energies

Article

Multi-State Reliability Assessment Model of Base-Load Cyber-Physical Energy Systems (CPES) during Flexible Operation Considering the Aging of Cyber Components

Zhaojun Hao [1], Francesco Di Maio [1,*] and Enrico Zio [1,2,3]

[1] Energy Department, Politecnico di Milano, 20156 Milan, Italy; zhaojun.hao@polimi.it (Z.H.); enrico.zio@polimi.it (E.Z.)
[2] Centre de Recherche sur les Risques et les Crises (CRC), MINES ParisTech/PSL Université Paris, 75272 Sophia Antipolis, France
[3] Department of Nuclear Engineering, Kyung Hee University, Seoul 17104, Korea
* Correspondence: francesco.dimaio@polimi.it

Abstract: Cyber-Physical Energy Systems (CPESs) are energy systems which rely on cyber components for energy production, transmission and distribution control, and other functions. With the penetration of Renewable Energy Sources (RESs), CPESs are required to provide flexible operation (e.g., load-following, frequency regulation) to respond to any sudden imbalance of the power grid, due to the variability in power generation by RESs. This raises concerns on the reliability of CPESs traditionally used as base-load facilities, such as Nuclear Power Plants (NPPs), which were not designed for flexible operation, and more so, since traditionally only hardware components aging and stochastic failures have been considered for the reliability assessment, whereas the contribution of the degradation and aging of the cyber components of CPSs has been neglected. In this paper, we propose a multi-state model that integrates the hardware components stochastic failures with the aging of cyber components, and quantify the unreliability of CPES in load-following operations under normal/emergency conditions. To show the application of the reliability assessment model, we consider the case of the Control Rod System (CRS) of a NPP typically used for a base-load energy supply.

Keywords: Cyber-Physical System (CPS); Nuclear Power Plant (NPP); Renewable Energy Source (RES); load-following; aging; multi-state model; Control Rod System (CRS)

check for updates

Citation: 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. https://doi.org/10.3390/en14113241

Academic Editor: Ricardo J. Bessa

Received: 17 April 2021
Accepted: 28 May 2021
Published: 1 June 2021

1. Introduction

Cyber-Physical Systems (CPSs) are systems that integrate cyber components within hardware systems in which physical processes take place [1]: when the processes relate to energy production, transmission and distribution, they are called Cyber-Physical Energy Systems (CPESs) [2]. With the penetration of Renewable Energy Sources (RESs) (e.g., wind, photovoltaic), CPESs are requested to provide flexible operation (e.g., load-following, frequency regulation) to adjust any sudden imbalance that may occur in the power grid, due to a high level of variability and uncertainty in the power generation by renewables [3]. This inevitably raises concerns on the reliability of CPESs traditionally used as base-load facilities, such as Nuclear Power Plants (NPPs). Indeed, given the stable steady-state energy supply demanded to the base-load CPESs, manoeuvring capabilities were designed for seldom operations, mainly triggered by safety needs (i.e., safe shutdowns) [4] and with limited safety margins or capabilities to satisfy flexible operation during frequent and fast-changing demand scenarios. Since the base-load CPESs are normally expected to operate under stable steady-state conditions, for which any change of the cyber part setting can be easily detected and corrected without losing control of the system [5–7], aging of cyber parts is not a concern, whereas under frequent, fast-changing transients, as it is in the case of load-following CPESs here considered, aging of the cyber part cannot be neglected [8].

Recently, dynamic reliability methods (e.g., Multi-State Physical Modelling (MSPM) [9], Petri Net [10], Bayesian Network [11]) based on dynamic models of CPSs are being increasingly developed to assess CPESs' reliability. For base-load CPESs like traditional NPPs that were not designed for flexible operation, efforts have been focusing on assessing the contribution to unreliability due to aging and stochastic failures of the hardware components, without considering the degradation and aging of the cyber components. On the other hand, as already said, the cyber components of any Cyber-Physical System (CPS) are sensitive parts of CPSs, because they control the physical processes, as shown in [12–14]: disturbances on the cyber components of a CPS can strongly affect its performance, especially during flexible operations where its functions are most active. Assuming the functionalities of cyber components of a CPES is quite important given the long operation time of CPESs, and their reliability is threatened by aging processes typical of cyber systems. Then, to assess the reliability of CPESs accounting for the aging and degradation of cyber systems, in [15] we proposed a multi-state model for describing the aging process driven by memory leakage [16,17], which leads to service rate decrease and, eventually, data-jamming in the mission queue [18,19], which, in turn, increases the memory request; in such conditions, the cyber system blocks its function, significantly increasing the control delay [20,21], deteriorating the system stability and controllability during transients, when the amount of memory available cannot satisfy the demand of the mission queue. In this paper, we elaborate on this modelling approach to propose a framework of analysis for complete support to the reliability assessment of CPES for load-following.

To demonstrate the use of the reliability assessment framework, we consider the Control Rod System (CRS) of a typical NPP. We apply the multi-state model [15] that integrates the hardware components' stochastic failures with the aging of cyber components, and quantify the unreliability of the CPS with respect to transients during load-following operations under normal/emergency conditions.

The remainder of paper is as follows: Section 2 presents the NPP case study considering both hardware components' stochastic failures and the aging of cyber components in load-following operation scenarios; Section 3 presents relative modelling works of cyber aging and the proposed multi-state model accounting for the cyber aging process; the reliability assessment procedure, embedding the multi-state model of Section 3, is presented in Section 4; the results of the application of the reliability assessment of Section 4 to the case study of Section 2 are reported and discussed in Section 5; and in Section 6, conclusions are drawn.

2. The Control Rod System

2.1. Control Rod System Description

The Control Rod System (CRS) (Figure 1) is an important system of a NPP, whose function is to adjust the insertion and withdrawal of control rods in the reactor core, so as to control the thermal power and, thus, the electric power generated [22–25] with a closed-loop feedback control (Figure 2). The CRS elementary scheme comprises of a sensor, a controller, a connecting network, an actuator, and a motor for rod movement, where r_k, u_k, y_k, e_k are the discrete reference, control, output, and error signals, respectively, at discrete time $t_k = kh$ [26], where h is the interval of sensor sampling and $k = 0, 1, 2, \dots$ is a discrete integer variable as a data-sampling sequence number. The feedback control loop accounts for a total delay time τ that sums up the network transmission (τ_{sc}, τ_{ca}) and controller processing (τ_c) delay times.

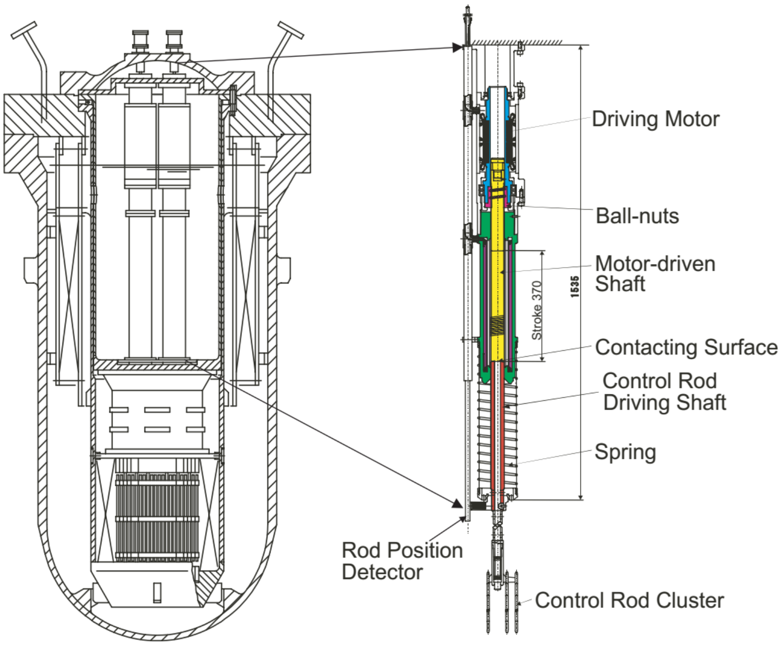

Figure 1. Structure of control rod mechanism.

