



HAL
open science

Post-Prognostics Decision Making in Distributed MEMS-Based Systems

Haithem Skima, Christophe Varnier, Eugen Dedu, Kamal Medjaher, Julien
Bourgeois

► **To cite this version:**

Haithem Skima, Christophe Varnier, Eugen Dedu, Kamal Medjaher, Julien Bourgeois. Post-Prognostics Decision Making in Distributed MEMS-Based Systems. *Journal of Intelligent Manufacturing*, 2019, 30 (3), pp.1125 - 1136. hal-02182819

HAL Id: hal-02182819

<https://hal.science/hal-02182819v1>

Submitted on 13 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Post-Prognostics Decision Making in Distributed MEMS-Based Systems

Haithem Skima · Christophe Varnier · Eugen Dedu · Kamal Medjaher · Julien Bourgeois

Received: date / Accepted: date

Abstract In this paper, the problem of using prognostics information of Micro-Electro-Mechanical Systems (MEMS) for post-prognostics decision in distributed MEMS-based systems is addressed. A strategy of post-prognostics decision is proposed and then implemented in a distributed MEMS-based conveying surface. The surface is designed to convey fragile and tiny micro-objects. The purpose is to use the prognostics results of the used MEMS in the form of Remaining Useful Life (RUL) to maintain as long as possible a good performance of the conveying surface. For that, a distributed algorithm for distributed decision making in dynamic conditions is proposed. In addition, a simulator to simulate the decision in the targeted system is developed. Simulation results show the importance of the post-prognostics decision to optimize the utilization of the system and improve its performance.

Keywords Prognostics and Health Management · Micro-Electro-Mechanical Systems · post-prognostics decision · distributed systems

1 Introduction

A failure in an engineering system results not only in a loss of timely services and productivity to the costumers, but also in safety and environmental problems, for example, aircraft crash due to engine failure, rail

accident due to bearing failure, etc. This risk emphasizes the need of maintaining engineering systems before a failure could happen. For that reason, maintenance strategies have progressed rapidly and shifted from unplanned breakdown maintenance to preventive maintenance, then to Condition-Based Maintenance (CBM) and recently to Predictive Maintenance (PM).

Unlike the traditional maintenance strategies (breakdown and preventive maintenance), for which interventions are performed *after* the occurrence of the failure or regardless of the system status, the CBM is based on the current health state of the system for deciding maintenance interventions [29]. In the case of PM, the current health state is projected into the future to predict future maintenance actions [29]. These two smart policies aim to improve the reliability, the availability and the security of the system while reducing its maintenance costs. Therefore, prognostics becomes a necessary step to anticipate and predict the time to the failure of a degrading equipment. The implementation of these maintenance strategies requires a scientific approach involving different tasks such as condition monitoring, modeling, analysis and decision making. These tasks can be performed within the Prognostics and Health Management (PHM) framework [30].

PHM is the combination of seven modules that collectively enable to link failure mechanisms with life management (Fig. 1). It is a discipline that deals with the study of a system failure mechanisms to better manage its health. Among the PHM modules, prognostics have attracted significant research interest due to the need of models for accurate prediction for different applications [30]. It is defined by the PHM community as the estimation of the Remaining Useful Life (RUL) of physical systems based on their health state and their future operating conditions. The RUL estimation can

H. Skima · C. Varnier · E. Dedu · J. Bourgeois
FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM
25000 Besançon, France.
E-mail: firstname.lastname@femto-st.fr

K. Medjaher
Production Engineering Laboratory (LGP), INP-ENIT
65000 Tarbes, France.
E-mail: kamal.medjaher@enit.fr

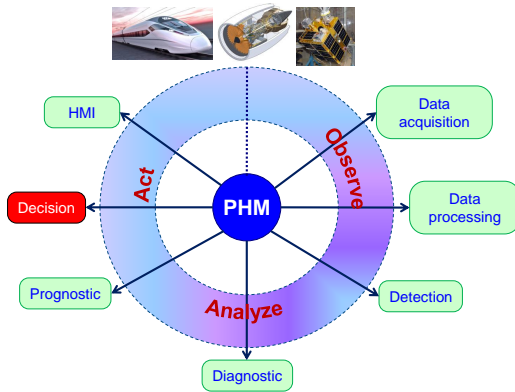


Fig. 1 Prognostics and Health Management cycle [26].

be done by using three main approaches [4]: data-driven prognostics [39], model-based prognostics [20] and hybrid prognostics [9].

