Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P5064-cd

A Data-Driven Framework for Optimized System Maintenance

Rim Kaddah

Institut de Recherche Technologique SystemX, Palaiseau, France E-mail: rim.kaddah@irt-systemx.fr

Dimitra Politaki

Institut de Recherche Technologique SystemX, Palaiseau, France E-mail: dimitra.politaki@irt-systemx.fr

Guillaume Doquet

Safran Tech, Magny-les-Hameaux, France E-mail: guillaume.doquet@safrangroup.com

Sin-Seok Seo

Safran Tech, Magny-les-Hameaux, France E-mail: sin-seok.seo@safrangroup.com

System maintenance is crucial to ensure the safety, reliability, and performance of modern systems. Effective maintenance reduces downtime, prevents unexpected failures, and extends the lifespan of equipment. With continuing monitoring though the deployment of digital twins we can develop new data driven approaches that are more efficient than traditional maintenance policies. Digital twin technology provides a real-time virtual model of a physical system, enabling continuous monitoring, diagnosis, and advanced analytics. In this study, we propose a framework combining predictive and prescriptive pipelines, utilizing machine learning techniques to tackle optimization problems. Specifically, we explore methods such as Smart Predict-then-Optimize (SPO) and we compare with the traditional approaches Predict-then-Optimize (PTO) for system maintenance. Different maintenance policies, including Condition-Based Maintenance (CBM), periodic CBM, and predictive maintenance, are applied within this framework. An illustrative example of battery maintenance demonstrates the practical implementation of these methodologies. By leveraging data-driven approaches, this framework enhances decision-making and helps prevent costly disruptions in critical systems.

Keywords: smart-predict-then-optimize, artificial intelligent, digital twin, system maintenance, industry 4.0/5.0 reliability/safety,system health management.

1. Introduction

In the era of Industry 4.0, the integration of digital twins (DTs) Grieves (2003) has revolutionized the way complex systems are monitored, managed, and maintained. A digital twin is a virtual representation of a physical system, enriched by real-time data and predictive algorithms to replicate the behavior and performance of its real-world counterpart. This technology offers unprecedented opportunities for improving system maintenance strategies, particularly in reducing costs and enhancing reliability. Maintenance operations, traditionally reactive or based on fixed schedules, are now evolving toward predictive and condition-based approaches, leveraging insights from digital twins to make data-driven decisions.

One of the key challenges in system maintenance is achieving a balance between minimizing the total cost of maintenance and maximizing system reliability. Maintenance activities, such as inspections and repairs, incur significant costs, including labor, downtime, and replacement parts. However, neglecting timely maintenance can lead to catastrophic failures, reduced productivity, and safety hazards. To address this, it is essential to develop optimization strategies that leverage the predictive capabilities of digital twins to perform maintenance actions based on the predicted State of Health (SoH) and/or Remaining Useful Life (RUL) of system components. By doing so, maintenance operations can be strategically planned to ensure system reliability while optimizing costs.

In this paper, we propose a novel approach to balance cost efficiency and system reliability in condition-based maintenance using Smart-Predict-and-Optimize (SPO). SPO serves as a comprehensive framework designed to facilitate maintenance decisions under the inherent uncertainties of predicting system State of Health (SoH).

This paper is organized as follows: Section 2 reviews related work on data-driven maintenance optimization techniques. Section 3 presents the problem description. Section 4 introduces the system model and presents its components. Section 5 presents maintenance optimization schemes, focusing on the use of estimators to drive maintenance decisions. In Section 6, we present our proposed SPO-based scheme. Section 7 provides numerical analysis and results, illustrating the effectiveness of the proposed approach. Finally, Section 8 concludes the paper, summarizing the key findings and outlining potential directions for future research.

2. Related work

Data driven optimization methods have been developed in order to better account for uncertainty in data. In the domain of maintenance, most work propose fitting stochastic processes Cai et al. (2023) and solving a stochastic optimization problem.