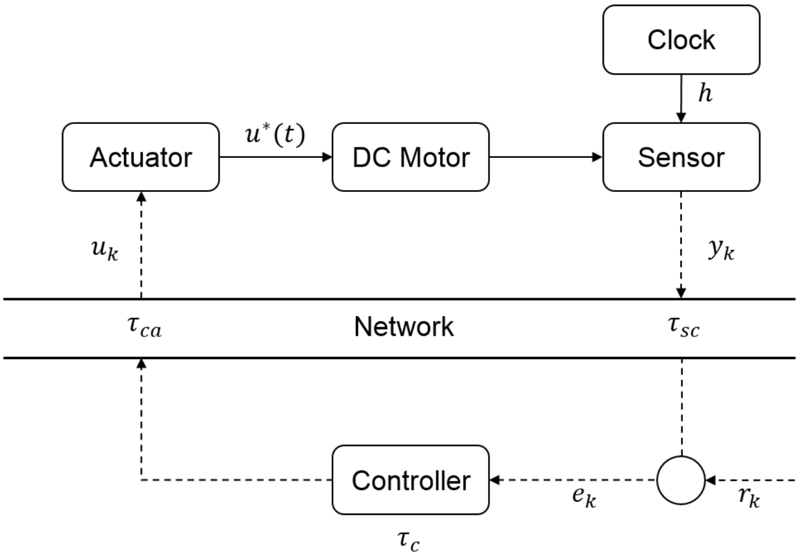

Figure 2. Closed-loop control of control rod system.

Without loss of generality, the CRS here considered comprises of (*i*) a typical digital Instrumental & Control (I&C) platform in single-controller mode with one CPU (for the controller module), and (*ii*) a typical DC motor (for the motor used to control the position of the control rods [22,27]). For the convenience of simulation, we use the discrete form of the transfer function (Equations (1) and (2) below) proposed in [26] between the DC motor and controller (where u and y are the control signal output of the controller and control rod's relative position controlled by the DC motor (considered as the percentage of energy output), respectively, and i is the index for the simulation step):

$$y_i = 1.944y_{i-1} - 0.944y_{i-2} + 0.004u_{i-1} + 0.0039u_{i-2} \tag{1}$$

$$u_i = u_{i-1} + 0.17e_i - 0.163e_{i-1} \tag{2}$$

2.2. *Load-Following Operation of the CRS*

By definition, load-following means adjusting the electricity generation to match the expected electricity load curve [28]. A complete load-following cycle consists of a power decrease from the normal power rate (P_n) of the CPES to a lower percentage of P_n ($\%P_n$), followed by a ramp to re-establish the lower power level to P_n [29]. As shown in Figure 3, different types of cycles can be envisaged in practical applications: "light cycles" with a limited power excursion (above $60\%P_n$) (dotted line); "deep cycles" with a large variation of power (below $60\%P_n$) (continuous line), and an "emergency cycle" with a large variation of power (below $50\%P_n$) by a high power change rate (dashed-dotted line) . For these three types of cycles, the lower power plateau can be either long or short, depending on the changes of demand on the grid side. The rate of power change from P_n to $\%P_n$ (and vice versa) depends on the energy CPS under analysis: in our case, we assumed a change rate of $5\%P_n$ per second for normal load-following conditions and $20\%P_n$ per second for a power decrease due to emergency conditions (compatible with the DC motor defined in Section 2.1) [4,22,27].

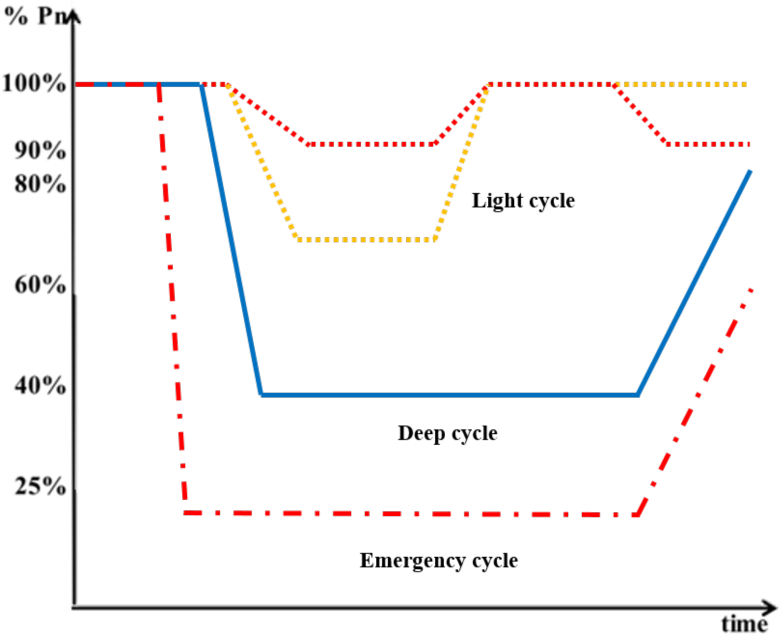

Figure 3. Load-following capability by cycle type.

In Table 1 [30], we can see that a typical PWR reactor is estimated to perform 100, 000 "light cycles" to $90\%P_n$, 100, 000 "light cycles" to $80\%P_n$, 15, 000 "deep cycles" to $60\%P_n$, 12, 000 "deep cycles" to $40\%P_n$, and 100 "emergency cycles" to $20\%P_n$ [4] during the plant lifetime. It results that the probability that a typical PWR NPP experiences any type of load-following cycle at each hour can be calculated (by the number of load cycles divided by total working hours in a NPP lifetime of 70 years), with results listed in the third column of Table 1.

Table 1. PWR reactor load-following capability [30].

Load Cycle	Number of Load Cycles	Probability
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 (emergency)	100	1.65×10^{-4}
No load-following	–	0.6297

3. Modelling of Cyber Systems Aging

In this Section, the multi-state model of the aging process of a CPS originally presented in [15] is briefly recalled and customized for application to the CRS of Section 2.

Aging of cyber systems manifests in performance degradation and failure rate increase of the software that drives the controller [17]. Cyber system aging is caused by some specific software faults/bugs, known as aging-related bugs [18] and activated by internal/external factors, causing errors that accumulate and propagate inside the system and finally lead to aging-related failures.

Memory leakage is a typical effect of cyber aging processes caused by internal errors, like unterminated processes that shrink the available amount of physical memory [18]. With memory leakage, data-jamming can occur, due to decreasing service rates that prevent the controller from processing or delivering data and tasks in due time, which results in (*i*) an accumulation of data in the mission queue, (*ii*) an increase of the memory request, and (*iii*) data packet loss, when the mission queue is full [19].

As a result, the cyber system becomes blocked when the amount of memory available cannot satisfy the demand of the mission queue, significantly increasing the control delay (τ_c) in processing data of the controller [20,31,32] and reducing controllability and stability of the controlled physical system [21], which increases risk of failure.

In the literature, modelling approaches of cyber aging are divided into two categories: measurement-based, and model-based [16,33]. With respect to measurement-based approaches, time-series analysis [34–36] and machine-learning methods [37,38] are used to forecast the system failure time by observing the performance degradation and resource consumption [39]. However, lacking in the generalization of systems, their data-driven characteristics make them hardly applicable to systems whose historical information is missing. With respect to model-based approaches, the cyber-aging system is commonly described as a Continuous-Time Markov Chain (CTMC) [6,40]. However, none comprehensively describes the causes, processes, and effects of cyber aging, such as service rate decrease and data-jamming.

In this work, we use CTMC to describe multiple performance degradation (i.e., service rate decrease) states embedded with a queueing model. With memory leakage, data-jamming has a higher probability of occurrence, and the system can be blocked more easily when the system cannot satisfy the memory demands, injecting high delays into the control loop which may make the system out of control. Thanks to its advantages over the mentioned approaches, (*i*) the chain of cyber aging phenomena is fully described; (*ii*) time-dependent blocking transition rates can be calculated by effects of memory leakage and the mechanism of data-jamming instead of constant transition rates with subjective assumptions, and (*iii*) considering the specialty of cyber aging, this model can be applied to simulate and explore system performances with high aging levels or under blocking conditions instead of directly assuming system failure.

