A Markov Decision Process framework for optimal operation of monitored multi-state systems

M. Compare[1,2], P. Marelli[1], P. Baraldi[1], E. Zio[1,2,3]
[1]*Department of Energy, Politecnico di Milano, Italy*
[2]*Aramis s.r.l, Milan, Italy*
[3]*Chair on Systems Science and the Energetic Challenge, Foundation Electricité de France at Ecole CentraleSupélec, France*

ABSTRACT: We develop a decision support framework based on Markov Decision Processes (MDPs) to maximize the profit from the operation of a Multi-State System (MSS). This framework enables a comprehensive management of the MSS, which considers the maintenance decisions together with those on the MSS operation setting, i.e., its loading condition and configuration. The decisions are informed by a condition monitoring system, which estimates the health state of the MSS components. The approach is shown with reference to a mechanical system made up of components affected by fatigue.

ACRONYMS AND SYMBOLS

AGAN	As Good As New
CM	Corrective Maintenance
MSS	Multi-State System
PM	Preventive Maintenance
a_j^i	j-th action applicable to component C^i, j=1, …, L+2
$\boldsymbol{a^i}$	Vector of actions a_j^i
A^i	Action taken on component C^i
$\boldsymbol{A}$	Vector of actions taken on the MSS
C^i	i-th component, i=1,…, N
C_{CM}	Cost of corrective maintenance, in arbitrary units
C_{PM}	Cost of preventive maintenance, in arbitrary units
d	Index of the degradation process, d=1,… D^i
D^i	Number of degradation processes affecting C^i
F^i	Performance level of C^i
F_j^i	Performance value associated to action a_j^i, in arbitrary units
$\overline{G}_\pi$	Expected return from following policy π, in arbitrary units
G_π	Return from following policy π, in arbitrary units
L	Number of load levels
N	Number of MSS components
Q_π	Action-value function from following policy π, in arbitrary units
r	Reward, in arbitrary units
S	Set of states of the MSS
S^i	Set of states of component C^i
s_d^i	State of degradation process d affecting component C^i
$\boldsymbol{s^i}$	Vector containing the states s_d^i

S_d^i	Failure state of component C^i with respect to degradation process d
t	Time instant, in arbitrary units
$\boldsymbol{T}_j^i$	Transition matrix associated to action a_j^i
$\boldsymbol{T}^i$	Transition matrix associated to action A^i
U	Performance of the MSS, in arbitrary units
V_π	Value function from following policy π, in arbitrary units
W	Production requirement, in arbitrary units
z_j^i	Binary variable indicating the action
$\boldsymbol{z^i}$	Vector of the binary variables z_j^i
γ	Discount factor
Γ	Reward function, in arbitrary units
π	Generic MSS management policy
π^*	Optimal policy

1 INTRODUCTION

Multi-State Systems (MSS) (Ding & Lisniansky, 2008; Liu et al., 2015; Wijnmalen and Hontelez, 1997; Zille et al., 2011) such as aircrafts (Hopp and Kuo, 1998), power grids (Bian and Gebraeel, 2014), water distribution networks (Shinstine et al., 2002), natural gas distribution systems (Wang et al., 2011), Navy Frigate (Tinga and Jannsen, 2013) are made up of interdependent elements working together to fulfill the system functions (Haurie and L'Ecuyer, 1982).

One interesting characteristic of MSSs lies in their reconfiguration capability: the loading conditions on some components can be changed in order to optimally respond to diverse operational settings, which depend on both internal (e.g., health state of the components) and external (e.g., working load) factors. For example, to operate a water distribution system while a pump is undergoing maintenance, it is possible to compensate the missing pumping rate by uploading other pumps in the system.

If the MSS is properly managed, considering also the impact of its configuration settings on the components failure behaviors (Wang and Chen, 2016), the MSS flexibility and adaptability to the operating conditions can result in a more safe, reliable and profitable operation. For example, (Harlow and Phoenix, 1978) showed that in bundles of fibers, the failure of some fibers causes overloads on the remaining ones, which finally results in an acceleration of the overall system failure.

Different approaches have been proposed to manage the operation of an MSS. For example, a multi-agent approach is developed in (Trappey et al., 2011) to maximize the profit and the reliability of a power transmission network, whereas (Hopp and Kuo, 1998) use Markov Decision Processes (MDPs, Sigaud and Buffet, 2010; Sutton and Barto, 1998) to address a maintenance management issue on aircraft engine components that are over-stressed by the overloading conditions led by severe turbulences.

Nowadays, the optimal management of an MSS can benefit from the application of Prognostics and Health Management (PHM) methods to detect, diagnose, and predict failures of components and systems (Baraldi and Zio, 2015; Sharp et al., 2015; Jardine et al., 2006; Zio, 2012). In principle, PHM allows for a significant reduction of the system unavailability through an efficient and agile maintenance management, capable of providing the right part to the right place at the right time, together with the necessary resources to perform the maintenance task (Compare and Zio, 2013; Grall et al., 2002; Pipe, 2008). Although it seems evident that PHM can contribute to the profitable management of a MSS, to the authors' best knowledge only a few works investigate how and to which extent. For example, (Zonta et al., 2014; Pozzi et al., 2010; Memarzadeh and Pozzi, 2016) have developed methods within the Partially Observable Markov Decision Process framework to estimate the value of the information provided by a PHM system installed on civil infrastructures, which also account for the uncertainty in the condition monitoring system outputs. However, the focus of these works is mainly on single components and does not consider the management of the system configuration. Rather, the objective of these works is to optimally set the inspections, in order to maximize their value of information.

On the other side, the application of MDP to maintenance optimization issue is not new. For example, (Papakostantinou & Shinozuka, 2014a), (Papakostantinou & Shinozuka, 2014b) and (Papakostantinou & Shinozuka, 2014c) develop POMDP-based methods to optimize maintenance and inspection policies on corroded structures. (Chan & Asgarpoor, 2006) propose MDPs for scheduled maintenance optimization on a generic

component affected by both random failures and failures due to deterioration. (Nielsen and Sørensen, 2014) compare MDPs to other approaches to maintenance decision making for wind turbines (for a review of the applications of MDPs to wind turbine facilities maintenance optimization, the interested readers can refer to (Dawid et al. 2015)). However, these works are concerned with single components, only, and cannot treat the management of a MSS, which requires accounting for the mutual interactions among the components and their settings.

Against this backdrop, we present here a study to support maintenance decision making in a setting where a condition monitoring system informs the decision maker about the health states of the MSS components. Accounting for this information, the optimal policy for managing the MSS working configurations, i.e. that which maximizes the MSS operation profit, is found.

For this, MDPs are used because of the ease of encoding the aleatory uncertainty of the degradation behaviors of the MSS components in the decision problem, which is not given by other optimization algorithms such as the evolutionary algorithms (e.g., Genetic Algorithms, e.g., Mitchell, 1998; Zio, 2009) or linear programming algorithms (e.g., Fang and Puthenpura, 1993).

The remainder of the paper is organized as follows. Section 2 presents the problem of interest; Section 3 proposes an MDP-based method; Section 4 is dedicated to the description of a case study in which we apply MDP to a MSS made up of 2 components; Section 5 provides the results to the case study; finally, Section 6 concludes the work.

2 PROBLEM STATEMENT

2.1 *Degradation model of the MSS*

Consider a MSS, which is made up of N components C^i, $i = 1, \ldots, N$, arranged according to the given structure function.

Every component C^i is affected by D^i independent degradation processes, which are individually modeled as multi-state Markov processes (Lisniansky and Levitin, 2003; Lisniansky, 2016), with S_d^i states each, $d = 1, \ldots, D^i$.

The overall state of component C^i at time t is given by vector $\boldsymbol{s}^i(t) = \left[s_1^i, \ldots, s_{D^i}^i\right]$, where $s_d^i \in \left\{1, \ldots, S_d^i\right\}$, $i = 1, \ldots, N$. Then, $\boldsymbol{s}^i(t)$ belongs to the Cartesian product $S^i = \times_{d=1,\ldots D^i} \left\{1, \ldots, S_d^i\right\}$, $\forall t$.

