Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

ADVANCED DYNAMIC SPARE PARTS INVENTORY MANAGEMENT UTILIZING MACHINE HEALTH DATA

J. Kruman[1], A. Kaufman[1], and Y. T. Herer[1]

[1]Faculty of Data and Decision Sciences. Technion Israel Institute of Technology, ISRAEL

ABSTRACT

This research presents a novel approach to spare parts inventory management by integrating real-time machine health data with dynamic, state-dependent inventory policies. Traditional static models overlook the evolving conditions of industrial machinery. Leveraging advanced digital technologies, such as those pioneered by Augury, our framework dynamically adjusts inventory levels, reducing costs and improving service. Using Markov chain modeling, simulation, and industry collaboration, we demonstrate up to 29% cost savings with state-dependent policies over static base-stock models. Sensitivity analysis confirms the robustness of these strategies.

1 INTRODUCTION

Effective spare parts inventory management is essential for continuous industrial operations. This research introduces a dynamic framework that leverages real-time machine health data, surpassing static models with fixed failure rate assumptions. By integrating real-time analytics, it improves inventory policies, reducing downtime and holding costs. Augury, a leader in machine health diagnostics, demonstrates how AI and IoT enable 24/7 monitoring for predictive maintenance, though these technologies remain underutilized in spare parts inventory management.

This study focuses on developing and testing dynamic policies for systems comprised of several machines organized in workstations—pairs of identical machines operating in a standby protocol—with one or two replaceable parts, demonstrating how real-time data integration enhances inventory management. Our system incurs both inventory holding costs and penalty costs. Through Markov chain modeling and simulation, we report up to 29% cost reduction when using state-dependent ordering policies compared to static policies. Future research will extend this model to third-party logistics (3PL) scenarios, incorporating machine learning to enhance decision-making.

The paper is organized as follows. It begins with the Literature review, followed by the Markov chain model introduction. A simulation study follows, followed by an extension to multiple component modeling, demonstrating the framework's scalability and broader applicability.

2 LITERATURE REVIEW

Our paper contributes to the literature on spare parts inventory management and the integration of machine health data, two important areas of research whose intersection has rarely been examined. We address this gap by leveraging live sensor data to model spare parts demand based on machine health state evolution within a Markov Decision Process (MDP) framework.

Spare parts inventory management poses unique challenges due to high demand variability and severe consequences of stock-outs; spare parts can constitute on average up to 2.5% of the equipment purchase price annually over a 30-year lifecycle (Zhang and Zeng 2017). Capital goods spare parts often experience irregular demand dictated by the health of the machines. Addressing these challenges requires integrating inventory management with predictive analytics and condition monitoring to ensure system reliability (Pinciroli et al. 2023; Van der Auweraer et al. 2019). Technological advancements have enabled real-time spare parts optimization in multi-echelon supply chains. IoT and AI tools now allow for dynamic programming and reinforcement learning to address uncertainty in parts needs and maintenance scheduling (Pinciroli et al. 2023). Industry 4.0 technologies such as predictive maintenance algorithms further enhance efficiency by reducing downtime and costs. However, practical applications remain limited, necessitating further research into their implementation (de Jonge and Scarf 2020). Despite these advancements, most current models are constrained by their reliance on static policies or simplified frameworks that do not fully exploit live machine health data for dynamic decision-making. This gap highlights the need for integrated models that dynamically adjust inventory policies based on real-time machine conditions.

The integration of machine health monitoring with spare parts inventory management has emerged as a transformative approach for limiting production downtime and reducing costs (Kritzinger et al. 2018). Condition-Based Maintenance (CBM) uses real-time condition monitoring to trigger maintenance actions based on signs of degradation. This minimizes unnecessary interventions while improving machine reliability. Remaining Useful Life (RUL) predictions further support preemptive maintenance actions like repairing machines or ordering spare parts (Dendauw et al. 2021). The integration of CBM with inventory management controls stock levels by incorporating RUL predictions into decision-making processes. Policies like the critical level policy prioritize corrective over preventive maintenance by reserving inventory for urgent repairs (Dendauw et al. 2021). Joint optimization models, such as Markov decision processes, synchronize maintenance scheduling with spare parts provisioning at the system level. These approaches account for dependencies between components in multi-unit systems, enabling cost-effective strategies like condition-based opportunistic preventive maintenance during corrective actions (Olde Keizer MCA et al. 2017; Zhang and Zeng 2017). While these studies illustrate the potential benefits of integrating CBM with inventory management, they often fall short of utilizing live sensor data to model machine health state evolution dynamically at the service of spare part inventory management. Filling this gap, our model utilizes machine health data from sensors to design efficient spare parts dynamic ordering policies that significantly reduce operations costs.

3 MODEL BUILDING

We begin by detailing our Markov chain model, which is based on the manufacturing operations of an oil refinery. We build our Markov chain model in three stages: we start with a model of a single machine, and then we expand our model to encompass workstations, defined as pairs of identical machines operating under a cold standby protocol. This configuration enhances operational continuity and introduces interdependencies between the active and standby machines. Next, we integrate these components into a system-wide model that includes multiple workstations and spare parts inventory management and incurs costs that we wish to minimize.

3.1 Machine

Our research is based on a model for a single machine using a five-state Markov chain, each state representing a distinct condition of the machine: (0) perfect condition, (1) needs monitoring, (2) needs adjustments, (3) imminent failure, and (4) failure. These states are mapped in a continuous time framework, reflecting real-world settings where machine conditions evolve over time. We define the state space of our continuous time Markov process to be $R_M = \{0,1,2,3,4\}$.