Although the benefits of the PHM are related to the decision module, research on how to use the prognostics results for decision is in its early stages. The purpose of the decision module of the PHM is to determine appropriate maintenance actions and to choose an appropriate system configuration in response to prognostics predictions [1, 2, 12, 21].

In this paper, we focus on using already obtained prognostics results to perform decision making. Our targeted system consists in a distributed MEMS-based conveyor designed to convey fragile and tiny micro-objects. It is a surface composed of an array of blocks that communicate together to fulfill a common mission, which is the transport of objects. It is important to notice that there is no work in the literature dealing with post-prognostics decision in distributed systems. In the targeted conveyor, the critical component that needs to be monitored and to anticipate its failures by calculating its RUL is the MEMS. To do so, in a previous work [36], the data acquisition, data processing and prognostics (RUL estimation) steps of the PHM cycle were applied to the MEMS used in the conveyor. The objective in this paper is to exploit the obtained prognostics results to ensure continuity of operation of our distributed conveyor, optimize its usage and increase its lifetime.

The paper is structured in six sections. Next section presents a brief literature review related to post-prognostics decision and the proposed strategy for distributed systems. The targeted distributed system and the decision algorithm are presented in Section 3. The developed simulator and the simulation results are presented respectively in Section 4 and 5. Finally, conclusions are drawn in Section 6.

2 Post-prognostics decision

In the PHM context, the decision making consists in exploiting the results of the prognostics step to determine the appropriate actions, such as maintenance interventions, mission reconfiguration, etc. The objective of this type of decision is to optimally use the available information to minimize costs and avoid failures.

Several research works dealing with the post prognostics decision were published. These works concern various applications, for example aerospace [2, 11], wind turbines [19, 38], batteries [32], and electronic systems [3, 34]. The three main decisions used in these applications are: 1) the maintenance optimization, 2) the control and 3) the missions reconfiguration, presented in the following.

2.1 Maintenance optimization

Most of the research works related to the decision part of the PHM focus on the maintenance optimization. This type of decision consists in using prognostics results, the current and future health state of components and the health state of the system for maintenance planning [11]. The main aim is to alert the user in time to plan optimally the necessary maintenance actions. The maintenance optimization using prognostics information has been used in several applications, such as electronic systems [3, 34], aerospace [2, 11] and wind turbines [5, 27].

For example, Camci *et al.* [11] proposed a tool for integrating PHM data with maintenance data. The PHM data are mainly the results of the prognostics step in the form of RUL. The maintenance data includes the resources needed for maintenance actions (personnel, hardware, tools, etc.), the resources available in the inventory, and the time to receive the resources ordered. The PHM process and the maintenance system are based on the return of the integration to update their data. Also, the proposed tool allows analyzing the information about the programmed missions to obtain more accurate RUL values since the mission profile affects the degradation rate of an equipment. In their work, Camci *et al.* [11] implemented this tool for fighter aircraft to validate its performance.

2.2 Control

The prognostics results are also used to improve the control of systems by determining immediate or rapid actions.

For example, Bole *et al.* [7] worked on the distribution of tasks based on the prognostics data to perform the control of a system. The idea consists in integrating the prognostics model into the control system. This latter uses the prognostics data to distribute the effort between the equipment of the system in order to better manage the risks generated by the uncertain estimates and the future performance of the system. To validate its performance, this methodology were applied to an autonomous vehicle subjected to degradation caused by thermal stresses. Further works on improving control using prognostics data may also be cited, such as the works of Bogdanov *et al.* [6] on servomotors and of Brown *et al.* [10] on the control of electro-mechanical actuators.

2.3 Missions reconfiguration

The data provided by the prognostics step can also be used to reconfigure the mission of a system depending on its health state. This type of decision is not yet sufficiently developed in the literature. However, there are some works that were proposed in specific contexts, including production scheduling [1], sensors network management [16], battery management [31] and the management of autonomous vehicles [37].

For example, Asmai *et al.* [1] have shown that knowing the RUL value can be very useful for the production scheduling. Indeed, this value gives information on the health state of the production equipment, which can be taken into account when launching new production tasks. This can prevent loss of production and waste of materials that can be caused by a failure occurring during the production. The decision using the prognostics information can take many forms, such as an immediate stop of the machine to avoid further damage, a continuation of normal production, a preventive maintenance intervention, or a re-scheduling of the production.

This paper deals with the post-prognostics decision in distributed systems. The main aim is to define the appropriate decision based on obtained prognostics results to optimize the usage of such systems.

