0.05 ± 0.18 (5)	1.43 ± 0.76 (5)	3.27 ± 0.86 (5)	–
Scheduled	0.42 ± 0.14 (4)	1.03 ± 0.47 (4)	1.85 ± 0.45 (3)	2.13 ± 0.48 (2)
Predictive	1.15 ± 0.05 (2)	0.45 ± 0.25 (2)	1.84 ± 0.43 (2)	2.01 ± 0.51 (3)
PPO	1.02 ± 0.07 (3)	0.53 ± 0.29 (3)	3.33 ± 0.79 (4)	–
PPO-aging	**1.41 ± 0.03 (1)**	**0.02 ± 0.01 (1)**	**0.03 ± 0.01 (1)**	**3.87 ± 1.36 (1)**

*In bold the best performance.

Table VI (in fact, smaller average RUL threshold means larger average maintenance interval and less interventions). From Tables VI and it can be noticed that the RL policies finds different RUL thresholds setting compared with PdM policy: RL policies have lower average RUL thresholds and also the thresholds of RL policies follow the weights of MLD listed in Table VI, which shows the relationship between components and system goal function (the larger weights show the stronger connections between components and goal function) (for further details see Ref. [42]). The RL policies are able to recognize the safety-related hardware components (large MLD weights, e.g., sensor p_{SG} (0.69), sensor P_{Th} (0.98) and control rods (0.58)) and sets higher RUL thresholds to maintain these hardware components in advance for preventing these safety-related components from failure, since they have high probability of leading to system severe shutdown (shown in Table VII, where the RL policies significantly decrease the severe shutdown caused by hardware component failures, compared to PdM policy).

Table VIII shows the comparison when cyber components failures and rejuvenation are of concern. The PPO-aging policy has the lowest cyber aging caused failures (0.03, on average). This is because it arranges controller rejuvenation more frequently (3.87 times, on average) than periodic rejuvenation policies (SM (2.13) and PdM (2.01)). Additionally, it exhibits the largest standard deviation (1.36), allowing it to handle the uncertainty of the aging process and accommodate different aging speeds. Even if the PPO policy can allocate just-in-time hardware components maintenance, the fact that it neglects cyber aging causes leads to too many failures (3.33 times, on average) and costs (0.42 of 95

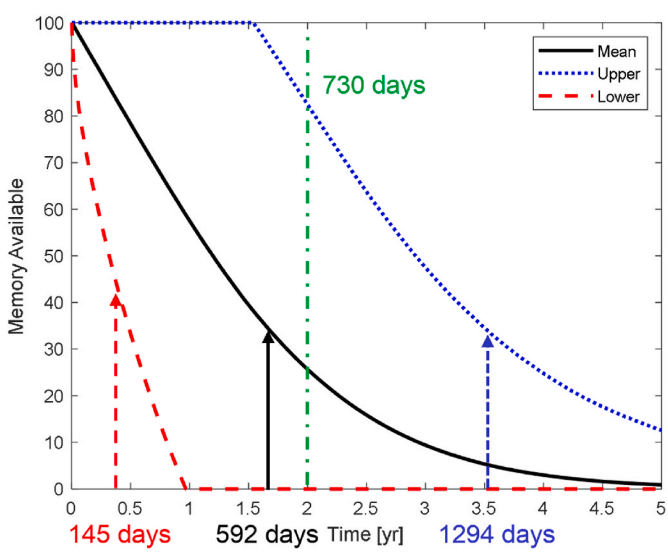

Fig. 5. Rejuvenation timing.

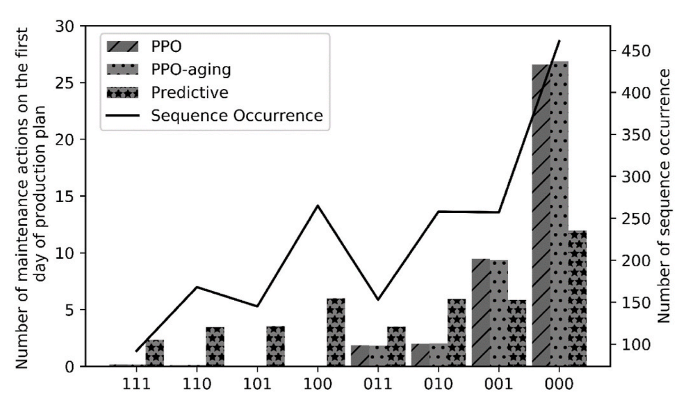

Fig. 6. Maintenance timing and power production demand sequence occurrence over 100 test sequences.