3.1. Memory Leakage

The system performance deteriorates stochastically and eventually reaches a blocking state when the available memory cannot satisfy the demand from the mission process queue; as shown in Figure 4, the leakage degradation process can be modeled as a continuous-time Markov Chain with state space $L = \{S_0, S_1, \ldots, S_n, B\}$, where state S_0 is the normal state, in which the system has the maximum memory capacity and performance; states $S_1 \sim S_n$ represent increasing degradation states of decreased memory available; state B is the blocking state; $\lambda_{i,i+1}$ $(i = 0, 1, \ldots, n-1)$ is the transition rate between degradation states S_i and S_{i+1}; $\lambda_{i,B}$ is the system-blocking transition rate from the i-th state S_i to blocking state B (if $i < j$, then $\lambda_{i,B} < \lambda_{j,B}$, which means that the worse the degradation state, the larger the transition rate to the blocking state).

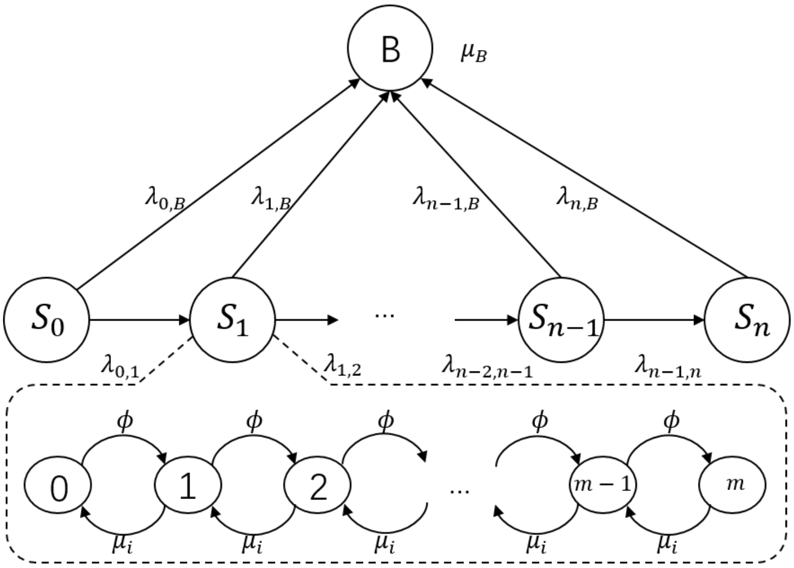

Figure 4. Multiple degradation states of cyber aging.

3.2. Data-Jamming

For each degradation state S_i, assuming a data arrival rate ϕ, an exponential service rate μ_i and a maximum capacity of task delivery queue equal to m, the continuous time Markov Chain of Figure 4 (below) can be used to model data-jamming, nested into the model of Figure 4 (above), where μ_i denotes the different service rates in different states S_i $(i = 0, 1, \ldots, k)$ (if $i < j$, then $\mu_i > \mu_j$), and the lowest service rate μ_B is that at blocking state B.

For each state S_i, the probability $P_{jam}(i, j)$ of j data-jamming in the queue at state S_i is [41]:

$$P_{jam}(i,j) = \frac{1-\phi/\mu_i}{1-(\phi/\mu_i)^{m+1}}\left(\frac{\phi}{\mu_i}\right)^j \\ i = 1, 2, \ldots, n \quad j = 0, 1, \ldots, m. \tag{3}$$

3.3. Calculation of the System-Blocking Transition Rate

As mentioned in Section 3.1, the probability of system-blocking $P_{i,B}$ from state S_i, and the corresponding blocking transition rate $\lambda_{i,B}$, depend on the current available memory $M(t)$ and on the memory request of the mission queue, which can be calculated with the values of the model parameters listed in Table 2.

$M(t)$ is estimated by assuming the transition time between degradation states (S_i and S_{i+1}) to be exponentially distributed with parameter $\lambda_{i,i+1}$ [16]. The Monte Carlo simulation is used to sample the transition times between states S_i and S_{i+1}, starting from the initial state S_0, and the available memory in each state is recorded at each transition time; repeating the simulation N_{mc} times, the mean value of the collected available memory at each time is taken as the available memory $M(t)$ at time t. Figure 5 shows one random trial of the simulation process (dashed line).

Table 2. Parameters for cyber aging model.

Parameter	Description	Value
n	Number of degradation states	3
$\lambda_{i,i+1}$	Transition rate between states S_i and S_{i+1}	5×10^{-5} [h^{-1}]
m	Maximum number of tasks	10
ϕ	Data coming rate	50 [s^{-1}]
μ_0	Service rate in state S_0	100 [s^{-1}]
μ_1	Service rate in state S_1	85 [s^{-1}]
μ_2	Service rate in state S_2	70 [s^{-1}]
μ_3	Service rate in state S_3	55 [s^{-1}]
μ_B	Service rate in state Blocking	30 [s^{-1}]
M	Total memory available	100 [Kb]
x	Memory request of each task	U(2,7) [Kb]
τ_{sc}, τ_{ca}	Transmission delay	N(13.1,5.7) [ms]

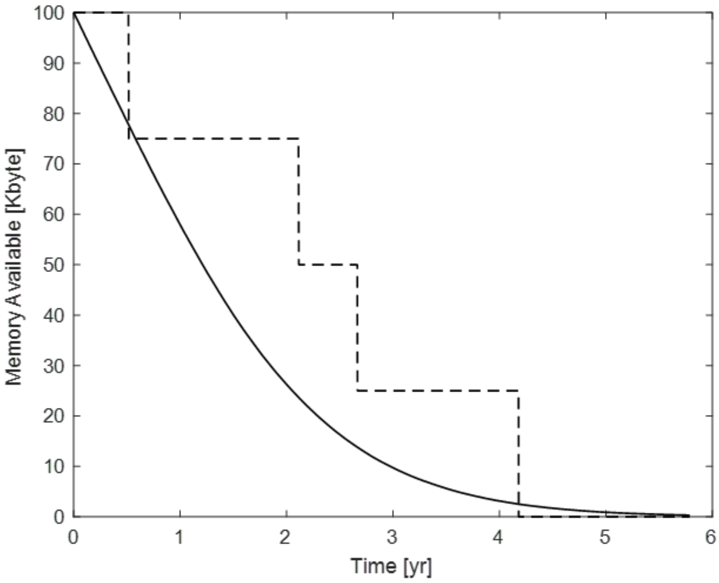

Figure 5. Decreasing available memory $M(t)$ (continuous line).

On the other hand, to estimate the memory request of the mission queue, we need to assume that each new data comes into the queue (with maximum capacity m) with a memory request which is a continuous random variable with density function $g(x)$ [40]: for any $0 < j \leqslant m$, let $g^{[j]}(x)$ be the density function for the total amount of j-independent resource requests, which is equal to the j-fold convolution of g [42].

$$g^{[1]}(x) = g(x),$$
$$g^{[j+1]}(x) = \int_0^x g^{[j]}(u)g(x-u)du, j \geqslant 1 \tag{4}$$

Let $G^{[j]}(x)$ be the corresponding cumulative distribution function of $g^{[j]}(x)$, $G^{[j]}(x) = \int_0^x g^{[j]}(u)du$. The conditional probability $\xi[j, M]$ that the system blocks with j data in the queue and M memory available upon the arrival of a new request can be calculated considering the system-blocking mechanism (i.e., the memory available cannot satisfy the memory request).

$$\begin{aligned}
&\xi[0,M]=1-G^{[1]}(M),\\
&\xi[j,M]=1-\frac{G^{[j+1]}(M)}{G^{[j]}(M)},\quad 1\leqslant j\leqslant m-1\\
&\xi[m,M]=1-G^{[m]}(M)
\end{aligned}\tag{5}$$

Combining the probability $P_{jam}(i,j)$ of j data-jamming in the queue at state S_i shown in Section 3.2 and the conditional probability $\xi[j,M]$ of system-blocking with j data, the probability $P_{i,B}(M)$ of system-blocking at each state with M available memory can be calculated as in Equation (6) below. Remembering that $M(t)$ is specific to each system, we can calculate $P_{i,B}(t)$ and the corresponding blocking transition rate $\lambda_{i,B}(t)$ (Figure 6) as in Equations (7) and (8) below.