2.4 Post-prognostics decision in distributed systems

After defining the post-prognostics decision, its different types and its main uses, this subsection is intended to position the contribution proposed in this paper in the post-prognostics decision. As presented before, most of the works dealing with the post-prognostics decision

focus on applications involving a single system or equipment. Contrary to these works, our work aims to go further in the sense that we consider a set of autonomous equipment which communicate and interact with each other using a communication network to fulfill a common mission, i.e. a distributed system.

Concerning the post-prognostics decision, such systems can have two levels: the module and the overall system. At the module level, it must be independent and able to assess in real time its health, estimate its RUL and auto-reconfigure depending on its health state to operate with the performance expected by the operator. Then, at the overall system level, modules communicate their health state and their RUL to their direct neighbors. This allows to detect degraded or failed modules in the system and reconfigure or adapt its mission based on module health states. Two types of post-prognostics decision can be used in this case: the control for the module level and missions reconfiguration for the overall system level.

This strategy can be applied to several modular applications such as sensor networks and modular robots [25]. In this work, this strategy is applied to a distributed MEMS-based conveyor presented in the next section. For clarity of presentation, only the mission reconfiguration is considered in this work. In the following, we assume that we have a control system at the module level.

3 Implementation in a distributed MEMS-based conveyor

3.1 System description

Most of the existing solutions to convey objects in production lines rely on contact-based technologies. However, these solutions are not appropriate for fragile and tiny micro-objects (medicines, micro-electronics parts, etc.), which can be easily damaged, contaminated or even scratched during conveying. Thus, conveyors based on contact-less air-jet technology, which avoid contact with conveyed objects, can be a solution in this case [23, 13, 18].

A conveyor generally consists of a single monolithic block dedicated to a specific task in a fixed environment. As a consequence, in case of failure or environment change, the conveyor will not be able to perform the dedicated task and has to be replaced. To address these issues, self-reconfigurable systems, which consist generally of small MEMS-based modules, can be used [24, 33].

A MEMS is a micro-system that integrates mechanical components using electricity as source of energy

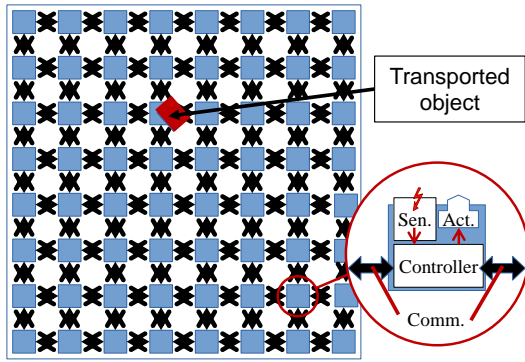


Fig. 2 General scheme of the conveying surface.

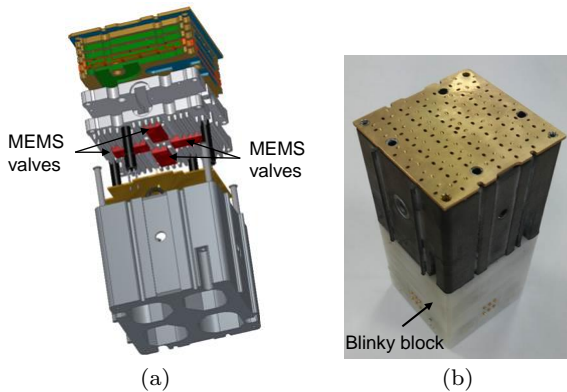


Fig. 3 (a) Design scheme of a smart block and (b) prototype of a manufactured smart block.

in order to perform measurement functions and/or operating in structure having micro-metric dimensions. Thanks to their miniaturization, low power consumption and tight integration with control and sense electronics, MEMS devices come in a wide variety of fields such as aerospace, automotive, bio-medical and communication technologies. Classical MEMS include accelerometers, gyroscopes, pressure sensors and micro-mirror arrays.

A self-reconfigurable conveying system is proposed by Boutoustous *et al.* [8]. It consists of a contact-less distributed MEMS-based conveying surface for safe and fast conveying of fragile and tiny micro-objects (Fig. 2). It is composed of an array of decentralized blocks, called smart blocks. In this conveyor, only one MEMS valve is used in each smart block to control the air flow.

In this paper, a similar conveyor is proposed, but with some modifications. We designed a new smart block (Fig. 3), in which four MEMS valves are used to control the air flow in the four directions (one MEMS valve for each direction) and a blinky block [22]. This latter allows a block to communicate with its four neighbors and integrates the prognostics and decision algorithms.