SPO framework was first introduced in Elmachtoub and Grigas (2020) as a novel framework addressing optimization cost efficiency. The main challenge of this approach is its computational complexity. However, it has the advantage of working in contexts where the decision rely on prediction models. Authors propose decision-aware prediction through the integration of a SPO loss function. A loss function is a mathematical function that measures the discrepancy between a model's predicted output and the actual target value. Using a custom SPO loss to fit a prediction model will force it to adapt its outputs to the optimization objective function. This framework has been applied to several domains (see for example Jiang and Ji (2024); Chu et al. (2023)) but very few papers have tackled maintenance optimization like Tian et al. (2023). In these works, prediction models are designed to estimate optimization costs.

To our knowledge, our proposed study is the first to analyze the impact of SPO-like scheme when the prediction model estimates a variable that do not appear in the objective function of the optimization problem but has a direct impact on perceived costs. We also analyze the impact of the parameters chosen on the design of effective decision modules.

3. Problem description

Given noisy monitoring data of a system, we aim to evaluate the robustness of Condition-based maintenance policies in ensuring overall system safety while minimizing costs.

Condition-based MaintenanceNF EN 13306 (2018) is an umbrella term for a class of maintenance policies where decisions to repair or replace components are taken based on estimations of system performance status. These estimations are often called *health indicators*, and can come in any number (say that the system is equipped with 100 sensors, providing a indicator vector in $\mathbb{R}^{100}$) and form (continuous, binary, categorical...).

It is often practical to summarize and condense the information contained in those health indicators into a single real-valued quantity called the system *State of Health (SoH)*. Typically, the SoH is normalized so that $SoH = 1$ indicates a system in perfect condition and $SoH = 0$ corresponds to definitive failure. Thereafter, one of the objectives of devised maintenance policies is to prevent the SoH from reaching 0.

Traditional approaches typically train the SoH estimator in an offline manner, independently from the maintenance policy. The main limitation of these approaches, no matter how sophisticated the policy, is that its quality is heavily dependent on the accuracy of the SoH estimator. For instance, if the SoH estimator is biased to over-estimate SoH (meaning the system is deemed healthier than it actually is), then catastrophic failure can occur. On the other hand, if SoH is frequently underestimated, then a typical maintenance scheme will be unnecessarily costly, with superfluous repair operations being scheduled.

4. System model

In practice, the health indicators used to estimate SoH are typically costly or unpractical to monitor in a perfectly continuous fashion. Instead, one performs *inspections* at given times (say, after 2 weeks of installing a new machine, then every 2 months thereafter). These inspections carry a cost, denoted c_I representing the time, materials and human labor expended to record the health indicators and estimate SoH. Hopefully, c_I is much lower than repair costs c_R, and even more than failure cost c_F.

In this paper, we suppose inspection happens at a regular time interval so we don't have to account for c_I. Hence, we have a system that operates over discrete time steps, denoted by $t = 0, 1, 2, \ldots, T$, where T is the time horizon.

Our system is represented in Figure 1 and comprises three main components: (1) a prediction module that produces estimates of SoH noted $\widehat{SoH(t)}$ for each time step t , (2) an optimization module that uses $\widehat{SoH(t)}$ and produces an optimal policy over time period T, and (3) an evaluation module that applies produced policy over T with real $SoH(t)$ values and compute incurred cost.

We differentiate between two maintenance optimization schemes: the traditional approach called Predict-then-Optimize (PTO) and our proposed scheme based on Smart-Predict-Then-Optimize (SPO) approach and called by the same name.

5. Maintenance optimization schemes

Each of the Predict-then-Opitmize (PTO) and Smart-Predict-Then-Optimize (SPO) have two main components: prediction and optimization. In the PTO case, prediction is done first then maintenance policy is optimized based on the prediction as follows.

