OPTIMAL REPLACEMENT AND FORECAST HORIZON IN MULTI-MACHINE SYSTEMS

Ezequiel HERNÁNDEZ-GARCÍA*, Adriana GONZÁLEZ-QUIROZ*† R. Israel ORTEGA-GUTIÉRREZ*, Rubén BLANCAS-RIVERA**.
*Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico Matemáticas, Mexico.
**Universidad de las Américas Puebla, Departamento de Actuaría, Física y Matemáticas, México.

ABSTRACT
This work addresses the optimal replacement problem in multi-machine systems under deterioration and increasing operational costs. We model the problem as a finite-horizon Markov Decision Process and analyze two configurations: parallel and series systems. Using dynamic programming, we compute optimal replacement policies and illustrate them with numerical examples. Additionally, we introduce the concept of forecast horizon to identify when optimal actions stabilize, providing a practical tool for decision-making.
KEYWORDS: optimal replacement, Markov control processes, forecast horizon.
MSC: 60J05, 90C39, 90C40

RESUMEN
Este trabajo aborda el problema de reemplazo óptimo en sistemas con múltiples máquinas sujetas a deterioro y costos operativos crecientes. El problema se modela como un Proceso de Decisión de Markov con horizonte finito, y se analizan dos configuraciones: sistemas en paralelo y en serie. Mediante programación dinámica, se calculan políticas óptimas de reemplazo que se ilustran con ejemplos numéricos. Además, se introduce el concepto de horizonte de pronóstico para identificar el punto a partir del cual las acciones óptimas se estabilizan, lo cual proporciona una herramienta práctica para la toma de decisiones.
PALABRAS CLAVE: reemplazamiento óptimo, procesos de control de Markov, horizonte de pronósticos.

1. INTRODUCTION

In industrial and production systems, an important decision is determining the appropriate time to replace aging machinery and equipment. This choice has a direct influence on operational efficiency, cost control, and the overall reliability of the system. The literature has explored this topic from various perspectives. For instance, [14] analyzes a discrete-time, finite-horizon replacement problem involving two machines. In [2], the authors consider a finite population of economically interdependent machines, which allows for the study of how replacement decisions interact in multi-unit systems. Another angle is presented in [1], where replacement policies are examined under a risk-sensitive, discounted cost framework. More application-oriented approaches have also been proposed, such as in [12], which models the

†

real-life decision process for replacing bus engines and demonstrates how such techniques can be used effectively in practice.

Despite the extensive literature on machine replacement problems, several aspects remain insufficiently explored. In particular, most existing works focus either on single-machine settings, simplified multi-unit systems, or specific cost structures, without explicitly addressing the combined analysis of parallel and series configurations within a unified framework. Furthermore, the interaction between machines in series systems, where the system performance depends on the joint condition of all units, has received comparatively less attention.

In contrast to these approaches, this paper develops a unified model for the optimal replacement of $\mathfrak{n}$ machines under both parallel and series configurations within a discrete-time Markov control framework. The main contribution of this work lies in the formulation and analysis of the series case, where replacement decisions are interdependent and lead to more complex decision structures. Additionally, numerical experiments are provided to illustrate the behavior of the proposed policies and to highlight their performance across different system configurations. These results allow for a clearer comparison of the impact of system structure on optimal replacement strategies. Finally, the proposed framework opens the possibility of extending the analysis to more general configurations, such as mixed systems combining series and parallel structures, which constitutes a natural direction for future research.

Within this framework, we study the problem of deciding when to replace machines in a system made up of $\mathfrak{n}$ units, each of which deteriorates over time and incurs higher operating costs as its condition worsens. The goal is to develop replacement strategies that minimize the total cost of operation over a finite time horizon, by choosing at each stage whether it is better to keep a machine running or to replace it with a new one. We examine two system configurations: one where machines operate independently (parallel case), and another where their operation is interdependent (series case), meaning the system's performance depends on the condition of all machines. A central focus of this work is the analysis of the series configuration, where the interaction between deterioration and replacement decisions creates more complex but also more realistic scenarios for decision-making.

The rest of the article is structured as follows. Section 2 introduces the main concepts and tools used throughout the work, including the discrete-time Markov control process framework, the definition of control policies, the formulation of the optimization problem, and the dynamic programming approach. This foundation builds on the classic references by Puterman [11] and Hernández-Lerma and Lasserre [4]. In Section 3, we delve into the optimal replacement problem for a system composed of $\mathfrak{n}$-machines, extending the analysis in [5, 6], which address related settings under a random planning horizon, the latter proposing rolling horizon procedures for its solution. Here, we examine two system configurations: the parallel case, where machines operate independently, and the series case, where interdependence between machines leads to joint replacement decisions. Section 4 presents numerical examples that illustrate how the system behaves under the proposed policies in both configurations. Section 5 focuses on the concept of forecast horizons, applying this idea to the series configuration to examine when replacement decisions tend to stabilize over time. The paper ends with a section of concluding remarks, highlighting the main contributions and outlining directions for future work.

2. MARKOV CONTROL PROCESSES

To formally analyze the optimal replacement problem, we adopt the framework of discrete-time Markov control models. These models provide a foundation for sequential decision-making under uncertainty, allowing the system to evolve over time according to probabilistic dynamics. In addition, this section presents the dynamic programming methodology for solving finite-horizon control problems.

Definition 2.1. *A* stationary discrete-time Markov control model *(MCM) is a quintuple*

$$\mathcal{M} = (S, A, A(x) : x \in S, Q, C), \tag{2.1}$$

where:

(a) S *is a finite set called the* state space.

