applied sciences

MDPI

Article

Enhancing Antifragile Performance of Manufacturing Systems through Predictive Maintenance

Oana Chenaru, Stefan Mocanu *, Radu Dobrescu and Maximilian Nicolae

Faculty of Automatic Control and Computers, University Politehnica of Bucharest, RO-060042 Bucharest, Romania

* Correspondence: stefan.mocanu@upb.ro

Abstract: Antifragility was introduced as a term no later than 10 years ago. As presented by Taleb, antifragility means that a system becomes more resilient and more robust with every harmful and/or stressful action it is confronted with. This paper is based on a study which aimed to use the concept of antifragility during the design stage of a self-improving system. This way, it is expected to obtain a fast adaptive system capable of functioning at optimal parameters even when it works under adverse conditions or faces unforeseen changes in the environment. Assuming that an antifragile system not only maintains its robust behavior when faced with stressful and harmful events but even benefits from them to optimize its performance, the paper offers a detailed description of the features that must be ensured when designing a self-improving antifragile manufacturing system. By ensuring the property of antifragility, complex manufacturing systems are much safer to exploit under uncertain conditions, which brings major benefits to the process management. Starting from consecrated solutions such as preventive maintenance (PvM) and predictive maintenance (PdM) and using techniques of artificial intelligence, we present the concept of antifragile maintenance (AfM).

Keywords: antifragile; resilience; robustness; predictive maintenance; preventive maintenance; self-improvement; self-adaptive; context awareness; uncertainties; decision-making; antifragile maintenance

Citation: Chenaru, O.; Mocanu, S.; Dobrescu, R.; Nicolae, M. Enhancing Antifragile Performance of Manufacturing Systems through Predictive Maintenance. *Appl. Sci.* **2022**, *12*, 11958. https://doi.org/10.3390/app122311958

Academic Editors: Paweł Sitek and Wilma Polini

Received: 28 September 2022
Accepted: 21 November 2022
Published: 23 November 2022

1. Introduction

Antifragility is a concept introduced and developed by Nassim Nicholas Taleb in his book "*Antifragile: Things That Gain from Disorder*" [1], not even 10 years ago. Of course, antifragility is opposite to fragility, but it is fundamentally different from the usual antonym for fragility, which is robustness. According to Taleb, antifragility is more than robustness (the capability to withstand or overcome adverse conditions and, therefore, recover from failure) and more than resilience (the capability to resist failure). By definition [1], antifragility is the feature of a system that increases in resilience or robustness as a result of the harmful effects of stressful actions: malfunctions, shocks, errors, noise, mistakes, disturbances, attacks, faults, or failures. In other words, the idea that stands behind antifragility is that some systems (either natural or artificial) can improve their performance and become better when subjected to various forms of stress.

Two years after the appearance of Taleb's book, Vincenzo De Florio published a work that somewhat systematizes the means by which an antifragile system can be designed, formulating an equation that has won its celebrity: "Antifragility = Elasticity + Resilience + Machine Learning" [2]. Machine learning can be defined as the capacity of a computer to learn from previous data, based on certain algorithms, with the goal of making predictions, classifications, or decisions.

Leaving aside the theoretical aspects, which are otherwise very rigorous, the paper proposes a scheme capable of self-optimizing system processing using a machine learning stage which succeeds to enhance the ability of the system to adjust to adverse environmental

conditions, so arguing that an antifragile system is able to learn while using flexible and resilient strategies. What is important in this approach is the idea of the self-optimization of the process in hostile environmental conditions, an idea resumed in another paper [3] that suggests the use of game theory to create a framework for modelling both the system and the environment as competitive opponents seeking to develop optimal behavioral winning strategies.

From this seminal point, the works on the design of some antifragile systems have gained expansion and offered various implementation solutions. However, most of the results refer to particular situations, especially in social and economic fields, starting from the simple idea that a substantial gain can be obtained only by assuming significant risks. This approach does not seem to be compatible with the management of industrial processes. Usually, the industry favors efficient, profitable solutions also characterized by the fact that they do not involve significant risks. Ideally, by using the antifragility approach, it would be desirable to obtain a system or process that produces a high reward, while reducing the risk exposure, but these two desires are difficult, if not impossible, to be fulfilled simultaneously.

In the aforementioned context, the problem to be solved is to avoid the negative effect of the uncertainties that can appear during the operations, especially due to environmental changes. Modern systems permanently monitor the status of a technological installation in operation and try to detect any anomalies in relation to the normal functioning state. The concept of a "digital twin" tries to address the virtual system that functions by mirroring the physical one. Anyway, for the "health" status of the system to be maintained, periodic checks are made, the worn parts are changed, and the working parameters are reconfigured to obtain the optimum performance. This operation is called preventive maintenance and ensures the robustness of the system. The limits of preventive maintenance are given by the lack of knowledge about the system's evolution, which is leading to uncertainties. However, if we were able to anticipate shortcomings, then the system would be more than robust; it would be antifragile. To become antifragile, we must prepare ourselves to face uncertainties; that is to say, something that cannot be predicted. It is known that through preventive maintenance (PvM), periodic inspections are planned with the aim of verifying and correcting deviations from the normal state and anticipating the detection of possible failures. The moments at which these interventions are performed are established using statistics based on historical data records and service intervals recommended by the manufacturers based on their design criteria. However, as the complexity of the manufacturing processes increases and, implicitly, the degree of uncertainty regarding context changes increases, preventive maintenance solutions using predefined scenarios have been gradually replaced with predictive maintenance (PdM) solutions, based on the real-time examination of assets' status, to detect behavioral anomalies that can lead to failures, but which do not have an obvious cause.

There is an example that we find convincing for associating PdM with antifragility. Formula 1 has achieved spectacular performances that are based on the setting of the motors according to the most important parameters of the environment. In each tour of the circuit, the minimum travel time is sought. However, as the evolution progresses, two variable elements emerge: weight reduction due to fuel consumption and tire wear. PvM would suggest the tires must be changed when the wear has reached magnitude X. PdM considers environmental conditions and calculates how to reconfigure the parameters as the tires wear out and calls for change only when the risk of explosion may occur. In other words, a stressor (wear) is used to maintain the optimum performance. The system is antifragile.

In their work, the authors of [4] present Europe's need to leap to Industry 6.0, and they define Industry 6.0 as "Ubiquitous, customer-driven, virtualized, antifragile manufacturing". The authors present a vision of how industry should evolve to meet various demands (environment friendly manufacturing, profitability, reduced emissions, resilience against various shocks, sustainability, and many others). The paper presents mostly a philosophical

approach, as the authors only show WHAT to expect and not HOW to do it. However, this study is a great starting point for tangible proposals.