Vectors $\boldsymbol{s}^i(t)$, $i = 1, \ldots, N$, are concatenated to form the MSS health state vector $\boldsymbol{s}(t) \in S = \times_{i=1,\ldots,N} S^i$, whose k-th element is:

$$s_k(t) = s_{k-q}^{i^*}, i^* = \min\left\{j \middle| \sum_{i=1}^{j} D^i \geq k, j \leq N\right\}$$

$$q = \begin{cases} \sum_{i=1}^{i^*} D^i & \text{if } i^* > 1 \\ 0 & \text{otherwise} \end{cases}$$

with S indicating the set of all the possible MSS states.

We assume that component C^i fails when any of its D^i degradation processes reaches its last state S_d^i, $d = 1, \ldots, D^i$, $i = 1, \ldots, N$.

2.2 *Management options for the MSS*

With respect to maintenance, we consider two types of actions: Preventive Maintenance (PM) actions, which are performed before component failure, and Corrective Maintenance (CM) actions, which are performed upon failure. The corresponding downtimes are considered as random variables obeying probability density functions (pdfs) f_{θ_p} and f_{θ_c}, respectively. These distributions are such that the downtime of a PM action is expected to

be shorter than that for CM: on the one hand, preventive actions avoid the failure propagation to other components, thus limiting the severity of the failure effects and the troubleshooting activities. On the other hand, PM enables performing timely arranged preventive actions, for which all the maintenance logistic support issues have already been addressed.

In an opportunistic view, we assume that both preventive and corrective maintenance actions restore the component to an As Good As New (AGAN) state with respect to all its degradation processes.

As mentioned before, we consider the situation in which the MSS components are continuously monitored and the system health state, $\boldsymbol{s}(t)$, is perfectly known (i.e., with no uncertainty) at every measurement acquisition time t=1, …, in arbitrary units. This information guides the decision about whether to take the action of performing maintenance or not.

Beside the actions relevant to maintenance, additional actions can be taken, which concern the setting of the operating performance of the components. Specifically, we assume that every component C^i, i = 1, …, N, can be operated at L different levels, which are associated to operating performance values $F_1^i,...,F_L^i$, where $F_k^i \geq F_l^i$ if $k \leq l$ (e.g., the production rate, the absorbed load, etc.). Furthermore, F_{L+1}^i and F_{L+2}^i are the performance values associated to preventive and corrective maintenance, respectively, and they are typically set to zero:

$$F_j^i = 0 \qquad \text{if } j=L+1, L+2.$$

Hence, the operating level of component C^i is indicated by $F^i \in \left\{F_1^i,...,F_L^i,F_{L+1}^i,F_{L+2}^i\right\}$, i=1, …, N.

The possible actions that the decision maker can take for component C^i are organized in vectors $\boldsymbol{a^i}$, i = 1, …, N:

$$\boldsymbol{a^i} = \left[a_1^i, a_2^i,..., a_L^i, a_{L+1}^i, a_{L+2}^i\right]$$

where a_j^i, j=1, …, L refers to setting component C^i at the operating level F_j^i, whereas the last two actions correspond to the decisions of preventively maintaining and repairing upon failure component C^i.

The action taken for component C^i is indicated by vector $\boldsymbol{z^i}$, which encodes the binary variables z_j^i, j = 1, …, L+2, i = 1, …, N:

$$z_j^i = \begin{cases} 1 & \text{if action } a_j^i \text{ is taken} \\ 0 & \text{otherwise} \end{cases}$$

The actions are mutually exclusive and exactly one out of the L+2 alternatives has to be taken for the i-th component:

$$\sum_{j=1}^{L+2} z_j^i = 1$$

The state of the component determines the actions that can be actually taken. Specifically, if component C^i is not failed, then it is able to work at any of the L possible load levels and any action can be taken on C^i except corrective maintenance, which requires the unit to be failed.

Formally, the constraints on the applicability of the actions are expressed as:

$$\sum_{j=1}^{L+1} z_j^i = \begin{cases} 0 & \text{if } \exists d \left| s_d^i = S_d^i, 1 \leq d \leq D^i \right. \\ 1 & \text{otherwise} \end{cases}$$

$$z_{L+2}^i = 1 - \sum_{j=1}^{L+1} z_j^i$$

Notice that in the considered model setting, the component performance is not influenced by the degradation level. Nonetheless, this relationship can be encoded through additional constraints. For example, to model that heavily degraded components cannot be operated at the highest performance levels, we can consider constraints such as:

$$z_j^i = \begin{cases} 0 & \text{if } j \leq 2 \wedge s_d^i \geq S_d^i - 3, d = 1,..., D^i \\ 1 & \text{otherwise} \end{cases}$$

which prevent running the i-th component at the two highest performance levels when any of its degradation mechanisms is in one among the three most degraded states.

The action taken by the decision maker for the MSS is represented by vector $\boldsymbol{A}=[A^1, A^2, \ldots, A^N]$, whose i-th entry is given by the scalar product:

$$A^i = \boldsymbol{a}^i \cdot \boldsymbol{z}^i$$

A deterministic policy π is a mapping function between state $\boldsymbol{s} \in S$ and action $\boldsymbol{A}$:

$$\boldsymbol{A} = \pi(\boldsymbol{s})$$

It is worth noticing that the decision about the action to be taken depends exclusively on the current system health state and not on time, being the degradation mechanisms described as Markov processes, i.e., memoryless processes for which the future evolution does not depend on the past but only on the present state. This explains why the time index is missing for the actions, i.e., $\boldsymbol{A} = \pi(\boldsymbol{s}) = \pi(\boldsymbol{s}(t))$.

2.3 *Degradation evolution*

Following policy π, an action is performed on every component at each decision time t and, consequently, a state transition occurs. The aleatory uncertainty in the consequence of action A^i is described by the transition matrix $\boldsymbol{T}^i = \sum_j z_j^i \cdot \boldsymbol{T}_j^i$, where $\boldsymbol{T}_j^i$ is a $\left(\sum_{d=1}^{D^i} S_d^i\right) \times \left(\sum_{d=1}^{D^i} S_d^i\right)$ block diagonal matrix:

$$\boldsymbol{T}_j^i = \begin{bmatrix} \boldsymbol{T}_{j,1}^i & \text{K} & & 0 \\ \text{M} & \text{O} & & \\ & & & \\ & & \text{O} & \text{M} \\ 0 & & \text{K} & \boldsymbol{T}_{j,D^i} \end{bmatrix}$$

whose d-th block, $\boldsymbol{T}_{j,d}^i$, is the $S_d^i \times S_d^i$ matrix containing the transition probabilities describing the evolution of the d-th degradation process in response to the selected action A^i, d=1, …, D^i, j=1, …, L+2, i = 1, …, N. Namely, the entry $(s_d^i(t), s_d^i(t+1))$ of $\boldsymbol{T}_{j,d}^i$ specifies the probability $\Pr\left[s_d^i(t+1) \middle| s_d^i(t), A^i\right]$ that the d-th degradation process, d=1,…,D^i, affecting the i-th component, i=1,…,N, evolves from state $s_d^i(t)$ to state $s_d^i(t+1)$.

The D^i degradation processes are independent on each other; then, the probability of having a transition from $\boldsymbol{s}^i(t) = \left[s_1^i(t),\ldots, s_{D^i}^i(t)\right]$ to $\boldsymbol{s}^i(t+1) = \left[s_1^i(t+1),\ldots, s_{D^i}^i(t+1)\right]$ upon taking action A^i, is given by:

$$\Pr\left[\boldsymbol{s}^i(t+1) \middle| \boldsymbol{s}^i(t), A^i\right] = \prod_{d=1}^{D^i} \Pr\left[s_d^i(t+1) \middle| s_d^i(t), A^i\right]$$

This definition is scaled up to the entire MSS for defining the probabilities that the MSS state at time t, $\boldsymbol{s}(t)$ changes to $\boldsymbol{s}(t+1)$ at the following time step, t+1, in response to action $\boldsymbol{A}$ as:

$$\Pr\left[\boldsymbol{s}(t+1) \middle| \boldsymbol{s}(t), \boldsymbol{A}\right] = \prod_{i=1}^{N} \Pr\left[\boldsymbol{s}^i(t+1) \middle| \boldsymbol{s}^i(t), A^i\right]$$

where, for simplicity, we are not considering the possible influences of the actions on one component on the degradation processes of the other components.

