A Sequential Decision Problem Formulation and Deep Reinforcement Learning Solution of the Optimization of O&M of Cyber-Physical Energy Systems (CPESs) for Reliable and Safe Power Production and Supply

Zhaojun Hao[1], Francesco Di Maio[1*], Enrico Zio[1,2]

[1]Energy Department, Politecnico di Milano, Milan, Italy
[2]Mines Paris, PSL Research University, CRC, Sophia Antipolis, France
e-mail: zhaojun.hao@polimi.it, francesco.dimaio@polimi.it, enrico.zio@polimi.it,
enrico.zio@mines-paristech.fr

Abstract: The Operation & Maintenance (O&M) of Cyber-Physical Energy Systems (CPESs) is driven by reliable and safe production and supply, that need to account for flexibility to respond to the uncertainty in energy demand and also supply due to the stochasticity of Renewable Energy Sources (RESs); at the same time, accidents of severe consequences must be avoided for safety reasons. In this paper, we consider O&M strategies for CPES reliable and safe production and supply, and develop a Deep Reinforcement Learning (DRL) approach to search for the best strategy, considering the system components health conditions, their Remaining Useful Life (RUL), and possible accident scenarios. The approach integrates Proximal Policy Optimization (PPO) and Imitation Learning (IL) for training RL agent, with a CPES model that embeds the components RUL estimator and their failure process model. The novelty of the work lies in *i*) taking production plan into O&M decisions to implement maintenance and operate flexibly; *ii*) embedding the reliability model into CPES model to recognize safety related components and set proper maintenance RUL thresholds. An application, the Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED), is provided. The optimal solution found by DRL is shown to outperform those provided by state-of-the-art O&M policies.

Keywords: Cyber-Physical Energy System (CPES), Operation & Maintenance (O&M), Deep Reinforcement Learning (DRL), Nuclear Power Plant (NPP), Optimization, Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED).

Abbreviations

The following abbreviations are used in this manuscript:

$\vec{C}_t$	Component state vector at time t
$\overrightarrow{MT}_t$	Times needed to complete the current maintenance vector at time t
$\vec{P}_t$	Production plan vector at time t
$\vec{R}_t$	RUL estimations vector at time t
$\vec{a}_t$	Action vector at time t
Π_{CM}	CM downtime
Π_{PM}	PM downtime
G_t	Revenue at time t
G_{water}	Water pump that regulates the feedwater mass flow rate
K_{base}	Base-load operation revenue
K_{load}	Load-following operation revenue
P_{Th}	Thermal power
$Q^{\pi}(s,a)$	Estimation of the expected future reward obtained by performing policy π, choosing action a in state s
R_l	Ground truth RUL of the l-th component
R_l^*	RUL estimation by PHM tools
$T_{L,cold}$	Cold leg lead temperature
T_M	Mission time
T_l	Ground truth failure time of the l-th component
T_{steam}	Steam temperature
U_{CM}	Cost for each downtime of CM
U_{PM}	Cost for each downtime of PM
U_{safe}	Cost of safe shutdown per unit of time
U_{severe}	Cost of severe shutdown per unit of time
W_t	Shutdown cost at time t
X_t	Maintenance cost at time t
p_{SG}	Steam Generator (SG) pressure
r_t	Reward at time t
y_{ref}	Controlled variable y reference value
$\vec{\theta}$	Policy search methods parameters

$\pi^*(a\|s)$	Optimal policy choosing action a in state s
ϵ_R	RUL estimation error
AGAN	As Good As New
ALFRED	Advanced Lead-cooled Fast Reactor European Demonstrator
CM	Corrective Maintenance
CPES	Cyber-Physical Energy System
CVaR	Conditional Value at Risk
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
GTST-MLD	Goal Tree Success Tree-Master Logic Diagram
IL	Imitation Learning
ML	Machine Learning
NPP	Nuclear Power Plant
O&M	Operation & Maintenance
PdM	Predictive Maintenance
PHM	Prognostic and Health Management
PI	Proportional Integral
PM	Preventive Maintenance
PPO	Proximal Policy Optimization
$\mathcal{R}$	The SDP reward function
RES	Renewable Energy Source
RL	Reinforcement Learning
RUL	Remaining Useful Life
SDP	Sequential Decision Problem
SM	Scheduled Maintenance
VaR	Value at Risk
CR	The control rods
L	Number of components
P	Power production plan
Sys_t	System state at time t
kv	Turbine admission valve that regulates the steam inlet mass flow rate
$\mathcal{A}$	SDP action space
$\mathcal{P}$	SDP transition probability
$\mathcal{S}$	SDP state space
γ	SDP discount factor
λ	Component failure rate

1. Introduction

Cyber-Physical Energy Systems (CPESs) are highly connected systems for energy production, transmission and distribution [1,2], for which high reliability and availability must be guaranteed by proper Operation & Maintenance (O&M) procedures [3,4].

Scheduled Maintenance (SM), e.g., on a time basis, is widely applied in industrial production [5,6]. On the other hand, the development of sensing and data analysis, and the advent of Prognostic and Health Management (PHM) techniques, have made it possible to collect and use condition monitoring data to estimate the components health states, and predict their Remaining Useful Life (RUL) [7–10], so as to enable the Predictive Maintenance (PdM) paradigm for just-in-time maintenance interventions that maximize system availability and minimize O&M costs [11,12].

On the other hand, the penetration of large shares of Renewable Energy Sources (RESs) onto the power grid, with their high degree of variability in power generation, challenges O&M to provide flexibility of operation (e.g., load-following [13]) for dealing with sudden imbalances between demand and production [14]. Then, O&M strategies should account for the components health state and Remaining Useful Life (RUL) [12,15–17], together with the variability of the power demand and generation over long-time horizons, ensuring flexible operation.

Recently, many researches have focused on O&M decision making: to name few, in [18] an artificial neural network is proposed to estimate the maintenance cost and, then used within a multi-agent Deep Reinforcement Learning (DRL) model to optimize decisions on large-scale systems; in [19] a Petri Net is applied to optimize offshore wind turbines O&M; in [20], a Bayesian Network maximizes a system supply capacity and gas supply reliability within a DRL scheme for maintenance planning. In all cases, however the fluctuations of the energy production and demands, their uncertainty, especially under increasing scenarios of penetration of RES specific production plans, have been overlooked. Also, the severity of the consequences of the CPES components failures is commonly neglected for simplicity.

In this paper, we formalize an optimization problem for such O&M strategies as a Sequential Decision Problem (SDP) to maximize productivity and safety, and provide flexible supply (load-following) to overcome the above mentioned limitations, i.e., we optimize the maintenance activities in light of the RUL of the CPES components, the severity of the consequences of their failures and the compliance with the operation plan (base-load or load-following) to satisfy the flexible operation needs while avoiding system shutdown caused by components failures. In a SDP, the goodness of the selected O&M action does not depend exclusively on the actual decision, but, rather, on the whole sequence of future decisions. To solve the SDP for the optimal O&M sequence of actions, we rely, as in [18–20], on Deep Reinforcement Learning (DRL), which is an extension of Reinforcement Learning (RL) and provides feasible application to complex systems [21,22]. RL has been applied to complex decision-making problems in many fields, including energy-related ones [23–32]. Indeed, tabular RL algorithms [33] allow finding the exact solution of SDPs in which the state and action spaces are small enough for the value function to be represented as tables. However, in most practical cases the computational cost of these algorithms is not compatible with the application to complex systems, whose state and action spaces are normally large due to the numerous components [33,34]. For this reason, we resort to DRL, which makes use of Deep Neural Networks (DNNs) to find approximate solutions [33]. In particular, we integrate the Proximal Policy Optimization (PPO) algorithm [35], which is one of the state-of-the-art approaches for DRL implementation, Imitation Learning (IL), which is a supervised learning approach [36] to pre-train the RL agent with a heuristic policy, and a CPES model that embeds the components RULs estimator and the components failure process model (i.e., the reliability model). A case study is provided concerning the Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED) [37]. This advanced Nuclear Power Plant (NPP) is designed precisely to offer flexible operation by providing the possibility of daily changing the power output between full (100%) power and 20% power levels. The main components of ALFRED, i.e., sensors, turbine admission valve, water pump and control rods, are considered equipped with RUL estimation capabilities. For the failure process, a Goal Tree Success Tree-Master Logic Diagram (GTST-MLD) reliability model is available [38,39]. In a nutshell, the novelty of the proposed approach lies in accounting for both the energy production plan and the CPES reliability model to inform O&M decisions that jointly consider production uncertainties and wear/tear of the CPES.

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

2. Problem Formulation

Let us consider a CPES whose load-following power production plan $P(t)$ to accommodate the RES fluctuations at each time $t = 1,2,\dots,T_M$ (the mission time), can span from full (100%) power (typically produced in base-load regime) to 20% (i.e., the minimum assumed in the daily cycles of load-following). For example, a load-following 100-60-100 cycle entails that in one day the load is 100% of the nominal power, then the load decreases by 40% to the 60% of the nominal power, then a power ramp is needed to re-establish the 100% full nominal power. Revenues are generated in both base-load and load-following operations, and are here indicated as K_{base} and K_{load}, respectively.