5.1. *Prediction*

For any regression model, whether based on polynomial fitting, traditional machine learning or deep learning, the SoH estimator is trained using a simple loss function, such as

$$\mathcal{L}(\widehat{SoH(t)}, SoH(t)) = ||\widehat{SoH(t)} - SoH(t)||_2^2,$$

where $\widehat{SoH(t)}$ denotes the estimated SoH at timestep t.

5.2. *Optimization*

Using SoH estimation over T, we study three condition-based maintenance models.

The most basic and commonplace form of condition-based maintenanceAli and Abdelhadi (2022) that we call CBM consists in defining a so-called repair threshold $\tau_R > 0$ s.t. taken action at time t is:

$$\begin{cases} \text{Repair if } SoH(t) \leq \tau_R \\ \text{Do nothing if } SoH(t) > \tau_R \end{cases}$$

Given a long-term time horizon T, the goal is then to find τ_R^* s.t. :

$$\begin{aligned} \tau_R^* &= argmin_{\tau_R \in]0,1[} \sum_{t=0}^{T} c_R(t) + c_F(t) \\ &= argmin_{\tau_R \in]0,1[} C_T(\tau_R) \end{aligned}$$

where $\forall t \in \{1..T\}$, a repair cost $c_R(t) = c_R + (t - t_F)^2$ if decision to repair is taken at timestep t ($c_R(t) = 0$ otherwise), and a failure cost $c_F(t) = c_F$ if failure occurs at timestep t, 0 otherwise. Using $(t - t_F)^2$ in $c_R(t)$ allows the optimisation to have a unique solution and forces the policy to choose the lowest value possible before failure if $c_F > c_R$. Given $\widehat{SoH(t)}$ built using historical data, this means that τ_R^* should be the lowest strictly positive value of $\widehat{SoH(t)}$. By abuse of notation, we will use $c_T(x)$ (or even $c_T(x, y)$) to express the total cost as a function of x (and y) depending on the variable in question.

Instead of simply defining τ_R, periodic condition-based maintenanceQuatrini et al. (2020) named pCBM allows joint optimization of τ_r with an *inspection interval* parameter denoted Δ_I. This decision scheme consists in $\forall t$:

$$\begin{cases} \text{Repair if } SoH(t) \leq \tau_R \text{ or } t \; mod \Delta_I = 0 \\ \text{Do nothing otherwise} \end{cases}$$

Given a long-term time horizon T, the goal is

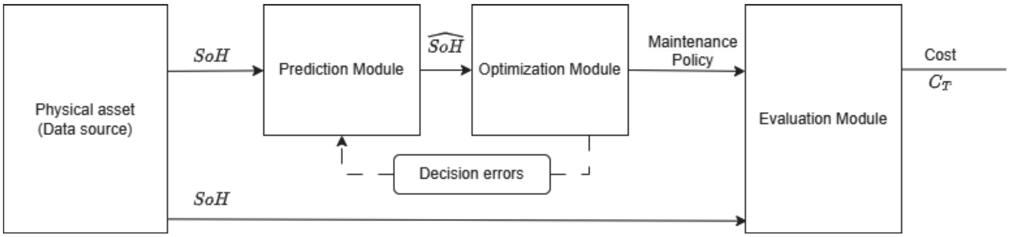


Fig. 1.: Our proposed pipeline

then to find (τ_r^*, Δ_I^*) s.t. :

$$(\tau_r^*, \Delta_I^*) = argmin_{\tau_r \in]0,1[, \Delta_I \in \mathbb{N}^*} \sum_{t=0}^{T} c_R(t) + c_F(t)$$
$$= argmin_{\tau_r \in]0,1[, \Delta_I \in \mathbb{N}^*} C_T(\tau_r, \Delta_I)$$

In other words, the goal of this scheme is to calibrate repair threshold and inspection interval together in order to minimize the long-term total maintenance cost $C_T(\tau_R, \Delta_I)$ of the system.