The performance U of the overall MSS depends on both the performance levels F^i of its components, i=1, …, N, and the MSS logic of operation. Then, action $\boldsymbol{A}$ yields the MSS performance:

$$U = f(F^1,\ldots, F^N) = f(\boldsymbol{A})$$

Obviously, larger performance levels entail larger loads and stresses and, thus, faster degradation paths.

2.4 *Reward function*

When action $\boldsymbol{A}$ is taken in state $\boldsymbol{s}(t)$ and the MSS has a transition in state $\boldsymbol{s}(t+1)$, a reward $r(t)$ is gained, which indicates how good the decision is to reach a pre-fixed goal (e.g., the operation profit):

$$r(t) = \Gamma(\boldsymbol{s}(t), \boldsymbol{A}, \boldsymbol{s}(t+1)) \in \mathbb{R}.$$

Functions f and Γ are specific of the case study considered.

The operation return of policy π is the value of the rewards gained from time t on:

$$G_\pi(t) = \sum_{\tau \geq t} \gamma^\tau r(\tau)$$

where γ is a discount factor, which determines the net present value of the future rewards (Sutton and Barto, 1998; van Otterlo, 2012).

Notice that the rewards depend on the actions taken and on the states between which the transitions occur, rather than on time, whereas the return value depends on both the policy adopted by the decision maker (i.e., the action to be associated to every state) and the stochastic evolutions of the degradation processes affecting the components. Therefore, to take into account the aleatory uncertainty in the degradation evolution from any initial state $\boldsymbol{s}_o$ under policy π, we focus on the expected return:

$$\overline{G}_\pi(\boldsymbol{s}_o) = E\left[G_\pi \middle| \boldsymbol{s} = \boldsymbol{s}_o\right] = E_\pi\left[r(0) + \gamma r(1) + \gamma^2 r(2) + \ldots \middle| \boldsymbol{s} = \boldsymbol{s}_o\right]$$

where $E_\pi[\cdot]$ indicates the expectation operator given that policy π is being followed.

In our setting, we assume that an income is gained only when the action taken enables the MSS to reach a performance value U larger than a pre-fixed level W, whereas costs are incurred when the action taken results in the MSS undergoing maintenance.

From the considerations above, it clearly appears that our objective is to find the optimal policy π^* (i.e., the management of the MSS configuration), which maximizes the profit deriving from the operation of the MSS:

$$\pi^*(\boldsymbol{s}) = \arg\max_{\boldsymbol{A}}(\overline{G}_\pi) \quad \forall \boldsymbol{s}$$

3 MARKOV DECISION PROCESSES

To find the optimal management policy regarding the MSS working configuration, we rely on Markov Decision Processes (MDPs), which require the definition of states, actions, transition probabilities and rewards introduced in the previous Section.

In the MDP framework, we want to estimate the value $Q_\pi(\boldsymbol{s}(t), \boldsymbol{A})$ of each state-action pair, which measures the expected return starting from state $\boldsymbol{s}(t)$, taking action $\boldsymbol{A}$ and thereafter following policy π (Sutton and Barto, 1998; van Otterlo, 2012):

$$Q_\pi(\boldsymbol{s}(t), \boldsymbol{A}) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r(t+k) \middle| \boldsymbol{s}(t), \boldsymbol{A}\right]$$

According to Bellman equation, this can be written as (Sutton and Barto, 1998; van Otterlo, 2012):

$$Q_\pi(\boldsymbol{s}(t), \boldsymbol{A}) = \sum_{\boldsymbol{s}(t+1)} \left(\Pr\left[\boldsymbol{s}(t+1) \middle| \boldsymbol{s}(t), \boldsymbol{A}\right] \cdot \left(\Gamma(\boldsymbol{s}(t), \boldsymbol{A}, \boldsymbol{s}(t+1)) + \gamma V_\pi(\boldsymbol{s}(t+1))\right) \right)$$

where $V_\pi(\boldsymbol{s}(t))$ specifies the expected return when starting from $\boldsymbol{s}$ at time t and following policy π thereafter, also denoted as the value of state $\boldsymbol{s}(t)$. Notice that both the value $V_{\pi^*}(\boldsymbol{s})$ and state-action pair $Q_{\pi^*}(\boldsymbol{s}, \boldsymbol{A})$ depend on state $\boldsymbol{s}$, rather than on time t.

To efficiently solve the Bellman Equation, we rely on the Value Iteration algorithm described in Appendix 1 (Sutton & Barto, 1998). This belongs to the dynamic programming algorithms family, which exploit the direct knowledge of the transition matrices. These algorithms, however, suffer from two main limitations:

- There may be cases in which the knowledge of the environment where decisions are taken does not justify the assumption of perfect knowledge of the transition matrices. In this case, sample-based Reinforcement Learning techniques must be applied (Sutton and Barto, 1998).
- Dynamic programming algorithms are affected by the curse of dimensionality issue: when the state-action space becomes very large, the computational burden is prohibitive. In this respect, (Mansour & Singh, 1999) and (Littman et al., 1995) analyze the complexity of the dynamic programming algorithms and, thus, give the bounds of their applicability to practical case studies To scale up, it is necessary to resort to complex value function approximation techniques.

Both issues will be tackled in future research work.

The readers interested in additional theoretical details on MDP can refer to (Sutton and Barto, 1998), (van Otterlo, 2012).

4 CASE STUDY

4.1 *Degradation model of the MSS*

Consider a MSS made up of N=2 identical pumps arranged in parallel configuration, whose design function is to supply the demanded flow rate.

Both components are subject to a single degradation process (i.e., $D^1 = D^2 = 1$), which is fatigue. This degradation process is discretized into $S_1^i = 15$ states, i = 1, 2, with state $s_1^i = 1$ corresponding to the As Good As New (AGAN) state and state $s_1^i = 15$ to the failure of the component. Then, the set of all the possible MSS states, S, contains 225 states.

Given that only one degradation process affects the two pumps, the state vector $\boldsymbol{s}^i$ reduces to a scalar, which will be denoted as s^i.

4.2 *Management options for the MSS*

Each pump can be operated at L=3 possible flow rate levels, therefore:

$$\boldsymbol{a}^i = \left[a_1^i, a_2^i, a_3^i, a_4^i, a_5^i\right] \; i = 1, 2.$$

where, a_1^i and a_2^i correspond to setting the pumping rate $F_1^i = 10$ and $F_2^i = 1$, in arbitrary units, respectively; a_3^i corresponds to the pump switch-off with $F_3^i = 0$, whereas a_4^i and a_5^i correspond to the PM and CM actions, respectively, which are associated to a null pumping rate.

4.3 *Degradation evolution*

In case of a_1^i and a_2^i, i=1, 2, the pump degradation mechanism evolves through stochastic paths that depend on the pumping rate and, thus, on the working load. Being the pumps affected by a single degradation process, a 15×15 transition matrix $\boldsymbol{T}^i$, i=1, 2, is associated to each action, whose (h,k) entry gives the probability of having a transition from state $s_1^i = h$ to state $s_1^i = k$, when action a_j^i is taken.

The values of $\boldsymbol{T}_j^i$, j=1, 2, have been derived by applying the procedure described in Appendix 2.

Notice also that the last row of $\boldsymbol{T}_1^i$ and $\boldsymbol{T}_2^i$ is set to 0 because it is not possible to take actions a_1^i and a_2^i while component C^i is in state $s_1^i = 15$, i=1, 2.

Choosing action $a_3^i = 0$ results in no flow delivered by pump i and, thus, no load exerted on the pump. Then, the degradation process does not evolve and, consequently, $\boldsymbol{T}_3^i = \boldsymbol{I}$.

The non-zero elements of $\boldsymbol{T}_4^i$ are those on the diagonal, which represent the probability values that the maintenance action is not completed within the reference time unit $\Delta t = 1$, and those on the first column, which are the probabilities that the pump exits the maintenance action within the time unit $t + \Delta t$, in the AGAN state (see Appendix 2). The values of $\boldsymbol{T}_4^i$ have been estimated by assuming that f_{θ_p} is an exponential distribution with repair rate $\mu_{\theta_p} = 0.7$, which gives $e^{-\mu_{\theta_p} \cdot \Delta t} = 0.5$.