$$P_{i,B}(M)=\sum_{j=0}^{m}P_{jam}(i,j)\cdot\xi[j,M]\quad i=1,2,\ldots,n\tag{6}$$

$$f_{i,B}(t)=\frac{dP_{i,B}(t)}{dt}\tag{7}$$

$$\lambda_{i,B}(t)=\frac{f_{i,B}(t)}{1-P_{i,B}(t)}\tag{8}$$

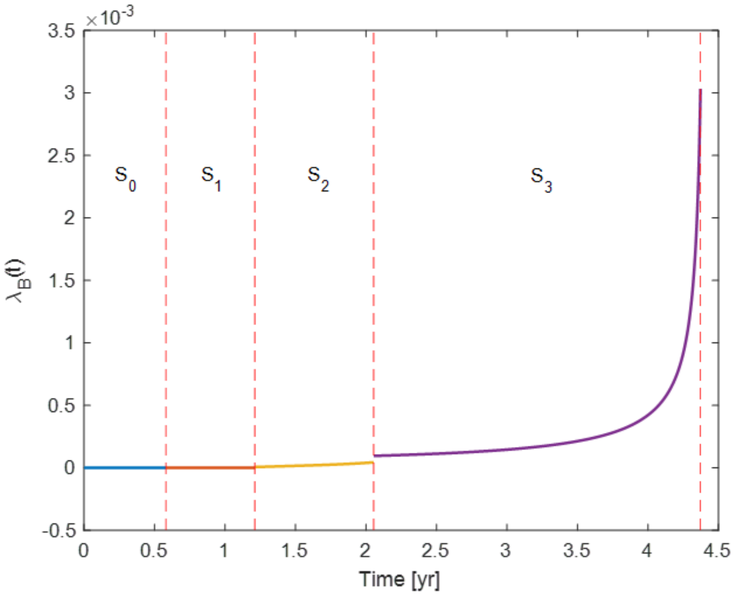

Figure 6. Transition rate of system-blocking.

3.4. Calculation of the Control Delay

The transmission delays τ_{sc} and τ_{ca} of Figure 2 are usually assumed constant for a specific network structure, whereas τ_c is dependent on the system state (blocking or non-blocking), and consists of the waiting time $\tau_{waiting}$ necessary for a data packet to be processed, and in the calculating time $\tau_{calculating}$. When the cyber system is in a non-blocking state $S_0\sim S_n$, the control delay equals to the sum of τ_{sc} and τ_{ca} (τ_c can be neglected); whereas when the cyber system is in the blocking state B, the low service rate causes data packet accumulation in the mission queue, significantly increasing τ_c, which results in an increase of the total control delay τ [43].

$$\tau_c=\tau_{waiting}+\tau_{calculating}\tag{9}$$

$$\tau = \begin{cases} \tau_{sc} + \tau_{ca}, & S_0 \sim S_n \\ \tau_{sc} + \tau_{ca} + \tau_c, & B \end{cases} \quad (10)$$

The signal delay calculation in a degraded CPS is sketched in Figure 7: the sensors sample the signals from the plant at the k-th sampling time; the control command signal finally reaches the actuator after delay τ: when the cyber system is in a non-blocking state $S_0 \sim S_n$, the total delay τ only accounts for τ_{sc} and τ_{ca}, commonly less than h; whereas when the cyber system is in the blocking state B, τ also accounts for τ_c, making τ larger than h.

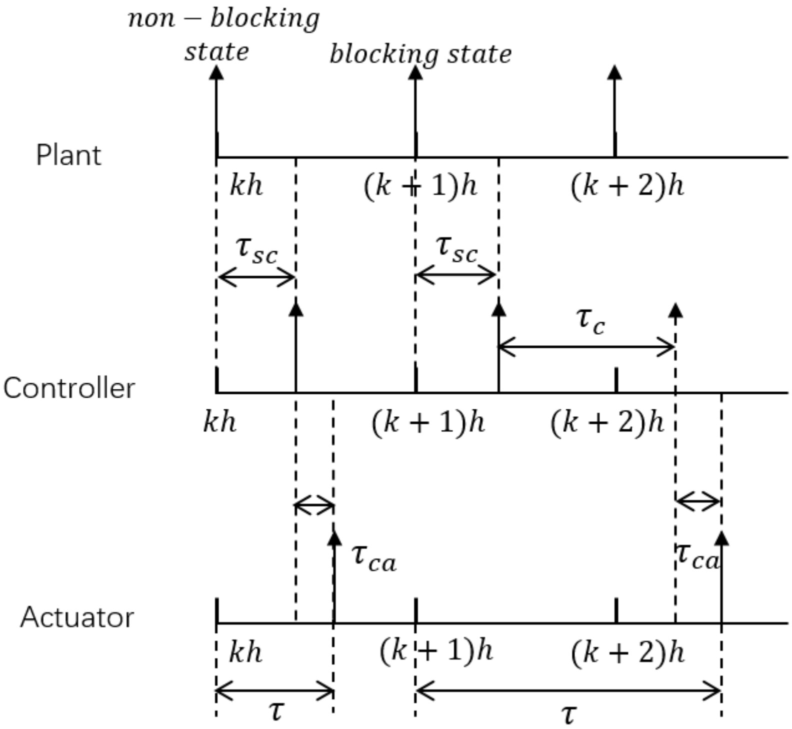

Figure 7. Delay of the CPS control.

4. Reliability Analysis of the CRS

In this Section, we show the procedure to calculate the reliability of the NPP CRS described in Section 2, while being used for flexible control during load-following operations. Cyber system aging is modelled as described in Section 3.3. The failure rates of the controller and DC motor are listed in Table 3 [44,45]. It is worth mentioning that we assume failure rates of both the controller and DC motor to be constant (and their failure times exponentially distributed) even though, in the literature, the lifetime of DC motor failure times are shown to change with temperature and to obey a Weibull distribution [46]. This assumption is here justified by the fact that the CRS is assumed to operate at a constant temperature.

The CRS is considered to be failed when the system response (power output) is out of the control safety boundary, that assumed to be smaller or larger than $2\%P_n$. Figure 8 shows examples of normal (continuous line) and failing (dotted line) load-following operations, which are both assumed to start at $t = 1$ s (the safety boundaries for a $100\%P_n$ to $60\%P_n$ power decrease are $[59.2\%P_n; 60.8\%P_n]$ (dashed-dotted line)).

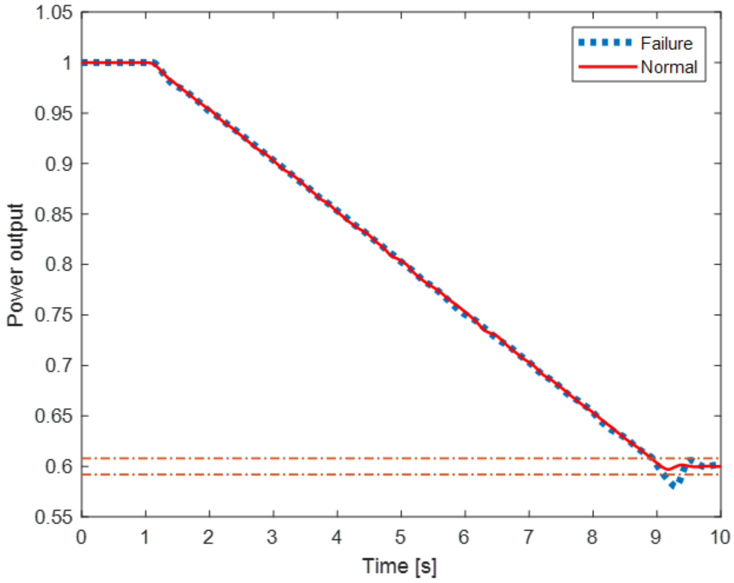

Figure 8. Examples of system load-following operations.

Table 3. Parameters for hardware stochastic failure.