Each machine's state transitions were parameterized using real data, capturing the practical dynamics observed in operational environments. As illustrated in Figure 1, a machine in a perfect state (state 0) can deteriorate, requiring monitoring and adjustments until it inevitably fails. Notably, not all transitions are possible. At any given time, a machine's health can degrade from state i to any state j where $j > i$. Additionally, it can transition from state $i \in \{1, 2\}$ to state $j \in \{0, 1\}$ when $j < i$ without requiring a spare part, possibly due to an adjustment or routine maintenance. A machine can only transition from state (3) to state (4) since no adjustments can improve the situation once a machine reaches state (3). Finally, a machine can only transition from state (4) to state (0), since a repaired machine returns to perfect condition. This transition requires a spare part, which will be incorporated into the model in Section 3.3. We assume that the transition times between states i and j follow an exponential distribution with rate (λ_{ij}) for ($i \neq$

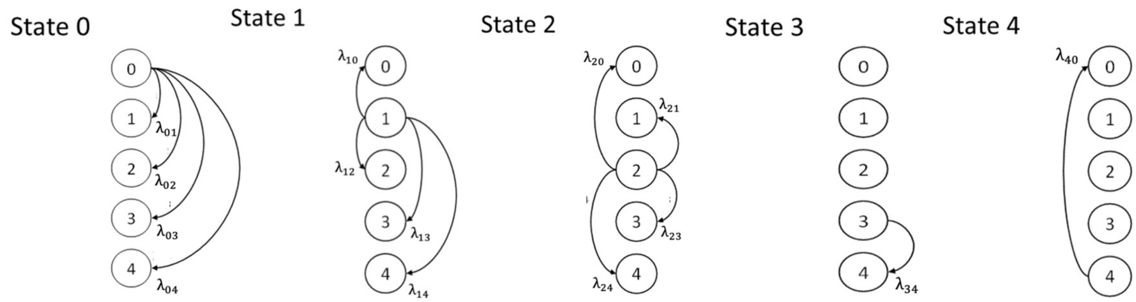

Figure 1: Transition rates for a single machine.

j). This rate depends on the machine's current state and its potential subsequent states. The possible transitions labeled with their rates are presented graphically in Figure 1.

3.2 Workstation

Building on the Markov chain model of a single machine introduced in the previous section, we define a second Markov chain model for a pair of identical machines following a cold standby protocol, with one machine operating actively while its counterpart remains idle, ready to take over operations if the first machine fails. We call this pair of machines a workstation. The workstation's dynamics, as illustrated in Figure 2, aim to maintain continuity of operations to the extent possible. The figure also illustrates interdependencies between the active and standby machines. This system allows for a seamless switch in operations without downtime, since the idle machine, if operational, activates immediately if and only if the working machine fails. Both machines, when active, are subject to the same probabilistic transition dynamics modeled by the machine Markov chain.

We represent the state of each workstation by an ordered pair (i, j) denoting the state of the active machine i followed by the state of the idle or failed machine j. The state (4,4), where both machines require repair, represent a total operational shutdown, which the system aims to avoid. Notably, the idle machines can only be in state 0 or state 4, and that states (4,0) and (0,4) denote the same situation since once the active machine fails the back-up machine takes over immediately. We define the state space of the Markov chain associated with a workstation as $R_W = \{(0,0), (1,0), (2,0), (3,0), (0,4), (1,4), (2,4), (3,4), (4,4)\}$. The state transitions in the workstation setting and the rates at which they occur are illustrated in Figure 3, providing a comprehensive view of how workstations react to changes in machine states. Note that the transition rate from state (4,4) to state (0,4) is $2\lambda_{40}$ as both machines in the workstation are being repaired, even though only the first of the two being repaired is recorded in the next transition. We denote by Λ the transition rate matrix of our workstations' Markov chain, where each element Λ_{ij} for $i \neq j$ represents the transition rate of a workstation from state i to state j, and $\Lambda_{ii} = -\sum_{i \neq j} \Lambda_{ij}$. The matrix Λ is a 9×9 square matrix, since the workstations can be in nine possible states.

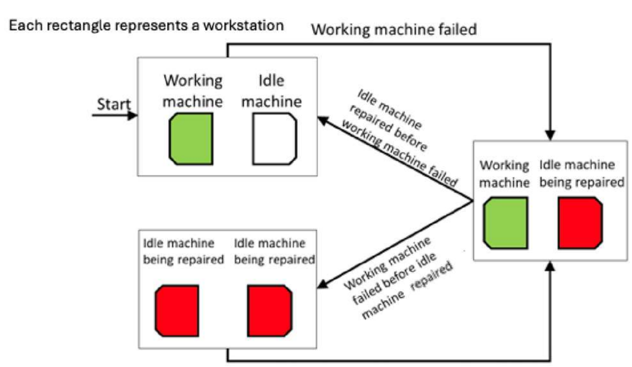

Figure 2: Workstation dynamics.

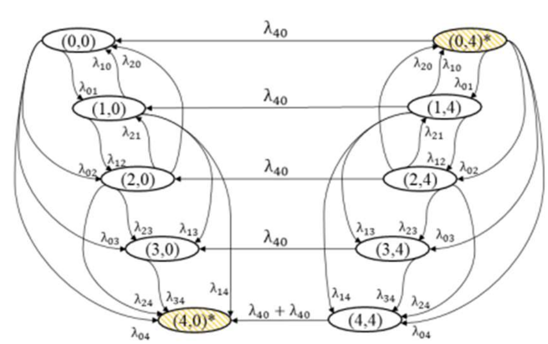

Figure 3: Workstation transition rates.

3.3 System

Building on the workstation Markov chain model detailed earlier, we define a comprehensive Markov chain model for our system, including managing spare parts inventory under a static policy and integrating the dynamics of multiple workstations. We introduce the notation W, which represents the number of workstations in our system and I, which represents the on-hand spare parts inventory and, by convention, includes the spare parts currently being used to repair machines. In addition, n_i represents the total number of machines, both working and idle, in state i for all $i \in R_M$, and $n_{(i,j)}$, which represents the number of workstations in state (i,j) for all $(i,j) \in R_W$. Recall that R_W has nine elements.

We denote by r the state of the system that is represented by the ordered ten tuple comprising the inventory level and the number of workstations in each state. We denote by I^r the inventory level when the system is in state r and by $n^r_{(i,j)}$ the number of workstations in state (i,j) when the system is in state r. That is, $r = \left(I^r; n^r_{(0,0)}, n^r_{(1,0)}, n^r_{(2,0)}, n^r_{(3,0)}, n^r_{(0,4)}, n^r_{(1,4)}, n^r_{(2,4)}, n^r_{(3,4)}, n^r_{(4,4)}\right)$. For simplicity, we often refer to the elements of this ten tuple without the superscript r. We note that $\sum_{(i,j)\in R_W} n_{(i,j)} = W$. The state space of the system, denoted by R_S, encompasses all possible states of the system defined by these ten tuples, constrained by the total number of workstations and the inventory replenishment policy.