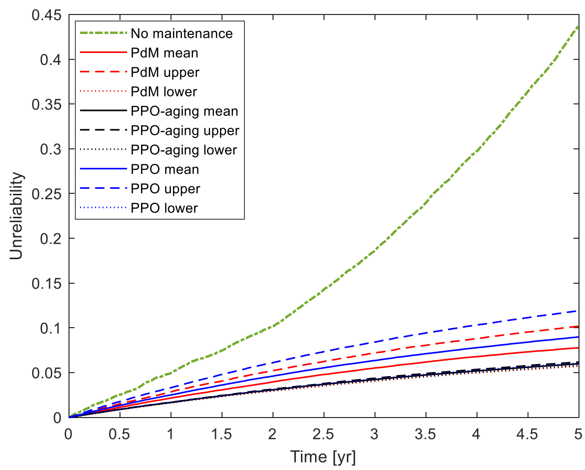

Fig. 7. System unreliability (95 % confidence interval).

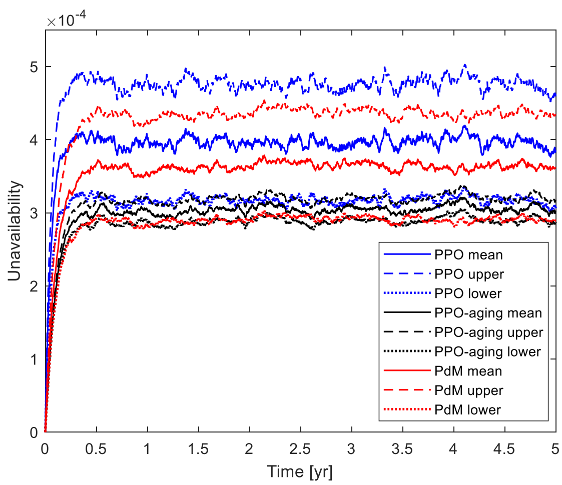

Fig. 8. System unavailability (95 % confidence interval).

% CVaR), as also summarized in Table V. Fig. 5 shows the effects of rejuvenation time of the PPO-aging policy and periodic rejuvenation policies with respect to the mean, upper and lower boundaries of 95 % confidence interval of the memory memory available in time. The dash-dotted line corresponds to the lower-upper boundary of the interval of memory available when a periodic rejuvenation (each 730 days) is adopted: most of the time the available memory is not enough, so that it may cause failures due to high level of cyber system aging and performance degradation. The PPO policy shows its capability of finding the proper rejuvenation timing (arrows) for the cyber system, with respect to the uncertainty of cyber system aging process, i.e., available memory decreasing speeds (145 days for the lower boundary, 592 days for the mean and 1294 days for the upper boundary).

In Fig. 6, the number of actions performed during specific power production plans are plotted for PPO, PPO-aging and PdM policies (slashed, dotted and star bars, respectively). Specifically, on the x-axis, the power production plans for J= 3 consecutive days are plotted (e.g., policy 110, standing for load-following operations on the first two days and, then, base-load operation on the third day), together with the frequency of occurrence of the production plan (continuous line), whose exact value can be calculated from the combination of load-following/base-load probabilities listed in Table III. It can be seen that the number of maintenance actions that the PdM policy chooses on the first day of the production plan follows the frequency of occurrence of the load-following sequences, which means that the PdM policy randomly chooses maintenance timing, neglecting the production plan, leading to a low performance in following the load. On the contrary, the PPO and PPO-aging policies (slashed and dotted bars) mostly arranges maintenance activities on base-load days and prefers 000 and 001 sequences than 010 and 011 sequences, to keep load-following operation as much as possible. This means that the RL agent chooses to implement the PM interventions on a base-load day. In other words, the RL agent can choose the actions considering the desired production plan (i.e., flexible operation) by optimizing the timing of maintenance activities. In particular, the RL agent postpones some of the maintenance activities from a load-following day to a base-load day, to respond to the production plan and to target the large profit objective.