(b) A *is a finite set called* action space.

(c) $A(x)$ *denotes the set of* admissible actions *when the system is in state* $x \in S$*. Let* $\mathbb{K} := \{(x, a) : x \in S, \, a \in A(x)\}$*, which is called the admissible state-action set.*

(d) Q *is a stochastic kernel on* S *given* $\mathbb{K}$*, known as the* transition law*. That is, for each* $(x, a) \in \mathbb{K}$*,* $Q(\cdot \mid x, a)$ *is a probability measure on* S*, and for each* $B \in \mathcal{B}(S)$*,* $Q(B \mid \cdot)$ *is a measurable function (where* $\mathcal{B}(S)$ *denotes the Borel* σ*-algebra on* S*).*

(e) $C : \mathbb{K} \longrightarrow \mathbb{R}$ *is the per-stage* cost function.

The dynamics of the process follow the scheme described below: at each time step $t \in \mathbb{N}$, the system is in a current state $X_t = x \in S$, and an action $A_t = a \in A(x)$ is selected. As a result, a cost $C(x, a)$ is incurred, and the system transitions to a new state $X_{t+1} = y \in S$ according to the transition law $Q(\cdot \mid x, a)$. This process continues sequentially until a given finite time horizon $N \in \mathbb{N}$ is reached.

Policies. A policy is a decision rule that specifies which action to take at each time step, based on the information available up to that point. That is, a policy determines how the decision-maker selects actions depending on the current state of the system (and possibly the history of past states and actions). To formally define this concept, we first introduce the notion of the history of the process.

Definition 2.2. *Let* $\mathbb{H}_0 := S$*, and define* $\mathbb{H}_t := \mathbb{K} \times \mathbb{H}_{t-1}$ *for each* $t \in \mathbb{N}$*. An element* $h_t \in \mathbb{H}_t$ *is given by*

$$h_t = (x_0, a_0, x_1, a_1, \ldots, x_{t-1}, a_{t-1}, x_t),$$

and represents the history up to time t*, where* $(x_i, a_i) \in \mathbb{K}$ *for all* $i = 0, 1, \ldots, t-1$*, and* $x_t \in S$*. The set* $\mathbb{H}_t$ *is referred to as the* space of observed histories *up to time* t*.*

Definition 2.3. *A* policy *is a sequence* $\pi = \{\pi_t : t = 0, 1, 2, \ldots\}$ *of stochastic kernels defined on the action space* A*, given the history* $\mathbb{H}_t$*, such that* $\pi_t(A(X_t) \mid h_t) = 1$*, for all* $h_t \in \mathbb{H}_t$ *and* $t \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$*. The set of all such policies is denoted by* Π*.*

Let us define the set of measurable selectors:

$$\mathbb{F} := \{f : S \to A \,|\, f \text{ is measurable and } f(x) \in A(x) \text{ for all } x \in S\}.$$

A sequence of functions $\pi = \{f_t \in \mathbb{F} : t = 0, 1, 2, \dots\}$ is called a *Markov policy.* If $f_t = f$ for all t, then π is referred to as a *stationary Markov policy.*
Given an initial state $x_0 = x \in S$ and any policy $\pi \in \Pi$, there exists a unique probability measure $\mathbb{P}_x^\pi$ on the canonical product space $\Omega := (S \times A)^\infty$, endowed with the product σ-algebra $\mathcal{F}$. This measure is induced by the triplet (x, π, Q), and its existence follows from the Ionescu-Tulcea extension theorem [4]. The corresponding expectation operator is denoted by $\mathbb{E}_x^\pi$. The quadruple $(\Omega, \mathcal{F}, \mathbb{P}_x^\pi, \{X_t\})$ defines the stochastic process known as a *Markov Control Process* (2.1).

Optimal Control Problem. Given the concept of a policy, a natural question arises: how can we evaluate its quality? This is done through a performance criterion, which quantifies the expected cost or reward associated with a policy. In this work, we adopt the following criterion for a stationary Markov control model.

Definition 2.4. *Let $\pi \in \Pi$ and $x \in S$. The* total expected discounted cost *with a finite horizon $N \in \mathbb{N}$ is defined as*

$$J(\pi, x) := \mathbb{E}_x^\pi \left[\sum_{t=0}^{N-1} \alpha^t C(X_t, A_t) \right], \tag{2.2}$$

where $\alpha \in (0, 1)$ is the discount factor.

Definition 2.5. *Given the performance criterion* (2.2)*, the* optimal value function *is defined as*

$$J^*(x) := \inf_{\pi \in \Pi} J(\pi, x), \quad x \in S.$$

Then, the *optimal control problem* consists in finding a policy $\pi^* \in \Pi$ such that

$$J(\pi^*, x) = J^*(x), \quad \forall x \in S.$$

In this case, π^* is called an *optimal policy.*

Dynamic Programming. Dynamic programming is a mathematical technique designed to solve sequential decision-making problems over multiple stages, where the objective is to minimize the total accumulated cost. The core idea of dynamic programming is to break down the N-stage optimal control problem into smaller subproblems, each involving fewer stages. The solutions to these subproblems are then combined to construct the global solution, progressively optimizing the cost over a reduced number of time periods. We now state the dynamic programming result for the optimal control problem defined above. For a detailed proof of this result, the reader may refer to standard references such as [11] or [4].