This paper aims to prove that ensuring the property of antifragility is the safest way to exploit complex manufacturing systems under uncertain conditions and that this philosophy can be applied generally in process management, using an association of emerging technologies borrowed from the techniques of artificial intelligence (machine learning, neural networks) and process control through computer networks (cloud computing, edge computing, big data) by performing PdM. We named the concept resulting from this association antifragile maintenance (AfM), which is a simplification of De Florio's mentioned formula and written as AfM = PdM + Learning. (In this way, we thank Vincenzo who suggested this interpretation in a private discussion; details about "Learning" can be found in Sections 4.3 and 5.)

The rest of this paper is organized as follows: Section 2 debates why self-improving systems can be seen as antifragile systems; Section 3 presents predictive maintenance in self-improving systems; Section 4 covers the challenges of designing self-improving systems; Section 5 is dedicated to uncertainties inside antifragile manufacturing systems; Section 6 presents a case study in which artificial errors are introduced to simulate uncertainties; and Section 7 is dedicated to our conclusions.

2. Self-Improving Systems as Antifragile Systems

The purpose of self-improvement in autonomous systems, which can also be called self-adapted systems, is dual: on the one hand, to improve their capacities for better management of the predicted critical situations and, on the other, to deal with unforeseen situations. Conceptually, a self-adapted system has two main components: the goals of the functionality (for which the system was developed, including control procedures for normal conditions) and the management of the functionality (for achieving goals under changing conditions, using an adaptation mechanism). Most approaches only address the first component, proposing solutions for realizing key self-functionalities such as self-configuration [5], self-healing [6], self-protection [7], and self-optimization [8].

Yet, the efficiency of the management function itself is ignored, and there are possibilities of the deterioration or malfunction of the components of the management system as well as the possibility of the dynamic modification of the objectives in function according to unanticipated events.

Different to common and well-known robust control engineering, denoted as resilience engineering from this point, antifragility engineering includes learning and knowledge acquisition processes as components of the dynamic system evolution. The essential difference is that an antifragile system has, as its main objective, the adaptation to unforeseen events and, in addition, to benefit from the new working conditions due to disturbances in order to improve their performance in similar circumstances that may occur in the future. We can, therefore, consider antifragile engineering as a superior form of ecological resilience engineering, related to the idea of dynamic balance in which the systems change and evolve when disturbed by changing the state after stress. In this direction, mechanisms of adaptation, self-organization, and self-improvement are responsible for enabling systems to learn and improve on past situations and take better advantage in future ones. In accordance with this wider form and considering interconnections between the environment, systems, and their emergent properties, a more comprehensive and antireductionist approach developed under the sign of antifragility becomes necessary.

To be able to hope in the building of antifragility, the simple design of a self-improving system is not enough. We also need to ensure the integration of such systems at different scales of a complex heterarchical structure. The integration task is proving to be difficult, especially in the case of complex systems containing numerous heterogeneous components with different properties. The control solutions implemented in such cases require a long time to design and analyze, so that in the overall performance of the system are reflected all the results expected at the level of each component. This challenge is even more difficult

when more autonomous reconfiguration and adaptation is needed during systems running in extremely dynamic and variable contexts. The inability to anticipate solutions for all the contextual circumstances and the lack of expertise required to optimize the selection of solutions already validated over time leads to the need for a system capable of continuously learning how to optimize the behaviour of the component subsystems to achieve their specific objectives. We will say such a system has not only "self-improvement" capability but also antifragile characteristics. The main problem for such systems is that by fulfilling particular objectives they are often confronted with conflicts and, therefore, they need to compromise to find solutions, which makes the decision-making activity much more difficult. Therefore, we think that the automatization of predictive maintenance (APM) is one of the steps needed to meet the objectives of antifragile engineering.

3. Predictive Maintenance in Self-Improving Manufacturing Systems

Specific to APM is the fact that it allows the integration and simultaneous exercise of control and maintenance procedures, respectively. In other words, the maintenance side expressed by PdM is proactive (PaM). PaM is a particular maintenance strategy that aims to detect the causes of failures and possibly avoid them or, if this is not possible, correct them. This strategy can be called conditional maintenance (CM) because it is performed only when required by the state of the system. As such, PaM aims at the timely diagnosis of a system that can degrade. On the other hand, this form of diagnosis is not enough because it provides only a warning (an alarm) and does not provide information about the remaining life of the system. For this reason, PdM complements the proactive side with a prognostic side, so that it can accurately predict when a system is expected to fail. In addition, the high dynamics of manufacturing processes run the risk that the old behaviors already learned become outdated. Reconfiguring them based on improved input data is not usually sufficient, so these system models need to be adapted and reconfigured to maintain and even improve their predictive performance.

An important advance in capitalizing on APM technology is the association with digital twins [9]. A DT allows the real-time monitoring of machine conditions but also the anticipation of behaviour based on prediction models without invasive techniques. DT-based simulation allows us to find a virtual counterpart for each component of a manufacturing system and, therefore, we can have a clear picture of all assets throughout their entire life cycle. In this way, the DT-based approach will use both real data provided by built-in sensors and data from virtual digital models, which allow the calculation of an essential parameter of PdM, namely, remaining useful life (RUL). Thus, the global monitoring system can detect behavioral anomalies of the components that could lead to a failure, a decrease in performance, or a decrease in the quality of the finished product. Therefore, the essential role of predictive maintenance consists of recognizing a behavioral anomaly at an early stage of the manufacturing process and then in providing high quality prediction models to reach the different targets of this process. In paper [10], the authors present an original work dedicated to the development of a sensor signal-based digital twin for intelligent machine tools. The paper covers, in detail, two different systems (DTCS—digital twin construction system and DTAS—digital twin adaptation system) that are involved in the construction and adaptation of the twin. In the final section, the authors conclude that a DT is a certain benefit for the IoT; however, they also consider that any DT must gather knowledge through machine learning techniques based on historical datasets and use that knowledge while receiving real-time information from sensors.

An interesting study investigating the connection between CPSs (cyber–physical systems) and smart manufacturing is presented in [11]. As the authors declare, the paper is "a comprehensive survey and analysis of the CPS treated as a combination of the IoT and the IoS". Although the paper aims to serve as a "theoretical basis and as a comprehensive framework for emerging manufacturing integration", one of the most important conclusions that can be drawn is that the integration of various concepts related to CPSs into manufacturing is still at the beginning. From this perspective, in the very near

future, we expect to see many proposals for architectures dedicated to smart and wisdom manufacturing. Although this paper falls into the same category, we admit this raises a new challenge: the need for (some form of) standardization. This was also noticed by the authors of [12] who propose a "unified architecture for IoT systems". They investigate the combination of DTs and the IoT and conclude that it is of high importance to take into consideration three aspects: the data, model, and service when a DT is in an application.