Parameter	Description	Value
$\lambda_{controller}$	Controller failure rate	8.01×10^{-6} [h^{-1}]
λ_{motor}	DC motor failure rate	9.50×10^{-6} [h^{-1}]

The CRS is considered to undergo maintenance during the refueling outage (every 18 months), as long as the components show decreasing performance [47]. In this paper, we assume (*i*) to maintain the controller and DC motor, alternately, every 18 months during the refueling outage, (*ii*) the maintenance activity on the controller clears all accumulated aging-related bug-caused errors (such as memory leakage) and aging-related cyber failures (as good as new (AGAN)).

The procedure for the reliability assessment proceeds as follows (sketched also in Figure 9):

1. Calculate system-blocking transition rate λ_B with the model described in [15] and the procedure summarized in Section 3.3;
2. Set: initial time $t = 0$, mission time $T_{miss} = 10^5$ h, simulation time step $dt = 1$ h, maintenance period $T_m = 12960$ h (18 months) and index of maintenance cycle $k_m = 1$;
3. Sample the DC motor and controller hardware failure times $T_{h,motor}$ and $T_{h,controller}$, respectively, from the exponential distributions whose rates are reported in Table 3;
4. Set the system failure time T_{hard} due to hardware stochastic failures: $T_{hard} = \min(T_{h,motor}, T_{h,controller})$;
5. Check whether the system must undergo maintenance:
    - If $t = k_m T_m$:
      (*i*) alternately maintain the DC motor and controller (AGAN policy), and resample the corresponding hardware failure time, $T_{h,motor}$ or $T_{h,controller}$;
      (*ii*) reset the system hardware failure time T_{hard} as step 4;
      (*iii*) set $k_m = k_m + 1$;
6. Check if the hardware stochastic failure time t exceeds T_{hard}:
    - If $t \geqslant T_{hard}$, record system failure time due to hardware stochastic failure in the failure time counter: $Cal(t) = Cal(t) + 1$, and jump to step 9;

- If $t < T_{hard}$:
  (i) sample load-following operation type L from the 3rd column in Table 1: if L is the index for which $F_{L-1} < R \leqslant F_L$, where $F_L = \sum_{l=0}^{L} P_l$, P_l is the load-following occurred probability and R is a random value sampled from the uniform distribution in $[0, 1]$; the load-following operation type L is obtained;
  (ii) sample the system-blocking time $T_{blocking} = -\ln(R)/\lambda_B(t - (k_m - 1)T_m/2)$, where R is another random value sampled from the uniform distribution in $[0, 1]$;
  (iii) if $T_{blocking} < dt$ (system transits to blocking state B), start to run the load-following simulation with the type sampled in i) as following steps (a) to (h):
  (a) Set: load-following simulation initial time $t' = 0$, mission time $T'_{miss} = 15$ s, time step $dt' = 0.002$ s, sample interval $h = 0.2$ s and sample iteration number $k = 1$, mission queue array Q with "first in first out" processing principle;
  (b) Set: initial system output $y_0 = 0$, error $e_0 = 1$ and control signals $u_0 = 0$;
  (c) Set: system reference input r according to different types of load-following operations (for example: $r = 1.05 - 0.05t'$ $(1 < t' < 9s)$ for load-following operations from P_n to $60\%P_n$);
  (d) Set $t' = t' + dt'$;
  (e) Calculate y_i according to Equation (1)
  (f) Check whether new data are collected from the sensors.
  If $t' = kh$:
    – Sample the calculation delay $\tau_{calculating}$ from the exponential distribution with parameter μ_B in Table 2;
    – Sample the transmission delay τ_{sc} and τ_{ca} from the Gaussian distributions with the parameters in Table 2;
    – Calculate the data waiting time $\tau_{waiting} = \sum Q_\tau[q]$, where q is the index of data waiting in the mission queue;
    – Calculate the total delay time τ for the k-th sample data y_i according to Equations (9) and (10);
    – Save y_i and $kh + \tau$ into mission queue Q as the k-th sample data and its processing end time;
    – Set $k = k + 1$
  (g) Check whether the actuator time t' for getting the new control signal $Q_{y_i}[1]$ (i.e., the first data in mission queue Q) exceeds the delay time $Q_\tau[1]$:
    – If $t' \geqslant Q_\tau[1]$, set $e_i = r_i - Q_{y_i}[1]$, calculate u_i according to Equation (2) and take the first data out of the mission queue;
    – If $t' < Q_\tau[1]$, set $e_i = e_{i-1}$ and $u_i = u_{i-1}$
  (h) Check whether the system output y_i, which refers to the system power output, exceed 2% of the power change above and below the reference values (i.e., the safety bounds):
    – If $|y_i - r_i| > 2\% \, of \, power \, change$, record the cyber failure time in the failure time counter: $Cal(t) = Cal(t + 1)$, and jump to step 9;
    – If $|y_i - r_i| \leqslant 2\% \, of \, power \, change$, repeat (d) to (h) until time t' exceeds T'_{miss}, and finish the simulation of load-following

7. $t = t + dt$;
8. Repeat steps 4 to 7 until time t exceeds T_{miss} for one simulation run;
9. Run N_c (e.g., 10^6 times) steps 2 to 8 and calculate the system unreliability simply as Cal/N_c.

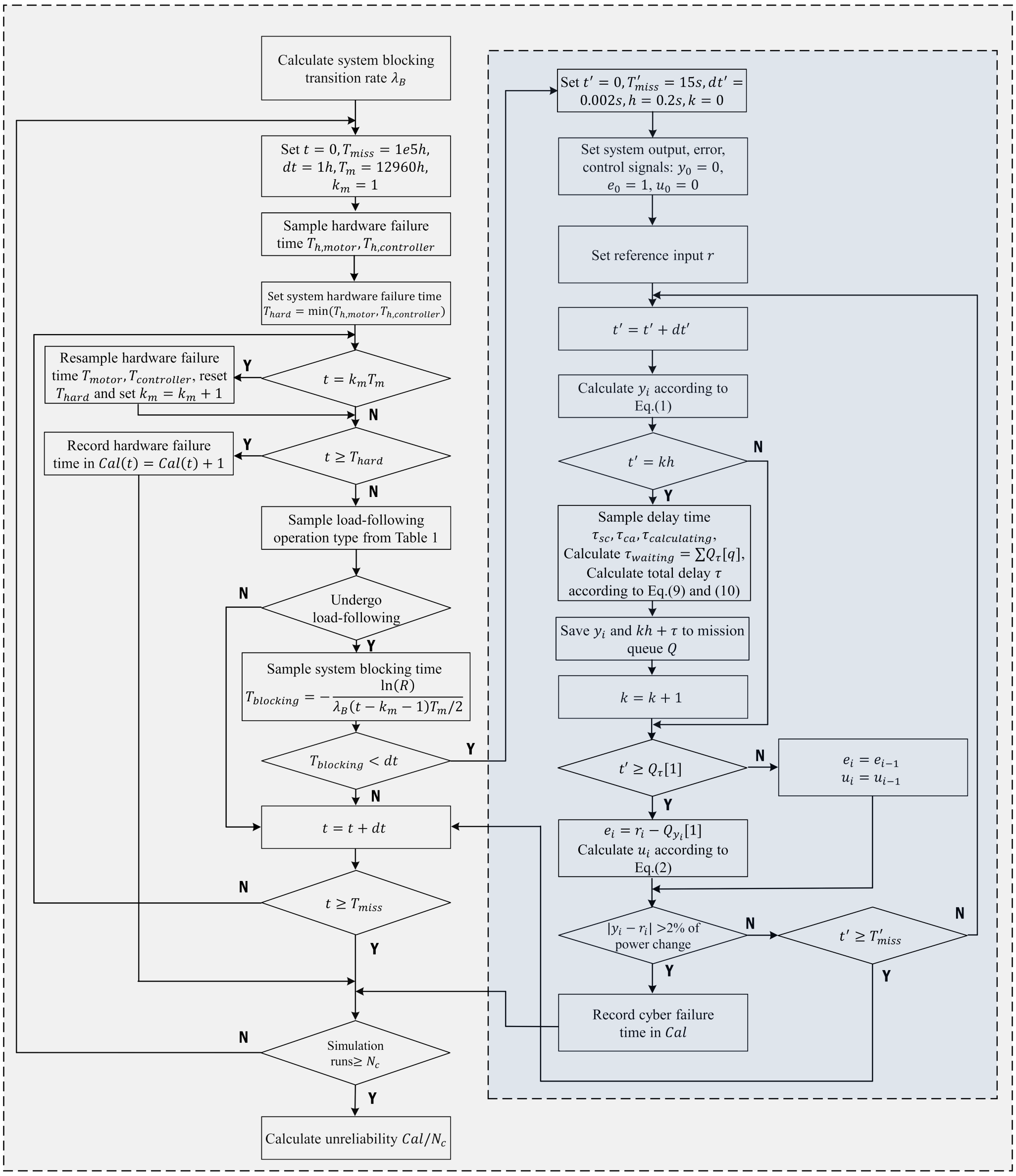
Calculate system blocking transition rate λ_B
Set $t = 0, T_{miss} = 1e5h$, $dt = 1h, T_m = 12960h$, $k_m = 1$
Sample hardware failure time $T_{h,motor}, T_{h,controller}$
Set system hardware failure time $T_{hard} = \min(T_{h,motor}, T_{h,controller})$
Resample hardware failure time $T_{motor}, T_{controller}$, reset T_{hard} and set $k_m = k_m + 1$
$t = k_m T_m$
Record hardware failure time in $Cal(t) = Cal(t) + 1$
$t \geq T_{hard}$
Sample load-following operation type from Table 1
Undergo load-following
Sample system blocking time $T_{blocking} = -\frac{\ln(R)}{\lambda_B(t - k_m - 1)T_m/2}$
$T_{blocking} < dt$
$t = t + dt$
$t \geq T_{miss}$
Simulation runs≥ N_c
Calculate unreliability Cal/N_c
Set $t' = 0, T'_{miss} = 15s, dt' = 0.002s, h = 0.2s, k = 0$
Set system output, error, control signals: $y_0 = 0$, $e_0 = 1, u_0 = 0$
Set reference input r
$t' = t' + dt'$
Calculate y_i according to Eq.(1)
$t' = kh$
Sample delay time $\tau_{sc}, \tau_{ca}, \tau_{calculating}$, Calculate $\tau_{waiting} = \sum Q_\tau[q]$, Calculate total delay τ according to Eq.(9) and (10)
Save y_i and $kh + \tau$ to mission queue Q
$k = k + 1$
$t' \geq Q_\tau[1]$
$e_i = e_{i-1}$ $u_i = u_{i-1}$
$e_i = r_i - Q_{y_i}[1]$ Calculate u_i according to Eq.(2)
$|y_i - r_i|$ >2% of power change
$t' \geq T'_{miss}$
Record cyber failure time in Cal
Y
N