Theorem 2.1.. *Let $J_0, J_1, ..., J_N$ be a sequence of functions defined on the state space S as follows:*

$$J_0(x) := 0, \quad x \in X$$

and for $t = 1, ..., N$, and $x \in S$

$$J_t(x) := \min_{a \in A(x)} \left[C(x,a) + \alpha \sum_{y \in S} J_{t-1}(y) Q(dy|x,a) \right]. \tag{2.3}$$

Then, for each $t = 1, ..., N$, there exists a selector $f_t \in \mathbb{F}$ such that $f_t(x) \in A(x)$ attains the minimum in (2.3) for every $x \in S$; that is,

$$J_t(x) = C(x, f_t(x)) + \alpha \sum_{y \in S} J_{t-1}(y) Q(dy|x, f_t(x)).$$

Therefore, the deterministic Markovian policy $\pi^* = \{f_1, f_2, ..., f_{N-1}\}$ is optimal, and the associated value function satisfies

$$J(\pi^*, x) = J^*(x), \quad \forall x \in S.$$

3. THE OPTIMAL REPLACEMENT PROBLEM FOR MULTI-MACHINE SYSTEMS

The optimal replacement problem analyzed in this work is presented below. We begin by introducing the fundamental elements of the model, including the definition of the state space, the action set, the set of admissible actions, and the cost structure. We then consider two distinct approaches: the parallel case, where the $\mathfrak{n}$-machines are assumed to operate independently, and the series case, where the $\mathfrak{n}$-machines exhibit interdependencies. The distinction between these two configurations will be reflected in the structure of the transition law.

We consider a system composed of $\mathfrak{n}$ identical machines, which may operate either independently or dependently on one another. The possible deterioration levels of each machine are denoted by $1, 2, \ldots, D$, where D is assumed to be a positive integer. Level 1 indicates that the machine is in perfect condition, and it is assumed that level i is better than level $i+1$, for all $i = 1, 2, \ldots, D-1$. At the beginning of each observation period, the system state can be represented by the vector $(d_1, d_2, \ldots, d_{\mathfrak{n}})$, where d_k, $k = 1, 2, \ldots, \mathfrak{n}$, indicates the deterioration level at which the $k-th$ machine is operating. Accordingly, the state space is defined as

$$S = \{(d_1, d_2, \ldots, d_{\mathfrak{n}}) | d_k \in \{1, 2, \ldots, D\}, k = 1, 2, \ldots, \mathfrak{n}\},$$

and its cardinality is $|S| = D^{\mathfrak{n}}$.

For the action space, we assume that at the beginning of each time period, two possible actions can be taken for each machine:

a) allow the $k-th$ machine, for $k = 1, 2, \ldots, \mathfrak{n}$, to continue operating at its current deterioration level, or

b) replace it with a new unit at a fixed cost $R > 0$.

To formalize the action space, we define the action vector as $(a_1, a_2, \ldots, a_\mathfrak{n})$, where $a_k \in \{0, 1\}$; the value $a_k = 0$ indicates that the $k-th$ machine continues to operate in its current state d_k, while $a_k = 1$ indicates that it is replaced. Thus, the action space is given by

$$A = A(x) = \{(a_1, a_2, \ldots, a_\mathfrak{n}) | a_k \in \{0, 1\}, k = 1, 2, \ldots, \mathfrak{n}\},$$

and its cardinality is $2^\mathfrak{n}$.

To define the cost structure, let $g : \{1, 2, 3, \ldots, D\} \to \mathbb{R}$ be a known function, where $g(i)$, $i = 1, 2, 3, \ldots, D$ represents the operating cost of a machine when it is in deterioration level i. The function g is assumed to be non-decreasing:

$$g(1) \leq g(2) \leq \ldots \leq g(D).$$

The stage cost function at any time t is defined as

$$C(X_t, A_t) = \sum_{k=1}^{\mathfrak{n}} \gamma(X_{k,t}, A_{k,t}),$$

where

$$\gamma(X_{k,t}, A_{k,t}) = \begin{cases} g(X_{k,t}) & \text{if } A_{k,t} = 0 \\ g(1) + R & \text{if } A_{k,t} = 1 \end{cases} \tag{3.1}$$

In this context, $X_{k,t}$ denotes the state (deterioration level) of the $k-th$ machine at time t, while $A_{k,t}$ indicates the action taken on that machine at the same time.

We now distinguish between two configurations: the parallel case, in which machines operate independently of one another, and the series case, where the functioning of each machine depends on the condition of the others.

The Parallel Case. In the context of the optimal replacement problem, the parallel case refers to the configuration in which the $\mathfrak{n}$ machines operate independently of one another. This independence implies that the decision to replace any given machine can be made by examining its individual deterioration state alone, without regard to the condition of the other machines. Consequently, the overall control problem decouples into $\mathfrak{n}$ separate and identical subproblems, one for each machine. This configuration is commonly found in systems such as data centers, where multiple servers operate independently and can be maintained or replaced without affecting the functionality of others, or in manufacturing settings with redundant production lines operating in parallel [3]. To model the system dynamics in the parallel configuration, we begin by specifying the deterioration process for an individual machine. Let $P^0 = (p^0_{i,j})_{D\times D}$ denote the transition matrix when a machine is *not replaced*. Here, $p^0_{i,j}$ represents the probability that a machine in deterioration level i transitions to level j in the next period. Since a machine cannot improve its condition spontaneously, we assume $p^0_{i,j} = 0$ for all $j < i$.