Through the blinky blocks, smart blocks communicate with each other to exchange information about

their health state to ensure a common goal, which is the transport of objects. The conveying principle consists in transferring objects from a start block to a final destination using controlled airflow controlled by the MEMS valves. To do so, all MEMS valves involved in conveying the objects have to be in a good health state and able to accomplish the mission. Then, the objective consists in using the information provided by the prognostics step to take appropriate decisions in order to minimize the risk of mission failure, anticipate MEMS failures, avoid the loss of the transported objects, optimize the utilization of the surface and maintain as long as possible a good level of performance. The post-prognostics decision consists in finding the best path that the object must take from the point of view of the health state of MEMS in the conveyor.

3.2 System Characteristics

The object of study is a conveying surface composed of m smart blocks denoted b_k , where $k \in \{1, 2, \dots, m\}$ is the number identifying the block in the surface. Each block contains four MEMS valves denoted $M_{k,i}$, where $k \in \{1, 2, \dots, m\}$ is the number identifying the block and $i \in \{1, 2, 3, 4\}$ is the number identifying the MEMS valve in the block. Thanks to the MEMS valves, each block is able to transfer objects to its neighbors in four directions (d_1, d_2, d_3, d_4). The directions correspond to the four sides of the square surface of the block.

The conveyor can be divided in two levels: 1) the smart block level and 2) the overall system level.

3.2.1 The smart block level

We consider that the health state of the conveyor is given by the health state of the smart blocks, which in turn is given by the health state of the MEMS valves inside. To evaluate the health state of the MEMS valve and predict its RUL, we have first to define its degradation model. This model was obtained in a previous work [36] and is generic for all the MEMS valves. It is related to the decrease in the magnitude of a physical parameter of the MEMS, called Health Indicator (HI). The projection of this HI can be exploited to predict the future behavior of the MEMS valve and estimate its RUL. For more details about how the degradation model is defined and the obtained prognostics results (RUL values), interested readers can refer to Skima *et al.* [36].

Each MEMS valve in the smart block is characterized by:

- A degradation model $HI_t(k, i)$. It represents the health state at time t of the MEMS $M_{k,i}$:

$$HI_t(k, i) = a. \exp(b.N_t(k, i)) + c. \exp(d.N_t(k, i)) \quad (1)$$

where $N_t(k, i)$ is the number of cycles performed by the MEMS $M_{k,i}$ up to time t and a , b , c and d are the parameters of the degradation model.

- A RUL value $RUL(k, i) = F(HI_t(k, i))$: the remaining useful life expressed in number of cycles. This value is estimated by using the degradation model and a state estimation tool (see [36]).
- A transfer time of the object $T(k, i)$: the time that takes an object to traverse a block and reach the next one. This parameter can be presented in two forms:
 - $T(k, i) = G(HI_t(k, i))$: it can be variable and depend on the degradation of the MEMS valve. The more degraded the MEMS is, the higher the transfer time is.
 - $T(k, i) = \text{const}$: it can be a constant value. This is related to the control decision. The MEMS valve is electro-thermally actuated and by increasing the input voltage, we can have the same performance (air pressure) even if the MEMS is degraded. Thus, by controlling the input voltage, we can maintain the same performance of the MEMS valve and then the same transfer time.

3.2.2 The overall system level

The conveying surface is composed of a set of m smart blocks. Each one is surrounded at most by four other blocks with which is able to communicate (send and receive information about the health state) thanks to its communication module (blinky block). Moreover, each block can transfer objects to its neighbors. This transfer is performed by the air flow controlled by the MEMS valves inside the block. Fig. 4 illustrates an example of a conveying surface composed of 9 smart blocks. For example, the block b_5 can communicate with its four neighbors b_2 , b_4 , b_6 and b_8 . An object located on the surface of the block b_5 can be transferred in directions d_1 , d_2 , d_3 and d_4 (respectively to blocks b_2 , b_4 , b_6 and b_8).

3.3 Mission of the system

The mission of the conveying surface consists in transporting objects by using a controlled air flow. This mission results in the search of the path allowing to go from a source block to a destination block. Thus, the conveying of an object can be characterized by a path that corresponds to a set of n blocks that participated in the

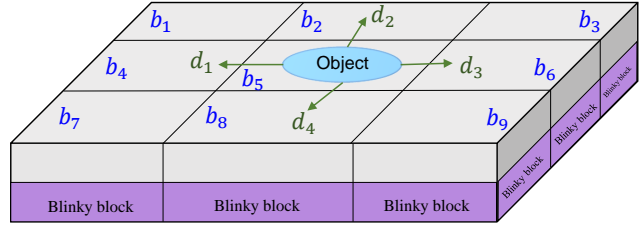


Fig. 4 Illustration of a conveying surface composed of 9 smart blocks.