As for actions a_1^i and a_2^i, the values of the last row of matrix $\boldsymbol{T}_4^i$ are all zeros, because the PM action is not applicable when component C^i is failed.

Finally, CM can be implemented only upon failure of the pump; then, the elements of $\boldsymbol{T}_5^i$ that are different from 0 are the first and last elements of the last row only, which define the probability of ending the CM action in the time unit and its complement to 1, respectively. The entries of $\boldsymbol{T}_5^i$ have been derived by assuming that f_{θ_c} is an exponential distribution with repair rate $\mu_{\theta_c} = 0.1$.

4.4 *Reward function*

Concerning the objective of the MSS, the two pumps have to supply a flow rate of at least W=11, in arbitrary units. Given that the pumps are set in parallel configuration, the overall output of the MSS is assumed to be the sum of the single pump outputs:

$$U = f(F^1, F^2) = F^1 + F^2$$

The threshold value W=11 requires both components to be working to not incur into loss of production.

Obviously, flow rate F^i is delivered by pump i only if the state entered at the next time step is not a failed state, i.e., $s^i(t+1) \neq S_1^i = 15$.

The MSS configurations ensuring that $U \geq W$ are those corresponding to the setting:

$$\sum_{i=1,2} \sum_{j=1,2} z_j^i - \prod_i z_2^i = 2 \quad (1)$$

This means that the pumps must not be running at the lowest levels at the same time. In this case study, when Eq. (1) is verified, that is when the flow rate provided by the MSS is at least equal to W, the reward function is:

$$r(t) = (I + O \prod_i z_1^i) \prod_i B^i(t) \quad (2)$$

where I=10 is the income for a Δt of operation, whereas O=1 is an additional benefit that is gained when both components operate at the highest pumping level, whereas B^i, i=1, 2 are Boolean variables such that:

$$B^i(t) = \begin{cases} 1 & \text{if } s^i(t+1) \neq S_1^i \\ 0 & \text{otherwise} \end{cases}$$

Moreover, we assume that the cost of maintenance is paid only upon completion of the activity. Therefore, when a maintenance action is implemented, its cost is not paid till the pump is restored in its AGAN state.

When Eq. (1) does not hold, the reward function becomes:

$$r(t) = P - (W - \sum_{i=1}^{N} F^i) - C_{CM}(\sum_{i=1}^{N} z_{L+2}^i \delta(s^i(t+1),1)) - C_{PM}(\sum_{i=1}^{N} z_{L+1}^i \delta(s^i(t+1),1)) + Syn(z_{PM}^1 z_{PM}^2 + z_{PM}^1 z_{CM}^2 + z_{CM}^1 z_{PM}^2) \prod_i \delta(s^i(t+1),1)$$

where P = -15 is a penalty that is incurred when the flow rate requirement W=11 is not delivered, $C_{CM} = 80$ is the cost, in arbitrary units, to perform a corrective action, $C_{PM} = 40$ is the cost to perform a preventive action, whereas $Syn = 10$ is the saving, in arbitrary units, owing to the synergy of performing maintenance actions at the same time on both components. Finally, δ is the Kronecker delta function:

$$\delta(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

For example, if $\boldsymbol{A} = [a_1^1, a_2^2]$, then the reward $r(t)$ is 10, whereas when both pumps deliver a low flow rate, i.e., $F^i = 1$, i=1, 2, $r(t)$ is -24. Yet, the rewards $r(t)$ for actions $\boldsymbol{A} = [a_4^1, a_4^2]$, $\boldsymbol{A} = [a_4^1, a_5^2]$ and $\boldsymbol{A} = [a_5^1, a_4^2]$ are equal to -96, -136 and -136, respectively.

Finally, a discount factor γ=0.99 is chosen to calculate the return G.

5 RESULTS AND COMMENTS

The optimal policy for the case study described above has been found by solving the MDP through the algorithm reported in Appendix 1, with a tolerance for convergence ε=10^{-4}. This took 270 seconds on an 8GB RAM machine, running an Intel Core i-7 processor @ 2.20 GHz. In this respect, notice that the computational times fast increase with the dimension of the state space, whereby other Reinforcement Learning algorithms need to be developed to address more complex case studies.

The optimal policy found is analyzed and validated by comparing its results with those of two traditional policies:

- π_1, the pumps work at the highest load level and undergo scheduled maintenance actions.
- π_2, the pumps work at the highest load level, with corrective maintenance only.

For all three policies (i.e., π^*, π_1, π_2), we have performed 10^7 Monte Carlo (MC) simulations of 10^3 time steps each, to estimate the corresponding return, starting from the initial state, $\boldsymbol{s}_0 = [s_1^1, s_1^2]$, in which both pumps are in the AGAN state.

Notice that the length of the time horizon for MC simulations has been set to 10^3 time steps, as it ensures that the weight given by the discount factor γ is such that the rewards collected from that point on are negligible. This setting on the time horizon is also validated by the fact that the expected profit of π^* given by $V_{\pi^*}(\boldsymbol{s}_o = [s_1^1, s_1^2]) = 922$ is almost equal to that estimated through the MC approach (i.e., 923). This proves that the contribution of the costs after 10^3 time steps is negligible. Furthermore, MC simulations allow us assessing the variability of the return G_π of every policy, in order to appraise the variability in the results obtained by applying the same policy π.

For policy π_1, we have estimated the expected return $\overline{G}_{\pi_1}$ for different values of the preventive maintenance intervals, in order to find its optimal value (Figure 1). As it can be seen from Figure 1, the time interval that maximizes the return $\overline{G}_{\pi_1}$ is 45, in arbitrary units. Notice that the scheduled preventive maintenance actions are performed independently on whether the component has failed and correctively repaired between two consecutive scheduled actions.

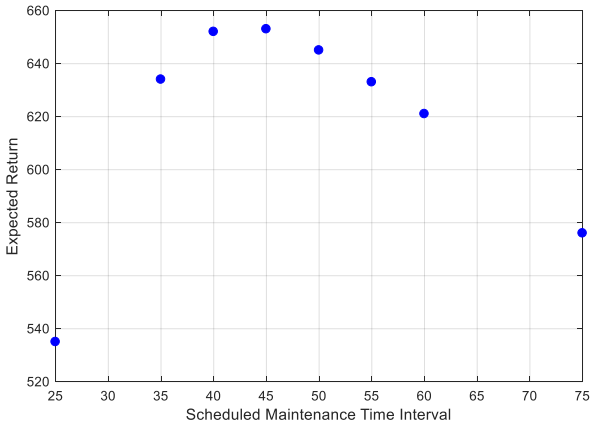

Figure 1 - Expected return for policy π_1 under different scheduled maintenance time intervals.

Figure 2 shows the expected return $\overline{G}_\pi$ for the three considered policies, together with the corresponding interval $\left[\overline{G}_\pi - \sigma_{G_\pi}, \overline{G}_\pi + \sigma_{G_\pi}\right]$, where the standard deviation refers to the values of G_π, and not to those of its estimator $\overline{G}_\pi$. From this Figure, we can see that the expected return $\overline{G}_{\pi^*}$ for the optimal policy π^* is the largest among the selected policies. Yet, the expected return $\overline{G}_{\pi_1}$ for the schedule maintenance policy π_1 is larger than that of policy π_2, $\overline{G}_{\pi_2}$ (Figure 2). This is mainly due to the synergy arising when performing preventive maintenance on both pumps, which reveals a better management strategy than that of running the components to failure and repairing them after the fact.