The CPES is made of L components: the generic l-th component, $l \in \Lambda = \{1, \dots, L\}$, is assumed to be equipped with PHM capabilities, which allow estimating its RUL. In [40–42], several advanced machine learning methods have been recently proposed to estimate the RUL, given the ground truth failure time T_l^* of the l-th component, is equal to:

$$R_l^* = T_l^* - t \tag{1}$$

And whose estimation provided by the PHM tool is:

$$R_l = R_l^* + \epsilon_R \tag{2}$$

where $\epsilon_R \sim N(0, \sigma_R)$ is a Gaussian noise representing the error of the RUL estimation [4,43].

Maintenance of the components is considered perfect, i.e., the component is restored as good as new (AGAN), and performed by a number of maintenance crews equal to the number of components in need of maintenance at the same time. The type of maintenance that is performed on the generic l-th component is: 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 due to PM and CM, Π_{PM} and Π_{CM} (typically $\Pi_{PM} < \Pi_{CM}$) are considered as a deterministic time period [44,45], and the costs for each downtime of maintenance are U_{PM} and U_{CM}, respectively. When a component fails, the system may undergo a safe shutdown or severe (damaged) shutdown, whose costs per unit of time are U_{safe} and U_{severe}, respectively.

For simplicity sake, but without loss of generality, we i) neglect backup components or safety-related protection systems (i.e., a component failure drives the system into failure, and CM is implemented), ii) assume that load-following operation can be implemented only when there are no components failed or under maintenance. It is important to point out that assumption *i*) neglects the fact that NPPs components are typically highly redundant for safety reasons, and so it allows providing conservative results, e.g., the upper boundary of the unreliability of the ALFRED control system; considering backup components can be done within a reliability analysis of the system, e.g., by Fault Trees or Reliability Block Diagram.

In this setting, the O&M problem can be formulated as a SDP defined by the set $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$, where:

- $\mathcal{S}$ is the state space, i.e., the set of variables describing the state of the system;
- $\mathcal{A}$ is the action space, i.e., the set of possible actions;
- $\mathcal{P}$ represents the transition probability, i.e., $\mathcal{P}(s'|s,a)$ is the probability of making a transition from state s to state s' by performing action a;

- $\mathcal{R}$ is the reward function, i.e., $\mathcal{R}(s'|s,a)$ is the reward received as a result of making a transition from state s to state s' by performing action a, and is used to update the O&M policy;
- $\gamma \in [0,1]$ is the discount factor, i.e., the factor used to evaluate the present value of future rewards.

The objective of solving the SDP is to define 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, in order to maximize the system profit over the mission time T_M. The state space $\mathcal{S}$, the action state $\mathcal{A}$, and the reward function $\mathcal{R}$, are defined in Sections 2.1, 2.2, and 2.3, respectively. In Section 2.4, the model of the CPES environment is described. Notice that, since in RL the learning agent directly interacts with the model of the environment, the explicit definition of the transition function $\mathcal{P}$ is not required.

2.1 State space $\mathcal{S}$

At each decision time t, the state space $\mathcal{S}$ is defined by the vector $\vec{s}_t = [\vec{R}_t, \vec{C}_t\, \overrightarrow{MT}_t, \vec{P}_t, Sys_t, t] \in \mathbb{R}^{3L+J+2}$, obtained appending the vectors of RUL estimations $\vec{R}_t = [R_1, R_2, \dots, R_L]$, the component state vector (operating, failed, CM and PM) $\vec{C}_t = [C_1, C_2, \dots, C_L]$, the vector of the times needed to complete the current maintenance $\overrightarrow{MT}_t = [MT_1, MT_2, \dots, MT_L]$, the production plan vector for consecutive J days from day t to day $t+J-1$ $(J=1,2,\dots,T_M-t+1)$ $\vec{P}_t=[P_0,P_1,\dots,P_{J-1}]$, and the system state (operating, PM, shutdown and failure).

2.2 Action space $\mathcal{A}$

At each decision time t, the maintenance actions space $\mathcal{A}$ is defined by the vector $\vec{a}_t=[a_1,\dots,a_l,\dots,a_L]$: if a decision is taken to maintain the l-th component, the corresponding a_l is set to 1, resulting in $\vec{a}_t=[0,\dots,0,a_l=1,0,\dots,0]$, or $\vec{a}_t=[0,\dots,0]$ otherwise.

2.3 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:

$$r_t=G_t-W_t-X_t \tag{3}$$

where G_t is the revenue (see Eq. (4) below), W_t is the cost when the system is under safe shutdown or severe shutdown (see Eq. (5) below) and X_t is the maintenance intervention cost (see Eq. (6) below).

G_t can be calculated as follows:

$$G_t=I_{base}\cdot K_{base}+I_{load}\cdot K_{load} \tag{4}$$

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:

$$W_t = I_{safe} \cdot U_{safe} + I_{severe} \cdot U_{severe} \tag{5}$$

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:

$$X_t = \sum_{l=1}^{L} I_l^{RUL>0} \cdot U_{PM} + I_l^{RUL=0} \cdot U_{CM} \tag{6}$$

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.

2.4 The environment model

Despite the agent may in principle find the optimal O&M policy by means of direct interactions with the real-world system, this turns out to be unfeasible in the case of CPES for economic, safety and time issues: the trial-and-error nature of the learning process consists in performing several times the actions suggested by the algorithm to explore the solution space, leading to economically inconvenient and unsafe system management in the early stage of the learning process (when they are not yet optimal); thus, the learning agent is typically trained using a white-box environment model of the system of interest [4].

The model here developed that reflects the complex response of the

CPES to failure scenarios depending on the large variety of system information (e.g., components RUL, components state, load-following operation plan), consists in a model of the system which can simulate its response in the scenarios by the components failures, and in the estimator of the components RUL, which provides the estimate of the RUL in the form of R_l of Eq.(2).

3. Reinforcement Learning algorithms

A schematic view of the RL procedure used in this paper is shown in Fig. 1. The decision maker is indicated as the agent and the system it interacts with is called environment: they interact continuously, the agent selecting actions and the environment responding to those actions with a reward that the agent tries to maximize over time [33]. 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 $\vec{C}_t$, the maintenance remaining times $\overrightarrow{MT}_t$, the production plan $\vec{P}_t$ and the system state Sys_t), and 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 iteratively repeating this procedure several times in a trial-and-error manner, 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 $E[\sum_{t=0}^{T_M} \gamma^t \cdot r_t(\vec{a}_t, \vec{s}_t, \vec{s}_{t+1})]$, where γ is the discount parameter of future rewards.

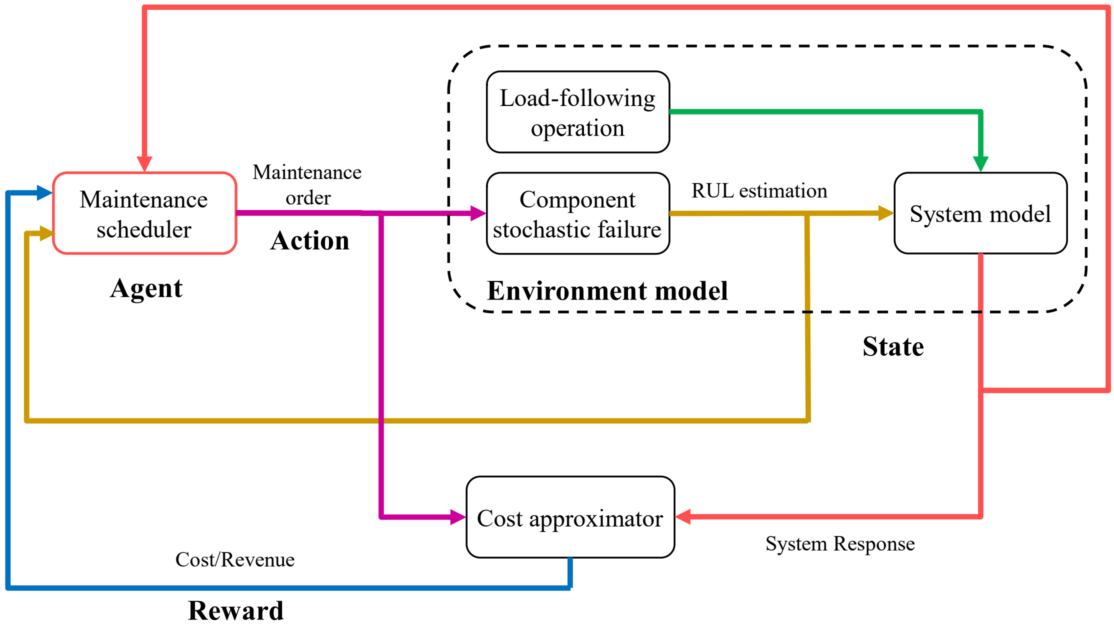


Fig. 1. Schematic representation of RL procedure.