Initially, the system operates under a standard static base-stock policy and we denote the base-stock level by S. Additional inventory policies will be investigated in Section 4.4. Using base-stock policies implies a maximum inventory level, denoted by I_{max} which depends on the base-stock level S as well as on the last five elements of system state r. For clarity, we introduce the notation r' that denotes the nine-tuple system state such that $r = (I; r')$. Thus, we have that $I^{r'}_{max} = S + n^{r'}_{(0,4)} + n^{r'}_{(1,4)} + n^{r'}_{(2,4)} + n^{r'}_{(3,4)} + 2n^{r'}_{(4,4)}$. For convenience, we define by $n^{r'}_i$ the total number of machines in state i, both working and idle, when the system is in state r'. Thus $I^{r'}_{max} = S + n^{r'}_4$. Defining O^r as the number of parts on order when the system is in state r, we have $O^r = I^r_{max} - I^r = S - I^{r'} + n^{r'}_4$.

When spare parts are limited and to help maintain workstation availability, machines are allocated spare parts according to a descending order of the state of the other machine at the workstation. That is, failed machines in workstations in state (4,4) will be given highest priority, failed machines in workstations in state (3,4) are at the next priority level and so on.

Our system continuous time Markov process is subject to three types of events:

- Transition between states without using a spare part: A workstation can change states following the deterioration or adjustment of the working machine. The transition rate to the next event of this type is given by $\left(n_{(0,0)} + n_{(0,4)}\right)\lambda(0) + n_1\lambda(1) + n_2\lambda(2) + n_3\lambda(3)$, where $\lambda(i) = \sum_j \lambda_{ij}$.
- Machine repair: As explained above, machine repair is subject to the availability of spare parts. Each machine in state 4 is repaired at rate λ_{40} only if there is an available spare part for it. If there are no available spare parts for failed machines, their transition rates will be 0 until the arrival of a spare part. As such, the transition rate to the next event of this type is $\min\{I, n_4\}\lambda_{40}$.
- Arrival of a spare part: Spare parts are supplied one at a time, with an exponential lead time and we denote by τ their average time of arrival i.e., the arrival rate of one spare part is $1/\tau$. The number of spare parts on order is $(O = S + n_4 - I)$. Thus, the arrival rate of the next spare part is O/τ.

When combining all the possible transition types, the total transition rate to the next event is equal to

$$\left(n_{(0,0)} + n_{(0,4)}\right)\lambda(0) + n_1\lambda(1) + n_2\lambda(2) + n_3\lambda(3) + min\{I, n_4\}\lambda_{40} + O/\tau. \qquad (1)$$

3.4 Cost

In this section, we detail the cost components that constitute the total yearly system cost in our spare parts inventory management model. We denote by p_r the steady state probability of being in state r. Let X be the random variable representing the yearly cost of the system. The expected value of the yearly cost is $E[X]$. Our goal is to minimize the yearly expected cost $E[X]$ by identifying and implementing an effective spare parts replenishment policy. The total yearly system cost is comprised of three components:

- The yearly inventory holding cost X_h is incurred for holding spare parts in inventory that are not being used to repair a failed machine and is paid at a rate of h dollars per unit per year. If all the spare parts on hand are being used to repair machines in state 4, then the system incurs a holding cost at a rate of 0. If, however, we have enough spare parts to repair all the machines in state 4, then at time t the inventory cost is being incurred at a yearly rate of $h(I^r - n_4^r)$. Thus, the expected yearly inventory holding cost is given by $E[X_h] = \sum_{(I,r')\in R_S} h\, p_r\big(I^r - n_4^{r'}\big)^+$.
- The yearly major penalty cost X_M arises when a workstation is in a complete failure state, leading to operational downtime. The penalty cost per workstation per unit time is denoted by M. The expected yearly major penalty cost $E[X_M]$ is given by $E[X_M] = \sum_{(I,r')\in R_S} M\, p_r n_{(4,4)}^{r'}$. We note that part of this cost is unavoidable and we will exclude this unavoidable component from our objective function.
- The yearly minor penalty cost X_m is incurred when a machine fails and is not being repaired. This penalty cost per machine per unit time is denoted by m. This cost, which includes the stress on the system due to the temporary loss of redundancy, is generally much lower than the major penalty cost but much higher than the holding cost. The yearly expected minor penalty cost is given by $E[X_m] = \sum_{(I,r')\in R_S} m p_r\big(n_4^{r'} - I^r\big)^+$.

Recall that a yearly major penalty cost is incurred when a workstation is in a complete failure state. Part of this major penalty cost X_M is unavoidable in the sense that even in a situation with unlimited spare parts, workstations still fail. We denote by $M_{S\to\infty}$ the expected yearly unavoidable major penalty cost, i.e., the major penalty cost with an infinite base-stock level. We determine the value of $M_{S\to\infty}$ by utilizing the stationary vector of the workstation Markov chain. When spare parts are always available, nothing is left to link the different workstations, thus a single system of W workstations can be viewed as W workstations operating independently. Using standard methods, we calculate the stationary probability of a workstation being in state (4,4) and denote it by $\pi_{(4,4)}$. Finally, to calculate $M_{S\to\infty}$—the long-term average yearly unavoidable major penalty cost for the entire system—we multiply $\pi_{(4,4)}$ by the number of workstations in the system (W) and the major penalty cost per unit time (M): $M_{S\to\infty} = \pi_{(4,4)} W M$.

We denote by X'_M the yearly average avoidable major cost, that is $X'_M = X_M - M_{S\to\infty}$, and by X' the average yearly total cost excluding this expected yearly unavoidable major cost $M_{S\to\infty}$. As such the total yearly expected system cost $E[X']$ is the sum of these three components:

$$E[X'] = E[X_h] + E[X'_M] + E[X_m].$$

Our goal is to determine the spare parts ordering policy that will minimize $E[X']$.