Similarly, let $P^1 = (p^1_{i,j})_{D\times D}$ be the transition matrix when the machine *is replaced*. In this case, the machine is reset to level 1 with certainty, so we define

$$p^1_{i,j} = \begin{cases} 1 & \text{if } j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

To define the system-level dynamics, consider the global state $i = (i_1, i_2, \ldots, i_{\mathfrak{n}}) \in S$ and an action vector $a = (a_1, a_2, \ldots, a_{\mathfrak{n}}) \in A$, where each $a_k \in \{0, 1\}$ indicates whether the k-th machine is replaced. The transition probability from state i to state $j = (j_1, j_2, \ldots, j_{\mathfrak{n}}) \in S$, under action a, is given by

$$Q^a_{i,j} = \prod_{k=1}^{\mathfrak{n}} p^{a_k}_{i_k, j_k},$$

which expresses the independence of machines in the parallel case.

The Series Case. In the context of the optimal replacement problem, the *series case* refers to a configuration in which the $\mathfrak{n}$ machines operate in a mutually dependent manner. In this setting, the decision to replace a given machine cannot be made in isolation; instead, it must consider the joint state of all machines in the system, since the functionality of the entire process depends on the performance of each individual component. This type of configuration arises frequently in production lines, where the failure or poor condition of a single station can disrupt the entire workflow, or in mission-critical systems such as medical equipment chains or aerospace subsystems, where the components must operate in coordinated sequence [10].
To define the transition kernel for the series configuration, we assume that the deterioration dynamics of each machine depend on the global state of the system, reflecting the interdependence among components. Let us denote the system state by $i = (i_1, i_2, \ldots, i_{\mathfrak{n}}) \in S$, where i_k indicates the deterioration level of machine k. Similarly, let $j = (j_1, j_2, \ldots, j_{\mathfrak{n}}) \in S$ be a possible successor state.

We define two transition matrices that govern the deterioration process depending on whether the machines are replaced or not.

a) The matrix $P^0 = (p^0_{i,j})_{D^{\mathfrak{n}} \times D^{\mathfrak{n}}}$ corresponds to the case in which no replacement occurs. The entry $p^0_{i,j}$ represents the probability that the system transitions from state i to state j under natural deterioration. In this configuration, transitions are constrained such that for each k, we have $j_k \geq i_k$, since machines cannot improve their deterioration levels spontaneously:

$$p^0_{i,j} = 0 \quad \text{if there exists } k \text{ such that } j_k < i_k.$$

b) The matrix $P^1 = (p^1_{i,j})_{D^{\mathfrak{n}} \times D^{\mathfrak{n}}}$ models the scenario in which all machines are simultaneously replaced. In this case, the system transitions deterministically to the fully renewed state:

$$p^1_{i,j} = \begin{cases} 1 & \text{if } j = (1, 1, \ldots, 1), \\ 0 & \text{otherwise.} \end{cases}$$

Given a state $i = (i_1, \ldots, i_{\mathfrak{n}}) \in S$ and an action $a = (a_1, \ldots, a_{\mathfrak{n}}) \in A$, the overall transition probability from i to $j = (j_1, \ldots, j_{\mathfrak{n}})$ is given by:

$$q^a_{i,j} = \mathbb{P}(X_{t+1} = j \mid X_t = i, A_t = a),$$

which must be specified for the series system. Unlike the parallel case, this probability is not simply the product of marginal transition probabilities for each machine, due to the coupling between them.

One possible modeling approach is to define:

$$q_{i,j}^{a} = \psi(i, j, a),$$

where ψ is a context-specific function that encodes the dependence structure. For example, deterioration in machine k may depend not only on its own level i_k but also on the maximum deterioration among preceding machines:

$$\psi(i, j, a) = \prod_{k=1}^{\mathfrak{n}} p_{i,j}^{(k,a_k)},$$

where $p_{i,j}^{(k,a_k)}$ depends on a systemic degradation rule (e.g., bottleneck or cascade failure models).

4. DYNAMIC PROGRAMMING ALGORITHM FOR REPLACEMENT OPTIMIZATION

Before presenting the numerical procedures, we highlight how the optimal replacement problem for multi-machine systems fits into the dynamic programming framework introduced in Theorem 2.1.. Specifically, for each stage $t = 1, \ldots, N$ and state $x = (x_1, \ldots, x_{\mathfrak{n}}) \in S$, the optimal value function satisfies the Bellman recursion:

$$J_t(x) = \min_{a \in A(x)} \left[C(x, a) + \alpha \sum_{y \in S} J_{t-1}(y)\, Q(y \mid x, a) \right], \tag{4.1}$$

where:

a) $C(x, a) = \sum_{k=1}^{\mathfrak{n}} \gamma(x_k, a_k)$ is the stage cost (see (3.1)),

b) $Q(y \mid x, a)$ is the system transition kernel under action a, reflecting either independent (parallel) or dependent (series) machine behavior.

This recursive structure forms the foundation of the backward computation used in the numerical algorithm presented below. To implement the optimal replacement strategy, we propose a dynamic programming algorithm tailored to finite-horizon Markov control processes. The procedure computes the optimal value function and the corresponding decision policy at each stage and for every possible system state. It is applicable to both the parallel and series configurations discussed earlier. **Algorithm 1** presents the step-by-step backward recursion, starting from the final stage and proceding toward the initial time period, in accordance with the dynamic programming formulation developed earlier (4.1). Although the series configuration has the same theoretical time complexity as the parallel case, it exhibits greater spatial complexity due to the need to store the full transition matrix Q, which has size $D^{\mathfrak{n}} \times D^{\mathfrak{n}}$ for each action. In contrast, the parallel case leverages the independence between machines to compute transitions on the fly, avoiding the storage of large matrices. This difference in memory requirements becomes critical as either the number of machines $\mathfrak{n}$ or the number of deterioration levels D increases, making the series configuration significantly more demanding in terms of both memory and computation in practical scenarios.