Figure 9. Flowchart of system unreliability calculation considering cyber aging and stochastic failures.

5. Results

5.1. Normal Condition

For comparison, we assess the reliability of CRS under three different modelling assumptions under normal load-following conditions (without considering 5-th row of Table 1):

1. Only hardware stochastic failures (i.e., by neglecting step 6 (*i*) to (*iii*) of the reliability assessment procedure described in Section 4);
2. Only cyber aging (i.e., by neglecting steps 3, 4 and 5 (*ii*) of the reliability assessment procedure described in Section 4);
3. Both hardware stochastic failures and cyber aging.

Figure 10 shows the result of the system unreliability estimation of normal load-following operations considering the three models mentioned above (only hardware stochastic failures in continuous line, only cyber aging in dashed lines and both hardware stochastic failures and cyber aging in the dashed-dotted line). It can be seen that:

- Hardware stochastic failures remain the principle cause of system failure;
- Each periodic maintenance (each 18 months) efficiently reduces the system unreliability;

- As CRS ages, longer delays are to be accommodated by the control loop, increasing the contribution of cyber aging to system failure, two years after the controller has undergone maintenance each time (with AGAN policy that clears all the aging-related errors);
- The largest contribution of cyber aging to system failure is recorded three years after maintenance.

Effects of cyber aging on CRS are, thus, not negligible and need to be accounted for in the reliability assessment. The difference between both stochastic and cyber aging (dashed-dotted) and only stochastic (continuous) curves clearly shows the contribution of cyber aging to the overall system reliability. Cyber aging is shown to account comparatively with hardware stochastic failures, and should be only in design and for operation. It should be noticed that it is thanks to the effective periodic (AGAN) maintenance assumed, that the unreliability is maintained to a low level. Additionally, it is important to notice that the mechanism of deterioration due to cyber aging (initially silent and negligible), abruptly becomes a priority to be addressed when implementing maintenance activities.

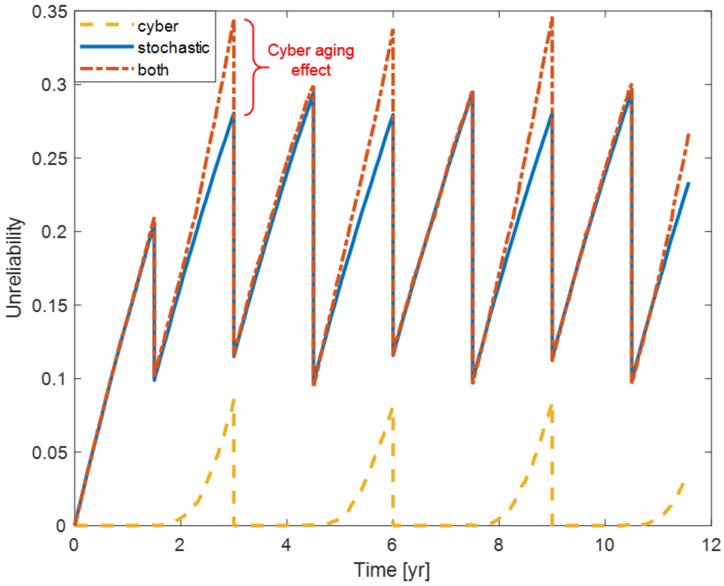

Figure 10. Result of system unreliability under a normal condition.

5.2. Emergency Condition

To show the effects of cyber aging in the emergency condition, we added an emergency cycle (5-th row in Table 1) into the same simulation framework presented in Section 5.1.

Figure 11 shows the system unreliability for load-following operations under emergency conditions. When considering emergency conditions, the effects of cyber aging are magnified, further showing the need to account for it in the reliability assessment of a CPES: in the Figure 11, the difference between the highest two curves shows the significant contribution of cyber aging to the system unreliability; with respect to Figure 10 (normal conditions), cyber aging (dashed line) has a larger contribution (around 0.28 at 3 years) to the system unreliability (whereas in Figure 10 the value is around 0.08); periodic maintenance (every 18 months) is still an efficient method to reduce the system unreliability; as CRS ages, longer delays are introduced into the control loop (as described in Section 3), which rapidly increase the contribution of cyber-aging-caused system failure (dashed line) that can be seen two years after each AGAN periodic maintenance.

Figure 12 further shows the results of system unreliability under an emergency condition (dashed line) compared with the normal condition (dotted line), both considering hardware stochastic failure and cyber aging. The results show that emergency conditions significantly increase the system unreliability and the CRS vulnerability to cyber aging, even if the occurrence of emergency transients is very rare. During emergency conditions, the large power change needs the CRS to be highly stable and controllable: in such rare conditions, the CPES integrity is undermined because the cyber aging makes the CRS more sensitive to delays that, under normal conditions, would have led to negligible effects.

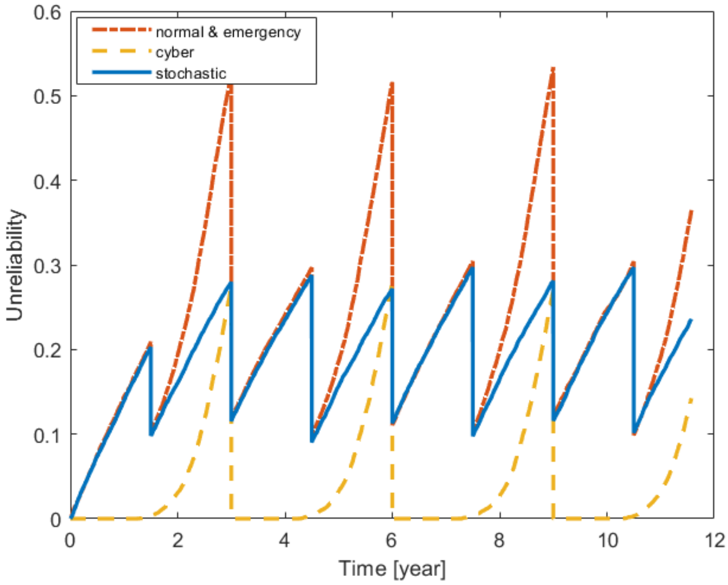

Figure 11. Result of system unreliability under the emergency condition.