3.5 Determining the Ordering Policy

If we could find the stationary distribution of the system Markov chain, then we would first determine the best base-stock level by evaluating the system cost for every reasonable base-stock level S. We would then look at dynamic policies of the type described below. These dynamic policies offer the potential to overcome the rigidity of static approaches by adapting to system changes in real time. While this approach is computationally feasible for very small instances of our system, it becomes impractical for medium-sized instances due to the rapid growth of the state space which renders exact solutions computationally intractable, as shown in Table 1. To address these limitations, we utilize simulation techniques.

Table 1:State space size, $|R_S|$ for various values of W and S.

	$S=0$	$S=1$	$S=2$	$S=4$	$S=8$	$S=16$
$W=4$	1,815	2,310	2,805	3,795	5,775	9,735
$W=8$	81,510	94,380	107,250	132,990	184,470	287,430
$W=16$	8,580,495	9,315,966	10,051,437	11,522,379	14,464,263	20,348,031
$W=32$	1.72×10^{9}	1.79×10^{9}	1.87×10^{9}	2.03×10^{9}	2.33×10^{9}	2.95×10^{9}
$W=64$	5.23×10^{11}	5.35×10^{11}	5.47×10^{11}	5.71×10^{11}	6.18×10^{11}	7.14×10^{11}

4 SIMULATION STUDY

In this section we detail our methodology and findings of the simulation process, aiming to evaluate various ordering policies. We begin by detailing the sequence of events in the simulation. We then define the required inputs, including both problem and simulation parameters, followed by a step-by-step explanation of the simulation procedure. We present the various policies tested—static, dynamic, and more complex rules—, with one approach achieving a notable 29% cost reduction. Finally, we describe sensitivity analysis we completed to assess the robustness of these results under varying conditions, ensuring the reliability and practical applicability of the findings.

4.1 Order of Events

The simulation order of events is as follows. (1) At the start of each cycle, a system state transition type and time is determined using the rates detailed earlier. (2) The system cost is updated based on the current state and the time that will pass until the transition. (3) The state of the system is updated. If a part arrives, the inventory is increased by one and the number of parts on-order is decreased by one; if a machine is repaired, the inventory is decreased by one and the number of workstations in each state is updated; if a machine deteriorates or improves, only the number of workstations in each state is updated. (4) Based on this updated system state, the new required stock level is determined, and as such the number of parts on order is updated. Following these updates, the system proceeds to the next state transition, marking the beginning of a new cycle.

4.2 Simulation Input

Before continuing to describe our simulation study, we specify the input parameters. We differentiate between problem parameters and simulation parameters. Later we discuss the sensitivity analysis performed on these problem parameters.

Simulation parameters: As discussed in the next section, we run the simulation for N batches, all of equal duration, and this duration as well as the warm-up period duration are given in Table 2.

Problem parameters: The problem parameters, as introduced in previous sections, are given in Table 2. Recall that Λ represents the workstation transition matrix. It is derived from a data set provided by Augury and is given below:

$$\Lambda = \begin{array}{c} \\ (0,0) \\ (1,0) \\ (2,0) \\ (3,0) \\ (0,4) \\ (1,4) \\ (2,4) \\ (3,4) \\ (4,4) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (0,0) & (1,0) & (2,0) & (3,0) & (0,4) & (1,4) & (2,4) & (3,4) & (4,4) \end{array} \\ \begin{pmatrix} -1 & 0.88 & 0.08 & 0.03 & 0.01 & 0 & 0 & 0 & 0 \\ 1.5 & -2 & 0.36 & 0.1 & 0.04 & 0 & 0 & 0 & 0 \\ 1.8 & 1.4 & -4 & 0.64 & 0.16 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 6 & 0 & 0 & 0 & 0 \\ 26 & 0 & 0 & 0 & -27 & 0.88 & 0.08 & 0.03 & 0.01 \\ 0 & 26 & 0 & 0 & 1.5 & -28 & 0.36 & 0.1 & 0.04 \\ 0 & 0 & 26 & 0 & 1.8 & 1.4 & -30 & 0.64 & 0.16 \\ 0 & 0 & 0 & 26 & 0 & 0 & 0 & 32 & 6 \\ 0 & 0 & 0 & 0 & 52 & 0 & 0 & 0 & -52 \end{pmatrix} \end{array}$$

Table 2: Problem and simulation parameters.

Parameter	Value
τ	0.25 years (3 months)
W	15 workstations
M	1,000,000 $\frac{\$}{\text{WS}\cdot\text{year}}$
m	10,000 $\frac{\$}{\text{WS}\cdot\text{year}}$
h	10 $\frac{\$}{\text{unit}\cdot\text{year}}$
N	10 batches
Warm-up period	10,000 years
Batch duration	100,000 years

4.3 Simulation Steps

At each simulation step, given the current state, we generate a list of all possible events and calculate the total exponential rate to the next event, given by Equation (1). We then sample the time to the next event as an exponential random variable with this rate. Finally, we determine the next event using the roulette sampling method, a stochastic selection method where each option is chosen with a probability proportional to its relative weight (Muguruma 2014), and update the state accordingly, following which we decide whether to order more spare parts or not.

After the warm-up period, data collection begins using the batch means method. Post simulation, we calculate the average yearly cost across all batches. For batch i, the average total yearly cost is denoted x_i.

We define $\bar{X}$ as the average yearly cost across all batches. We also calculate the sample standard deviation σ_x as well as the standard error of the mean $\sigma_{\bar{X}}$. In addition, we define $\bar{X}_h$, $\bar{X}'_M$ and $\bar{X}_m$, respectively, as the average yearly inventory, major and minor costs across all batches.

4.4 Simulation Investigation

In this section we explore inventory management policies aimed at minimizing total costs in a dynamic production environment. We first analyze static base-stock policies, followed by state-dependent dynamic policies, which adapt inventory levels in real-time based on system conditions and achieve significant cost reductions compared to static policies. Finally, we introduce more complex dynamic policies that incorporate proactive adjustments while addressing practical constraints like order cancellation and more than double the cost savings achieved by simpler dynamic strategies. Through simulation, we assess the effectiveness of these approaches and their potential for improved cost efficiency and operational responsiveness.