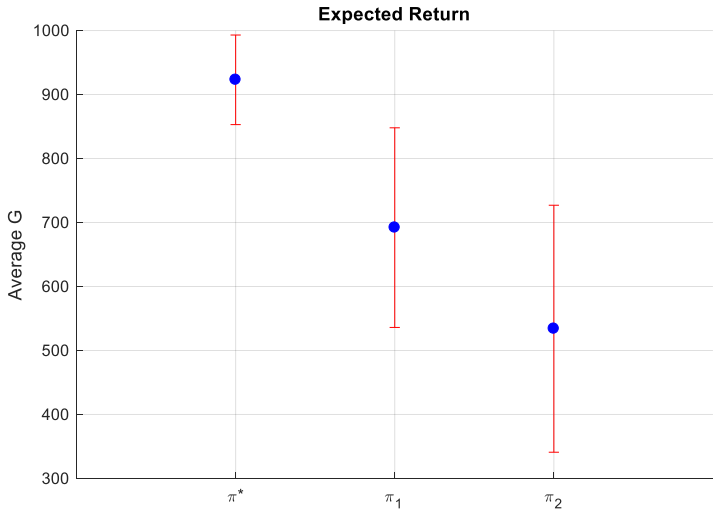

Figure 2 - Expected return for the selected policies

5.1 *Characteristics of the optimal policy*

Figure 3 shows the average number of times that the different actions have been taken throughout the Monte Carlo simulations when the MSS is obeying the optimal policy π^*. We can see that the additional benefit gained when both components run on high load (see Eq. (2)) is not large enough to justify the choice of this action for the whole time horizon. Indeed, the most frequent action (i.e., $\boldsymbol{A} = \left[a_1^1, a_2^2\right]$, $\boldsymbol{A} = \left[a_2^1, a_1^2\right]$, which corresponds to U=11) allows the MSS fulfilling the requirement W and, thus, avoiding penalties while degrading in a slower way compared to that of the setting in which both pumps are delivering the largest flow rate (i.e., $\boldsymbol{A} = \left[a_1^1, a_1^2\right]$).

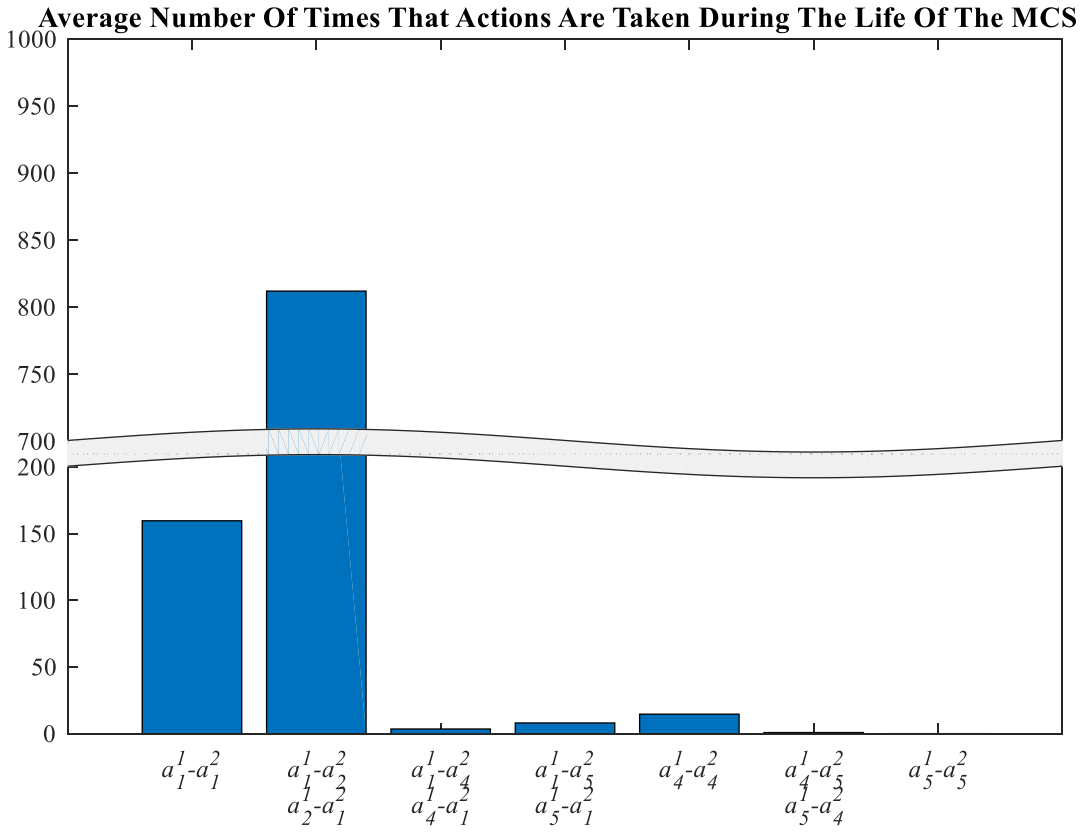

Figure 3 - Number of times that actions are taken during the life of the MSS for the optimal policy π^*.

Figure 4 and Figure 5 allow comparing the results of Figure 3 with the corresponding ones of policies π_1 and π_2, respectively. As expected, the most frequent action in both cases is $\boldsymbol{A} = \left[a_1^1, a_1^2\right]$ (i.e., largest flow rate level

on the pumps), which leads to a very fast degradation and, thus, to a number of corrective activities larger than that of π^* (i.e., actions $\boldsymbol{A}=\left[a_5^1,a_5^2\right]$, $\boldsymbol{A}=\left[a_1^1,a_5^2\right]$, $\boldsymbol{A}=\left[a_5^1,a_1^2\right]$), especially in case of π_2.

Obviously, policy π_1 requires performing a number of PM actions (i.e., $\boldsymbol{A}=\left[a_4^1,a_4^2\right]$, $\boldsymbol{A}=\left[a_1^1a_4^2\right]$ and $\boldsymbol{A}=\left[a_4^1,a_1^2\right]$), which allow reducing the number of corrective interventions, as it is confirmed by Table 1. This reports the average total number of maintenance actions carried out for the different policies: if we compare π_1 and π_2, we can see that the total number of maintenance actions in π_1 is almost 50% of that of π_2. This means that preventive actions are strongly beneficial for the profit of the MSS operation.

The optimal policy π^* requires the smallest number of maintenance activities, the most of which being preventive actions: this highlights that the optimal management of the MSS yields a reduction of maintenance costs because the total number of actions is minimized, while avoiding the large downtimes due to corrective actions.

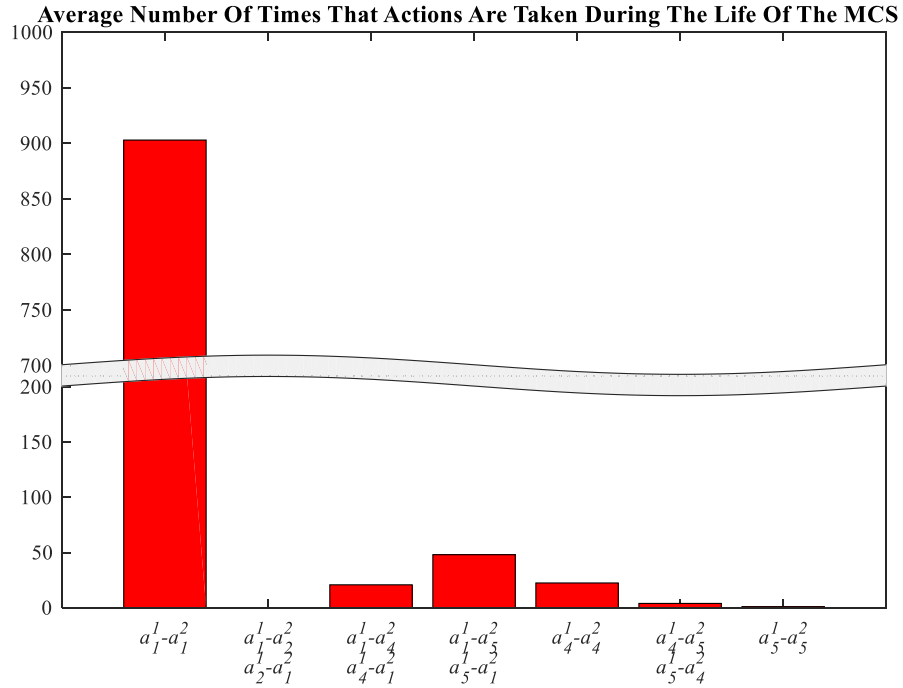

Figure 4-Number of times that actions are taken during the life of the MSS for the scheduled maintenance policy π_1.