In Fig. 7, the comparison between no maintenance policy (dash-dotted line), PPO, PPO-aging and PdM policies is shown with respect to the unreliability curve: it can be seen that the PPO-aging strategies achieve the lowest mean unreliability, in comparison with the unreliability achieved when the other strategies are adopted. The PdM policy cannot adapt to most cyber aging conditions with respect to the fixed periodic rejuvenation setting, resulting in the larger variance and the second lowest mean unreliability. Although PPO policy performs well in hardware maintenance as PPO-aging policy, neglecting cyber aging leads to a lot of unexpected failures (see also Table VIII), causing the largest unreliability. The same occurs for the unavailability (shown in Fig. 8). In conclusion, we can claim that the low unreliability (i.e., high productivity) and positive response to the production plan make the PPO-aging policy the highest profits and lowest CVaR (shown in Table V).

7. Conclusions

In this paper, we illustrate the SDP formalization of the O&M optimization in CPESs that operate flexibly to accommodate the fluctuations in production brought by penetration of RESs into the power grid and the uncertainty in power demand, considering the hardware components failure and cyber system aging. The DRL-based approach is used to solve the SDP, in which an agent-neural network is trained by interacting with the CPES model to search for the optimal O&M action to be performed on the basis of the available information (e.g., production plan, component RUL, component state, maintenance remaining time, system state, cyber system available memory, cyber system operating times and state).

The proposed approach has been applied to an advanced NPP design, ALFRED, and shown to be capable of providing an optimized O&M policy based on the RUL of the CPES components, the severity of the consequences of their failures and the aging level of the cyber system to avoid unexpected system safe/severe shutdown. It is necessary to point out that system safe/severe shutdowns are significantly decreased by embedding cyber aging model to help the RL agent to adaptively allocate rejuvenation to different aging speed and taking advantage of the system reliability model by GTST-MLD to recognize the safety-related components and set higher RUL thresholds. The policy considering cyber aging proposed here can outperform the state-of-practice policies (CM, SM, PdM and PPO without considering cyber aging) and keep the production availability and profitability high (and the costs low).

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Zhaojun Hao reports financial support was provided by China Scholarship Council.