If Δ_I is too small, the primary consequence is to perform unnecessarily many inspection/repair operations, thereby inflating the total maintenance cost. Conversely, if τ_R is too low and/or Δ_I is too large, then the main risk is for system failure to occur in between two inspections. The first scenario carries a financial risk, whereas the second carries a safety risk, which is typically unacceptable in many critical industrial applications (e.g. aircraft engines, or public transportation). Given $\widehat{SoH(t)}$ from historical data, this means that τ_R^* should be the lowest strictly positive value of $\widehat{SoH(t)}$ and Δ_I^* the largest period before $\widehat{SoH(t)}$ reaches 0.

The condition-based maintenance can be enhanced by adding a *predictive* component responsible for estimating SoH at futur timesteps $t+1, t+2, ..., T$ so that a repair operation can be scheduled in advance, before $SoH(t) = 0$ or more commonly before a Remaining Useful Life (RUL) is exhausted. This scheme is called predictive maintenance (PdM) and consists in, given a time horizon T, finding time steps t over which a maintenance action should be scheduled. This translated in the following objective :

$$(x_t, ..., x_T) = argmin_{x_t \in \{0,1\}} \sum_{t=0}^{T} c_R(t) + c_F(t)$$
$$= argmin_{x_t \in \{0,1\}} C_T(x_t)$$

where a binary variable $x_t = 1$ if a repair action is scheduled at time t and $x_t = 0$ otherwise. In this case, SoH prediction is used to estimate the time series at hand to model the evolution of the SoH in future time steps. If $c_R < c_F$, the optimal policy will decide to repair $x_t = 1$ when $\widehat{SoH(t+1)} = 0$.

6. Our proposed SPO-based scheme

Smart Predict then Optimize (SPO), as mentioned in Section 2 is a method that refines the prediction based on the decision-making task through an iterative process between prediction and optimization.

In this work, we use this method to propose a direction for building robust maintenance policies, where SoH estimator and maintenance policy are integrated in a single pipeline.

In our adaptation of the *Smart Predict and Optimize (SPO)* scheme, we propose to learn a *policy-adjusted SoH estimator* $\widehat{SoH_P}$ s.t.

$$L(\widehat{SoH_P}(t), SoH(t)) = \lambda * ||\widehat{SoH_P}(t) - SoH(t)||_2^2$$
$$+(1-\lambda) * \mathcal{L}_{\text{SPO+}}(\widehat{SoH_P}(t), SoH(t))$$

where $\lambda \in [0, 1]$ and $\mathcal{L}_{\text{SPO+}}(\widehat{SoH_P}(t), SoH(t))$ is calculated using the SPO+ loss defined in Elmachtoub and Grigas (2020) as follows:

$$\mathcal{L}_{\text{SPO+}}(\widehat{SoH_P}(t), SoH(t)) =$$
$$max_\tau\{C_T(SoH(t), \tau) - 2\, C_T(\widehat{SoH_P}(t), \tau)\}$$
$$+2\, C_T(\widehat{SoH_P}(t), \tau*) - C_T(SoH(t), \tau*)$$

where $C_T(SoH(t), \tau)$ is the optimization cost computed for a given $SoH(t)$ data and maintenance strategy τ, and $\tau*$ is the optimal maintenance policy.

Then, the maintenance policy itself is optimized to minimize total cost C_T based on $\widehat{SoH_P}(t)$.

The policy-adjusted SoH estimator directly takes into account the incurred maintenance cost of the associated policy to make its prediction. Training of this estimator and optimization of the linked maintenance policy becomes an iterative process, alternating training/optimization steps between the two components. It is important to notice that we choose to still include the traditional loss $||\widehat{SoH_P}(t) - SoH(t)||_2^2$, weighted by a given λ against the maintenance cost objective, which is weighted by a given $1 - \lambda$. This hybrid loss corresponds to finding a compromise between SoH regression accuracy (estimating system health as precisely as possible) and maintenance cost efficiency (making the best decisions possible, *given a certain policy*). This tradeoff is crucial so that the physical meaning of this new metric is preserved : it cannot drift away too far from the actual SoH without paying a heavy penalty.