In general, RL algorithms can be classified into three groups: *policy search*, *value function* and *actor-critic* methods [22]. Policy search methods directly look for the optimal policy, $\pi^*(a|s)$, by updating the parameters, $\vec{\theta}$, of a parameterized policy, $\pi(a|s;\vec{\theta})$, through which optimal actions are selected. These methods typically converge to a local optimum rather than to the global optimum [21].

Differently, value function methods learn the value of selecting a particular action when being in a particular state, $Q^\pi(s,a)$, which is an estimation of the expected future reward obtained by performing action a

in state s, and, then following the policy $\pi(a|s)$. In this way, the optimal policy, $\pi^*(a|s)$, is the one that maximizes the action-value function $Q^{\pi^*}(s,a)$:

$$\pi^*(a|s) = argmax_a \, Q^{\pi(a|s)}(s,a) \qquad (7)$$

Actor-Critic methods learn both the value function and the policy in an attempt to combine the strong points of value function and policy search methods [22]. Actor-Critic methods consist of two models: the critic, which learns the value function, and the actor, which learns the policy by updating the parameters in the direction suggested by the critic.

In the simplest cases, i.e., those in which the state and action spaces are small, tabular RL can be implemented. In tabular RL the learning agent is represented as a table which stores the state-value (goodness of policy) or action-value (goodness of action in the state). Although tabular RL leads to find the exact optimal solution, its computational cost makes the applications unfeasible to complex systems characterized by large or continuous state and/or action spaces [34]. Then, function approximation has been introduced to approximate the state-value function or the action-value function [46]. In principle, linear approximation can be implemented for function approximation using different basis functions, e.g., polynomial basis or Fourier basis [23]. Deep Neural Networks (DNNs) have recently been successfully used for non-linear function approximation, within a

DRL framework. Indeed, the use of DNNs in continuous and high-dimensional state spaces makes it possible to extract hidden features, which enable the DRL agent to overcome the uncertainty and partial observability of the environment.

In this work, we adopt the state-of-the-art RL algorithm, Proximal Policy Optimization (PPO) [35] to optimize the O&M strategy of a CPES. PPO is an actor-critic algorithm, which aims at stabilizing the policy optimization by constraining the gradient updates, in the attempt to monotonically improve the policy. The main idea is to avoid too large policy updates, which can increase the probability of accidental performance collapses. PPO is considered relatively easy to implement and tune, and despite its simplicity, it has been shown able to outperform many state-of-the-art approaches on discrete and continuous benchmarks [35] and on several applications in different research fields, such as supply chains [47], autonomous vehicles [48] and power production plants [4,30–32].

Since in complex system applications the state space is very large, it can be hard for the agent to discover the optimal policy $\pi^*(a|s)$ in an efficient way starting from a random initialization of the neural network. This problem has been tackled by including domain knowledge in the learning process, using methods such as reward shaping [49] and state-action similarity solutions [50]. In this work, we resort to Imitation

Learning (IL) [36], in particular, Behavioral Cloning [51], to generate trajectories with a heuristic policy, for pre-training the agent to reproduce the heuristic policy in a supervised learning framework. In other words, the heuristically generated trajectories are used as training data for the policy neural network to learn to pair the state $\vec{s}_t$ and the chosen action $\vec{a}_t$. Then, the agent is fine-tuned using RL to explore new policies and discover the optimal one. The interested reader may refer to [36,52–54] for a detailed description of the DRL framework here adopted.

4. Case Study: The Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED)

As a promising technology capable of meeting the Generation IV goals of Nuclear Power Plants (NPPs), ALFRED [55,56], a conceptual reactor design within the European nuclear community, has been the subject of various studies of reactor design [37,57–59], control design [60,61], and reliability and risk analysis [62,63]. ALFRED is designed to operate in a flexible way [64–66] and is expected to reach operational conditions of an industrially deployable small modular lead fast reactor around the year 2035-2040 [55], making it a perfect candidate of NPPs to be considered for coping with the variability of RESs, within a load-following schedule.

The control of ALFRED is implemented by means of four feedforward and Proportional Integral (PI)-based feedback control loops (see Fig. 2)

[37], that keep four variables $\vec{y}$ (steam temperature T_{steam}, Steam Generator (SG) pressure p_{SG}, cold leg lead temperature $T_{L,cold}$ and thermal power P_{Th}) controlled at reference values $\vec{y}_{ref}$ in full power nominal condition, and within the safety thresholds ($\vec{L}_y$ and $\vec{U}_y$) in any other operational condition (see Table I). The control system is here simplified as composed of $L = 7$ hardware components (4 sensors for the variables T_{steam}, p_{SG}, $T_{L,cold}$ and P_{Th}, and 3 actuators for the turbine admission valve (kv), the water pump (G_{water}) and the control rods (CR)). Such control system is not only exposed to components stochastic failures, but also to cyber failures that can contribute to the ALFRED unreliability [63,67,68]; however in this work, the components are considered subjected only 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 $\sigma_R = 10$ days (see Eqs. (1) and (2)). The failure time T_l^* of each component is sampled from an exponential distribution; the failure rates for the components are listed in Table II.

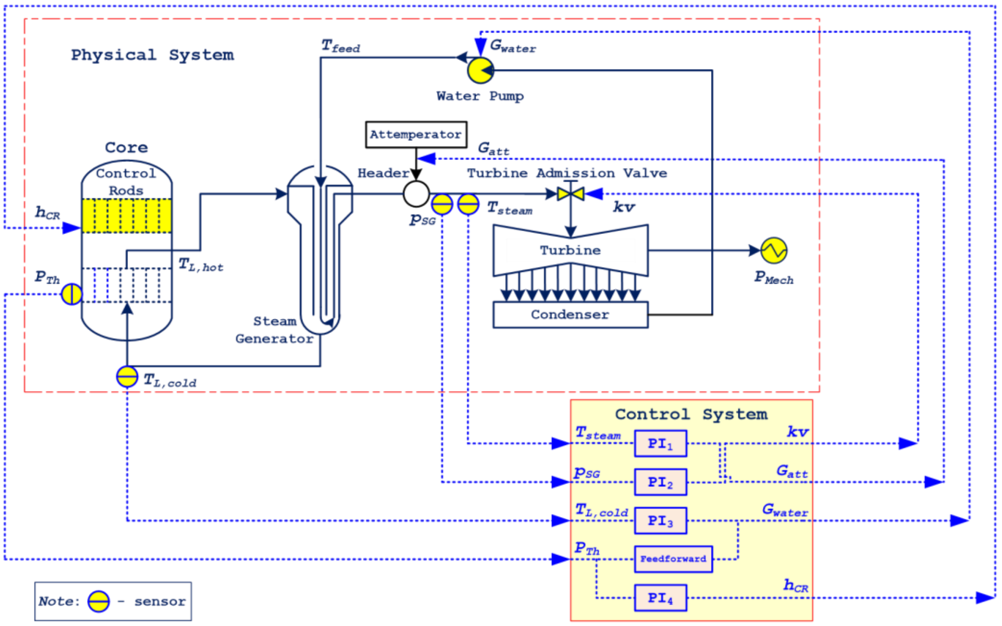


Fig. 2. ALFRED reactor control system [63].

TABLE I. Reference value and safety thresholds of the controlled variables [57]

Controlled variable, y	Reference value at full power nominal condition, $\boldsymbol{y_{ref}}$	Lower safety threshold, $\boldsymbol{L_y}$	Upper safety Threshold, $\boldsymbol{U_y}$
T_{steam} (°C)	450	/	550
p_{SG} (Pa)	180E5	170E5	190E5
$T_{L,cold}$ (°C)	400	350	/
P_{Th} (W)	300E6	270E6	330E6

TABLE II. Components failure rates [69].

Failure rate/occurrence probability	Value
Sensor failure rate λ_{sensor}	6.20E-3/Year
Turbine admission valve (kv) failure rate λ_{kv}	6.57E-4/Year
Water pump (G_{water}) failure rate λ_{water}	1.14E-2/Year
Control rods (CR) failure rate λ_{CR}	5.30E-3/Year

We assume that *i*) the production plan $\vec{P}_t$ (base-load or load-following with respect to the probabilities listed in Table III (the load-following cycle ranges from 100% to 40% of the normal power for normal operation conditions, whereas it can drop to 20% of the normal power, as explained in [70])) for $J = 2$ successive days is known, i.e., $\vec{P}_t = [P_0, P_1, P_2]$, *ii*) the maintenance durations Π_{PM} and Π_{CM} are considered as deterministic time periods $\Pi_{PM} = 1.25$ days [71] and $\Pi_{CM} = 3\ .37$ days [72], respectively, *iii*) the daily revenues and maintenance costs of PM and CM are as listed in Table IV.

TABLE III. NPP load-following capability [68,73].

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

Load Cycle	Number of Load Cycles in 70 years lifetime	Probability per day
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 IV. Daily revenues and maintenance costs [71,74,75].