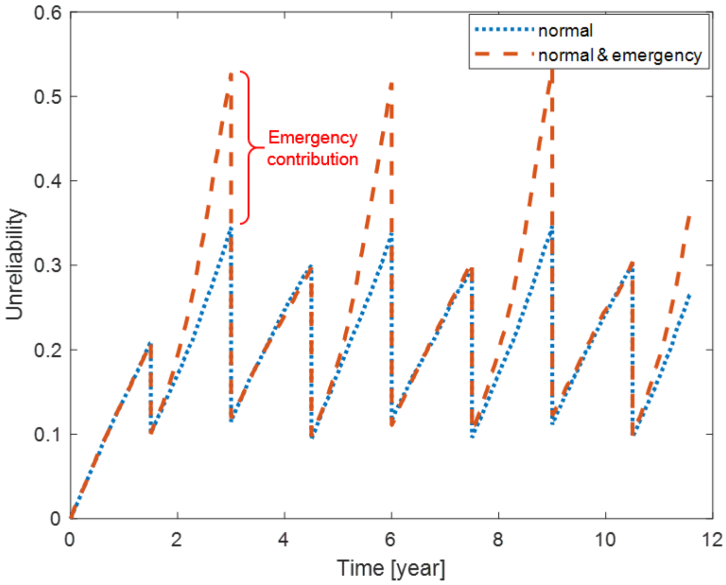

Figure 12. Results comparison between normal and emergency conditions.

6. Conclusions

In this paper, a previously proposed multi-state model that integrates memory leakage, data-jamming, and a control delay to describe cyber system aging processes of a CPS was considered within a MC-based reliability assessment framework for CPESs typically used as the base-load, to assess the effects of cyber aging when dealing with flexible operation (e.g., load-following).

We took the CRS of a NPP as a case study, which consists of a PI controller, a DC motor, and connecting network. The result shows that: hardware stochastic failure is the main reason for system failure; the periodic maintenance (assumed AGAN) can efficiently reduce the system unreliability, for both causes of stochastic failures and cyber aging; with gradual deterioration of the control rod system and larger delays in the control loop, cyber aging starts contributing significantly, up to at most about 27% of system unreliability; the emergency condition with a lower occurrence probability contributes more than the normal condition and increases up to, at most, about 48% of the system unreliability.

Cyber aging can, then, be an important, non-negligible cause of unreliability in base-load CPES used for flexible operation, especially during emergency conditions. Effective preventive maintenance on the cyber system must be planned to mitigate the aging effects, together with the effective control of an energy dispatch at different base-load CPESs with different aging profiles to avoid system failure during transients.

Author Contributions: All authors have equally contributed to the work. Z.H.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization and writing (original draft preparation, review and editing); F.D.M.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization and writing (original draft preparation, review and editing); E.Z.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization and writing (original draft preparation, review and editing). All authors have read and agreed to the published version of the manuscript.

Funding: The participation of Enrico Zio has been funded by "Smart maintenance of industrial plants and civil structures by 4.0 monitoring technologies and prognostic approaches—mac4pro", sponsored by the call BRIC-2018 of the National Institute for Insurance against Accidents at Work—INAIL. This research was also funded by China Scholarship Council (CSC).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

$\lambda_{controller}$	Controller failure rate
$\lambda_{i,B}$	Transition rate from state S_i to blocking state B
$\lambda_{i,i+1}$	Transition rate between state S_i and state S_{i+1}
λ_{motor}	DC motor stochastic failure rate
μ_B	Blocking service rate
μ_i	Non-blocking service rate
ϕ	Data arrival rate
τ	Total delay time
τ_c	Controller processing delay time
$\tau_{calculating}$	Calculation time of data in mission queue
τ_{ca}	Transmission delay between controller and actuator
τ_{sc}	Transmission delay between sensor and controller
$\tau_{waiting}$	Waitting time of data to be processed in mission queue
$\xi[j, M]$	Conditional probability of system-blocking with j data in the queue and M memory available
B	system-blocking state
Cal	Counter of system failure times
dt	Simulation time step
e	System error signal
g	Probability density function of memory requested by a data sample
G	Cumulative distribution function of memory requested by a data sample
h	Sensor sampling interval
i	Index of simulation steps
k	Data sampling sequence number
k_m	Index of maintenance action
M	Total memory available
m	Maximum number of tasks
n	Number of degradation states
N_c	Number of simulation runs to calculate system unreliability
N_{mc}	Number of simulation runs to calculate memory curve

P_n	Normal power rate
$P_{i,B}$	Probability of system-blocking from state S_i
P_{jam}	Probability of data-jamming
q	Index of data waiting in the mission queue
Q	Mission queue array
Q_τ	Delay time of the data waiting in mission queue Q
Q_{y_i}	Value of the data waiting in mission queue Q
r	System reference input
R	Random value sampled from uniform distribution in $[0,1]$
S_i	System normal or degradation state
T_m	Maintenance period
$T_{blocking}$	system-blocking time
$T_{h,controller}$	Controller stochastic failure time
$T_{h,motor}$	DC motor stochastic failure time
T_{hard}	System failure time due to hardware stochastic failure
T_{miss}	Mission time
u	Control signal output
x	Memory request of each task
y	System power output
$CPES$	Cyber-Physical Energy System
CPS	Cyber-Physical System
CRS	Control Rod System
$I\&C$	Instrumental & Control
NPP	Nuclear Power Plant
PWR	Pressurized Wawter Reactor
RES	Renewable Energy Source

References

1. Baheti, R.; Gill, H. Cyber-physical systems. *Impact Control Technol.* **2011**, *12*, 161–166.
2. 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. [CrossRef]
3. 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. [CrossRef]
4. Lokhov, A. *Technical and Economic Aspects of Load Following with Nuclear Power Plants*; NEA OECD: Paris, France, 2011.
5. Koutras, V.P.; Platis, A.N.; Gravvanis, G.A. On the optimization of free resources using non-homogeneous Markov chain software rejuvenation model. *Reliab. Eng. Syst. Saf.* **2007**, *92*, 1724–1732. [CrossRef]
6. Trivedi, K.S.; Vaidyanathan, K.; Goseva-Popstojanova, K. Modeling and analysis of software aging and rejuvenation. In Proceedings of the 33rd Annual Simulation Symposium (SS 2000), Washington, DC, USA, 16–20 April 2000; pp. 270–279.
7. Tipsuwan, Y.; Chow, M.Y. Network-based controller adaptation based on QoS negotiation and deterioration. In Proceedings of the IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 37243), Denver, CO, USA, 29 November–2 December 2001; Volume 3, pp. 1794–1799.
8. Rajkumar, S.M.; Chakraborty, S.; Dey, R.; Deb, D. Online delay estimation and adaptive compensation in wireless networked system: An embedded control design. *Int. J. Control Autom. Syst.* **2020**, *18*, 856–866. [CrossRef]
9. Di Maio, F.; Colli, D.; Zio, E.; Tao, L.; Tong, J. A multi-state physics modeling for estimating the size-and location-dependent loss of coolant accident initiating event probability. In *2017 International Topical Meeting on Probabilistic Safety Assessment and Analysis, PSA 2017*; American Nuclear Society: La Grange Park, IL, USA, 2017; Volume 2, pp. 1185–1192.
10. Lee, D.Y.; Choi, J.G.; Lyou, J. A safety assessment methodology for a digital reactor protection system. *Int. J. Control. Autom. Syst.* **2006**, *4*, 105–112.
11. Boudali, H.; Dugan, J.B. A continuous-time Bayesian network reliability modeling, and analysis framework. *IEEE Trans. Reliab.* **2006**, *55*, 86–97. [CrossRef]
12. Wang, W.; Di Maio, F.; Zio, E. Three-loop Monte Carlo simulation approach to Multi-State Physics Modeling for system reliability assessment. *Reliab. Eng. Syst. Saf.* **2017**, *167*, 276–289. [CrossRef]
13. Wang, W.; Di Maio, F.; Zio, E. Adversarial Risk Analysis to Allocate Optimal Defense Resources for Protecting Cyber–Physical Systems from Cyber Attacks. *Risk Anal.* **2019**, *39*, 2766–2785. [CrossRef]
14. 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. [CrossRef]
15. Hao, Z.; Di Maio, F.; Zio, E. A Multi-State Model of the Aging Process of Cyber-Physical Systems. In Proceedings of the 30th European Safety and Reliability Conference, ESREL 2020, Venice, Italy, 1–5 November 2020.