4.4.1 Calculation of Unavoidable Cost Through Simulation

As explained earlier, there is a certain proportion of our expected annual major penalty cost $E[X_M]$, that we denote by $M_{S\to\infty}$, that will be incurred even with unlimited inventory, and it can be evaluated by using the steady-state probability vector of our workstation Markov chain model. Solving the system of equations described in Section 4.4.5. with the problem instance defined above, we find that $M_{S\to\infty} = 19.87/\text{year}$. We corroborate this result by running our simulation for a very large value of S. For example, for $S = 100{,}000$ we find an average yearly major penalty cost $\bar{X}_M$ across all batches of \$20.37 with a \$2.75 standard error. This finding is in line with the value of $M_{S\to\infty}$ computed analytically, a simple statistical test—one-sided t-test—confirmed.

We evaluate policies' effectiveness by comparing their cost after deducting this unavoidable expected yearly major penalty cost $M_{S\to\infty}$ from the average yearly major penalty cost $\bar{X}_M$. We note that due to the randomness of our simulation, when scenarios present a very low amount of downtime, we might observe a small negative average yearly avoidable major penalty cost $\bar{X}_M - M_{S\to\infty}$. We opted to display this small negative amount in the subsequent result tables for transparency. We denote by $\bar{X}'_M$ the average yearly avoidable major cost, that is $\bar{X}'_M = \bar{X}_M - M_{S\to\infty}$, and by $\bar{X}'$ the average yearly total cost excluding this expected yearly unavoidable major cost $M_{S\to\infty}$: $\bar{X}' = \bar{X}_h + \bar{X}_m + \bar{X}_M - M_{S\to\infty} = \bar{X}_h + \bar{X}_m + \bar{X}'_M$.

4.4.2 Static Policies

We start by testing static base-stock policies to determine which value of the base-stock level S would lead to the lowest average yearly total cost. Table 3 presents the results of these static policies. We note that the average yearly major and minor penalty costs are both monotonically decreasing in S while the average yearly inventory holding cost is monotonically increasing in S. Balance is achieved at the optimal base-stock level denoted by $S^* = 3$. This intermediate base-stock level strikes the best balance between minimizing penalties and managing inventory holding costs. We also note that beyond S^*, the average yearly inventory holding cost increases by h for every unit added to the base-stock level S, since it is rarely needed to repair failed machines. From now on, for this problem instance, we use a static policy with $S^* = 3$ and average yearly total cost $\bar{X}' = 33.61$ as our baseline, and we test various dynamic policies against this baseline.

Table 3: Static policy results for base-case.

S	X'	$\sigma_{x'}$	$\sigma_{\bar{X}'}$	X'_M	X_m	X_h
0	5,820.30	76.56	24.21	2,033.39	3,786.91	0.00
1	838.76	19.29	6.10	216.29	615.64	6.83
2	104.90	5.61	1.78	20.11	68.51	16.28
3	**33.61**	**2.60**	**0.82**	**1.82**	**5.57**	**26.22**
4	37.11	2.74	0.87	0.51	0.38	36.21
5	46.71	2.75	0.87	0.48	0.02	46.21
6	56.69	2.75	0.87	0.48	0.00	56.21

One advantage of having information regarding machine health is that it allows us to make decisions in real time according to the actual system state. Unlike the static policy, a dynamic policy can alternate between different order-up-to levels according to the current system state—maintaining enough inventory to avoid paying penalty costs yet not over-stocking.

4.4.3 State-Dependent Dynamic Policies

We tested several dynamic policies, in which order-up-to quantities are determined as a function of the system state r' see Kaufman (2023) for details. We introduce the notation $S^{r'}$ which denotes a state-dependent base-stock level that can deviate up or down from the optimal static policy according to the system state, using the base-stock level S^* as a starting point. That is, for the optimal static policy, $S^{r'} = S^*$ for all r'. For example, dynamic policy #1 increases the base-stock level based on the number of machines in state 3, allowing for a more responsive inventory system that adapts to changes in machine conditions, that is $S^{r'} = S^* + \lceil n_3^{r'}/2 \rceil$, where $\lceil x \rceil = \min\{n \in \mathbb{Z}: n \geq x\}$. Recall that the static policy already orders spare parts for any machine in state 4. All the dynamic policies we investigated through simulation are given in the Table 4, together with the simulation results.

Dynamic policy #1 surpasses the static approach by 12.2%. Using a one-sided t-test we verify that this result is statistically significant (with a p-value less than 0.05). Overall, the most effective dynamic policies feature adaptive mechanisms that fine-tune inventory levels in response to operational fluctuations, thereby reducing the average yearly total cost.

We note that up to now, the only dynamic policies developed and tested were those that do not cause order cancellation. For example, in Dynamic Rule #2 we have $S^{r'} = S^* + n_3^{r'}$ so when a machine in state 4 in a workstation in state (3,4) is repaired, $n_3^{r'}$ remains unchanged and so does $S^{r'}$. However, because of this constraint, we observe that even the best of these dynamic policies (#1) lead to excess inventory when machines do not fail for long periods. To design more efficient policies, we decided to open our study to ordering policies that would, without any added mechanism, allow order cancellation, and we implement a no-order cancellation mechanism, as explained next.

4.4.4 More Complex Dynamic Policies:

Building upon the insights derived from the previously discussed Dynamic Rules results, we developed a more advanced set of dynamic rules. In this section, we outline the components of these newly formulated rules and present an analysis of the simulation outcomes when they are implemented in our system.

In our new rules we use a different decision variable. Up to now, the various ordering policies determined the base-stock levels, $S^{r'}$, based on the current state. We now use the number of parts on order O as our decision variable. We also define the desired quantity of spare parts to order, denoted as $\hat{O}$, which plays a crucial role in the implementation of our no-order cancellation mechanism. We note that in certain situations, for example when a single workstation is in state (4,4) ($n_{(4,4)}^{r'} = 1$) and there are no spare parts in inventory ($I^r = 0$), one might decide to order a large quantity of spare parts to maximize the chances of receiving one quickly, only to cancel the remaining orders for spare parts once one arrives, thereby saving on inventory holding costs. However, this scenario is unrealistic, as suppliers would be unable to depend on orders. We therefore restrict ourselves to policies that do not cancel orders, by setting the new number of parts on order to be the maximum between the current number of parts on order and the desired number of parts to order $\hat{O}$. With this added mechanism, we update step (4) of the simulation order of events as follows: (4') based on the updated system state, the desired number of parts to order is updated. If this

Table 4: Dynamic rules and simulation results.