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

The ALFRED system model we use in the RL environment is a goal-oriented logical model, based on the Goal Tree Success Tree-Master Logic Diagram (GTST-MLD) (see Appendix B). Once a component is failed, the ALFRED system is considered being in safe shutdown and not able to continue operating. After initializing the components state and propagating the component failure through the GTST-MLD (the interested reader may refer to [69,76] for implementation details), the GTST-MLD reliability model can evaluate the system response with respect to whether the component failure leads the four controlled variables (T_{steam}, p_{SG}, $T_{L,cold}$ and P_{Th}) out of the safety thresholds, which is considered to be a system failure leading to severe consequences. In other words, the system structure and functionality are described by the hierarchical framework GTST and the system response to the components failure are simulated by the MLD in a transparent way [38,39]. Thus, the GTST-MLD system model can be considered as a white-box model which can be applied as the RL environment model and used to simulate the interactions with the agent.

As RL agent, based on the settings in [4,13], 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-reward triplets 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 [4].

5. Results

For a fair comparison of the PPO (GTST-MLD) RL, 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 4), *iv*) a PPO RL (same RL without GTST-MLD, as the one shown in [77]). 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), respectively.

In this paper, we use Conditional Value at Risk (CVaR) 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 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). In other words, CVaR provides a measure of the extent of the losses that might be suffered beyond the VaR cut-off threshold [78].

We rely on a Monte Carlo simulation approach to calculate the 95% CVaR with respect to 100 different test sequences (for each strategy), in which we simulate the sequence of O&M decisions, collect the losses (including maintenance cost, safe shutdown cost, severe shutdown cost and load-following operation unfulfillment cost (i.e., the difference between load-following and base-load profits)): the lower the CVaR estimate, the lower the losses and the less the number of safe/severe shutdowns. 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 and PM actions needed in the sequence mission time.

TABLE V. 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 $[10^9\text{euro}]$ (Ranking)	95% CVaR $[10^9\text{euro}]$ (Ranking)	Average number of CM (Ranking)	Average number of PM (Ranking)
Corrective	0.09 ± 0.13 (5)	1.41 ± 0.88 (5)	38.12 ± 5.64 (5)	-
Scheduled	0.53 ± 0.12 (4)	0.93 ± 0.53 (4)	24.32 ± 1.98 (4)	60.47 ± 6.55 (4)
Predictive	1.18 ± 0.07 (3)	0.30 ± 0.17 (3)	$\mathbf{0.03 \pm 0.01}$ **(1)**	44.13 ± 5.86 (3)
PPO	1.39 ± 0.02 (2)	0.04 ± 0.03 (2)	0.05 ± 0.03 (3)	42.03 ± 3.98 (2)
PPO (GTST-MLD)	$\mathbf{1.44 \pm 0.02}$ **(1)**	$\mathbf{0.01 \pm 0.01}$ **(1)**	0.04 ± 0.02 (2)	$\mathbf{41.97 \pm 4.06}$ **(1)**

* In bold the best performance

From Table V, it can be noticed that the CM and SM policies, which are commonly used [79,80], cause a large number of components failures, leading to an average of 38.12 and 24.32 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). The PdM, PPO and PPO (GTST-MLD) policies perform better than CM and SM (PPO (GTST-MLD) has the highest profit), due to the exploitation of the information on the health state of the components: these three policies allocate just-in-time PM actions (44.13, 42.03 and 41.97 on average, respectively) to avoid system dysfunction (0.03, 0.05 and 0.04 on average, respectively) and, therefore, the consequent CM. The number of PM actions of PPO (42.03) and PPO (GTST-MLD) (41.97) are slightly smaller than PdM (44.13), due to the smaller average RUL thresholds (35 days for PdM policy, 31.2 days and 31.4 days on average for PPO and PPO (GTST-MLD) policies, respectively) shown in Table VI (in fact, smaller average RUL threshold means larger average maintenance interval and less interventions). From Table VI, it can be noticed that even if the PPO and PPO (GTST-MLD) agents are pre-trained with the same PdM policy, the optimized RL agent founds different RUL thresholds setting: the thresholds of PPO policy are close to the average value (31.2 days), whereas the thresholds of PPO (GTST-MLD) (31.4 days) follow the weights of MLD listed in Table VI, which shows the relationship between components and system goal function (e.g., the MLD weight linking sensor p_{SG} and goal function p_{SG} control (0.69) means that when the sensor p_{SG} fails, there is 0.69 probability that the controlled variable p_{SG} will be out of the safety boundary, causing system severe shutdown) (for further details see Appendix A). The PPO (GTST-MLD) recognizes the safety-related components with larger MLD weights (sensor p_{SG} (0.69), sensor P_{Th} (0.98) and control rods (0.58)) and sets higher RUL thresholds (sensor

p_{SG} (46.2 days), sensor P_{Th} (52.6 days), and control rods (43.3 days)) to maintain these components in advance for preventing these safety-related components from failure, since they have high probability of leading to system severe shutdown. The average number of safe shutdowns and severe shutdowns over 100 test sequences are listed in Table VII. With the components safety importance information (the GTST-MLD weights) and reasonable setting of the component RUL (higher RUL threshold for larger weights components), the PPO (GTST-MLD) efficiently avoids system severe shutdown (0.00 ± 0.01, leading to the lowest 95% CVaR 0.01 ± 0.01, shown in Table V), whereas PdM and PPO policies suffer severe shutdown times (0.03 ± 0.01 and 0.03 ± 0.02, respectively).

TABLE VI. Components RUL thresholds of maintenance interventions and corresponding GTST-MLD weights.

Components	RUL threshold of PPO policy [days]	RUL threshold of PPO (GTST-MLD) policy [days]	GTST-MLD weights			
			$\boldsymbol{T_{steam}}$ control	$\boldsymbol{p_{SG}}$ control	$\boldsymbol{T_{L,cold}}$ control	$\boldsymbol{P_{Th}}$ control
Sensor T_{steam}	33.5	27.9	0	0	0	0
Sensor p_{SG}	32.1	46.2	0.35	0.69	1.54E-5	0.12
Sensor $T_{L,cold}$	29.7	27.7	0	0.09	0	0
Sensor P_{Th}	28.8	52.6	0.11	0.72	0	0.98

Components	RUL threshold of PPO policy [days]	RUL threshold of PPO (GTST-MLD) policy [days]	GTST-MLD weights			
			$\boldsymbol{T_{steam}}$ control	$\boldsymbol{p_{SG}}$ control	$\boldsymbol{T_{L,cold}}$ control	$\boldsymbol{P_{Th}}$ control
Turbine admission valve (kv)	30.4	28.1	0	0	0	0
Water pump (G_{water})	32.7	28.5	0	0	0	2.50E-3
Control rods (CR)	29.9	43.3	0.06	0.58	0	0.05
Average RUL threshold	31.2	31.4	--			

TABLE VII. Performance of the tested strategies in terms of average number of safe/severe shutdowns in 100 test sequences.

Maintenance strategy	Average number of safe shutdowns (Ranking)	Average number of severe shutdowns (Ranking)
Predictive	$\mathbf{0.01 \pm 0.01}$ **(1)**	0.03 ± 0.01 (2)
PPO	0.02 ± 0.02 (2)	0.03 ± 0.02 (3)
PPO (GTST-MLD)	0.04 ± 0.02 (3)	$\mathbf{0.01 \pm 0.01}$ **(1)**

In Fig. 3, the number of actions performed during specific power production plans are plotted for PPO (GTST-MLD), PPO and PdM policies (slash, 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 II. 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 RL policy (PPO (GTST-MLD) and PPO, slash and dotted bars, respectively) 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 postpone the PM interventions from a load-following day to a base-load day, to accommodate the frequency of occurrence of the preferred production plans. In other words, the RL agent chooses the actions in light of the desired operation plan (i.e., flexible operation) by optimizing the timing of maintenance activities.

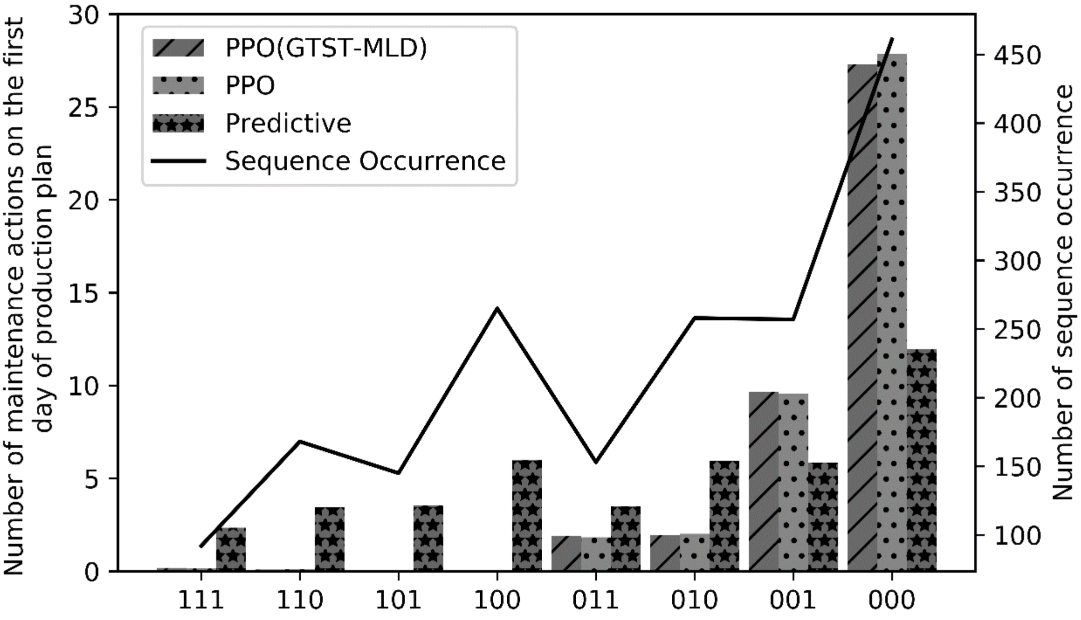


Fig. 3. Maintenance timing and power production demand sequence occurrence over 100 test sequences.