Algorithm 1 Dynamic programming algorithm for computing the optimal value function and policy.

```
Input: Horizon N, deterioration levels D, number of machines n, state space S, action space A,
  transition matrix Q, cost vector c, replacement cost R, discount factor α
Output: Value matrix J, policy matrix P
J ← zero matrix of size (N + 1, D^n)                              ▷ Optimal value function
P ← zero matrix of size (N, D^n)                                  ▷ Optimal policies
for j = 0 to D^n − 1 do
    J[N, j] ← 0
end for
for i = N − 1 to 0 do                                             ▷ Backward over stages
    for j = 0 to D^n − 1 do                                       ▷ For each state
        vp ← zero vector of size 2^n                              ▷ Expected value vector
        C ← zero vector of size 2^n                               ▷ Immediate cost vector
        v ← zero vector of size 2^n                               ▷ Total cost vector
        for s = 0 to 2^n − 1 do                                   ▷ For each action
            vp[s] ← 0
            for k = 0 to D^n − 1 do                               ▷ Expected value of J at next stage
                vp[s] ← vp[s] + J[i + 1, k] × Q[s, j, k]
            end for
            C[s] ← 0
            for k = 0 to n − 1 do                                 ▷ Immediate cost computation
                if A[s, k] == 0 then
                    C[s] ← C[s] + c[S[j, k] − 1]
                else
                    C[s] ← C[s] + c[0] + R
                end if
            end for
            v[s] ← C[s] + α × vp[s]                               ▷ Total expected cost
        end for
        J[i, j] ← v[0]                                            ▷ Initialize minimum value
        P[i, j] ← 1                                               ▷ Initial optimal action
        for s = 1 to 2^n − 1 do                                   ▷ Search for optimal action
            if v[s] < J[i, j] then
                J[i, j] ← v[s]
                P[i, j] ← s + 1
            end if
        end for
    end for
end for
Return J, P
```

5. NUMERICAL EXAMPLES

To illustrate the application and performance of the optimal replacement strategy for multi-machine systems, this section presents a set of numerical examples based on the dynamic programming formulation developed earlier. We consider both the parallel and series configurations, comparing the resulting optimal policies and value functions under different parameter settings.

5.1. Numerical Results for the Parallel Machine Case

We now illustrate the optimal replacement strategy for the parallel configuration through a numerical example under a discounted finite-horizon setting. Consider a system with two identical machines ($\mathfrak{n} = 2$) and three levels of deterioration ($D = 3$). The deterioration dynamics when a machine is not replaced are governed by the transition matrix:

$$P^0 = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0 & 0.3 & 0.7 \\ 0 & 0 & 1 \end{pmatrix}.$$

The operation cost vector is $g = (2, 3, 7)$, the replacement cost is $R = 4$, the horizon is $N = 5$, and the discount factor is $\alpha = 0.2$.

Given that the terminal cost is initialized as $J_0(x) = 0$ for all $x \in S$, the first backward iteration of the dynamic programming algorithm yields

$$\begin{aligned} J_1(x) &= \min_{a \in A(x)} \left[C(x,a) + \alpha \sum_S J_0(y) Q(dy|x,a) \right] \\ &= \min_{a \in A(x)} \left[C(x,a) \right]. \end{aligned}$$

since the value function at stage 0 is zero. For instance, for the initial state $(1, 1)$, we evaluate the cost of all possible actions:

$$\begin{aligned} J_1(1,1) &= \min \left\{ C((1,1),(0,0)), C((1,1),(0,1)), C((1,1),(1,0)), C((1,1),(1,1)) \right\} \\ &= \min \left\{ 4, 8, 8, 12 \right\} = 4. \end{aligned}$$

Thus, the optimal value at stage 1 for state $(1, 1)$ is $J_1(1, 1) = 4$, and the optimal action is $(0, 0)$, meaning that neither machine is replaced.

By applying the backward algorithm iteratively up to stage $t = 5$, we obtain the optimal value functions $J_t(x)$ and the corresponding policies for each state $x \in S$ and time step t. The results, summarized in Table 1, were computed in `Python 3.10` using Algorithm 1.
From Table 1, the structure of the optimal policy can be summarized as follows:

- When machines are in deterioration levels 1 or 2, it is optimal not to replace them at any stage.
- Machines in deterioration level 3 should be replaced immediately, regardless of the stage.