References

[1] E. Zio, The future of risk assessment, Reliab. Eng. Syst. Saf. (2018), https://doi.org/10.1016/j.ress.2018.04.020.
[2] L. Pinciroli, P. Baraldi, G. Ballabio, M. Compare, E. Zio, Optimization of the operation and maintenance of renewable energy systems by deep reinforcement learning, Renew. Energy 183 (2022) 752–763.
[3] E. Gursel, B. Reddy, A. Khojandi, M. Madadi, J.B. Coble, V. Agarwal, V. Yadav, R. L. Boring, Using artificial intelligence to detect human errors in nuclear power plants: a case in operation and maintenance, Nucl. Eng. Technol. (2022), https://doi.org/10.1016/j.net.2022.10.032.
[4] E. Zio, Prognostics and Health Management (PHM): where are we and where do we (need to) go in theory and practice, Reliab. Eng. Syst. Saf. 218 (2022), 108119.
[5] P. Baraldi, F. Mangili, E. Zio, Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data, Reliab. Eng. Syst. Saf. 112 (2013) 94–108.
[6] F. Di Maio, P. Baraldi, E. Zio, R. Seraoui, Fault detection in nuclear power plants components by a combination of statistical methods, IEEE Trans. Reliab. 62 (2013) 833–845.
[7] L.M. Elshenawy, M.A. Halawa, T.A. Mahmoud, H.A. Awad, M.I. Abdo, Unsupervised machine learning techniques for fault detection and diagnosis in nuclear power plants, Prog. Nucl. Energy 142 (2021), 103990.
[8] G. Qian, J. Liu, Development of deep reinforcement learning-based fault diagnosis method for rotating machinery in nuclear power plants, Prog. Nucl. Energy 152 (2022), 104401.
[9] Z. Welz, J. Coble, B. Upadhyaya, W. Hines, Maintenance-based prognostics of nuclear plant equipment for long-term operation, Nucl. Eng. Technol. 49 (2017) 914–919, https://doi.org/10.1016/j.net.2017.06.001.
[10] M. Compare, P. Baraldi, E. Zio, Challenges to IoT-enabled predictive maintenance for industry 4.0, IEEE Internet Things J. 7 (2019) 4585–4597.
[11] H. Peng, Y. Wang, X. Zhang, Q. Hu, B. Xu, Optimization of preventive maintenance of nuclear safety-class DCS based on reliability modeling, Nucl. Eng. Technol. 54 (2022) 3595–3603, https://doi.org/10.1016/j.net.2022.05.011.
[12] H.A. Gohel, H. Upadhyay, L. Lagos, K. Cooper, A. Sanzetenea, Predictive maintenance architecture development for nuclear infrastructure using machine learning, Nucl. Eng. Technol. 52 (2020) 1436–1442, https://doi.org/10.1016/j.net.2019.12.029.
[13] T. Jiejuan, M. Dingyuan, X. Dazhi, A genetic algorithm solution for a nuclear power plant risk–cost maintenance model, Nucl. Eng. Des. 229 (2004) 81–89.
[14] A.W. Labib, M.N. Yuniarto, Maintenance strategies for changeable manufacturing, in: Changeable and Reconfigurable Manufacturing Systems, Springer, 2009, pp. 337–351.
[15] L. Pinciroli, P. Baraldi, G. Ballabio, C. Compare, E. Zio, Deep reinforcement learning for optimizing operation and maintenance of energy systems equipped with phm capabilities, in: Proceedings of the Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference, 2020.
[16] M.T. Kartal, A. Samour, T.S. Adebayo, S.K. Depren, Do nuclear energy and renewable energy surge environmental quality in the United States? New insights from novel bootstrap Fourier Granger causality in quantiles approach, Prog. Nucl. Energy 155 (2023), 104509.
[17] G. Chen, M. Li, Y. Zou, H. Xu, Analysis of load-following operation characteristics of liquid fuel molten salt reactor, Prog. Nucl. Energy 150 (2022), 104308.
[18] B. Tjahjono, C. Esplugues, E. Ares, G. Pelaez, What does industry 4.0 mean to supply chain? Procedia Manuf. 13 (2017) 1175–1182.
[19] Z. Hao, F. Di Maio, L. Pinciroli, E. Zio, Optimal prescriptive maintenance of nuclear power plants by deep reinforcement learning, in: Proceedings of the Proceedings of the 32nd European Safety and Reliability Conference, 2022.
[20] V. Holmgren, General-purpose Maintenance Planning Using Deep Reinforcement Learning and Monte Carlo Tree Search, 2019.
[21] M. Grottke, R. Matias, K.S. Trivedi, The fundamentals of software aging, in: Proceedings of the 2008 IEEE International Conference on Software Reliability Engineering Workshops (ISSRE Wksp), Ieee, 2008, pp. 1–6.
[22] K.S. Trivedi, K. Vaidyanathan, K. Goseva-Popstojanova, Modeling and analysis of software aging and rejuvenation, in: Proceedings of the Proceedings 33rd Annual Simulation Symposium (SS 2000), IEEE, 2000, pp. 270–279.
[23] W. Wang, A. Cammi, F. Di Maio, S. Lorenzi, E. Zio, A Monte Carlo-based exploration framework for identifying components vulnerable to cyber threats in nuclear power plants, Reliab. Eng. Syst. Saf. 175 (2018) 24–37.