In the aforementioned context, the achievement of PdM objectives is based on three pillars: (i) data gathering and preprocessing; (ii) the integration of maintenance knowledge in rigorous mathematical models; and (iii) maintenance planning based on RUL calculations. We insist that preprocessing should not be neglected, as it is crucial in establishing trust in the data in the process of transitioning from potential information to consistent real information. Only in this way can the condition of the machine be predicted as a result of the simulation on virtual models without stopping the operation of the machines. The great advantage of PdM is the ability to make decisions in the shortest time without reaching the preset intervals for performing the revision. Even if, at first glance, the success of a PdM program depends on the performance of the predictive algorithm, in reality, the challenge is to determine the optimal moment in which to apply the interventional decision. It should be noted that the way in which an antifragile system is designed, by increasing its resilience in difficult conditions, is fully in line with multiobjective optimization because all the essential criteria pursued by optimal control (efficiency of asset management, robust behaviour, and minimum intervention time) are complementary and nonconflicting. For modern manufacturing systems, the tendency is to achieve integrated planning (IP) that synchronizes manufacturing planning and maintenance planning with predictive maintenance capabilities.

We already mentioned in the introduction the differences between PdM and PvM. Recall that PvM ensures robustness and PdM ensures resilience, but neither of them successfully cope with uncertainties. Although they use real-time collected data for prediction, PdM algorithms allow decisions based on finding specific patterns in historical data and cannot correctly interpret sudden and unexpected changes in context. It requires an additional ability to adapt to these changes, and we see this possibility through learning techniques, thus reaching AfM. AfM respects all of the PdM characteristics but adds the ability to adapt to unexpected changes in the operating environment, as it is also context-aware. Therefore, a system that has AfM can be called an antifragile self-improving system (ASIS).

4. Key Challenges for Designing Antifragile Self-Improving Systems

The conception of an ASIS requires the development and adoption of self-adaptive features from the early stages of design in a different approach than the classic one, which starts from the desire to satisfy certain user requirements. This new approach requires permanent feedback after the partial validation of each functional component, which also implies a specialized communication protocol. Each validation must certify the capacity of the system to mitigate the impact of unforeseen incidents. The following properties are essential in the design of a suitable self-improving system.

4.1. Distributed System Architecture

Two architectural solutions are mainly considered for the development of distributed and autonomous complex systems: service-oriented architectures (SOAs) and multiagent systems (MASs) [13]. SOAs are a mature technology, which have proven their advantages in any application that integrates both distributed or separated software components, being "enabled by technologies and standards that facilitate components communication and cooperation over a network" [14]. However, the design of a service-oriented architecture is of a top-down type, which implies, on the one hand, a meticulous stage of study of as many evolutionary scenarios as possible and, on the other hand, the exclusion of "surprises", i.e., uncertain events, as much as possible. Unlike SOAs, the MAS architecture design, a specific structure of artificial intelligence, is based on a bottom-up approach in which

the collaboration of the agent-type entities located at the lowest level of the hierarchy is essential to cope with the unpredictable changes in the context. MASs also provide various elements of structural organization which can form holons, groups, teams, hierarchies, etc. Being more natural and intuitive, MASs have been the subject of numerous works that highlight the propensity for self-organization and self-improvement [15–17]. However, the use of MASs is subject to restrictions particularly due to the limited support for ensuring MAS scalability and modularity.

For this reason, we recommend as the best solution for the design of self-improving systems, the combination of the MAS and SOA styles proposed in [18] and defined as "agent-services". This solution will maintain all the valuable characteristics of the agents, including various cooperation and coordination mechanisms, goal-based planning, and flexible organization and, at the same time, will highlight the main SOA facilities for software engineering based on modularity, reusability, and interoperability. In our vision, a self-improving system should be a set of agent-service teams (ASTs), each with different specific strengths and capabilities. An AST comprises many agent-service entities (ASEs) that cooperate to reach a common target. The use of ASTs provides a way for distributed problem-solving that will lead to a faster and more strongly argued decision-making process, considering that ASEs can also benefit from collective learning procedures.

4.2. Flexible System Architecture

Especially in the case of AST-based architecture, flexibility must be an intrinsic property of a self-improving system. By providing flexibility, such a system receives several features which allow it to adapt both behaviorally and structurally. The main feature of a flexible architecture is the adjustable autonomy. Obviously, the agents must behave autonomously, but an adjustable autonomy can modify the proactivity of the ASE by either increasing it in favor of the component "agent" or decreasing it in favor of the component "service". At the same time, the flexibility allows the ASE to dynamically modify its role according to the objective pursued and then to dynamically modify its behaviour according to the role it is playing in the current activity.

4.3. Autonomous and Collaborative Learning

The learning process is essential for acquiring the necessary knowledge in the decision-making process, especially for a self-improvement system which must be able to collect its own training data and learn from it, using both autonomous and collaborative learning mechanisms. The current tendency is to use machine learning techniques, with three of them being adequate to ensure the main objectives in the learning process for self-improved systems: eliminating the incorrect knowledge from the already learned ones, promoting self-motivated learning, and ensuring the correctness and relevance of the learned knowledge. These techniques are: supervised learning (SL) which uses feedback observations to build models, unsupervised learning (UL) which allows the learning of patterns from a set of observations without any explicit feedback, and reinforcement learning (RL) which allows software agents to take action in an environment based on a procedure which maximizes the cumulative reward.

4.4. Distributed Self-Management

As we have already mentioned, autonomous self-improved systems must have the capacity to improve their management procedure in real time; in other words, they must show a capacity for self-management. More precisely, ASISs need a dedicated service management system which corresponds to a service-oriented organization. The peculiarity of an SMS for an ASIS is that it has two components: one that refers to the management of the application, ensuring its functionality, and one that refers to the self-management subsystem itself, ensuring the coherence of the relationship of the components of the managed subsystem under the conditions in which they change their dynamic goals.

4.5. Context-Aware Modelling

For an ASIS, interaction with the environment is a permanent source of information based on which context-sensitive adaptation algorithms can be developed. The best solution is to develop appropriate context-aware models that accurately represent both the current state of the operating process/system and the status of the context. As new knowledge is acquired, it must be evaluated on these context-aware models to certify to what extent their interpretation in the decision-making process ensures that the performance criteria are met and the compatibility with standards is satisfied. The most important challenge, however, remains the ability to cope with unforeseen situations, i.e., dealing with uncertainties due to contextual changes.

A very interesting and relevant study on *context awareness* is depicted in [19]. The authors present the key elements of context awareness and outline the importance of context modelling in possessing a good understanding between the system, elements, and the environment. In addition, context adaptation is proposed as a mean for adjusting an application's behavior, so an expected response can be obtained. In [20], the authors present a method for developing context-aware systems. This study is based on a practical project dedicated to international container shipping. In order to avoid false or partially correct assumptions about elements of the context, the authors present a methodology for identifying relevant and irrelevant elements of a context. Whereas the study proposes three steps for getting familiar with the context, observing elements that define the context, and determining the rules for system adaptation, it has a limited field of applicability. In paper [21], the authors address the problem of context awareness from the perspective of business process management (BPM). Although the authors identify some useful approaches to context awareness, they conclude that "most BPM methods are not context-specific—or at least they do not state in which contexts they can or should be applied".