Initial State	J_1	f_1	J_2	f_2	J_3	f_3	J_4	f_4	J_5	f_5
(1,1)	4	(0,0)	5.4	(0,0)	5.6824	(0,0)	5.7358	(0,0)	5.7477	(0,0)
(1,2)	5	(0,0)	6.72	(0,0)	6.9784	(0,0)	7.0317	(0,0)	7.0427	(0,0)
(1,3)	8	(0,1)	9.1	(0,1)	9.3812	(0,1)	9.4361	(0,1)	9.4469	(0,1)
(2,1)	5	(0,0)	6.72	(0,0)	6.9784	(0,0)	7.0317	(0,0)	7.0427	(0,0)
(2,2)	6	(0,0)	8.04	(0,0)	8.2744	(0,0)	8.3276	(0,0)	8.3387	(0,0)
(2,3)	9	(0,1)	10.42	(0,1)	10.6772	(0,1)	10.732	(0,1)	10.7429	(0,1)
(3,1)	8	(1,0)	9.1	(1,0)	9.3812	(1,0)	9.4361	(1,0)	9.4469	(1,0)
(3,2)	9	(1,0)	10.42	(1,0)	10.6772	(1,0)	10.732	(1,0)	10.7429	(1,0)
(3,3)	12	(1,1)	12.8	(1,1)	13.08	(1,1)	13.1364	(1,1)	13.1471	(1,1)

Table 1: Optimal Value and Policy Functions for Each State and Stage.

5.2. Numerical Results for the Series Machine Case

We now illustrate the optimal replacement strategy for the series configuration, where the operation of each machine depends on the condition of the others. In this setting, the system state is defined as an ordered pair (i_1, i_2), where $i_k \in \{1, 2, 3\}$ denotes the deterioration level of the $k-th$ machine, $k = 1, 2$. There are nine possible system states, given by the Cartesian product $S = \{1, 2, 3\} \times \{1, 2, 3\}$.

Unlike the parallel configuration, the deterioration dynamics in the series model are stochastically dependent. The degradation of one machine affects the deterioration probability of the other. This interaction is modeled through conditional transition probabilities, where the likelihood of transitioning to a worse state depends on the current state of both machines.

Transition Rule. Let $p_{i\to j}^{(m)}$ be the probability that a machine in state i transitions to state j given that the other machine is in state m. We define the following conditional probabilities:

Current State i	Other Machine m	$p_{i\to i}^{(m)}$	$p_{i\to i+1}^{(m)}$
1	1	0.6	0.4
1	3	0.3	0.7
2	1	0.3	0.7
2	3	0.1	0.9
3	*	1.0	0.0

Remark 1. *In the last row of the table, the symbol "$*$" under Other Machine m indicates that the transition probabilities for a machine in deterioration level* 3 *are independent of the state of the other machine. This reflects the fact that level* 3 *is modeled as an absorbing state; once a machine reaches this level, it remains there with probability one, regardless of external conditions.*

The joint transition probability for the system is computed as the product of the conditional transitions of both machines. For instance, starting from state $(1, 1)$, the following transitions occur:

$$P\left((1,1)\to(1,1)\right)=0.6\times 0.6=0.36,$$
$$P\left((1,1)\to(1,2)\right)=0.6\times 0.4=0.24,$$
$$P\left((1,1)\to(2,1)\right)=0.4\times 0.6=0.24,$$
$$P\left((1,1)\to(2,2)\right)=0.4\times 0.4=0.16.$$

Extending this computation to all states yields the full transition matrix P^0, whose rows represent the transition probabilities under natural deterioration (i.e., when no replacements are made).

$$P^0=\begin{pmatrix} 0.36 & 0.24 & 0 & 0.24 & 0.16 & 0 & 0 & 0 & 0 \\ 0 & 0.18 & 0.42 & 0 & 0.12 & 0.28 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.18 & 0.42 & 0 & 0.12 & 0.28 & 0 \\ 0 & 0 & 0 & 0 & 0.03 & 0.27 & 0 & 0.07 & 0.63 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{pmatrix}$$

Each row of the matrix P^0 defines the transition probabilities from a given system state to all possible future states. Using **Algorithm 1**, we compute the optimal value functions $J_t(x)$ and the corresponding optimal actions $f_t(x)$ for all $x\in S$ and for each stage $t=1,\ldots,5$. The results are presented in Table 2.

Initial State	J_1	f_1	J_2	f_2	J_3	f_3	J_4	f_4	J_5	f_5
(1,1)	4.0000	(0,0)	4.9600	(0,0)	5.2365	(0,0)	5.2966	(0,0)	5.2966	(0,0)
(1,2)	5.0000	(0,0)	6.5000	(0,0)	6.8298	(0,0)	6.8919	(0,0)	6.8919	(0,0)
(1,3)	8.0000	(0,1)	9.6000	(0,1)	9.9200	(0,1)	9.9840	(0,1)	9.9840	(0,1)
(2,1)	5.0000	(0,0)	6.3800	(0,0)	6.7391	(0,0)	6.7963	(0,0)	6.7963	(0,0)
(2,2)	6.0000	(0,0)	8.1600	(0,0)	8.3826	(0,0)	8.4298	(0,0)	8.4298	(0,0)
(2,3)	9.0000	(0,1)	10.1600	(0,1)	10.3826	(0,1)	10.4298	(0,1)	10.4298	(0,1)
(3,1)	8.0000	(1,0)	9.6000	(1,0)	9.9200	(1,0)	9.9840	(1,0)	9.9840	(1,0)
(3,2)	9.0000	(1,0)	10.1600	(1,0)	10.3826	(1,0)	10.4298	(1,0)	10.4298	(1,0)
(3,3)	12.0000	(1,1)	13.6000	(1,1)	13.9200	(1,1)	13.9840	(1,1)	13.9840	(1,1)

Table 2: Optimal values and actions for the series case with dependent transitions.

The results obtained for both the parallel and series configurations reveal consistent patterns in the structure of the optimal replacement policy:

a) In both models, the optimal strategy does not recommend replacement when machines are in deterioration levels one or two, regardless of the time stage.

b) Replacement becomes optimal only when a machine reaches the third (most severe) level of deterioration, emphasizing the cost-benefit balance between continued operation and preventive replacement.