Rule	$S^{r'}$	S	$\bar{X}'$	$\sigma_{x'}$	$\sigma_{\bar{X}'}$	$\bar{X}'_M$	$\bar{X}_m$	$\bar{X}_h$	% Improv.
Dynamic rule #1	$S^{r'} = S^* + \lceil n_3^{r'}/2 \rceil$	**3**	**29.5**	**2.5**	**0.8**	**-0.2**	**1.7**	**28.0**	**12.2**
Dynamic rule #2	$S^{r'} = S^* + n_3^{r'}$	3	29.7	1.9	0.6	0.2	1.4	28.1	11.8
Dynamic rule #3	if $n_{(3,0)}^{r'} > 0 : S^{r'} = S^* - 1$	3	30.6	3.0	1.0	0.9	1.8	27.95	9.02
	else: $S^{r'} = S^* - 2$								
Dynamic rule #4	$S^{r'} = \max\{S^*, n_3^{r'} + \lceil S^*/2 \rceil\}$	3	31.4	3.2	1.0	1.0	4.0	26.4	6.6
Dynamic rule #5	$S^{r'} = S^* + \lfloor n_3^{r'}/2 \rfloor$	3	31.9	3.0	1.0	1.4	4.1	26.4	5.2
Dynamic rule #6	$S^{r'} = \max\{S^*, n_3^{r'} + \lfloor S^*/2 \rfloor\}$	3	33.1	2.4	0.8	1.6	5.3	26.2	1.5
Dynamic rule #7	$S^{r'} = \max\{S^*, n_{(3,0)}^{r'}\}$	3	33.6	2.7	0.9	1.8	5.5	26.2	0.1
Dynamic rule #8	$S^{r'} = \max\{S^*, n_3^{r'}\}$	3	33.6	2.7	0.9	1.8	5.5	26.2	0.1
Dynamic rule #9	$S^{r'} = S^* - 1 + n_3^{r'}$	3	46.6	1.3	0.8	5.0	23.4	18.1	(38.6)
Dynamic rule #10	*if* $n_{(3,0)}^{r'} > S^* + 1 : S^{r'} = S^* + 1$	3	52.1	4.1	1.3	6.4	27.8	18.0	(55.0)
	else if $n_{(3,0)}^{r'} > S^* : S^{r'} = S^*$								
	else : $S^{r'} = S^* - 1$								

desired number of parts to order is greater than the outstanding number of parts on order, then the number of parts on order is increased accordingly. Otherwise, the number of parts on order is unchanged.

We define Lack (L) as the number of spare parts needed to cover all machines in states 3 or 4, and Excess (E) as the surplus inventory beyond this requirement, that it $L = \max(n_3 + n_4 - I, 0)$ and $E = \max(0, I - n_3 - n_4)$. When $E = L = 0$, the inventory is perfectly aligned with the number of machines in states 3 and 4. When $L > 0$, additional spare parts are required and we are willing to risk added inventory costs by ordering more parts than required, reducing lead time and increasing the chance of parts arrival before failures occur. However, when $E > 0$, there is an excess of E parts beyond short-term needs and we ideally wish to order less parts. We introduce ℓ and e, adjustable parameters that give a weight to, respectively, L and E. The desired order quantity $\hat{O}$, is then given by $\hat{O} = S + \ell L - eE$. Such policies are designed to ensure that orders are appropriately adjusted to reflect shortfalls or excesses, while also ensuring that orders, once placed, do not get canceled.

We tested these policies for various values of S, e and ℓ on our simulation model and compiled the results in Table 5. We observe that the average yearly total cost is lower for (e, ℓ) policies using $S = 3$ as a basis. Table 6 summarizes results for various values of e and ℓ with the base case $S = 3$. The results demonstrate significant enhancements in managing the spare parts inventory by reducing unnecessary stock levels, enhancing response times to machine failures, and ultimately decreasing overall operational costs. The most effective configuration was identified as $S = 3$, $e = 2$ and $\ell = 3$ which balances inventory, minimizing average yearly costs and ensuring availability for necessary repairs, achieving 29.1% reduction in average yearly total cost when compared to our baseline static policy. These findings highlight the need for adaptable inventory policies to meet dynamic production demands. The simulations offer key insights into policy effectiveness, guiding more efficient inventory strategies.

Table 5: Summary of $\bar{X}'$ for various values of S, e and ℓ.

		ℓ			
S	e	1	2	3	4
2	1	46.57	33.42	27.40	26.55
2	2	77.62	46.94	35.10	31.96
3	1	29.65	29.73	29.29	29.27
3	2	28.46	25.80	**23.83**	25.06
3	3	36.17	30.67	28.49	28.47
4	1	38.24	38.25	38.08	38.09
4	2	26.51	25.99	25.48	25.54
4	3	25.62	24.90	24.93	25.54
4	4	30.99	28.12	28.73	29.35

Table 6:Top non-Markovian dynamic policies results for base-case $S = 3$.

S	e	ℓ	$\bar{X}'$	$\sigma_{x'}$	$\sigma_{\bar{X}'}$	Major penalty	Minor penalty	Inventory holding	Percentage improvement
3	1	1	29.65	1.92	0.61	0.15	1.39	28.11	11.8
3	1	2	29.73	2.2	0.7	0.71	0.85	28.17	11.5
3	1	3	29.29	1.98	0.63	0.42	0.59	28.28	12.9
3	1	4	29.27	1.77	0.56	0.38	0.46	28.43	12.9
3	2	1	28.46	2.85	0.9	1.15	6.85	20.46	15.3
3	2	2	25.8	1.47	0.47	0.77	4.16	20.87	23.2
3	**2**	**3**	**23.83**	**1.54**	**0.49**	**-0.57**	**2.91**	**21.50**	**29.1**
3	2	4	25.06	1.23	0.39	0.62	2.12	22.32	25.4
3	3	1	36 .17	3 .20	1 .011	3 .26	13 .50	19 .41	(7 .6)
3	3	2	30 .67	2 .02	0 .38	2 .47	8 .02	20 .17	8 .8
3	3	3	28.49	1.67	0.53	1.58	5.62	21.29	15.2
3	3	4	28.47	2.11	0.67	1.64	4.15	22.68	15.3