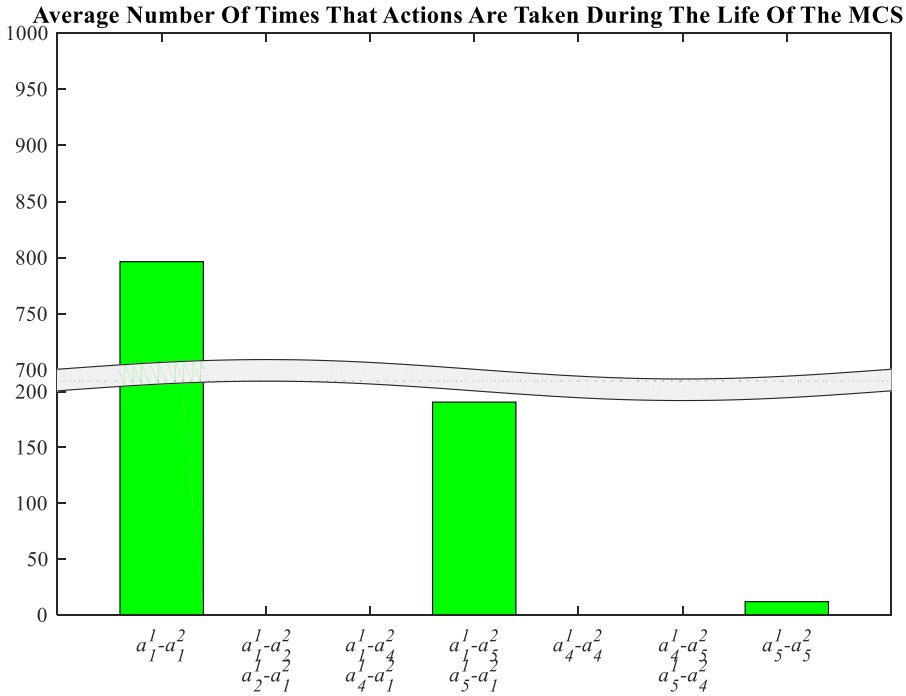

Figure 5-Number of times that actions are taken during the life of the MSS for the corrective maintenance policy π_2

	π^*	π_1	π_2
N° of PM $\{a_1^1 - a_4^2\} + \{a_4^1 - a_1^2\} + 2\{a_4^1 - a_4^2\} + \{a_4^1 - a_5^2\} + \{a_5^1 - a_4^2\}$	34	70	0
N° of CM $\{a_1^1 - a_5^2\} + \{a_5^1 - a_1^2\} + \{a_4^1 - a_5^2\} + \{a_5^1 - a_4^2\} + 2\{a_5^1 - a_5^2\}$	9	54	214

Table 1 - Total number of PM and CM actions taken on average for each policy.

Finally, it is worth noticing that in this case study the variability of the return in case of policies π_1 and π_2 is larger than that of policy π^*. This result can be justified by considering the instantaneous availability of the system with respect to the threshold *W*, which is defined as the probability that the MSS is able to supply, at any time instant, the due flow rate *W* (Figure 6).

Moreover, the behavior of the instantaneous availability corresponding to π^* is less oscillating than that of policies π_1 and π_2, because of the distribution of the loads on the pumps that minimizes their risk of failure while ensuring that the production request *W* is met. This results in a larger steady state availability. If we consider policy π_1, we can see periodical sharp reductions in the availability behavior, which correspond to the scheduled maintenance activities. These make the MSS unable to provide the necessary output to satisfy the production request *W*. These peaks are not present in case of π_2, because the units are not preventively stopped and repaired. In this case, there is a first reduction of the availability corresponding to the failure of one pump (around *t*=90), followed by an increase due to the maintenance activity that resets the pump into operation.

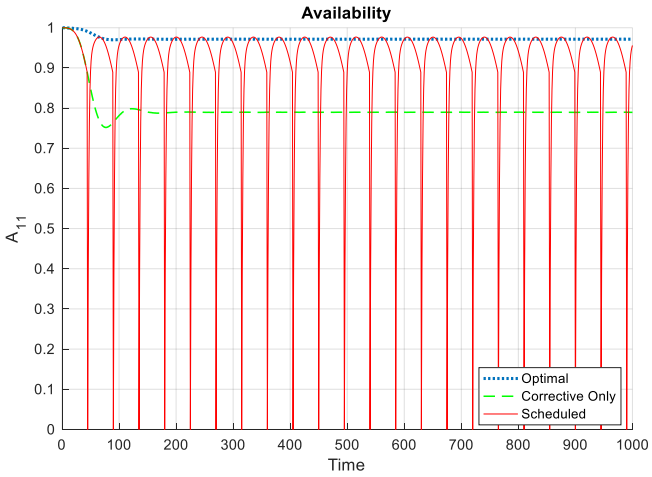

Figure 6 - Instantaneous availability of the MSS performing under the chosen policies.

6 CONCLUSION

The profitable operation of MSSs requires managing the component operational settings together with their preventive and corrective maintenance activities. On the one hand, when components run at high performance levels, the MSS operation profit increases; on the other hand, this results in accelerated failure behaviors, which require more frequent preventive actions and corrective actions.

To maximize the profit from the operation of a MSS, we have developed a decision support framework based on MDPs, which enables a comprehensive management of the MSS whereby maintenance decisions are taken together with those on the MSS operational settings, based on condition monitoring information about the health state of the MSS components.

To set the MDP, a mathematical framework has been developed, which relies on the definition of the system state, the corresponding possible actions with their effects and the reward function.

An application is shown to a MSS made up of 2 parallel pumps, which must deliver a minimum flow rate to not incur into losses. The results provided by the MDP have been compared to those of two other management policies and shown superior.

MDPs have proven to be a powerful tool in the field of condition-based management of MSS, enabling for improvements in terms of income and availability.

Future research will be aimed at developing Reinforcement Learning for identifying the management policy of MSSs with large state spaces, also in the situation in which the component degradation levels cannot be exactly known.

7 REFERENCES

[1] Baraldi P., Zio E., 2015, Guest Editorial, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, Vol. 229, Issue 4, pp. 277 – 278.

[2] Bian L., Gebraeel N., 2014, Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions, *IIE Transactions*, 46:5, 470-482.

[3] Cadini, F., Zio E., Avram D., 2009, Monte Carlo-based filtering for fatigue crack growth estimation, *Probabilistic Engineering Mechanics*, Vol. 24, Issue 3, pp. 367-373.

[4] Chan, G.K., Asgarpoor, S., 2006, Optimum maintenance policy with Markov processes, *Electric Power Systems Research*, Vol. 76, pp. 452–456

[5] Ding Y., Lisnianski A., 2008, Fuzzy universal generating functions for multi-state system reliability assessment, *Fuzzy Sets and Systems*, Vol. 159, Issue 3, pp. 307-324.

[6] Dawid, R., McMillan, D., Revie, M. 2015, Review of Markov Models for Maintenance Optimization in the Context of Offshore Wind, Annual Conference of the Prognostics And Health Management Society 2015.

[7] Fang S.-C., Puthenpura S., 1993, Linear Optimization and Extensions: Theory and Algorithms. Prentice Hall, Upper Saddle River, NJ.

[8] Grall A., Bérenguer C., Dieulle L., 2002, A condition-based maintenance policy for stochastically deteriorating systems, *Reliability Engineering & System Safety*, Vol. 76, Issue 2, pp. 167-180.

[9] Harlow D. G., Phoenix, S. L., 1978, The Chain-of-Bundles Probability Model for the strength of fibrous materials I: Analysis and Conjectures, *Journal of Composite Materials*, Vol. 12, Issue 2, pp. 195-214

[10] Haurie, A., L'Ecuyer, P., 1982, A stochastic control approach to group preventive replacement in a multicomponent system, *IEEE Transactions on Automatic Control*, Vol. 27, Issue 2, pp. 387-393.

[11] Hopp, W. J., Kuo, Y.-L., 1998, An optimal structured policy for maintenance of partially observable aircraft engine components, *Naval Research Logistics*, Vol. 45, pp. 335-352.

[12] Jardine A. K.S., Lin D., Banjevic D., 2006, A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing*, Vol. 20, Issue 7, 2006, pp. 1483-1510.