4.6. Distributed Decision-Making

As an essential element in adapting to the sudden environmental changes sensed and filtered by a context-aware system, the decision-making process is subject to the same challenges already mentioned regarding decentralized control in ASE coordination, valid for both an ASIS as a whole and for its associated SMS. We will insist, however, that to comply with the numerous constraints of organization and functioning in real time, the decision system must have the capacity to avoid the inherent conflicts that arise from the simultaneous requests to perform the functions of self-improvement and, respectively, of self-management and to compensate for the lack of information caused by the impact of uncertainties.

5. Dealing with Uncertainties in Antifragile Manufacturing Systems

The increasing complexity of the challenges faced by production systems, in general, and manufacturing systems far exceeds the ability to discover the cause and make even provisional causal assumptions to be subject to error. Therefore, it is increasingly important to find methods of assessing solutions that can face uncertainties arising in situations of high complexity in which no single cause, but several simultaneous causes, can be identified. As a first step, we must be able to modify the approach of the working mode through adaptation obtained by learning about uncertainty. Learning becomes essential to offer the knowledge necessary for decision-making in uncertainty. In [22], the authors describe a simple but intuitive model of the repartition of knowledge in a four-quadrant representation of a knowledge-centric view of uncertainty. The first quadrant Q1, named KK (from "Known Knowns"), represents the knowledge stored in the system based on previous data and observations. By learning, this knowledge can be renewed, and, at the same time, the outdated or incorrect knowledge can be eliminated. The second quadrant Q2, named KU (from "Known Unknowns"), contains several deficiencies in knowledge due either to incomplete information or insufficient expertise. The ambiguity due to the presence of these gaps prevents a correct interpretation of the data and, as such, represents a danger

in decision-making. The purpose of learning in Q2 is to deduce from the observations the possible behavioral tendencies of the system and establish a consolidated control strategy. The third quadrant Q3, named UK (from "Unknown Knowns"), refers to knowledge validated in previous acquisitions but which was not used either because of blockages or it could not be referenced. The learning process in this quadrant aims to unlock this knowledge and, eventually, to transfer it (sharing) from Q3 to Q1. The fourth quadrant Q4, named UU (from "Unknown Unknowns"), refers to uncertainty which is unknown for the system. Learning in this quadrant is performed by evaluating the results of changes in the data acquired from previous events. It is primarily about the detection of anomalies but just as important is the detection of evolution trends of the parameters in a preferential direction (gradient). This is the location of the antifragile approach to decision-making.

We will specify that the process by which an ASIS placed in Q4 tries to solve situations of indeterminacy through training based on historical data refers only to epistemic uncertainties, which, unlike the uncertainties produced by the variability of natural processes, are not random. An epistemic uncertainty results from a lack of knowledge, either because not enough data were collected or the flow of information in the dynamics of the system is fluctuating. This means that an ASIS, which improves through the accumulation of new knowledge, can reduce or even eliminate uncertainties. Moreover, by the fact that an ASIS can identify hazards through a logical process of understanding the uncertainties in critical infrastructures, it can also assess the vulnerabilities associated with each hazard and implicitly assess the effectiveness of the risk reduction measures.

From the implications of the learning process, the knowledge model suggests that one can deliberately introduce some errors without serious repercussions, and this can allow the rapid detection of weaknesses and provide a learning basis for how to remove these errors. Failure injection provides a basis for the predictive analysis of risk and vulnerability, which will augment the intimate knowledge on the system. It is true that the priority given to indeterminism to the detriment of a meticulous investigation of the deterministic causes of errors can be questioned. Our answer is that we have a limited capacity to discover this determinism and that, for the antifragile approach, the bet on combating indeterminism is the indicated one. Of course, this approach is mainly reflected in the transposition of uncertainty based on the degree of confidence in the predictive maintenance algorithms.

One of the major challenges in PdM is validation of the prognostic methods, which require the comparison of estimated failure times with the observed failures. In fact, this is the main element that creates the difference between PdM and PvM. The performance of a forecast model can only be tested when a failure occurs. PvM policies, as they greatly reduce the number of failures observed, significantly complicate the validation of the prognostic method. Solutions for checking the status of a system at predefined fixed intervals become insufficient in the case of increasingly complex manufacturing processes. In the case of PdM, which permanently provides additional information about the level of system degradation or the occurrence of behavioral anomalies, information about the possibility of a failure is already available before the actual failure occurs. The limitations of the probabilistic approaches are mainly due to three causes: (i) model uncertainty (there is no perfect model; the accuracy of the model depends on the volume of knowledge and the level of detail); (ii) data uncertainty (the use of expert data, accuracy, and relevance in the data acquisition); and (iii) the changing context (events cannot be known with certainty because of their continuous change).

We have already mentioned in Section 3 that an ASIS has in principle nonconflicting optimization objectives and, as such, can successfully solve multiobjective optimization (MOPs) problems. On the other hand, we discussed only well-defined objectives, unaltered by uncertainties. The problem is complicated if the control techniques aim to identify optimal solutions in an uncertain environment. In MOPs, uncertainties are due either to disturbances in the variables involved in the decision-making processes or unpredictable changes in the environmental parameters. In this context, we want to develop methods that determine the most robust solutions, i.e., the least sensitive to disturbances. However,

in this situation, we reach again the situation in which we have to accept a compromise between achieving optimal performance and robustness, respectively. An interesting solution, which we decided to adopt, is to combine a PdM procedure with the method presented in [23], whose objective is to minimize the weighted amount of quality robustness (QR) and robustness of the solution, which in our paper we call functional robustness (FR). The antifragile MOP solution we discuss below starts from the design of a model that integrates a PdM procedure with the dynamic phases of the production plan and aims for the optimization of two objectives—maximizing the QR and FR parameters.

The main instrument that ensures balance in the process of simultaneous optimization of the production scheduling and maintenance policy is a time buffer that must be inserted to deal with uncertainties that may cause damages or failures, which, from a mathematical point of view, represents a stochastic optimization procedure with discrete and continuous variables. In summary, the steps of the optimization procedure in the simple case of a single machine system are:

1. Establishing a set of tasks (jobs) with a prefixed processing time, with all jobs available at the initial moment.
2. Establishing, based on the exploitation indicators processed in the previous stages, the law of distribution of the probability of failure.
3. Establishing the weighting coefficient which reflects the increase in the probability of defect due to the increase in the operating life of the machine (aging).
4. Establishing the time of the production horizon in which we must carry out the maintenance procedure, after which the machine is restored to the same operating conditions that it had at the initial time, i.e., it becomes a machine "like a new one". (We must remember that during the execution of the maintenance, work jobs cannot be performed on the machine.)
5. Setting the minimization objective, namely, the weighted amount of QR and FR, which is determined by the sequence of jobs, the maintenance interval, and the buffer time in the program.
6. Solving the minimization problem which ensures a solution S_1 for QR and a solution S_2 for FR. A surrogate measure for S_1 with the value V_1 is selected to minimize the total deviation of the recalculated time of the current program from that of the initial program, and, respectively, a surrogate measure for S_2 with the value V_2 is selected in order to serve to minimize the total completion time. The final goal of the optimization problem is to obtain the solution that minimizes the function $f = w_1V_1 + w_2V_2$, with $w_1 + w_2 = 1$, where w_1 and w_2 are weights assigned according to priorities given to each individual objective. This is a classic problem of the weighted sum method for multi-objective problems [24] reduced to two objectives This method allows us to systematically change weights, and each objective optimization determines a different optimal solution.