Our intuition is that the estimation of this new policy-adjusted SoH should yield almost identical results to a traditional SoH estimator when the maintenance policy is well-calibrated. However, when the maintenance policy would make a costly mistake (e.g. ask for repair when none is needed or let the system go to failure), then policy-adjusted SoH is able to strategically "lie" to the maintenance policy by temporarily drifting away from the true SoH to prevent that mistake from happening.

If τ_R is dangerously low (risk of failure), then $\widehat{SoH_P}$ will artificially be underestimated to trigger a repair decision and keep the system safe. Conversely, if using dynamic inspection intervals and Δ_I is currently too small (superfluous costly inspections), then $\widehat{SoH_P}$ can compensate by overestimating system health a bit so that the dynamic inspection interval increases Δ_I in the next iteration. $\widehat{SoH_P}$ should then stop its overestimation. We will investigate this behavior in the experimental campaign of section 7.

7. Numerical analysis

To evaluate the performance and behavior of our proposed SPO schemes compared to traditional PTO schemes, we generate datasets that simulate battery capacity degradation. We focus on synthetic data because publicly available real-world datasets are limited and often lack sufficient run-to-failure data

7.1. *Data generation scheme*

The data generation process involves simulating a population of batteries, with each battery having a typical lifetime measured in cycles. For reach battery, we draw a lifetime value from a normal distribution with an average lifetime of 700 (in our experiments) and a standard deviation of 100. The lifetime represents the duration (in cycles) a battery will last before it fails. We use a linear degradation model that will produce a time series for SoH for one battery that starts with $SoH(t) = 100\%$ for $t = 0$ and ends with $SoH(t) = 0\%$ for t equals to the battery lifetime value. This should reflect typical wear and tear. To introduce more realism into the simulation, noise is added to the data to model inspection errors. In our experiments, we suppose a noise that follows a Gaussian distribution, $\mathcal{N}(0,1)$. The noise is generated using a fixed random seed to ensure that the results are reproducible in future simulations.

For a population of 100 batteries, this produces batteries that last on average 696 cycles, with a standard deviation of 98 cycles. The minimum number of cycles observed is 367 cycles and the maximum number of cycles is 879 cycles.

7.2. *Settings*

The generated data is transformed for prediction into a set of features used by the prediction model. In this paper, to predict $SoH(t)$, we use the cycle number, $SoH(t-1)$ and $SoH(t-2)$. We then split the data into a training set and a test set with a ratio of 80% and 20% respectively.

For the prediction, we implement a Linear Regression(LR) Seber and Lee (2003) model. In the case of PTO, this model is trained using the loss function of section 5.1. For SPO, we use the SPO+ loss function of section 6 with $\lambda = 0.8$.

We use training data to produce the optimal policies of the optimization schemes described in Section 5.2: CBM, $pCBM$ and PdM.

Test data is used to evaluate the efficiency of the produced policies.

In this analysis, we consider a long enough time horizon $T = 900$ over which a single repair decision (of replacing the battery) can be made at a certain t based on the maintenance policy at hand. Given the decision, we compute the cost incurred over time horizon T based on whether the system failed before getting the chance to repair or not. To test the impact of the cost values, we define various scenarios, each characterized by a different order of magnitude difference between the cost of repair and the cost of failure. These scenarios are represented in table 1.

Table 1.: Cost structure for each scenario.