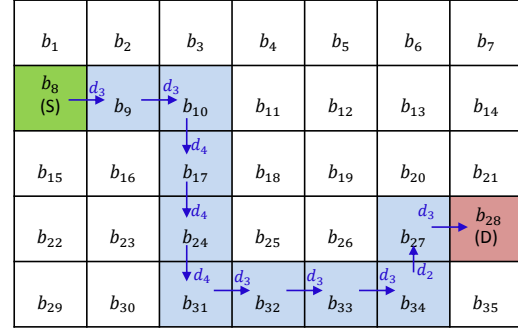


Fig. 5 Illustration of a path between the source block S and the destination block D .

transport of the object, $path = \{S, b_{k_2}, \dots, b_{k_{n-1}}, D\}$, where S corresponds to the source block b_{k_1} and D to the destination block b_{k_n} . The index j , with $j \in \{1, \dots, n = \text{length}(path)\}$, is the order of the block on the path. For example, b_{k_j} is the block number j on the path. A path is valid when b_{k_j} and $b_{k_{j+1}}$ are two neighboring blocks. Since two consecutive blocks on the path are neighbors, we can then deduce the unique direction between these two blocks that we denote by d_i , where $i \in \{1, 2, 3, 4\}$. Fig. 5 illustrates an example of a path on a given surface. For example, in this path denoted $path = \{b_8, b_9, \dots, b_{27}, b_{28}\}$, d_4 is the direction to go from the block number 10 on the surface (b_{10} to the block 17 on the surface (b_{17}).

We defined the following two metrics on the path, namely the RUL and the transfer time of this path:

- the path RUL value ($RUL(path)$) corresponds to the minimum of all the RUL values of the MEMS valves that participated in conveying the object on this path:

$$RUL(path) = \min_{j=1, \dots, n} RUL(k_j, i) \quad (2)$$

- the path transfer time ($Time(path)$) is the sum of all the transfer times of the MEMS valves that participated in conveying the object on this path:

$$Time(path) = \sum_{j=1}^n T(k_j, i) \quad (3)$$

The objective of a conveying mission is to maximize the lifetime of the surface, optimize its utilization and

improve its performance. This can result in the maximization of the RUL of the path and the minimization of the transfer time of the objects from the source S to the destination D . In the next section, an algorithm to solve this problem is presented.

3.4 Decision algorithm

Finding an optimal path from a start block to a destination one on the conveying surface is similar to some classical problems in the graph theory. The conveyor can be modeled as a weighted undirected graph, where each vertex represents a block with four edges connected to its neighbors. Each edge in the graph has two weights, which are the RUL and the transfer time.

The minimization of the transfer time is similar to the search of the shortest path in a graph and the maximization of the RUL is similar to the search of the maximum flow in a network. In the literature, several algorithms are proposed to find solutions to these classical problems, for example Dijkstra [14] and Bellman-Ford [28] algorithms for the time minimization, and Ford-Fulkerson [17] and Edmonds-Karp [15] algorithms for RUL maximization. However, in our case, the challenge is to find a path between a source and a destination that maximizes the RUL and minimizes the transfer time.

Among the available algorithms, the Dijkstra's algorithm can be used to solve this problem. Indeed, this algorithm meets our need in term of transfer time minimization, but not in term of RUL maximization. Therefore, we adapted it to maximize the RUL in addition to the minimization of the transfer time.

The modified Dijkstra's algorithm, presented in Algorithm 1, finds an optimal path with maximum RUL, and in case of equal paths (same path RUL), it chooses the path that has the minimum transfer time (fastest path). Note that, the obtained algorithm maintains the same complexity as the original Dijkstra's algorithm.