16. Du, X.; Qi, Y.; Hou, D.; Chen, Y.; Zhong, X. 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, Seattle, WA, USA, 20–24 July 2009; Volume 1, pp. 240–245.
17. Huang, Y.; Kintala, C.; Kolettis, N.; Fulton, N.D. Software rejuvenation: Analysis, module and applications. In Proceedings of the Twenty-Fifth International Symposium on Fault-Tolerant Computing, Pasadena, CA, USA, 27–30 June 1995; pp. 381–390.
18. Grottke, M.; Matias, R.; Trivedi, K.S. The fundamentals of software aging. In Proceedings of the 2008 IEEE International Conference on Software Reliability Engineering Workshops (ISSRE Wksp), Seattle, WA, USA, 11–14 November 2008; pp. 1–6.
19. Garg, S.; Puliafito, A.; Telek, M.; Trivedi, K. Analysis of preventive maintenance in transactions based software systems. *IEEE Trans. Comput.* **1998**, *47*, 96–107. [CrossRef]
20. Cloosterman, M.B.; Van de Wouw, N.; Heemels, W.; Nijmeijer, H. Stability of networked control systems with uncertain time-varying delays. *IEEE Trans. Autom. Control.* **2009**, *54*, 1575–1580. [CrossRef]
21. Åström, K.J.; Wittenmark, B. *Computer-Controlled Systems: Theory and Design*; Courier Corporation: North Chelmsford, MA, USA, 2013.
22. Divandari, M.; Hashemi-Tilehnoee, M.; Khaleghi, M.; Hosseinkhah, M. A novel control-rod drive mechanism via electromagnetic levitation in MNSR. *Nukleonika* **2014**, *59*, 73–79. [CrossRef]
23. Yoritsune, T.; Ishida, T.; Imayoshi, S. In-vessel type control rod drive mechanism using magnetic force latching for a very small reactor. *J. Nucl. Sci. Technol.* **2002**, *39*, 913–922. [CrossRef]
24. Yuanqiang, W.; Xingzhong, D.; Huizhong, Z.; Zhiyong, H. Design and tests for the HTR-10 control rod system. *Nucl. Eng. Des.* **2002**, *218*, 147–154. [CrossRef]
25. Bakhri, S. Investigation of Rod Control System Reliability of Pwr Reactors. *KnE Energy* **2016**, *1*, 94–105. [CrossRef]
26. Tipsuwan, Y.; Chow, M.Y. Control methodologies in networked control systems. *Control Eng. Pract.* **2003**, *11*, 1099–1111. [CrossRef]
27. Divandari, M.; Hashemi-Tilehnoee, M.; Asgari-Ziarati, B.; Hosseinkhah, M.; Sabagh, K. Minimizing torque ripple in a brushless DC motor with fuzzy logic: Applied to control rod driving mechanism of MNSR. *Nucl. Sci. Tech.* **2015**, *26*, 10601-010601.
28. Lazarev, G.; Hrustalyov, V.; Garievskij, M. 1. Non-baseload Operation in Nuclear Power Plants: Load Following and Frequency Control Modes of Flexible Operation. *Nucl. Energy Ser.* **2018**, *1*, 173.
29. Bruynooghe, C.; Eriksson, A.; Fulli, G. Load-following operating mode at Nuclear Power Plants (NPPs) and incidence on Operation and Maintenance (O&M) costs. *JRC Rep.* **2010**, *5*, JRC60700.
30. Ludwig, H.; Salnikova, T.; Stockman, A.; Waas, U. Load cycling capabilities of german nuclear power plants (NPP). *VGB Powertech* **2011**, *91*, 38–44.
31. Yue, D.; Han, Q.L.; Peng, C. State feedback controller design of networked control systems. In Proceedings of the 2004 IEEE International Conference on Control Applications, Taipei, Taiwan, 2–4 September 2004; Volume 1, pp. 242–247.
32. Peng, C.; Tian, Y.C.; Tade, M.O. State feedback controller design of networked control systems with interval time-varying delay and nonlinearity. *Int. J. Robust Nonlinear Control IFAC-Affil. J.* **2008**, *18*, 1285–1301. [CrossRef]
33. Bovenzi, A.; Cotroneo, D.; Pietrantuono, R.; Russo, S. Workload characterization for software aging analysis. In Proceedings of the 2011 IEEE 22nd International Symposium on Software Reliability Engineering, Hiroshima, Japan, 29 November–2 December 2011; pp. 240–249.
34. Li, L.; Vaidyanathan, K.; Trivedi, K.S. An approach for estimation of software aging in a web server. In Proceedings of the International Symposium on Empirical Software Engineering, Nara, Japan, 3–4 October 2002; pp. 91–100.
35. Grottke, M.; Li, L.; Vaidyanathan, K.; Trivedi, K.S. Analysis of software aging in a web server. *IEEE Trans. Reliab.* **2006**, *55*, 411–420. [CrossRef]
36. Magalhães, J.P.; Silva, L.M. Prediction of performance anomalies in web-applications based-on software aging scenarios. In Proceedings of the 2010 IEEE Second International Workshop on Software Aging and Rejuvenation, San Jose, CA, USA, 2 November 2010; pp. 1–7.
37. Cassidy, K.J.; Gross, K.C.; Malekpour, A. Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers. In Proceedings of the International Conference on Dependable Systems and Networks, Washington, DC, USA, 23–26 June 2002; pp. 478–482.
38. Alonso, J.; Belanche, L.; Avresky, D.R. Predicting software anomalies using machine learning techniques. In Proceedings of the 2011 IEEE 10th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 25–27 August 2011; pp. 163–170.
39. Cotroneo, D.; Natella, R.; Pietrantuono, R.; Russo, S. A survey of software aging and rejuvenation studies. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **2014**, *10*, 1–34. [CrossRef]
40. Bao, Y.; Sun, X.; Trivedi, K.S. A workload-based analysis of software aging, and rejuvenation. *IEEE Trans. Reliab.* **2005**, *54*, 541–548. [CrossRef]
41. Bolch, G.; Greiner, S.; De Meer, H.; Trivedi, K.S. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
42. Trivedi, K. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2001.

43. Long, M.; Wu, C.H.; Hung, J.Y. Denial of service attacks on network-based control systems: Impact and mitigation. *IEEE Trans. Ind. Inform.* **2005**, *1*, 85–96. [CrossRef]
44. Chyou, Y.P.; Yu, D.D.; Cheng, Y.N. Performance validation on the prototype of control rod driving mechanism for the TRR-II project. *Nucl. Eng. Des.* **2004**, *227*, 195–207. [CrossRef]
45. Iida, H.; Imayoshi, S.; Morimoto, K.; Watanabe, M.; Komada, N.; Takeshita, T. Long-term stability of Sm_2Co_{17}-type magnets for control rod drive mechanism (CRDM) in a nuclear reactor. *IEEE Trans. Magn.* **1995**, *31*, 3653–3655. [CrossRef]
46. Song, K.; Shi, J.; Yi, X.; Xie, Y.; Liu, G.; Lu, M. Accelerated Life Data Analysis for Control Rod Drive Mechanism Coil. In Proceedings of the 2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Beijing, China, 15–17 August 2019; pp. 940–943.
47. Greene, R. Aging assessment of BWR control rod drive systems. *Nucl. Saf.* **1992**, *33*, 87–99.