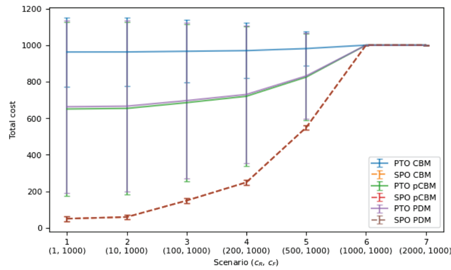
Scenario	1	2	3	4	5	6	7
C_R	1	10	100	200	500	1000	2000
C_F	1000	1000	1000	1000	1000	1000	1000

7.3. *Results*

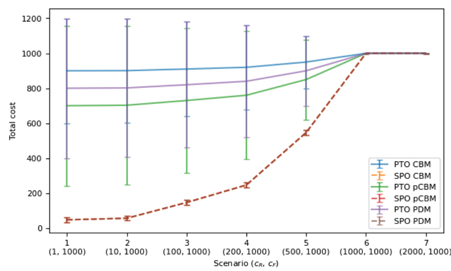
Figures 2 show for each scenario the mean and standard deviation of the total costs recorded when the pipeline(see figure 1) is applied on training data and testing.

Like expected, we can see that average total cost increases when the gap between c_R and c_F decreases. This means that if the cost of failure is not set high enough, letting the system fail becomes a possible option for optimization.

For both training and test data, the SPO schemes show lower costs compared to the PTO schemes. We also highlight that SPO schemes show very low standard deviation values and hence have more stable total costs for all datasets. Since in PTO, prediction is done offline, it makes it suitable on average but fail to deal with steeper that average SoH curves (i.e., degradation curves).



(a) Results for the training dataset



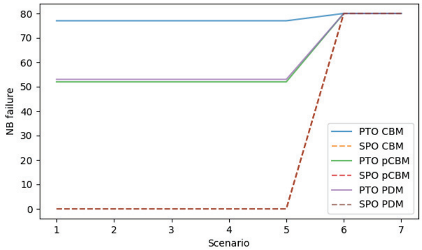
(b) Results for the test dataset

Fig. 2.: Average and standard deviation of total cost for all time series per scenario.

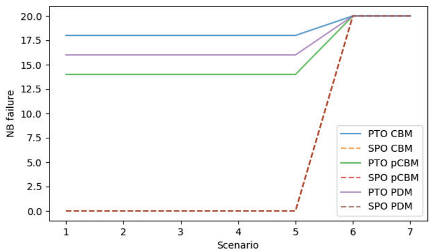
This also is shown in figures 3 where the total number of failure of PTO based schemes is higher than the ones based on SPO.

Looking at total costs and total number of failures, all SPO schemes have close values.

Table 2 presents prediction error. Since in PTO, linear regression uses MSE as a loss function, LR model for PTO is the same across all PTO schemes and all scenarios. For SPO, prediction is directly tied to the optimization scheme considered, so it is more likely that a different prediction model is produced by each optimization scheme(CBM, pCBM and PdM). We can see that SPO schemes can degrade the quality of the prediction to better serve the optimization model. In scenarios 1, 2, 3, and 4, given that c_R is much lower than c_F, the overall behavior remains similar for SPO PDM, as shown by the MSE and MAE results. In Scenario 6, the costs of repair and failure are equal, so it is more beneficial for the system to fail. In Scenario 7, both pCDM and CDM indicate that failure is cheaper than repair, making failure the more cost-effective option. From the



(a) Results for the training dataset



(b) Results for the test dataset

Fig. 3.: Total number of failure per scenario.

Table 2.: Prediction Mean Square Error(MSE) and Mean Absolute Error(MAE) for each model per scenario