In addition, it must be noted that, even if the RL policy can preserve the load-following operations with a more aggressive O&M policy, that requires a smaller average RUL threshold, the resulting ALFRED unreliability and unavailability estimation are larger for PPO (GTST-MLD) (dotted line) and PPO (dashed-dotted line) than for PdM O&M (dashed line) (see Fig. 4 and Fig. 5): as expected, compared with the no-maintenance policy (continuous line), all three policies significantly decrease the unreliability, with PdM the lowest unreliability and PPO (GTST-MLD) and PPO slightly larger unreliability value. The reason is that the smaller average RUL thresholds of the PPO (GTST-MLD) and PPO may lead to a larger number of unexpected safe/severe shutdowns than PdM. But due to the larger RUL thresholds setting for safety-related components, PPO (GTST-MLD) can avoid part of the unexpected safe/severe shutdowns, leading to a lower unreliability than the PPO policy. The same occurs for the unavailability.

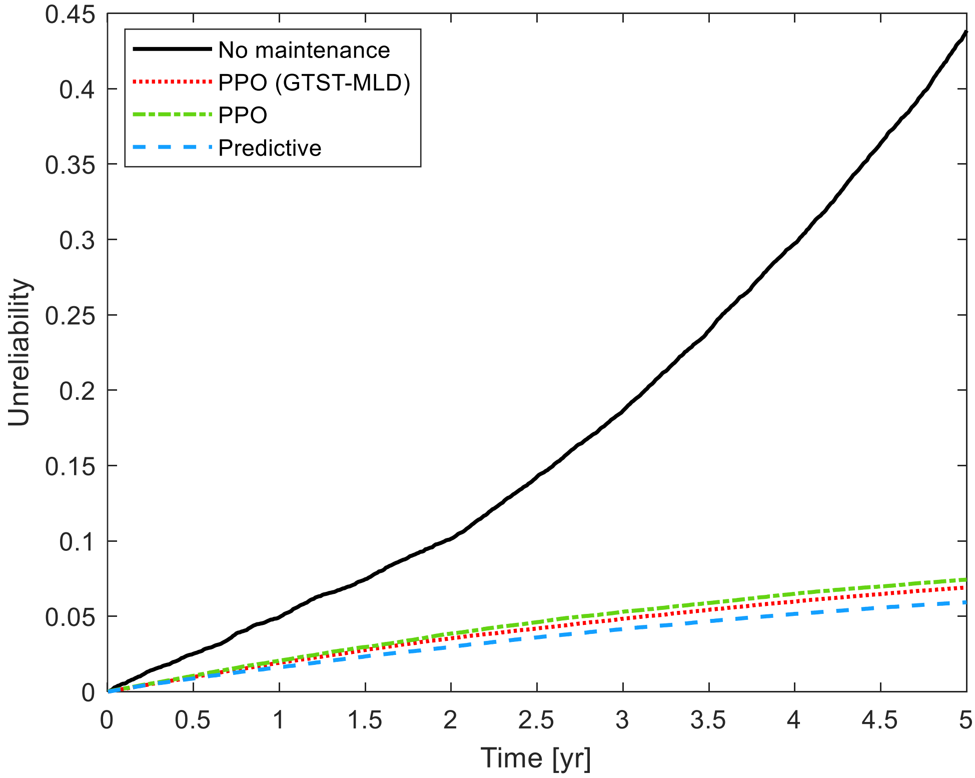


Fig. 4. System unreliability.

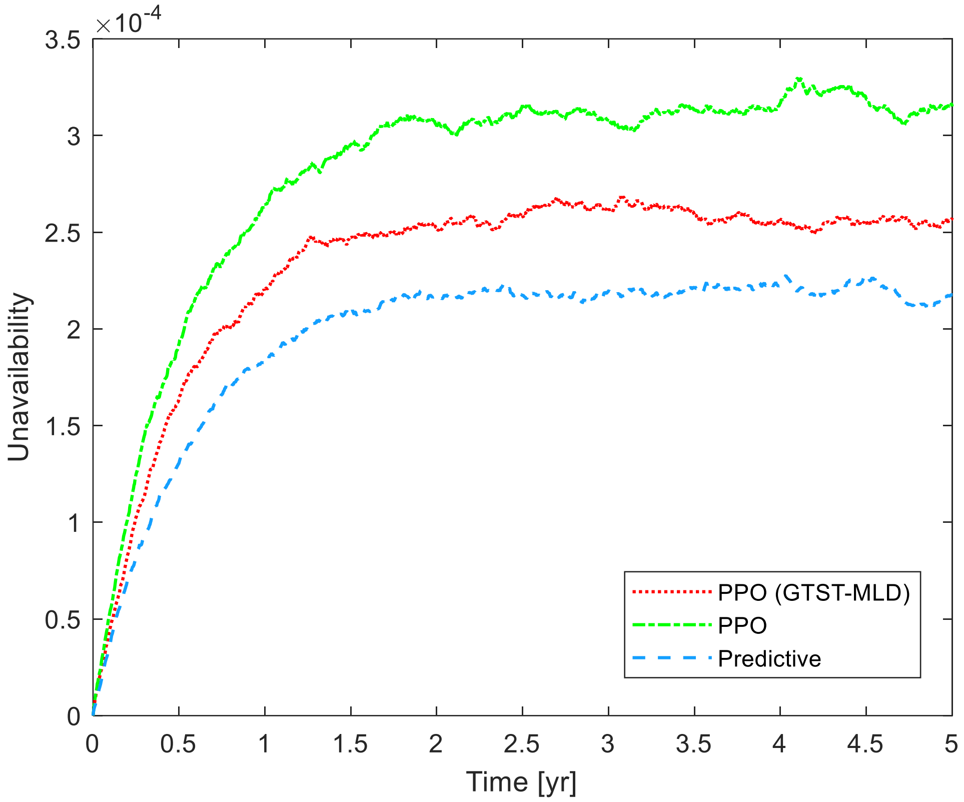


Fig. 5. System unavailability.

6. Conclusions

In this paper, we have illustrated the SDP formalization of the O&M optimization in CPESs that must operate flexibly to accommodate the fluctuations in production coming from the penetration of RESs into the power grid and the uncertainty in power demand, for providing reliable and safe power production and supply. A novel DRL-based approach has been developed to solve the SDP, in which an agent-neural network is trained by interacting with the CPES model that embeds the system failure process model to search for the optimal policy, i.e., choose the best 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) and learning from the set of previous maintenance activities performed.

The proposed approach has been applied to an advanced NPP design, ALFRED, and shown to be capable of providing an optimized O&M policy that tends to dynamically arrange the maintenance interventions on the base-load days, to preserve flexible operation as much as possible, i.e., the proposed approach optimizes the maintenance activities in light of the RUL of the CPES components, the severity of the consequences of their failures and the compliance with the operation plan (base-load or load-following) to satisfy the flexible operation needs while avoiding system shutdown caused by components failures. With the system reliability model by GTST-MLD, the DRL-based approach can recognize the safety-related components and set higher RUL thresholds to prevent system severe shutdown due to their critical failures. The DRL-based policy proposed here can outperform the state-of-practice policies (CM, SM, PdM and PPO without GTST-MLD) and keep the production availability and profitability high (and the costs low).

Future works will regard:

- Train the RL agent to obtain the proper maintenance activity considering not only the components stochastic failures, but also cyber aging and cyber failures

- Due to the variability of the dynamic energy market and RES, the fluctuation of energy price and power generation will affect the profit, with clear effects on the selection of the O&M strategy. Thus, the integral analysis and joint prediction of energy price, power generation and demands should be taken into consideration.
- Besides the failure of physical and cyber parts of CPESs, the effects of the external environment should also be accounted for in the O&M strategy, for example, the unavailability of cooling water due to climate change and abnormal weather conditions [81].

Appendix A. Imitation Learning and Reinforcement Learning

Imitation learning (IL) is a type of supervised machine learning technique which is used for various tasks, including control, decision making and manipulation, and has shown promising results in several fields such as robotics, autonomous driving, and gaming [51].