4.4.5 Sensitivity Analysis

We now apply the same methodology across different scenarios to evaluate the robustness of our methodology. We selected five scenarios to provide a balanced representation of different system characteristics, chosen to encompass a wide range of system sizes, lead times, and levels of penalty costs, ensuring a comprehensive analysis across diverse operational conditions. Unless otherwise specified, the problem parameters are as stated in Table 2. The simulation parameters are unchanged. For each scenario, we first ran our simulation under the standard base-stock policies with various values of our base-stock level S, S^*. We then ran the same dynamic policies as with our base case scenario, using S^* for each scenario, and finally tested the more complex dynamic policies to find the optimal vales for e and ℓ. We summarize the parameters of our five alternative scenarios and sensitivity analysis results in Table 7.

The results of the sensitivity analysis demonstrate that the more complex policy with $e = 2$ and $\ell = 3$ consistently outperforms both base-stock policies and the basic dynamic policies. Across all tested scenarios, the (2,3) policy exhibits significantly higher efficiency, highlighting its robustness and effectiveness in managing varying conditions compared to the simpler alternative approaches.

Table 5: Summary of alternative scenarios and results.

Scenario	τ	M	m	h	W	Transition Rates	S^*	X' Base Stock	X' Dynamic Policy	%	X' $(2,3)$ Policy	%
0: Baseline	3	1M	10K	10	15	Λ	3	33.6	29.5	12.2	23.8	29.1
1: Shrunk	2.5	0.8M	7.5K	8	12	$\Lambda - 10\%$	3	23.7	20.3	14.3	17.6	25.5
2: Inflated	3.5	1.2M	12.5K	12	18	$\Lambda + 10\%$	4	43.8	43.6	0.3	36.3	17.0
3: Larger System	2	0.9M	9K	9	20	Adjusted rates	3	27.6	26.3	4.8	20.9	24.2
4: Smaller System	4	1.5M	5K	15	10	Adjusted rates	3	44.9	42.9	4.3	32.9	26.8
5: Scalability	1	2M	2K	5	25	Adjusted rates	3	14.8	14.0	5.3	12.6	14.6

5. EXTENSION TO MULTIPLE-COMPONENT MODELING

In this section, we extend our model to handle multiple independent components, introducing a capacity constraint and adapting the cost structure accordingly, allowing us to analyze more realistic scenarios.

We extend our model to include K types of spare parts. We add superscript (k) to refer to component k when modeling multi-components systems. For example, $S^{(k)}$ is the base-stock level for component k. We introduce a storage constraint that takes into consideration a limited resource at the facility. This constraint only applies to spare parts that are not assigned to already failed machines. We denote by S_{max} the maximum number of spare parts of all types that can be stored in inventory at any time: $\sum_{k=1}^{K}\left(O^{(k)} + I^{(k)} - n_4^{(k)}\right) \le S_{max}$. The incurred yearly total system cost can be written as: $X' = \sum_{k=1}^{K}\left(X_h^{(k)} + X_M'^{(k)} + X_m^{(k)}\right)$. As in the one-part model described earlier, the expected yearly total system cost, that we look to minimize, is given by:

$$\mathbb{E}[X'] = \sum_{k=1}^{K}\left(\mathbb{E}[X_h^{(k)}] + \mathbb{E}[X'^{(k)}_M] + \mathbb{E}[X_m^{(k)}]\right), \text{ where:}$$

$$\mathbb{E}[X_h^{(k)}] = h^{(k)}\mathbb{E}[(I^{(k)} - n_4^{(k)})^+] = \sum_{r\in R_S^{(k)}} h^{(k)} p_r^{(k)} (I^{r,(k)} - n_4^{r,re(k)})^+,$$

$$\mathbb{E}[X'^{(k)}_M] = M^{(k)}\mathbb{E}[n_{(4,4)}^{(k)}] - M^{(k)}_{S^{(k)}\to\infty} = \sum_{r\in R_S^{(k)}} M^{(k)} p_r^{(k)} n_{(4,4)}^{r,(k)} - M^{(k)}_{S^{(k)}\to\infty} \text{ and}$$

$$\mathbb{E}[X_m^{(k)}] = m^{(k)}\mathbb{E}[(n_4^{(k)} - I^{(k)})^+] = \sum_{r\in R_S^{(k)}} m^{(k)} p_r^{(k)} (n_4^{r,(k)} - I^{r,(k)})^+.$$

The overall optimization model can be formulated as follows:

$$\min \mathbb{E}[X']$$

$$subject\ to \sum_{k=1}^{K}\left(O^{(k)} + I^{(k)} - n_4^{(k)}\right) \leq S_{max}$$

When the storage constraint is active, we need to ensure we first order the most critically needed spare parts. We do so by comparing the criticality level of each type of spare parts. There are various ways to represent the criticality level and for this first exploration of the K-part model we define it as the difference between the number of machines in state 4 and the level of on-hand inventory—excluding spare parts assigned to already failed machines. If we have enough inventory to fix all the machines in state 4, we set the criticality level to 0. We denote as $c^{(k)}$ the criticality level of type k spare part, that is, $c^{(k)} = \max\left(0, n_4^{(k)} - I^{(k)}\right)$. This criticality rule is a simple one and does not take into consideration the state of the machine's workstation. We use it here to verify that our methodology works in the multi-part model, and we will explore more complex criticality rules in the future.

The order of events for the K-part model is as follows. Steps (1) through (4) remain unchanged. If we can update all the number of parts on order without going over S_{max}, we do so. If not, we order in decreasing criticality level. In case of a criticality level tie between two or more spare parts, we choose randomly which part to order next. Following these updates, the system proceeds to the next state transition, marking the beginning of a new cycle.