Scenario	LR Model for	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
all	PTO	0.148	0.969
1	SPO CBM	4.686	5.698
1	SPO pCBM	3.765	5.117
1	**SPO PDM**	0.774	2.068
2	SPO CBM	2.106	3.848
2	SPO pCBM	2.106	3.848
2	**SPO PDM**	0.774	2.068
3	SPO CBM	36.222	15.690
3	SPO pCBM	4.308	5.282
3	**SPO PDM**	0.774	2.068
4	SPO CBM	42.733	17.170
4	SPO pCBM	15.466	10.248
4	**SPO PDM**	0.774	2.068
5	SPO CBM	21.585	12.098
5	SPO pCBM	148.108	31.702
5	**SPO PDM**	0.774	2.068
6	**SPO CBM**	0.148	0.971
6	**SPO pCBM**	0.148	0.971
6	**SPO PDM**	0.148	0.971
7	SPO CBM	1.927	3.349
7	**SPO pCBM**	1.021	2.600
7	**SPO PDM**	1.021	2.600

results in Table 2, among all SPO schemes, SPO PDM seem to be the best at keeping MSE and MAE the lowest over all scenarios. This means that SPO PDM respects the most the “physical” meaning of the SoH indicator.

8. Conclusion and perspectives

In this paper, we propose an adaptation of the Smart Predict-then-Optimize framework to tackle maintenance optimization. We compare the performance given by this approach with the traditional Predict-then-Optimize method. In our comparison, we consider three maintenance strategies : Condition-Based Maintenance, periodic Condition-Based Maintenance, and predictive maintenance. We show through numerical experiments that Smart Predict-then-Optimize improves performance by reducing total costs. This is because the scheme can adapt prediction objectives to effectively account for uncertainty, aligning them with the specific decision-making strategy employed. SPO models are hence “safer” and more robust than PTO ones.

From a methodological perspective, as a future work, our aim is to explore the behavior of SPO when other types of prediction models are involved, such as machine learning or deep learning. Different models interact with loss functions in distinct ways, influencing their ability to converge effectively. By systematically comparing multiple models, we can identify the most suitable approach that balances accuracy, efficiency, and alignment with specific problem constraints.

We can also extend our approach that deals with a static inspection period to allow and optimize a dynamic one. Inspections can then be more or less frequent depending on system health. When considering an inspection cost, this improvement aims to reduce total costs by inspecting as sparsely as possible while maintaining overall safety.

Another research perspective is to tackle multi-objective optimization. In this work, we use a single-objective formulation of the maintenance optimization problem, mixing failure cost with financial cost. In applications where safety is paramount, this should not be done and the two criteria need to remain somewhat separate as safety criterion is to be prioritized over financial cost.

Acknowledgement

This work has been supported by the French government under the ”France 2030” program, as of the SystemX Technological Research Institute.

References

Ali, A. and A. Abdelhadi (2022). Condition-based monitoring and maintenance: State of the art review. *Applied Sciences 12*, 688.

Cai, Y., R. H. Teunter, and B. de Jonge (2023). A data-driven approach for condition-based maintenance optimization. *European Journal of Operational Research 311*(2), 730–738.

Chu, H., W. Zhang, P. Bai, and Y. Chen (2023). Data-driven optimization for last-mile delivery. *Complex & Intelligent Systems 9*(3), 2271–2284.

Elmachtoub, A. N. and P. Grigas (2020). Smart predict, then optimize. *arXiv preprint*. `https://arxiv.org/abs/1710.08005`.

Grieves, M. (2003). Digital twin: Manufacturing excellence through virtual factory replication. *CIRP Annals - Manufacturing Technology 51*(1), 1–4.

Jiang, Z. and R. Ji (2024). Optimising hurricane shelter locations with smart predict-then-optimise framework. *International Journal of Production Research*.

NF EN 13306 (2018, January). Maintenance - maintenance terminology. Standard, Association Française de NORmalisation (AFNOR).

Quatrini, E., F. Costantino, G. D. Gravio, and R. Patriarca (2020). Condition-based maintenance—an extensive literature review. *Machines 8*, 31.

Seber, G. A. F. and A. J. Lee (2003). *Linear Regression Analysis*. Wiley.

Tian, X., R. Yan, Y. Liu, and S. Wang (2023). A smart predict-then-optimize method for targeted and cost-effective maritime transportation. *Transportation Research Part B: Methodological 172*, 32–52.