In this work, IL is applied to pre-train the O&M decision making agent before the Reinforcement Learning (RL) step [4]. After IL, the agent model learns to perform the O&M decision task by observing and imitating the behavior of the given heuristic Predictive Maintenance (PdM) strategy. This is typically done by training the agent (neural network) to predict the actions $\vec{a}_t$ that the heuristic PdM strategy would take in a given CPES state $\vec{s}_t$ and get the reward r_t. The implementation of Imitation learning can be summarized as:

1. Collect heuristic trajectories: this involves obtaining the heuristic PdM strategy state-action-reward triplets $(\vec{s}_t, \vec{a}_t, r_t)$ through each step of N_{IL} times of CPES environment model simulation within the mission time (in this work, $N_{IL} = 500$), which are fed to the agent model as the supervised training dataset.
2. Define the agent model: this involves defining and initializing the agent neural network. In this work, the neural network has two hidden layers of 64 neurons.
3. Train the model: this involves feeding the training data (state-

action-reward triplets) into the model and training it using a supervised learning algorithm.

The pseudocode of IL is shown in Fig. A1.

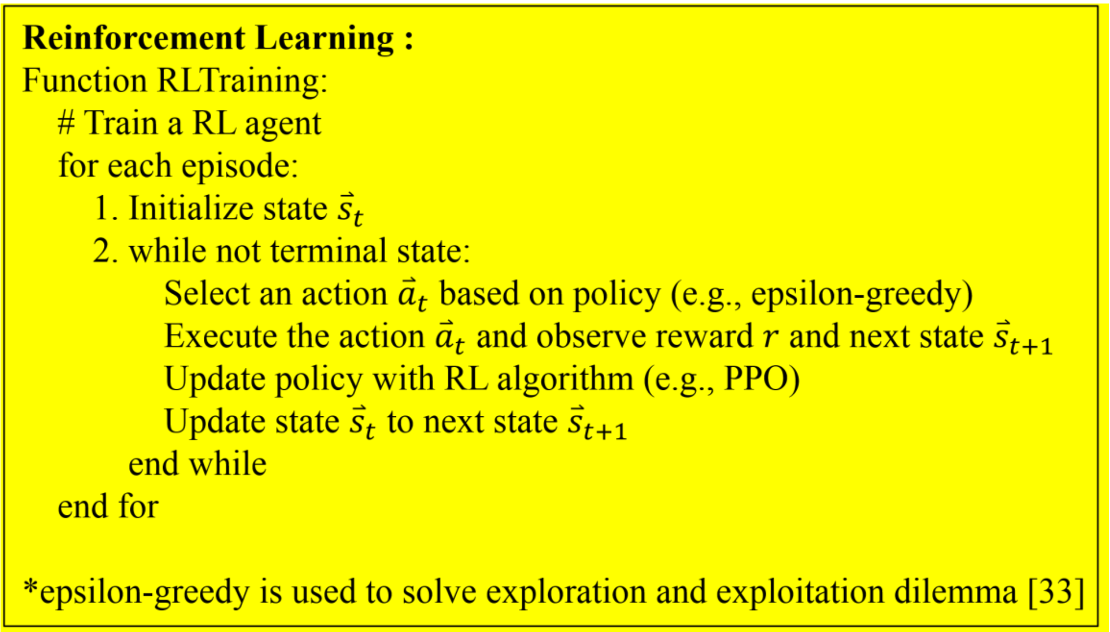
```
Imitation Learning:
Function CreateDataset ($N$):
    # Create a dataset of state-action-reward pairs ($\vec{s}_t, \vec{a}_t, r_t$)
    1. for $N_{IL}$ times:
            Run system simulation until reaching the mission time and
            record ($\vec{s}_t, \vec{a}_t, r_t$) to dataset at each simulation step
    2. Return dataset

Function ImitationLearning (dataset, model)
    # Train the agent to mimic expert behavior using supervised learning
    1. Initialize agent neural network with random weights
    2. for each iteration:
            Sample batch of state-action-reward triplets ($\vec{s}_t, \vec{a}_t, r_t$) from dataset
            Calculate $loss$ between heuristic PdM actions $\vec{a}_t$ and agent neural
            network prediction
            Update model weights using gradient descent and $loss$
        end for
    3. Return model
```

Fig. A1. Pseudocode for IL.

After pre-training the agent by IL, the RL can be implemented as follows: at each decision time t, the agent receives a representation of the environment state $\vec{s}_t$, and the agent will select an action $\vec{a}_t$ based on the action selection policy (e.g., epsilon-greedy [33]). The environment system model simulates the system response $\vec{s}_{t+1}$ to the selected action $\vec{a}_t$, and returns the corresponding numerical reward r_t to the agent. Then, the agent policy is updated with RL algorithm (in this work, PPO is implemented [35]) and the current state is set to $\vec{s}_{t+1}$. By iteratively repeating this procedure several times, the agent reaches the optimal policy.

The pseudocode of implementing training of RL is shown in Fig. A2.



```
Reinforcement Learning :
Function RLTraining:
    # Train a RL agent
    for each episode:
        1. Initialize state $\vec{s}_t$
        2. while not terminal state:
                Select an action $\vec{a}_t$ based on policy (e.g., epsilon-greedy)
                Execute the action $\vec{a}_t$ and observe reward $r$ and next state $\vec{s}_{t+1}$
                Update policy with RL algorithm (e.g., PPO)
                Update state $\vec{s}_t$ to next state $\vec{s}_{t+1}$
            end while
    end for

*epsilon-greedy is used to solve exploration and exploitation dilemma [33]
```

Fig. A2. Pseudocode for RL training.

Appendix B. The Goal Tree Success Tree-Master Logic Diagram (GTST-MLD) Model

The Goal Tree Success Tree-Master Logic Diagram (GTST-MLD) method has been proposed to analyze the scenarios generated by the combination of stochastic failures of the hardware components and malicious, intentional attacks to the cyber elements of a CPES [69,76]. The GTST-MLD provides a comprehensive modeling of the system response. It does so by decomposing the system logic from the point of view of goal functions in the Goal Tree (GT), down to the components and functions that they provide, represented in the Success Tree (ST) and MLD [82]. The GTST-MLD for ALFRED is shown in Fig. B1.

The MLD weights [38] represent the strength of the relationship between components and functions. To overcome the subjectivity of expert judgment in assigning the weights $CF_{c,f}$ (relationship between main components and subfunctions), a simulation-based inference method is proposed in [38] based on Bracketing Order Statistics [83] to estimate the weights probabilistically. The estimated weights are listed in Table B1 [38]. As commonly done when using GTST-MLD, after initialization and propagation step, the top goal function fulfillment can be simulated considering the AND/OR logic gates that define the relationships among subfunctions and the top goal function [38,69,76].