6. Method for Injection of Artificial Errors in a Virtual Environment

Increasingly sophisticated methodologies are available nowadays to determine the causes of the failures of flexible manufacturing lines. Of these, the most widely used seems to be the failure mode and effects analysis (FMEA), most often associated with criticality analysis (CA). The use of FMEA allows both the identification of failures and the actions required to remove them, usually based on rules of priority in the execution of corrections. The testing and validation of the method was carried out on the testbed called SMART Flexible Assembly System within the research project CIDSACTEH (http://cidsacteh.upb.ro accessed on 1 September 2022). The main objective of the project is the use of advanced modelling and simulation technologies for the performance assessment of manufacturing mechatronic lines. The logistic support for performing the tests is a laboratory model for a flexible assembly line of industrial products with five workstations (Stations 1 to 5), as presented in Figure 1.

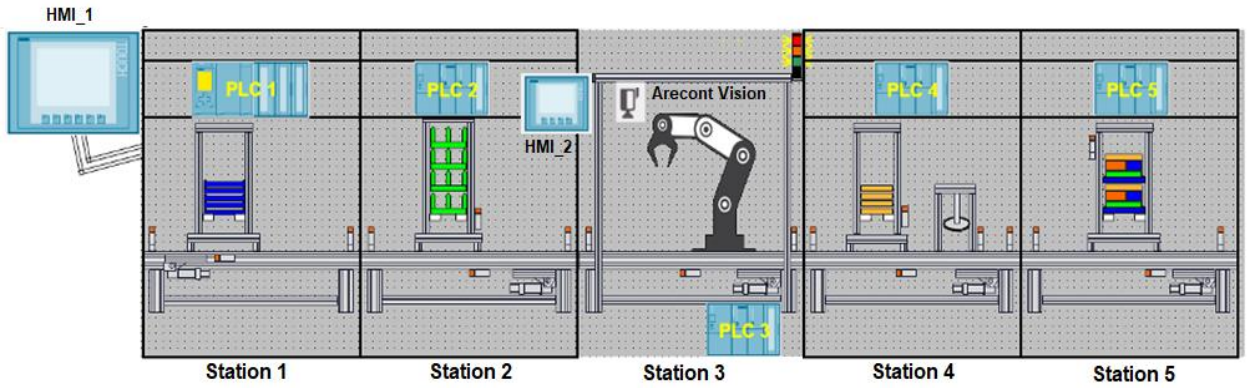

Figure 1. Block diagram of the mechatronic assembly line.

The technological flow consists of successive sequences of movement of a support on which the component elements of a product are mounted. The actual assembly is performed using a robotic arm in the central workstation (No. 3).

Our method proposes the use of a virtual environment for injecting and evaluating the impact of artificial errors on the system's antifragility. This method uses FSM (finite state machine) representation of the process correlated with FMEA (failure mode and effect analysis) to dynamically quantify this impact.

The method involves two steps: (1) building a behaviour model in the virtual environment, and (2) an analysis of the model under artificial error injection.

Process modelling (Figure 2) is a DT representation of the system, with all of the phases, nodes, and dependencies. This can be performed through an FSM where different diagrams capture different detail levels of the plant as well as dependencies between different nodes, allowing a nested top–bottom approach. Each node represents either a phase, a branch, or a final element. A risk factor is assigned to each element, taking into consideration the severity of the possible failures, the occurrence, and detectability probability. The risk factor of a phase or branch consists in the sum of all of the included elements. According to this data, a criticality matrix is built to represent element failure along with the occurrence and severity.

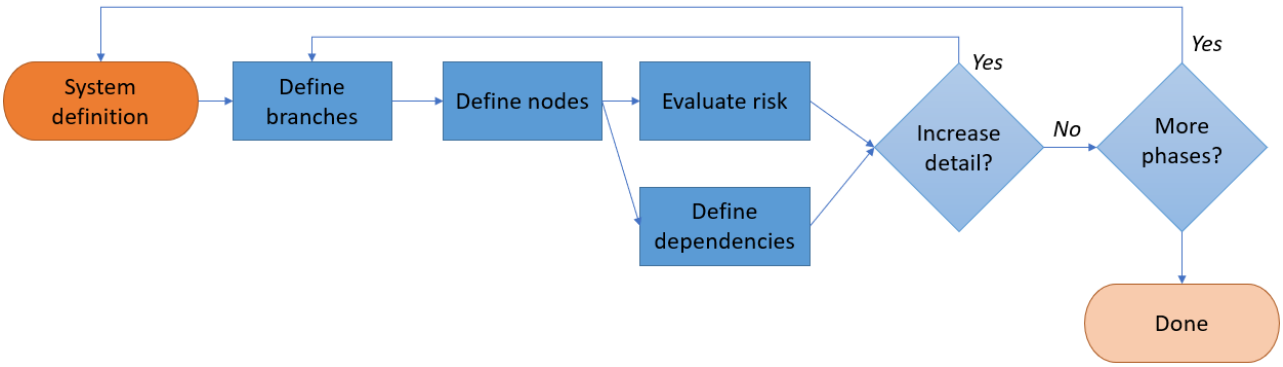

Figure 2. Building the behaviour model.

For each station of the manufacturing line, an FSM representation was built. For example, Station 1 (Figure 3) was modelled using 10 states and 12 transitions. The states are represented by: *nr_pf*, the number of products and *nr_pi*, the number of components for each product, given by the PLC; two inductive sensors, SP1 and SP2, showing the product entered or exited the conveyor belt; one RFID sensor, RFID1, to identify the stop position; one optical sensor, SO1, to check the pallet availability in the rack; two feedback sensors, SF1 and SF2, which confirm the element is in the correct position and can be released from the stack; a capacitive sensor, SC, for confirming the element reached the belt; and

B2_free, a parameter confirming the next station accepts the new elements. The transitions represented by elements P1 to P12 check the cumulative conditions required for each step for the element to be correctly processed until it leaves the station.