Despite this shared policy behavior, there are notable differences in the value functions and algorithmic complexity:

a) The series configuration exhibits slightly higher value functions in most states and stages. This is due to the dependency structure in the deterioration dynamics: the failure of one machine increases the likelihood of deterioration in the other, resulting in higher expected long-term costs.

b) The optimal actions remain stable across time stages. For a given state, the same action is consistently recommended from early to late stages, indicating a robust policy structure under interdependent dynamics.

c) From a computational perspective, the series model requires greater storage and computation time, as the transition probabilities depend jointly on the states of all machines. This dependency leads to a non-factorizable transition structure, unlike the parallel case, where transitions can be treated independently for each machine.

In summary, while the optimal policy is structurally similar in both models, the series configuration introduces additional risk and complexity, both in operational and computational terms. These observations naturally raise the question of how stable the optimal actions remain over time. To address this, we turn to the concept of the *forecast horizon*, which helps determine when future information stops affecting current decisions. The following section introduces this idea and describes how to compute it through discrepancy measures and the systematic elimination of suboptimal actions.

6. FORECAST HORIZON IN THE OPTIMAL REPLACEMENT PROBLEM

In stochastic dynamic decision problems with discounted cost criteria, it is often observed that the optimal action at the initial stage stabilizes after a finite number of steps. This phenomenon, known as the *forecast horizon*, plays a crucial role in reducing computational complexity and in supporting the use of rolling-horizon implementations.

In our optimal machine replacement model, the state and action spaces are both finite, and the deterioration process follows a Markovian dynamic. Within this framework, the forecast horizon refers to the minimum number of stages needed for the optimal action in a given initial state $x \in S$ to match the action prescribed by the infinite-horizon solution. Our goal is to identify the smallest such value N^* for which the optimal decision at the initial stage remains unchanged when the planning horizon is extended further.

To address this problem, we adopt the discrepancy-based approach proposed by Hernández-Lerma and Lasserre [4]. The method works by gradually eliminating suboptimal actions through a comparison of their discrepancy values at each stage with a computable threshold. As the process continues, actions that fail to meet the criterion are discarded. Once only one admissible action remains, it is identified as the optimal choice, and the iteration count at that point determines the forecast horizon for the initial state.

In our replacement problem, the state space $S = \{1, 2, \ldots, D\}^n$ and the action space $A = \{0, 1\}^n$ are both finite. Within this structure, the proposed method offers a systematic way to determine the forecast horizon N^* and to identify the optimal action a^* for any given initial state. To do this, we work with the optimal value function $J_n(x)$, which represents the cost-to-go from state x over a planning horizon of n stages. For each state $x \in S$ and action $a \in A(x)$, we define the discrepancy function

$$D_n(x, a) := C(x, a) + \alpha \sum_{y \in S} J_{n-1}(y) Q(y|x, a) - J_n(x),$$

which measures how much worse an action a performs compared to the optimal choice at stage n. Actions with large discrepancy values are flagged as suboptimal and can be excluded from future consideration.

We summarize the procedure in **Algorithm 2**, which depends on the following parameters:

a) $c_{\max} := \max_{(x,a) \in \mathbb{K}} C(x, a)$ is an upper bound on the one-stage cost.

b) $\gamma := \alpha\beta$, with $0 < \beta < 1/\alpha$. In our implementation, we set $\beta = 1/(l\alpha)$ for some $l > 1$, e.g., $l = 5$.

Furthermore, $count_m$ is an auxiliary function that counts the number of optimal actions at each stage.

Numerical Example

With the help of Algorithm 2, we compute the forecast horizon for all possible initial states using the parameters $\beta = \frac{1}{l\alpha}$ with $l = 5$, and $\gamma = \alpha\beta$. The results are presented in Table 6..

Initial state	Forecast horizon	Optimal action
(1,1)	2	(0,0)
(1,2)	3	(0,1)
(1,3)	3	(1,1)
(2,1)	3	(1,0)
(2,2)	2	(1,1)
(2,3)	3	(1,1)
(3,1)	3	(1,1)
(3,2)	3	(1,1)
(3,3)	2	(1,1)

Table 3: Forecast horizon and optimal actions for the parallel configuration

For example, in the initial state $(1, 1)$, the forecast horizon is 2 and the optimal action is $(0, 0)$, meaning that no replacement is needed. In contrast, for the state $(1, 2)$, the forecast horizon is 3 and the optimal action is $(0, 1)$, indicating that the second machine should be replaced. The values for the remaining states can be interpreted similarly as shown in Table 6..