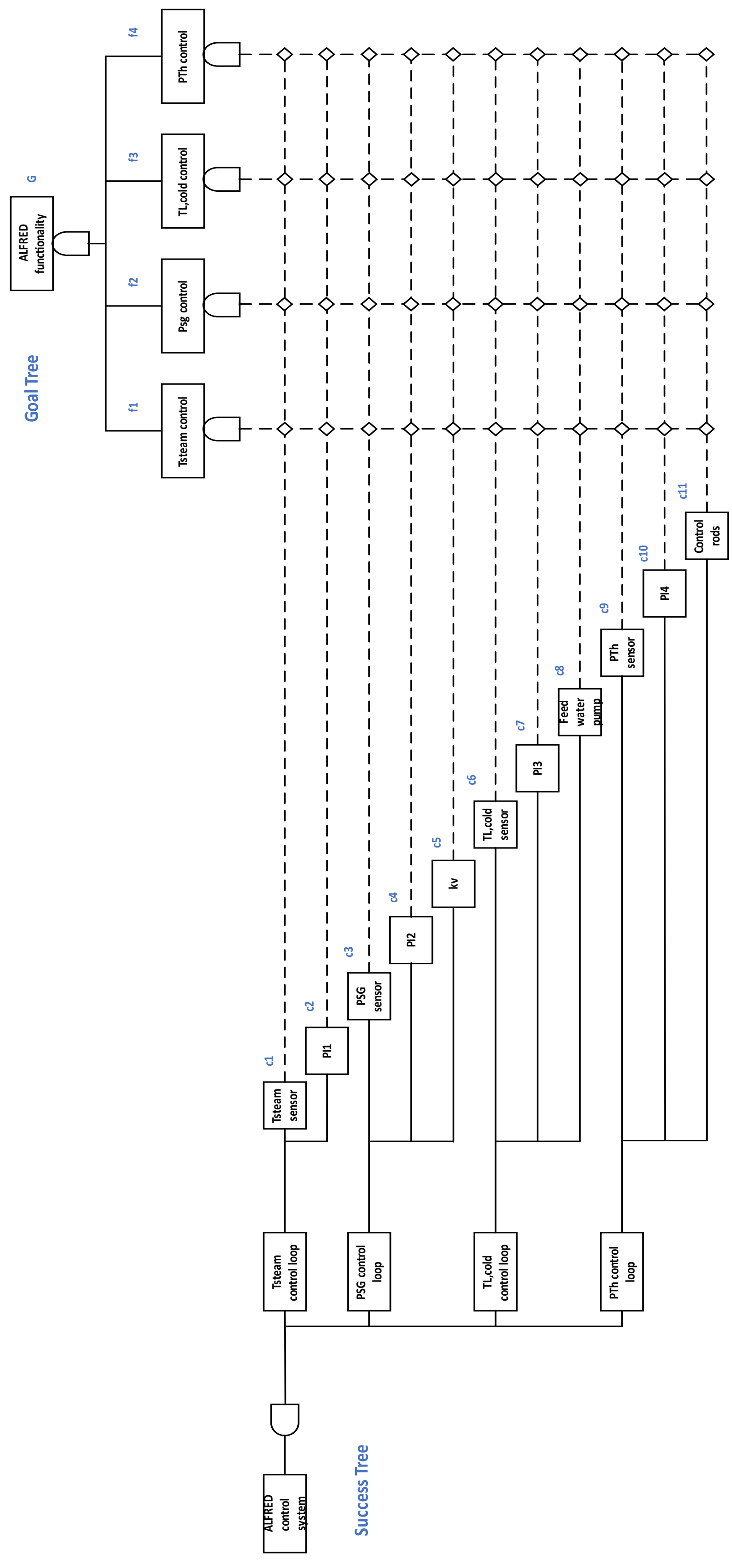


Fig. B1. GTST-MLD for ALFRED

TABLE B1. Weights between main components and subfunctions

	$\boldsymbol{T_{steam}}$ control	$\boldsymbol{p_{SG}}$ control	$\boldsymbol{T_{L,cold}}$ control	$\boldsymbol{P_{Th}}$ control
Sensor for T_{steam}	0	0	0	0
Sensor for p_{SG}	0.35	0.69	1.54E-5	0.12
Sensor for $T_{L,cold}$	0	0.09	0	0
Sensor for P_{Th}	0.11	0.72	0	0.98
Turbine admission valve (kv)	0	0	0	0
Water pump (G_{water})	0	0	0	2.50E-3
Control rods (CR)	0.06	0.58	0	0.05

References:

1. Zio, E. Challenges in the vulnerability and risk analysis of critical infrastructures. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 137–150.
2. Zio, E. The future of risk assessment. *Reliab. Eng. Syst. Saf.* **2018**, doi:10.1016/j.ress.2018.04.020.
3. Lee, J.; Bagheri, B.; Kao, H.-A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manuf. Lett.* **2015**, *3*, 18–23.
4. Pinciroli, L.; Baraldi, P.; Ballabio, G.; Compare, M.; Zio, E. Optimization of the Operation and Maintenance of renewable energy systems by Deep Reinforcement Learning. *Renew. Energy* **2022**, *183*, 752–763.
5. Jiejuan, T.; Dingyuan, M.; Dazhi, X. A genetic algorithm solution for a nuclear power plant risk–cost maintenance model. *Nucl. Eng. Des.* **2004**, *229*, 81–89.
6. Labib, A.W.; Yuniarto, M.N. Maintenance strategies for changeable manufacturing. In *Changeable and Reconfigurable Manufacturing Systems*; Springer, 2009; pp. 337–351.
7. Zio, E. Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108119.
8. Baraldi, P.; Mangili, F.; Zio, E. Investigation of uncertainty

treatment capability of model-based and data-driven prognostic methods using simulated data. *Reliab. Eng. Syst. Saf.* **2013**, *112*, 94–108.

9. Zio, E.; Di Maio, F. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliab. Eng. Syst. Saf.* **2010**, *95*, 49–57.

10. Di Maio, F.; Baraldi, P.; Zio, E.; Seraoui, R. Fault detection in nuclear power plants components by a combination of statistical methods. *IEEE Trans. Reliab.* **2013**, *62*, 833–845.

11. Rothwell, G. *Economics of nuclear power*; Routledge, 2018; ISBN 1317511786.

12. Compare, M.; Baraldi, P.; Zio, E. Challenges to IoT-enabled predictive maintenance for industry 4.0. *IEEE Internet Things J.* **2019**, *7*, 4585–4597.

13. Pinciroli, L.; Baraldi, P.; Ballabio, G.; Compare, C.; Zio, E. 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.

14. Pierobon, L.; Casati, E.; Casella, F.; Haglind, F.; Colonna, P. Design methodology for flexible energy conversion systems

accounting for dynamic performance. *Energy* **2014**, *68*, 667–679.

15. Ustundag, A.; Cevikcan, E. *Industry 4.0: managing the digital transformation*; Springer, 2017; ISBN 3319578707.
16. Tjahjono, B.; Esplugues, C.; Ares, E.; Pelaez, G. What does industry 4.0 mean to supply chain? *Procedia Manuf.* **2017**, *13,* 1175–1182.
17. Okoh, C.; Roy, R.; Mehnen, J.; Redding, L. Overview of remaining useful life prediction techniques in through-life engineering services. *Procedia Cirp* **2014**, *16*, 158–163.
18. Nguyen, V.-T.; Do, P.; Vosin, A.; Iung, B. Artificial-intelligence-based maintenance decision-making and optimization for multi-state component systems. *Reliab. Eng. Syst. Saf.* **2022**, *228*, 108757.
19. Saleh, A.; Chiachío, M.; Salas, J.F.; Kolios, A. Self-adaptive optimized maintenance of offshore wind turbines by intelligent Petri nets. *Reliab. Eng. Syst. Saf.* **2023**, *231*, 109013.
20. Fan, L.; Su, H.; Wang, W.; Zio, E.; Zhang, L.; Yang, Z.; Peng, S.; Yu, W.; Zuo, L.; Zhang, J. A systematic method for the optimization of gas supply reliability in natural gas pipeline network based on Bayesian networks and deep reinforcement learning. *Reliab. Eng. Syst. Saf.* **2022**, *225*, 108613.
21. Grondman, I.; Busoniu, L.; Lopes, G.A.D.; Babuska, R. A survey of actor-critic reinforcement learning: Standard and natural policy

gradients. *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.* **2012**, *42*, 1291–1307.

22. Konda, V.R.; Tsitsiklis, J.N. Actor-critic algorithms. *Adv. Neural Inf. Process. Syst.* **2000**, 1008–1014.

23. Li, Y. Deep reinforcement learning. *Submitt. Publ.* **2018**, doi:10.18653/v1/p18-5007.

24. Compare, M.; Bellani, L.; Cobelli, E.; Zio, E.; Annunziata, F.; Carlevaro, F.; Sepe, M. A reinforcement learning approach to optimal part flow management for gas turbine maintenance. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2019**, doi:10.1177/1748006X19869750.

25. Rocchetta, R.; Bellani, L.; Compare, M.; Zio, E.; Patelli, E. A reinforcement learning framework for optimal operation and maintenance of power grids. *Appl. Energy* **2019**, *241*, 291–301, doi:10.1016/j.apenergy.2019.03.027.

26. Mnih, V.; Silver, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv Prepr. arXiv1312.5602 (2013).* 1–9.

27. Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In Proceedings of the International conference on machine learning; PMLR, 2016; pp. 1329–1338.

28. Deng, Y.; Bao, F.; Kong, Y.; Ren, Z.; Dai, Q. Deep direct

reinforcement learning for financial signal representation and trading. *IEEE Trans. neural networks Learn. Syst.* **2016**, *28*, 653–664.

29. Neto, W.F.; Cavalcante, C.; Do, P. Deep Reinforcement Learning-Based Maintenance Decision-Making for a Steel Production Line. In Proceedings of the Proceedings of the 31st European Safety and Reliability Conference; Castanier, B., Cepin, M., Bigaud, D., Berenguer, C., Eds.; Research Publishing, Singapore, 2021; pp. 2611–2618.
30. Perera, A.T.D.; Kamalaruban, P. Applications of reinforcement learning in energy systems. *Renew. Sustain. Energy Rev.* **2021**, *137,* 110618.
31. Fang, J.; Hu, W.; Liu, Z.; Chen, W.; Tan, J.; Jiang, Z.; Verma, A.S. Wind turbine rotor speed design optimization considering rain erosion based on deep reinforcement learning. *Renew. Sustain. Energy Rev.* **2022**, *168*, 112788.
32. Ganesh, A.H.; Xu, B. A review of reinforcement learning based energy management systems for electrified powertrains: Progress, challenge, and potential solution. *Renew. Sustain. Energy Rev.* **2022**, *154*, 111833.
33. Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*; MIT press, 2018; ISBN 0262352702.

34. Tavares, A.R.; Chaimowicz, L. Tabular Reinforcement Learning in Real-Time Strategy Games via Options. In Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games (CIG); IEEE, 2018; pp. 1–8.

35. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv Prepr. arXiv1707.06347* **2017**.

36. Ho, J.; Gupta, J.K.; Ermon, S. Model-free imitation learning with policy optimization. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016; 2016; Vol. 6, pp. 4036–4046.

37. Ponciroli, R.; Bigoni, A.; Cammi, A.; Lorenzi, S.; Luzzi, L. Object-oriented modelling and simulation for the ALFRED dynamics. *Prog. Nucl. Energy* **2014**, *71*, 15–29.

38. Di Maio, F.; Mascherona, R.; Wang, W.; Zio, E. Simulation-based goal tree success tree for the risk analysis of cyber-physical systems. In Proceedings of the 29th European Safety and Reliability Conference, ESREL 2019; Research Publishing Services, 2020; pp. 4122–4129.