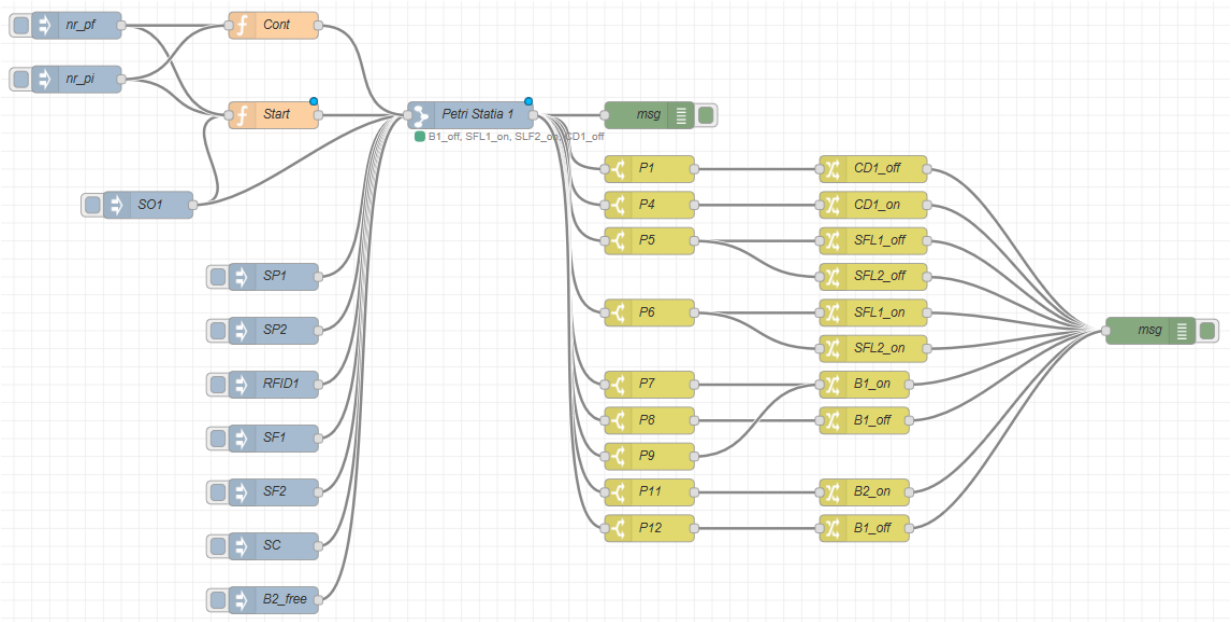

Figure 3. Behaviour model example for Station 1.

The FMEA analysis was applied on Station 1, considering for each element possible the failure modes, and causes and effects, and assigning a risk ranking with a factor between 1 and 10 for the severity of the event, the probability of occurrence, and the ease of detecting it (measured by detectability). By multiplying all three indices, we obtained a risk factor which varied from 1 to 1000. The risk value was denoted by an RPN (risk priority number), as one can see in Table 1. This step was performed offline, with the identification of the failure modes and assignment of the associated risk ranking determined according to a specialist's experience and recommendation.

Starting from the FMEA analysis in our method, we considered, at the beginning, the same occurrence index for all of the elements with the value of 1, thus making the initial risk of the operation lower, corresponding to a proper operation (as shown in Table 1). The risk index for an individual element was computed as the maximum value between all of the RPNs associated with that element. By overlapping the risk indices over the elements represented in the FSM diagram, we can perform real-time computation of the overall risk factor as the sum of all of the possible risks of all of the linked elements, according to the state of each element. In a manufacturing line where reconfiguration is possible, these values should be computed for all possible links and set for each element through the normal behaviour of the manufacturing line.

The method for introducing artificial errors involves investigating how a failure can affect the overall system operation and if it can be overcome by selecting an alternative path. It allows fast identification and classification of system weaknesses, providing a tool not only for increased plant reliability but also for better action planning. For this, we considered in the analysis both the functional dependencies between the system elements, according to the existing process models, as well as the propagation impact, according to the real-time data. Thus, depending on the error, multiple branches may be affected. The method illustrated in Figure 4 considers evaluating multiple available paths and computing as a severity factor. If the impact of this artificial error is major, alternative paths are searched in the DT model.

Table 1. FMEA analysis of station components (an extract), considering minimum occurrence risk.

Critical Component	Failure Mode	Failure Cause	Failure Effects	Initial Risk Ranking			
				Severity (S)	Occurrence (O)	Detectability (D)	Risk Priority Number (RPN)
1. Programmable logical controller (PLC)	Shut down	Instant power line failures	Line stop	8	1	4	32
2. Programmable logical controller (PLC)	Operation error	Delayed maintenance	Line stop	7	1	9	63
3. Programmable logical controller (PLC)	Communication error	Improper connection	Line stop	7	1	8	56
4. TP1500 Comfort HMI trainer	Shut down	Instant power line failures	Under optimal operation	4	1	4	16
5. Conveyor belt with asynchronous AC motor	Motor failure	Broken rotor bars Stator faults	Line stop	8	1	4	32
6. Conveyor belt with asynchronous AC motor	Shut down	Instant power line failures	Line stop	8	1	4	32
7. Conveyor belt with asynchronous AC motor and IO	Excessive vibration	Unbalance Misalignment	Improper assembly	5	1	6	30
8. SINAMICS G120—1AC 230 V	Shut down	Instant power line failures	Line stop	8	1	4	32
9. SINAMICS G120—1AC 230 V	High fluctuation in output power	Fluctuation of power supply Partial failure	Under optimal operation	6	1	8	48
10. Compact pneumatic pallet storage unit	Actuator moving abnormally slow	Air leak or squeezed tube	Low productionImproper alignment	3	1	4	12
11. RFID—SIEMENS RF300	Unidentified parts	Broken module	Production stop	4	1	9	36

We applied the method to artificial failure injection on our manufacturing line by simulating a high fluctuation in the output of the frequency converter. This was acknowledged by setting an occurrence index for this event in Table 1 to the maximum level of 10 and recomputing the risk priority number of each element of the normal path as well as for alternative paths, if available. We did not change the occurrence index between elements where there was a dependency relationship such as an event affecting only the task completion. For elements which were in a propagation relationship with the simulated faulty element, we used a medium occurrence value determined in the offline FMEA analysis. These values and their corresponding risk factors can be recomputed during the operation according to the fault events and required maintenance procedures. As detailed in Figure 4, an alternative path can be selected, considering this event will have a lower impact on those elements. This can be achieved either if the artificial error was not applied to an element which is part of the new path or if the propagation impact is lower. Following the same approach, the method can be extended to support multiple error injections.