In our simulation we focused on systems with two ($K = 2$) independent components. Our system is composed of $W^{(1)}$ workstations with machines depending on spare parts of type 1 and $W^{(2)}$ workstations with machines depending on spare parts of type 2. System parameters for spare parts of type 1 and spare parts of type 2 are set to be the same as for the one-part model described earlier. We employ the same methodology as for the one-part model: we already know that the optimal base-stock level for static policies is $S^* = 3$. As such we decide to run the simulation with an inventory constraint $S_{max} = 5$ so that the joint constraint is active even in the optimal static policy. We first run the simulation with the static policy to which we add both the constraint and criticality-based allocation mechanism. We then run the simulation with the same Markovian dynamic rules as with the one-part model, again integrated with our constraint and allocation mechanism. Finally, we test the (e, ℓ) dynamic policy with $e = 2$ and $\ell = 3$. The results of these simulations are given in Table 8. We note that, as for the one-part model, the more complex (e, ℓ) rule results in the greatest savings, with an average yearly total cost 26.40% lower than with the static policy.

Table 8: Top Policies Results for 2-Parts Model for S_{max} =5.

Policy	X'	$\sigma_{x'}$	$\sigma_{\bar{X}'}$	X'_M	X_m	X_h	Improvement %
Base-Stock	130.0	8.2	2.6	17.6	70.0	42.5	-
Dynamic Rule #4	128.4	7.5	2.4	16.6	69.3	42.5	1.26
Dynamic Rule #5	128.4	7.3	2.3	16.7	69.2	42.5	1.26
Dynamic Rule #9	112.7	4.3	1.4	16.1	61.0	35.6	13.31
Dynamic Rule #10	112.6	4.9	1.6	15.7	61.3	35.7	13.35
$(e, \ell) = (2, 3)$	95.7	6.6	2.1	11.2	46.9	37.6	26.4

CONCLUSION

This paper highlights the potential of integrating advanced inventory management with machine health monitoring for spare parts provisioning. Future research will explore more complex systems, including multiple identical or varied parts and interdependent components managed across multiple locations by a 3PL. Machine learning could further enhance predictive capabilities, enabling adaptive policies for real-time decision-making. These advancements support robust, scalable inventory frameworks applicable across diverse contexts. Addressing these challenges will lay the groundwork for innovative solutions that enhance operational efficiency and resilience in complex industrial environments.

REFERENCES

Dendauw, P., T. Goeman, D. Claeys, K. De Turck, D. Fiems, and H. Bruneel. 2021. "Condition-based Critical Level Policy for Spare Parts Inventory Management". *Computers & Industrial Engineering* 157.

Eryilmaz, S. 2017. "The Effectiveness of Adding Cold Standby Redundancy to a Coherent System at System and Component Levels". *Reliability Engineering & System Safety* 165: 331–335.

Grimmett G. and D. Stirzaker. 2020. Probability and Random Processes. *Oxford University Press*.

Hu Y., X. Miao, Y. Si, E. Pan, and E. Zio. 2022. "Prognostics and Health Management: a Review from the Perspectives of Design, Development and Decision". *Reliability Engineering & System Safety* 217:108063.

de Jonge B. and P.A. Scarf. 2020. "A Review on Maintenance Optimization". *European Journal of Operational Research* 285:805-824.

Kaufman, A. 2023. "Dynamic Spare Parts Inventory Management Utilizing Machine Health Data*"*. Technion Israel Institute of Technology.

Kritzinger W., M. Karner, G. Traar, J. Henjes, and W. Sihn. 2018. "Digital Twin in Manufacturing: A Categorical Literature Review and Classification". *IFAC-PapersOnLine* 51(11):1016-1022.

Kulshrestha N., S. Agrawal, and D. Shree. 2024. "Spare Parts Management in Industry 4.0 Era: a Literature Review". *JQME* 30,1.

Muguruma, F. 2014. "Roulette Model of Systematic Sampling". *American Journal of Applied Mathematics and Statistics* Vol. 2, No. 5, 344-351.

Olde Keizer M.C.A., R.H. Teunter, and J. Veldman. 2017. "Joint Condition-based Maintenance and Inventory Optimization for Systems with Multiple Components". *European Journal of Operational Research* 257:209-222.

Pinciroli L., P. Baraldi, and E, Zio. 2023. "Maintenance Optimization in Industry 4.0". *Reliability Engineering & System Safety* 234:109204.

Schreibfeder J. 2017. Achieving Effective Inventory Management. *Effective Inventory Management Inc*. Sixth Edition.

Si X.S., W. Wang, C.H. Hu, and D.H. Zhou. 2011. "Remaining Useful Life Estimation – A Review on Statistical Data-driven Approaches". *European Journal of Operational Research* 213:1-14.

Van der Auweraer S., R.N. Boute, and A.A. Syntetos. 2019. "Forecasting Spare Part Demand with Installed Base Information: a Review". *International Journal of Forecasting* 35:181-196.

Wang J., Q. Qiu, and H. Wang. 2021. "Joint Optimization of Condition-based and Age-based Replacement Policy and Inventory Policy for a Two-unit Series System". *Reliability Engineering & System Safety* 205:107251.

Zhang X. and J. Zeng. 2017. "Joint Optimization of Condition-based Opportunistic Maintenance and Spare Parts Provisioning Policy in Multiunit Systems". *European Journal of Operational Research* 262:479–498.

AUTHOR BIOGRAPHIES

JENNIFER KRUMAN is a PhD candidate in the Faculty of Data and Decision Science at Technion Israel Institute of Technology. With 14 years of industry experience and an educational background from Columbia University and French engineering school Ecole des Ponts, Jennifer is passionate about optimization. Her email address is jennifer.k@campus.technion.ac.il.

AVITAL KAUFMAN completed a M.Sc. in the Faculty of Data and Decision Science at Technion Israel Institute of Technology in 2023 and currently works as a logistics specialist at ZIM, in Israel. Her email address is avital.ka@technion.ac.il.

YALE T. HERER Yale T. Herer, B.S. (1986), M.S. (1990), Ph.D. (1990) is a Professor in the Faculty of Data and Decision Sciences at the Technion – Israel Institute of Technology. Yale's research interest can be broadly defined as covering Production Planning and Control. Yale has won various prizes including a 1996 IIE Transactions Best Paper Award, a 2008 IBM Faculty Award, and INFORM's 2013 Daniel H. Wagner Prize for Excellence in Operations Research Practice. His email address is yale@technion.ac.il.