[24] Z. Hao, F. Di Maio, E. Zio, A multi-state model of the aging process of cyber-physical systems, in: Proceedings of the 30th European Safety and Reliability Conference, ESREL 2020 and 15th Probabilistic Safety Assessment and Management Conference, PSAM15 2020, Research Publishing, Singapore, 2020, pp. 2241–2248.
[25] Y. Huang, C. Kintala, N. Kolettis, N.D. Fulton, Software rejuvenation: analysis, module and applications, in: Proceedings of the Twenty-Fifth International Symposium on Fault-Tolerant Computing. Digest of Papers, IEEE, 1995, pp. 381–390.
[26] D. Cotroneo, R. Natella, R. Pietrantuono, S. Russo, A survey of software aging and rejuvenation studies, ACM J. Emerg. Technol. Comput. Syst. 10 (2014) 1–34.
[27] R.S. Sutton, A.G. Barto, Reinforcement Learning: an Introduction, MIT press, 2018. ISBN 0262352702.
[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, 2017 *arXiv Prepr. arXiv1707.06347*.
[29] J. Ho, J.K. Gupta, S. Ermon, Model-free imitation learning with policy optimization, in: Proceedings of the 33rd International Conference on Machine Learning, vol. 6, ICML 2016, 2016, pp. 4036–4046.
[30] Z. Hao, F. Di Maio, E. Zio, Dynamic reliability assessment of cyber-physical energy systems (CPEs) by GTST-MLD, in: Proceedings of the 2021 5th International Conference on System Reliability and Safety (ICSRS), IEEE, 2021, pp. 98–102.
[31] Z. Hao, F. Di Maio, E. Zio, Modelling the Aging Process of a Cyber Physical System, 2019.
[32] Z. Hao, F. Di Maio, E. Zio, Multi-state reliability assessment model of base-load cyber-physical energy systems (CPES) during flexible operation considering the aging of cyber components, Energies 14 (2021) 3241.
[33] R. Ponciroli, A. Bigoni, A. Cammi, S. Lorenzi, L. Luzzi, Object-oriented modelling and simulation for the ALFRED dynamics, Prog. Nucl. Energy 71 (2014) 15–29.
[34] K.J. Åström, B. Wittenmark, Computer-controlled Systems: Theory and Design, Courier Corporation, 2013. ISBN 0486284042.
[35] X. Du, Y. Qi, D. Hou, Y. Chen, X. Zhong, Modeling and performance analysis of software rejuvenation policies for multiple degradation systems, in: Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference, vol. 1, IEEE, 2009, pp. 240–245.
[36] Y.-J. Lin, J.-M. Yang, R.-Y. Wang, Y.-X. Yang, Research on common cause fault evaluation model of RTS based on β-factor method, in: Proceedings of the International Symposium on Software Reliability, Industrial Safety, Cyber Security and Physical Protection for Nuclear Power Plant, Springer, 2022, pp. 590–599.
[37] Z.-G. Wu, J. Zhu, X.-B. Yu, Reliability analysis of tripping solenoid valve power system based on dynamic fault tree and sequential Monte Carlo, in: Proceedings of the International Symposium on Software Reliability, Industrial Safety, Cyber Security and Physical Protection for Nuclear Power Plant, Springer, 2022, pp. 148–158.
[38] N. Vanvuchelen, J. Gijsbrechts, R. Boute, Use of proximal policy optimization for the joint replenishment problem, Comput. Ind. 119 (2020), 103239, https://doi.org/10.1016/j.compind.2020.103239.
[39] S. Ross, J.A. Bagnell, Efficient reductions for imitation learning, J. Mach. Learn. Res. 9 (2010) 661–668.
[40] Z. Hao, F. Di Maio, E. Zio, Optimal prescriptive maintenance of nuclear power plants by deep reinforcement learning, in: Proceedings of the 32nd European Safety and Reliability Conference, ESREL, 2022, p. 2022.
[41] G. Terol, Porous Media Approach in CFD Thermohydraulic Simulation of Nuclear Generation-IV Lead-Cooled Fast Reactor ALFRED, 2021.
[42] F. Di Maio, R. Mascherona, E. Zio, Risk analysis of cyber-physical systems by GTST-MLD, IEEE Syst. J. 14 (2019) 1333–1340.
[43] S. Zhang, M. Du, J. Tong, Y.-F. Li, Multi-objective optimization of maintenance program in multi-unit nuclear power plant sites, Reliab. Eng. Syst. Saf. 188 (2019) 532–548.
[44] S. Martorell, A. Sánchez, S. Carlos, V. Serradell, Simultaneous and multi-criteria optimization of TS requirements and maintenance at NPPs, Ann. Nucl. Energy 29 (2002) 147–168.
[45] H. Ludwig, T. Salnikova, A. Stockman, U. Waas, Load cycling capabilities of German nuclear power plants (NPP), VGB PowerTech 91 (2011) 38–44.
[46] O. Eungse, L. Kangdae, Y. Sungok, Evaluation of commercial digital control systems for NPP I&C system upgrades, in: Proceedings of the Transactions of the, Korean Nuclear Society Spring Meeting, 2007.
[47] International Atomic Energy Agency *Non-baseload Operations in Nuclear Power Plants: Load Following and Frequency Control Modes of Flexible Operation*, IAEA, 2018. ISBN 9201108168.
[48] R.T. Rockafellar, S. Uryasev, Conditional value-at-risk for general loss distributions, J. Bank. Finance 26 (2002) 1443–1471.