As one can observe in Table 2, the risk index for the PLC, HMI, storage unit, and RFID sensor is represented by the maximum value between all of the possible events associated with each of these elements, as identified in Table 1. A failure of the frequency converter at Station 1 will change the occurrence value for this element to 10, thus making the new risk factor 480. As the cause can be represented by a fluctuation in the power line, this

would affect all frequency convertors and conveyor belt motors but with a lower risk index. This risk index is computed based on the offline values for the occurrence factor, as identified in the FMEA analysis. For example, for the conveyor belt, the occurrence factor for motor failure under the event of motor failure will be set to 4, so the new risk index will become 128. The index factor for FREQ2 is computed also considering the initial occurrence factor of 5, thus resulting in a risk index of 240. By determining the new risk indexes (higher than previous in the case of component #3, #4, #5, and #6), we consider the learning phase produces visible results. In addition, by determining the different factors for the new indexes, the system reacts differently to different stressors and provides data for the decision-making process. Although research to validate antifragile design control solutions is in its infancy, the facilities offered by the DT-based CIDSACTEH testbed are promising and encouraging. To date, the performance of adaptive control procedures and physical controls has been assessed using HIL (hardware in the loop) and SIL (software in the loop) techniques—see details in [25]. Now, the research aims to increase this performance by addressing antifragile responses to the effect of uncertainties due to changes in the environment.

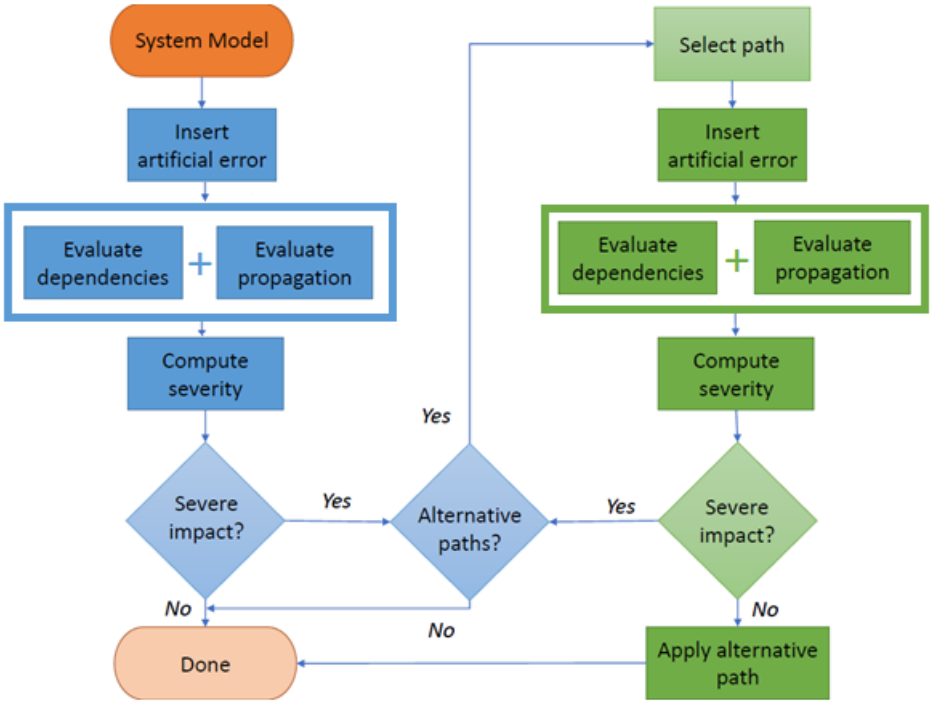

Figure 4. Method for artificial failure injection and analysis.

Table 2. Risk index of station components (an extract), considering artificial failure injection.

Critical Component	Identification	Initial Risk Index	New Risk Index
1. Programmable logical controller (PLC)	PLC	63	63
2. TP1500 Comfort HMI trainer	HMI	16	16
3. Conveyor belt with asynchronous AC motor (Station 1)	B1	32	128
4. Conveyor belt with asynchronous AC motor (Station 2)	B2	32	128
5. SINAMICS G120—1AC 230 V (Station 1)	FREQ1	48	480
6. SINAMICS G120—1AC 230 V (Station 2)	FREQ2	48	240
7. Compact pneumatic pallet storage unit (Station 1)	SO1	12	12
8. RFID—SIEMENS RF300 (Station 1)	RFID1	36	36

7. Conclusions

Finally, we mention a phrase rather jokingly said by Taleb: " ... is better to be dumb and antifragile than smart and fragile". It is obvious that it is even better, i.e., optimal to be "smart and antifragile". In this work, we tried to present the possibility of designing such systems, which we called antifragile self-improving systems (ASISs). We have shown that ASISs are a special category of complex adaptive systems that permanently maintain performance at optimum parameters, extending the operating time as much as possible by assuming calculated risk forms. This is recommended because traditional risk management is not able to predict and reduce incidents that lead to malfunction and stopping processes.

From a hardware architecture perspective, an ASIS should be a MAS, with many collaborative agent-services entities (ASEs) that cooperate to reach a common goal in agent-service teams (ASTs). The main benefit of an AST is that it enables collective learning, leading to faster problem-solving and better decision-making. From a software architecture perspective, an ASIS must be a scalable system consisting of separate and isolated processes running on multiple servers and communicating through an external computer network. As both malicious and benevolent processes that take too long to activate are difficult to detect and repair, these processes are designed to stop as soon as an error occurs, and, after isolation, their functionality should be replaced immediately to mitigate the effects of unwanted incidents and avoid the spread of errors that cause failure.

Although managerial concepts were mentioned in the first five sections in this paper in relation to predictive maintenance, we aimed to focus on the automatization of predictive maintenance (APM) which is as a needed step to meet the objectives of antifragile engineering. In addition, APM should be seen as a means to meet and overcome the limits of robust/resilient design. Thus, we consider that antifragile engineering is an improved form of robust and resilient engineering, and we see this to be more of a technical than managerial concept. The first five sections approached management only to prepare the automatization technical solutions. Due to the limited functionalities available at the manufacturing line we had at our disposal, we could only implement and test the antifragile response obtained by learning to stress generated by the artificial error injection method. The seemingly bizarre solution to achieving antifragility consisted of experimental injections of artificial defects into a manufacturing system (it is advisable to simulate this in a virtual environment) to detect and eliminate hidden vulnerabilities and ensure the isolation of the wrong behaviour processes. The introduction of artificial defects, thus, appears as positive feedback with a catalytic character to accelerate self-improvement actions. In fact, artificial error injection should be seen only as a learning basis and a means to analyze the response of the system to stressors.

The present results are represented by the scenario and the preparation of antifragile responses to stressors. We consider this to be an incipient stage in which antifragility can be obtained through "learning" based on artificial error injection. We tried to implement a system capable of working well under uncertain conditions, based on learning from previous experiences. In our vision, the evolution of the indexes presented in Table 2 is an incipient form of antifragility as the system became able to show different responses to the effects of environment uncertainties. For the future development of the study, the challenge will be to confirm the antifragile response when more stressors are added.