39. Hao, Z.; Di Maio, F.; Zio, E. 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.

40. Zhang, J.; Jiang, Y.; Li, X.; Huo, M.; Luo, H.; Yin, S. An adaptive remaining useful life prediction approach for single battery with unlabeled small sample data and parameter uncertainty. *Reliab. Eng. Syst. Saf.* **2022**, *222*, 108357.

41. Zhang, J.; Li, X.; Tian, J.; Jiang, Y.; Luo, H.; Yin, S. A variational local weighted deep sub-domain adaptation network for remaining useful life prediction facing cross-domain condition. *Reliab. Eng. Syst. Saf.* **2023**, *231*, 108986.

42. Zhang, J.; Jiang, Y.; Wu, S.; Li, X.; Luo, H.; Yin, S. Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. *Reliab. Eng. Syst. Saf.* **2022**, *221*, 108297.

43. Liu, J.; Zio, E.; Hu, Y. Particle filtering for prognostics of a newly designed product with a new parameters initialization strategy based on reliability test data. *IEEE Access* **2018**, *6*, 62564–62573.

44. Lin, Y.-J.; Yang, J.-M.; Wang, R.-Y.; Yang, Y.-X. 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.

45. Wu, Z.-G.; Zhu, J.; Yu, X.-B. 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.

46. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv Prepr. arXiv1312.5602* **2013**.

47. Vanvuchelen, N.; Gijsbrechts, J.; Boute, R. Use of Proximal Policy Optimization for the Joint Replenishment Problem. *Comput. Ind.* **2020**, *119*, 103239, doi:10.1016/j.compind.2020.103239.

48. Guan, Y.; Ren, Y.; Li, S.E.; Sun, Q.; Luo, L.; Li, K. Centralized Cooperation for Connected and Automated Vehicles at Intersections by Proximal Policy Optimization. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12597–12608, doi:10.1109/TVT.2020.3026111.

49. Mataric, M.J. Reward Functions for Accelerated Learning. In Proceedings of the Machine Learning Proceedings; Morgan Kaufmann Publishers, Inc., 1994; pp. 181–189.

50. Rosenfeld, A.; Taylor, M.E.; Kraus, S. Leveraging human knowledge in tabular reinforcement learning: A study of human subjects. In Proceedings of the Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17);

2017; pp. 3823–3830.

51. Ross, S.; Bagnell, J.A. Efficient reductions for imitation learning. *J. Mach. Learn. Res.* **2010**, *9*, 661–668.

52. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An introduction to deep reinforcement learning. *arXiv Prepr. arXiv1811.12560* **2018**.

53. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38.

54. Li, Y. Deep reinforcement learning: An overview. *arXiv Prepr. arXiv1701.07274* **2017**.

55. Alemberti, A. Status of the ALFRED Project. In Proceedings of the ESNII Biennial conference; pp. 17–19.

56. Alemberti, A.; Frogheri, M.; Mansani, L. The lead fast reactor: demonstrator (ALFRED) and ELFR design. **2013**.

57. Alemberti, A.; Caramello, M.; Frignani, M.; Grasso, G.; Merli, F.; Morresi, G.; Tarantino, M. ALFRED reactor coolant system design. *Nucl. Eng. Des.* **2020**, *370*, 110884.

58. Castelluccio, D.M.; Grasso, G.; Lodi, F.; Peluso, V.G.; Mengoni, A. Nuclear data target accuracy requirements for advanced reactors: The ALFRED case. *Ann. Nucl. Energy* **2021**, *162*, 108533.

59. Grasso, G.; Petrovich, C.; Mikityuk, K.; Mattioli, D.; Manni, F.;

Gugiu, D. Demonstrating the effectiveness of the European LFR concept: the ALFRED core design. In Proceedings of the International Conference on Fast Reactors and Related Fuel Cycles: Safe Technologies and Sustainable Scenarios (FR13), Paris, France; 2013.

60. Ponciroli, R.; Cammi, A.; Lorenzi, S.; Luzzi, L. Control approach to the load frequency regulation of a Generation IV Lead-cooled Fast Reactor. *Energy Convers. Manag.* **2015**, *103*, 43–56.

61. Ponciroli, R.; Cammi, A.; Della Bona, A.; Lorenzi, S.; Luzzi, L. Development of the ALFRED reactor full power mode control system. *Prog. Nucl. Energy* **2015**, *85*, 428–440.

62. Wang, W.; Di Maio, F.; Zio, E. Considering the human operator cognitive process for the interpretation of diagnostic outcomes related to component failures and cyber security attacks. *Reliab. Eng. Syst. Saf.* **2020**, *202*, 107007.

63. Wang, W.; Cammi, A.; Di Maio, F.; Lorenzi, S.; Zio, E. A Monte Carlo-based exploration framework for identifying components vulnerable to cyber threats in nuclear power plants. *Reliab. Eng. Syst. Saf.* **2018**, *175*, 24–37.

64. TEROL, G. Porous media approach in CFD thermohydraulic simulation of nuclear generation-IV lead-cooled fast reactor ALFRED. **2021**.

65. Bragg-Sitton, S.M.; Boardman, R.; Ruth, M.; Zinaman, O.; Forsberg, C. *Rethinking the future grid: integrated nuclear renewable energy systems*; National Renewable Energy Lab.(NREL), Golden, CO (United States), 2015;

66. Chou, Q.B. Characteristics and maneuverability of CANDU nuclear power stations operated for base-load and load-following generation. *IEEE Trans. Power Appar. Syst.* **1975**, *94*, 792–801.

67. Aldemir, T.; Stovsky, M.P.; Kirschenbaum, J.; Mandelli, D.; Bucci, P.; Mangan, L.A.; Miller, D.W.; Sun, X.; Ekici, E.; Guarro, S. Dynamic reliability modeling of digital instrumentation and control systems for nuclear reactor probabilistic risk assessments. *Nureg/Cr-6942. Washington, DC US Nucl. Regul. Comm.* **2007**.

68. Hao, Z.; Di Maio, F.; Zio, E. Multi-State Reliability Assessment Model of Base-Load Cyber-Physical Energy Systems (CPES) during Flexible Operation Considering the Aging of Cyber Components. *Energies* **2021**, *14*, 3241.

69. Di Maio, F.; Mascherona, R.; Zio, E. Risk analysis of cyber-physical systems by GTST-MLD. *IEEE Syst. J.* **2019**, *14*, 1333–1340.

70. Lokhov, A. Technical and economic aspects of load following with nuclear power plants. *NEA, OECD, Paris, Fr.* **2011**, *2*.

71. Zhang, S.; Du, M.; Tong, J.; Li, Y.-F. Multi-objective optimization

of maintenance program in multi-unit nuclear power plant sites. *Reliab. Eng. Syst. Saf.* **2019**, *188*, 532–548.

72. Martorell, S.; Sánchez, A.; Carlos, S.; Serradell, V. Simultaneous and multi-criteria optimization of TS requirements and maintenance at NPPs. *Ann. Nucl. Energy* **2002**, *29*, 147–168.

73. Ludwig, H.; Salnikova, T.; Stockman, A.; Waas, U. Load cycling capabilities of german nuclear power plants (NPP). *VGB powertech* **2011**, *91*, 38–44.

74. Eungse, O.; Kangdae, L.; Sungok, Y. 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.

75. International Atomic Energy Agency *Non-baseload operations in nuclear power plants: load following and frequency control modes of flexible operation*; IAEA, 2018; ISBN 9201108168.

76. Ferrario, E.; Zio, E. Goal Tree Success Tree–Dynamic Master Logic Diagram and Monte Carlo simulation for the safety and resilience assessment of a multistate system of systems. *Eng. Struct.* **2014**, *59*, 411–433.

77. Hao, Z.; Di Maio, F.; Pinciroli, L.; Zio, E. 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.

78. Rockafellar, R.T.; Uryasev, S. Conditional value-at-risk for general loss distributions. *J. Bank. Financ.* **2002**, *26*, 1443–1471.

79. Stenström, C.; Norrbin, P.; Parida, A.; Kumar, U. Preventive and corrective maintenance–cost comparison and cost–benefit analysis. *Struct. Infrastruct. Eng.* **2016**, *12*, 603–617.

80. International Atomic Energy Agency *Maintenance Optimization Programme for Nuclear Power Plants*; Nuclear Energy Series; INTERNATIONAL ATOMIC ENERGY AGENCY: Vienna, 2018; ISBN 978-92-0-110916-3.

81. Linnerud, K.; Mideksa, T.K.; Eskeland, G.S. The impact of climate change on nuclear power supply. *Energy J.* **2011**, doi:10.5547/ISSN0195-6574-EJ-Vol32-No1-6.

82. Hu, Y.-S.; Modarres, M. Evaluating system behavior through dynamic master logic diagram (DMLD) modeling. *Reliab. Eng. Syst. Saf.* **1999**, *64*, 241–269.

83. Lehmann, E.L.; Casella, G. *Theory of point estimation*; Springer Science & Business Media, 2006; ISBN 0387227288.