[13] Lisnianski, A., Laredo D., Haim, H.B., 2016, Multi-state Markov Model for Reliability Analysis of a Combined Cycle Gas Turbine Power Plant, *2016 Second International Symposium on Stochastic Models in Reliability Engineering, Life Science and Operations Management (SMRLO)*, Beer Sheva, pp. 131-135.

[14] Lisnianski, A., Levitin G., 2003, *Multi-State System Reliability: Assessment, Optimization and Applications*, Chapter 1, pp.15-50, World Scientific, New Jersey, London, Singapore, Hong Kong.

[15] Littman, M.L., Dean, T.L., Kaelbling, L.P., 1995, On the Complexity of Solving Markov Decision Problems, Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95, Montreal, Canada, pp. 394-402.

[16] Liu C., Chen N., Yanj J., 2015, New method for multi-state system reliability analysis based on linear algebraic representation, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, pp.469-482.

[17] Mansour, Y., Singh, S., 1999, On the complexity of policy iteration, Proceeding UAI'99 Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Stockholm, Sweden — July 30 - August 01, 1999, pp. 401-408

[18] Memarzadeh, M., Pozzi, M., 2016, Value of information in sequential decision making: Component inspection, permanent monitoring and system-level scheduling, *Reliability Engineering and System Safety*, Vol. 154, pp. 37–151.

[19] Mitchell M., 1999, *An introduction to genetic algorithms*, MIT Press, Cambridge, MA, USA.

[20] Nielsen, J.S., Sørensen, J.D., 2014, Methods for Risk-Based Planning of O&M of Wind Turbines, *Energies*, Vol. 7, pp. 6645-6664.

[21] Papakostantinou K.G., Shinozua, M., 2014a, Planning optimal inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory, Reliability Engineering and System Safety, Vol. 130, pp. 202-213.

[22] Papakostantinou K.G., Shinozua, M., 2014b, Planning optimal inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation, Reliability Engineering and System Safety, Vol. 130, pp. 214-224.

[23] Papakostantinou K.G., Shinozua, M., 2014c, Optimum inspection and maintenance policies for corroded structures using partially observable Markov decision processes and stochastic, physically based models, Probabilistic Engineering Mechanics, Vol. 37, pp. 93-108.

[24] Pipe, K., 2008, Practical prognostics for condition based maintenance, *International Conference on Prognostics and Health Management, PHM 2008*.

[25] Pozzi, M., Zonta, D., Wang, W., Chen, G., 2010, A framework for evaluating the impact of structural health monitoring on bridge management. *Proc. of 5th International Conf. on Bridge Maintenance, Safety and Management (IABMAS2010)*, Philadelphia, July 11–15, 2010, pp. on CD.

[26] Sharp, M., Coble, J., Nam, A., Hines, J.W., Upadhyaya, B., 2015, Lifecycle Prognostics: Transitioning between information types *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, Vol. 229, Issue 4, pp. 279-290.

[27] Shinstine, D. S., Ahmed, I., Lansey, K. E., 2002, Reliability/availability analysis of municipal water distribution networks: Case studies. *Journal of Water Resources Planning and Management*, Vol. 128, Issue 2, pp. 140-151.

[28] Sigaut O., Buffet O., 2010, Markov Decision Processes in Artificial Intelligence, Chapters 1-2, Wiley.
[29] Sutton, R. S., Barto, A. G., 1998, *Introduction to Reinforcement Learning (1st ed.)*. MIT Press, Cambridge, MA, USA.
[30] Tinga, T.,Jannsen, R., 2013, The interplay between deployment and optimal maintenance intervals for complex multi-component systems *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, Vol. 227, Issue: 3, pp. 227-240.
[31] Trappey A. J., Trappey C. V., Ni W-C, 2013, A multi-agent collaborative maintenance platform applying game theory negotiation strategies, *Journal of Intelligent Manufacturing*, Vol. 24, Issue 3, pp. 613-623.
[32] van Otterlo, M., Wiering, M., 2012, *Reinforcement Learning: State-of-the-Art*, Chapter 1, pp. 3-42, Springer.
[33] Wang, R., Chen, N., 2016, A survey of condition-based maintenance modeling of multi-component systems, *2016 IEEE International Conference on Industrial Engineering and Engineerin*g Management *(IEEM)*, Bali, Indonesia, pp. 1664-1668.
[34] Wang S., Hong L., Chen X., Zhang J., Yan Y., 2011, Review of interdependent infrastructure systems vulnerability analysis, *2011 2nd International Conference on Intelligent Control and Information Processing*, Harbin, pp. 446-451.
[35] Wijnmalen D. J. D., Hontelez J. A. M., 1997, Coordinated condition-based repair strategies for components of a multi-component maintenance system with discounts, *European Journal of Operational Research*, Vol. 98, Issue 1, pp. 52-63.
[36] Zille V., Bérenguer C., Grall A., Despujols A., 2011, Modelling multicomponent systems to quantify reliability centred maintenance strategies, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, Vol. 225, Issue 2, pp. 141 – 160.
[37] Zio E., 2009, *Computational Methods for Reliability and Risk Analysis*, Series on Quality, Reliability and Engineering Statistics: Volume 14, world Scientific, Singapore.
[38] Zio E., 2012, Prognostics and Health Management of Industrial Equipment, *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, IGI Global, pp.333-356.
[39] Zio, E. and Compare, M., 2013, Evaluating maintenance Policies by quantitative modeling and analysis, *Reliability Engineering and System Safety*, Vol. 109, pp. 53–65.
[40] Zonta D., Glisic, B., Adriaenssens, S., 2014, Value of information: Impact of monitoring on decision-making, *Structural Control and Health Monitoring*, Vol. 21, pp. 1043–1056.

8 APPENDIX 1

Initialize Q=0 for each state $\boldsymbol{s} \in S$, $\boldsymbol{A}$ available in $\boldsymbol{s}$.

Repeat

$\Delta = 0$

for each $\boldsymbol{s} \in S$

for each $\boldsymbol{A}$ available when in $\boldsymbol{s}$

q($\boldsymbol{s}$,$\boldsymbol{A}$)←$Q(\boldsymbol{s},\boldsymbol{A})$

$$Q(\boldsymbol{s},\boldsymbol{A}) = r(\boldsymbol{s},\boldsymbol{A},\boldsymbol{s}') + \gamma \max_{s',A'}(Q(\boldsymbol{s},\boldsymbol{A}',\boldsymbol{s}'))$$

% $\boldsymbol{s}'$, $\boldsymbol{A}'$ denote, respectively, all the states that can be reached from $\boldsymbol{s}$ when taking $\boldsymbol{A}$, and the actions that are available in the new state.

$$\Delta = \max(\Delta, \left| q(\boldsymbol{s},\boldsymbol{A}) - Q(\boldsymbol{s},\boldsymbol{A}) \right|)$$

Until $\Delta < \varepsilon$ (small positive number)

Output: Q, $\pi(\boldsymbol{s}) = \arg\max_{A}(Q)$

9 APPENDIX 2

In our study, the components undergo fatigue, a degradation process that causes microscopic cracks to grow when cyclic loads are applied. Once a crack reaches a critical size, it will propagate suddenly and the unit will fracture.

The evolution of the depth of the crack can be modeled by means of the Paris-Erdogan law (Cadini et al., 2009):

$$x_{k+1} = x_k + e^{\omega_k C}(\beta\sqrt{x_k})^n \Delta N$$

where x denotes the crack depth, N the load cycles, C and n are constants related to the material properties and β can be derived from experimental data, ω is a white Gaussian noise to represent the stochasticity in the evolution of the degradation mechanism and k is the current time step of the process (Cadini et al., 2009). This equation is used to simulate a large number of failure histories.

Then, we partition the crack depth [0,100] into bins representing the discrete health states of the degradation process. This allows us estimating the transition rates λ_{ij} from state *i* to state *j* through the maximum likelihood method:

$$\lambda_{ij} = \frac{n_{ij}}{n_{departures}}$$

where n_{ij} denotes the number of times that, starting from state *i*, the component ended in state *j* and $n_{departures}$ is the total number of times that the component departed from state *i*. Then, assuming a time step dt=1, the transition probability from state *i* to state *j* is: $p_{ij}=\lambda_{ij}dt$.