Although ASISs become stronger through over-adaptation for stressors, they are resilient only up to a certain limit. If this limit is exceeded, the systems may be severely damaged or even collapse. Therefore, the positive feedback loops with beneficial effect must be controlled by stronger negative feedback loops, which maintain the stability of the system as a whole.

As we said, in practice, you cannot be dumb and antifragile. ASISs are smart and antifragile.

Author Contributions: Conceptualization, R.D. and O.C.; methodology, O.C., S.M., R.D. and M.N.; software, O.C. and M.N.; validation, O.C., M.N. and S.M.; formal analysis, R.D.; investigation, R.D., M.N. and S.M.; resources, R.D.; data curation, O.C, M.N. and S.M.; writing—original draft preparation O.C., R.D. and S.M.; writing—review and editing, R.D. and S.M.; visualization, O.C. and M.N.; supervision, R.D.; project administration, R.D.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Romanian Ministry of Education and Research under grant 78PCCDI/2018-CIDSACTEH.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Taleb, N.N. *Antifragile: Things That Gain from Disorder*; Random House: New York, NY, USA, 2012; ISBN 9781400067824.
2. De Florio, V. Antifragility = Elasticity + Resilience + Machine Learning Models and Algorithms for Open System Fidelity. *Procedia Comput. Sci.* **2014**, *32*, 834–841. [CrossRef]
3. De Florio, V. On resilient behaviors in computational systems and environments. *J. Reliab. Intell. Environ.* **2015**, *1*, 33–46. [CrossRef]
4. Annanperä, E.; Jurmu, M.; Kaivo-oja, J.; Kettunen, P.; Knudsen, M.; Lauraéus, T.; Majava, J.; Porras, J. From Industry X To Industry 6.0: Antifragile Manufacturing For People, Planet, And Profit With Passion. In *White Paper No. 5/2021*; Allied ICT Finland (AIF), 2021; Available online: https://www.alliedict.fi/downloads/ (accessed on 1 September 2022).
5. Shi, J.; Sha, M. Parameter Self-Configuration and Self-Adaptation in Industrial Wireless Sensor-Actuator Networks. In Proceedings of the IEEE Infocom 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 658–666. [CrossRef]
6. Tao, M.; Shaukat, A.; Tao, Y. Modeling foundations for executable model-based testing of self-healing cyber-physical systems. *Softw. Syst. Model.* **2019**, *18*, 2843–2873.
7. Elgendi, I.; Hossain, F.; Jamalipour, A.; Munasinghe, K. Protecting Cyber Physical Systems Using a Learned MAPE-K Model. *IEEE Access* **2019**, *7*, 90954–90963. [CrossRef]
8. Klemets, J.R.; Hovd, M. Accounting for dynamics in self-optimizing control. *J. Process Control.* **2019**, *76*, 15–26. [CrossRef]
9. Grieves, M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication. In *White Paper 1411.0*; Florida Institute of Technology: Melbourne, FL, USA, 2017; pp. 1–7.
10. Ghosh, A.K.; Ullah, A.M.M.S.; Teti, R.; Kubo, A. Developing sensor signal-based digital twins for intelligent machine tools. *J. Ind. Inf. Integr.* **2021**, *24*, 100242. [CrossRef]
11. Yao, X.; Zhou, J.; Lin, Y.; Li, Y.; Yu, H.; Liu, Y. Smart manufacturing based on cyber-physical systems and beyond. *J. Intell. Manuf.* **2019**, *30*, 2805–2817. [CrossRef]
12. Jiang, Z.; Guo, Y.; Wang, Z. Digital twin to improve the virtual-real integration of industrial IoT. *J. Ind. Inf. Integr.* **2021**, *22*, 100196. [CrossRef]
13. Passos, D.; Coelho, H.; Sarti, F. From Resilience to the Design of Antifragility. In Proceedings of the Eighth International Conference on Performance, Safety and Robustness in Complex Systems and Applications, Athens, Greece, 22–26 April 2018; pp. 7–11.
14. Feygenson, O.; Feygenson, N. Modern TRIZ and the concept of antifragility. Friends, enemies or frenemies? In Proceedings of the MATRIZ TRIZfest 2018 International Conference, Lisbon, Portugal, 13–15 September 2017.
15. Marshalla, C.J.; Roberts, B.; Grenn, M. Adaptive and automated reasoning for autonomous system resilience in uncertain worlds. In *Disciplinary Convergence in Systems Engineering Research*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 799–812.
16. Bellman, K.; Gruhl, C.; Landauer, C.; Tomforde, S. Self-Improving System Integration—On a Definition and Characteristics of the Challenge, In Proceedings of the 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W), Umea, Sweden, 16–20 June 2019.
17. Mao, X.; Dong, M.; Zhu, H. Towards multiple-layer self-adaptations of multi-agent organizations using reinforcement learning. In *Novel Design and Applications of Robotics Technologies*; IGI Global: Hershey, PA, USA, 2019; pp. 66–95.
18. Baruwal Chhetri, M.; Uzunov, A.; Nepal, S.; Kowalczyk, R. Self-Improving Autonomic Systems for Antifragile Cyber Defence: Challenges and Opportunities. In Proceedings of the 2019 IEEE International Conference on Autonomic Computing (ICAC), Umeå, Sweden, 16–20 June 2019; pp. 18–23.
19. Temdee, P.; Prasad, R. Elements of Context Awareness. In *Context-Aware Communication and Computing: Applications for Smart Environment*; Springer Series in Wireless Technology; Springer: Cham, Switzerland, 2018. [CrossRef]

20. van Engelenburg, S.; Janssen, M.; Klievink, B. Designing context-aware systems: A method for understanding and analysing context in practice. *J. Log. Algebraic Methods Program.* **2019**, *103*, 79–104. [CrossRef]
21. Brocke, J.V.; Baier, M.; Schmiedel, T.; Stelzl, K.; Röglinger, M.; Wehking, C. Context-Aware Business Process Management—Method Assessment and Selection. *Bus. Inf. Syst. Eng.* **2021**, *63*, 533–550. [CrossRef]
22. Cleden, D. *Managing Project Uncertainty*; Routledge: New York, NY, USA, 2017.
23. Cui, W. Approximate Approach to Deal with the Uncertainty in Integrated Production Scheduling and Maintenance Planning. *J. Shanghai Jiaotong Univ. (Sci.)* **2020**, *25*, 106–117. [CrossRef]
24. Kim, I.Y.; de Weck, O.L. Adaptive weighted sum method for multi-objective optimization: A new method for Pareto front generation. *Struct. Multidiscip. Optim.* **2006**, *31*, 105–116. [CrossRef]
25. Dobrescu, R.; Chenaru, O.; Florea, G.; Geampalia, G.; Mocanu, S. Test Methodology for Hardware-in-Loop Assessment of Control Architectures. In Proceedings of the 24th International Conference on System Theory, Control and Computing, Sinaia, Romania, 8–10 October 2020. (*in press*).