Algorithm 2 Computation of the Forecast Horizon and Optimal Action

Input: S, A, D, x, J, c, R, α, γ, $c_{\max}$, n, `count_m`
Output: N^*, a^* ▷ Forecast horizon and optimal action
$i \leftarrow 1$ ▷ Initialize stage counter
$A_x \leftarrow$ set of admissible actions in state x
$Ag_x \leftarrow A_x$ ▷ Auxiliary array of candidate actions
while `count_m`$(Ag_x) \neq 1$ **do** ▷ Repeat until only one action remains
    $z \leftarrow 0$ ▷ Auxiliary index
    for each $a \in A_x$ **do**
        Compute discrepancy: $D_i(x,a) := C(x,a) + \alpha \sum_{y \in E} J[i-1,y] \cdot Q[y|x,a] - J[i,x]$
        if $D_i(x,a) \geq \dfrac{2c_{\max} \cdot \gamma^{i-1}}{1-\gamma}$ **then**
            $Ag_x[z] \leftarrow -1$ ▷ Eliminate action
        end if
        $z \leftarrow z + 1$
        if `count_m`$(Ag_x) = 1$ **then**
            break ▷ Stop if only one action remains
        end if
    end for
    $i \leftarrow i + 1$ ▷ Advance to the next stage
end while
for each j in the range of length(Ag_x) **do**
    if $Ag_x[j] \neq -1$ **then**
        $a^* \leftarrow Ag_x[j]$ ▷ Retrieve the surviving action
    end if
end for
Return: $N^* = i - 1$, a^*

7. CONCLUDING REMARKS

In this work, we studied the optimal replacement problem for systems consisting of $\mathfrak{n}$ machines under two operational settings: parallel and series. In both cases, we modeled the control process using finite-horizon Markov Control Processes with a total discounted cost criterion and solved it through dynamic programming. The parallel configuration assumes independent deterioration of each machine, which simplifies the decision-making process by allowing localized replacement policies. On the other hand, the series configuration introduces dependencies between machines, as the failure of a single unit compromises the functioning of the entire system.

An important contribution of this analysis is the incorporation of the forecast horizon concept. By employing a discrepancy-based elimination method [4], we were able to determine how many planning steps are needed before the optimal decision stabilizes. This is especially useful from a practical standpoint, since it shows that in many cases, a relatively short planning horizon is enough to make reliable decisions. The results align with intuitive expectations: machines are generally not replaced unless their deterioration level is high. Moreover, we observed that the tendency to delay replacement increases as future costs are discounted more heavily. Interestingly, machines in worse condition tend to reach stable decisions more quickly, while those in better shape require a longer planning horizon to determine the right time for replacement.

In summary, this study demonstrates how tools from Markov Decision Processes, combined with horizon prediction techniques, can be applied to maintenance problems involving multiple components. Future research directions may involve incorporating stochastic replacement times, modelling each machine as a Markov stopping game [8, 9, 13], as well as accounting for variability in the discount factor [7].

RECEIVED: SEPTEMBER, 2025.
REVISED: APRIL, 2026.

REFERENCES

[1] ÁVILA-GODOY, G., BRAU, A., AND FERNÁNDEZ-GAUCHERAND, E. (1997): Controlled markov chains with discounted risk-sensitive criteria: applications to machine replacement In **Proceedings of the 36th IEEE Conference on Decision and Control**, volume 2, pages 1115–1120. IEEE.

[2] CHILDRESS, S. AND DURANGO-COHEN, P. (2005): On parallel machine replacement problems with general replacement cost functions and stochastic deterioration **Naval Research Logistics (NRL)**, 52(5):409–419.

[3] GEURTSEN, M., ADAN, J., AND AKÇAY, A. (2023): Integrated maintenance and production scheduling for unrelated parallel machines with setup times **Flexible Services and Manufacturing Journal**, pages 1–34.

[4] HERNÁNDEZ-LERMA, O. AND LASSERRE, J. B. (2012): **Discrete-time Markov control processes: basic optimality criteria**, volume 30 Springer Science & Business Media.

[5] ILHUICATZI-ROLDAN, R. AND CRUZ-SUAREZ, H. (2012): Optimal replacement in a system of n-machines with random horizon **Proyecciones (Antofagasta)**, 31(3):219–233.

[6] ILHUICATZI-ROLDÁN, R. AND CRUZ-SUÁREZ, H. (2013): Rolling horizon procedures for the solution of an optimal replacement problem of n-machines with random horizon **Investigación Operacional**, 34(2):105–117.

[7] ILHUICATZI-ROLDÁN, R., CRUZ-SUÁREZ, H., AND CHÁVEZ-RODRÍGUEZ, S. (2017): Markov decision processes with time-varying discount factors and random horizon **Kybernetika**, 53(1):82–98.

[8] LÓPEZ-RIVERO, J., CAVAZOS-CADENA, R., AND CRUZ-SUÁREZ, H. (2022): Risk-sensitive markov stopping games with an absorbing state **Kybernetika**, 58(1):101–122.

[9] LÓPEZ-RIVERO, J., CRUZ-SUÁREZ, H., AND CAMILO-GARAY, C. (2024): Nash equilibria in risk-sensitive markov stopping games under communication conditions **AIMS Mathematics**, 9(9):23997–24017.

[10] NIU, J., YAN, R., AND ZHANG, J. (2025): Preventive replacement policies of parallel/series systems with dependent components under deviation costs **Reliability Engineering & System Safety**, 260:111033.

[11] PUTERMAN, M. L. (2014): **Markov decision processes: discrete stochastic dynamic programming** John Wiley & Sons.

[12] RUST, J. (1987): Optimal replacement of gmc bus engines: An empirical model of harold zurcher **Econometrica: Journal of the Econometric Society**, pages 999–1033.

[13] TORRES-GOMAR, M. A., CAVAZOS-CADENA, R., AND CRUZ-SUÁREZ, H. (2024): Denumerable markov stopping games with risk-sensitive total reward criterion **Kybernetika**, 60(1):1–18.

[14] ZACARÍAS-ESPINOZA, G., CRUZ-SUÁREZ, H., AND VENEGAS-PÉREZ, L. A. (2010): Control óptimo de dos máquinas usando políticas de reemplazo In **Novena Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI**, page 159.