In our work, we performed 10^5 Monte Carlo simulations both for the load setting a_1^i, in which the load cycle ΔN is 10, and for the load setting a_2^i, whose load cycle is 1. Each simulation starts with a crack depth of $x_0 = 0.1$, in arbitrary units, and ends when the threshold for failure, $x_f = 100$ is reached. The number of bins is 15, which corresponds to the discrete number of states that was chosen for the study. The parameters characterizing the law of Paris-Erdogan can be found in (Cadini et al., 2009).

Below we report the estimated transition matrices $\boldsymbol{T}_1^i$ and $\boldsymbol{T}_2^i$, together with $\boldsymbol{T}_4^i$ and $\boldsymbol{T}_5^i$.

With respect to matrix $\boldsymbol{T}_2^i$, notice that the transition probabilities are not always decreasing with increasing degradation states. For example, the probabilities $\boldsymbol{T}_2^i(1,10)$, $\boldsymbol{T}_2^i(1,13)$ and $\boldsymbol{T}_2^i(1,15)$, of entering states $s_1^i = 10,13,15$ from state $s_1^i = 1$, *i=1, 2*, respectively, are smaller than $\boldsymbol{T}_2^i(1,12)$ and $\boldsymbol{T}_2^i(1,14)$ (i.e., the probabilities of reaching states $s_1^i = 12,14$, respectively). This is a consequence of the application of the Monte Carlo method to estimate the transition rates between states: transitions from the least degraded states to those highly degraded are rare events and, thus, their correct estimations require a large number of MC simulations. In this case study, $\boldsymbol{T}_2^i(1,10)$, $\boldsymbol{T}_2^i(1,13)$ and $\boldsymbol{T}_2^i(1,15)$ are set to 0 because the MC simulations have never experienced any of these transitions.

Notice that we could have applied a different MDP algorithm encoding the Monte Carlo simulation in the estimation of the value functions, instead of estimating off-line the transition matrices (e.g., Sutton and Barto, 1998). Our choice is motivated by a twofold justification. On one side, the algorithm proposed in this study is easier to implement, whilst the computational burdens of the two MDP solutions are similar to each other. On the other side, this work is a preliminary study on MDPs for managing systems with PHM equipped components, and in the future research work, we expect to address issues in which the transition matrices are known.

$$T_1^i = \begin{bmatrix}
0.968 & 0.029 & 0.002 & 4e-4 & 2e-4 & 9e-5 & 6e-5 & 3e-5 & 2e-5 & 2e-5 & 1e-5 & 7e-6 & 7e-6 & 4e-6 & 2e-5 \\
 & 0.899 & 0.087 & 0.007 & 0.002 & 0.001 & 6e-4 & 3e-4 & 2e-4 & 1e-4 & 1e-4 & 7e-5 & 5e-5 & 5e-5 & 2e-4 \\
 & & 0.866 & 0.113 & 0.011 & 0.004 & 0.002 & 0.001 & 6e-4 & 4e-4 & 3e-4 & 2e-4 & 2e4 & 1e-4 & 6e-4 \\
 & & & 0.840 & 0.131 & 0.015 & 0.006 & 0.003 & 0.002 & 9e-4 & 6e-4 & 4e-4 & 3e-4 & 2e-4 & 0.001 \\
 & & & & 0.819 & 0.145 & 0.019 & 0.006 & 0.003 & 0.002 & 0.001 & 8e-4 & 5e-4 & 4e-4 & 0.002 \\
 & & & & & 0.801 & 0.158 & 0.021 & 0.008 & 0.004 & 0.002 & 0.001 & 9e-4 & 7e-4 & 0.003 \\
 & & & & & & 0.785 & 0.167 & 0.023 & 0.009 & 0.005 & 0.003 & 0.002 & 0.001 & 0.004 \\
 & & & & & & & 0.772 & 0.175 & 0.026 & 0.01 & 0.005 & 0.003 & 0.002 & 0.006 \\
 & & & & & & & & 0.759 & 0.182 & 0.029 & 0.011 & 0.006 & 0.003 & 0.009 \\
 & & & & & & & & & 0.747 & 0.189 & 0.03 & 0.012 & 0.066 & 0.014 \\
 & & & & & & & & & & 0.736 & 0.195 & 0.033 & 0.013 & 0.023 \\
 & & & & & & & & & & & 0.726 & 0.200 & 0.035 & 0.038 \\
 & & & & & & & & & & & & 0.718 & 0.204 & 0.078 \\
 & & & & & & & & & & & & & 0.708 & 0.292 \\
 & & & & & & & & & & & & & & 0
\end{bmatrix}$$

$$T_2^i = \begin{bmatrix}
0.995 & 0.004 & 2e-5 & 4e-6 & 1e-6 & 4e-7 & 2e-7 & 9e-8 & 1e-7 & 0 & 4e-8 & 4e-8 & 0 & 4e-8 & 0 \\
 & 0.987 & 0.013 & 1e-4 & 3e-5 & 1e-5 & 5e-6 & 2e\text{-}6 & 8e-7 & 8e-7 & 1e-6 & 4e-7 & 2e-7 & 2e-7 & 4e-7 \\
 & & 0.982 & 0.018 & 2e-4 & 6e-5 & 2e-5 & 1e-5 & 5e-6 & 3e-6 & 1e-6 & 7e-7 & 9e-7 & 9e-7 & 3e-6 \\
 & & & 0.977 & 0.022 & 4e-4 & 9e-5 & 4e-5 & 1e-6 & 9e-5 & 5e-6 & 3e-6 & 2e-6 & 5e-7 & 4e-6 \\
 & & & & 0.973 & 0.026 & 5e-4 & 1e-4 & 5e-5 & 2e-5 & 1e-5 & 8e-6 & 3e-6 & 3e-6 & 1e-5 \\
 & & & & & 0.970 & 0.029 & 7e-4 & 2e-4 & 7e-5 & 4e-5 & 1e-5 & 1e-5 & 5e-6 & 1e-5 \\
 & & & & & & 0.967 & 0.032 & 8e-4 & 2e-4 & 8e-5 & 5e-5 & 2e-5 & 1e-5 & 3e-5 \\
 & & & & & & & 0.963 & 0.035 & 0.001 & 3e-4 & 1e-4 & 5e-5 & 3e-5 & 6e-5 \\
 & & & & & & & & 0.961 & 0.037 & 0.001 & 3e-4 & 1e-4 & 5e-5 & 1e-4 \\
 & & & & & & & & & 0.958 & 0.04 & 0.001 & 3e-4 & 1e-4 & 2e-4 \\
 & & & & & & & & & & 0.955 & 0.042 & 0.001 & 4e-4 & 4e-4 \\
 & & & & & & & & & & & 0.953 & 0.044 & 0.002 & 9e-4 \\
 & & & & & & & & & & & & 0.951 & 0.046 & 0.003 \\
 & & & & & & & & & & & & & 0.948 & 0.052 \\
 & & & & & & & & & & & & & & 0
\end{bmatrix}$$

$$T_4^i = \begin{bmatrix}
1 & & & & & & & & & & & & & & \\
0.5 & 0.5 & & & & & & & & & & & & & \\
0.5 & & 0.5 & & & & & & & & & & & & \\
0.5 & & & 0.5 & & & & & & & & & & & \\
0.5 & & & & 0.5 & & & & & & & & & & \\
0.5 & & & & & 0.5 & & & & & & & & & \\
0.5 & & & & & & 0.5 & & & & & & & & \\
0.5 & & & & & & & 0.5 & & & & & & & \\
0.5 & & & & & & & & 0.5 & & & & & & \\
0.5 & & & & & & & & & 0.5 & & & & & \\
0.5 & & & & & & & & & & 0.5 & & & & \\
0.5 & & & & & & & & & & & 0.5 & & & \\
0.5 & & & & & & & & & & & & 0.5 & & \\
0.5 & & & & & & & & & & & & & 0.5 & \\
0 & & & & & & & & & & & & & & 0
\end{bmatrix}$$

$$T_5^i = \begin{bmatrix}
0 & \Lambda & & & & & & & & & & & & & \\
M & 0 & & & & & & & & & & & & & \\
 & & O & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & O & & \\
 & & & & & & & & & & & & & 0 & \\
0.1 & & & & & & & & & & & & & & 0.9
\end{bmatrix}$$