The optimized criteria need to be ordered according to their importance. In Algorithm 1, the path RUL is maximized as a principal criterion and the transfer time is minimized if there is more than one path with the same path RUL value. Recall that the path RUL value is the minimum RUL value on the path, cf. (2), and the path transfer time is the sum of all MEMS transfer time on the path, cf. (3). If we want to minimize the transfer time as a principal criterion, the relaxation part in the algorithm (line 19) should be changed to:

```
if timeThroughU <  $T_s[w]$  or (timeThroughU ==
 $T_s[w]$  and RULThroughU >  $R_s[w]$ )
```

Algorithm 1 Modified Dijkstra's algorithm.

```
1: function ModifiedDijkstra (Graph,  $s$ )
2: for each vertex  $w$  in the Graph do
3:    $T_s[w] = \infty$  // time from the source  $s$  to  $w$ 
4:    $R_s[w] = 0$  // RUL from the source  $s$  to  $w$ 
5:    $P[w] = \text{undefined}$  // previous block in the path
6: end for
7:  $T_s[s] = 0$  // transfer time from the source to the source
8:  $R_s[s] = \infty$  // RUL from the source to the source
9:  $Q =$  initially contains the  $s$  vertex //  $Q$  is a priority
   queue
10: while  $Q$  is not empty do
11:    $u =$  vertex in  $Q$  with the biggest RUL in  $R_s[]$ 
12:   remove  $u$  from  $Q$ 
13:   for each neighbor  $w$  of  $u$  do
14:      $\Delta_x = w_x - u_x$ 
15:      $\Delta_y = w_y - u_y$ 
16:      $\Delta = \Delta_y + 2 + 2\Delta_x$  // number of the MEMS allowing
   to move toward the neighbor
17:     timeThroughU =  $T_s[u] + u.getT[\Delta]$ 
18:     RULThroughU =  $\min(R_s[u], u.getR[\Delta])$ 
19:     if RULThroughU >  $R_s[w]$  or (RULThroughU ==
 $R_s[w]$  and timeThroughU <  $T_s[w]$ ) then
20:        $R_s[w] =$  RULThroughU
21:        $T_s[w] =$  timeThroughU
22:        $P[w] = u$ 
23:       add  $w$  to  $Q$ 
24:     end if
25:   end for
26: end while
27: end function
```

This algorithm can be implemented in the blinky block of each smart block as a decision maker. In the next section, it is used to simulate post-prognostics decision in the distributed conveying surface.

4 The DiMEMS Simulator

To show the importance of the post-prognostics decision making in the distributed MEMS-based conveyor (resulting in a longer life for the surface), we have developed *DiMEMS Simulator*, a simulator written in Java programming language and which is multi-threaded. Once launched, it allows to choose the dimensions of the conveying surface, the number of objects to introduce on the surface, their source(s), their destination(s) and the principal criterion (RUL or transfer time). It creates the surface with random values for both criteria in each block.

In a previous work [35], the simulator used only 1 MEMS per block and used a simpler, linear degradation model; also, this work does not analyze surface lifetime, but the evolution of the best path during conveying.

In this paper, the simulator is updated to use four MEMS per block and the degradation model previously described. Each time a MEMS $M_{k,i}$ participates at conveying an object, its number of cycles $N(k, i)$ is incre-

mented. As a consequence, its HI value $HI(k, i)$ decreases, its $RUL(k, i)$ decreases and its transfer time $T(k, i)$ increases. Hence, RUL and transfer time of blocks change dynamically.

At the beginning of the simulation, each block stores a matrix of the same size as the surface. Each cell of this matrix maps to the corresponding block in the surface and contains the RUL and the transfer time for each of its four MEMS. Initially, the cell of its own block has the right values of RUL and transfer time, and all the other cells contain 0 for both criteria. Before starting the simulation, each block communicates with its four neighbors and sends them its matrix. When a neighbor receives the matrix, it compares values in this matrix with its matrix and stores the maximum, since the RUL and transfer time cannot be less than or equal to 0. After some time, all the blocks have the same matrix which contains the right values of the surface.

Once this step is finished, the first object is sent in the surface. Blocks execute asynchronously the algorithm shown in Algorithm 2 (Fig. 6). If the block detects the presence of an object, it executes the modified Dijkstra's algorithm and sends the object to the next block according to the result that it finds. Thus, the RUL and the transfer time of the MEMS which participates at conveying the object change. Then, it sends its updated matrix with the new values to its four neighbors. When a neighbor receives the matrix, it compares values in this matrix with its matrix and stores the minimum RUL as this criterion can keep the same value or decrease and the maximum transfer time as it can keep the same value or increase. We call this step *diffusion*. If a block does not detect an object, it continues to send its matrix to its neighbors (diffusion). Thus, blocks have always an updated matrix. Fig. 7 shows the steps performed by each block. The big advantage of being asynchronous is that the surface does not need a global clock for all the blocks, which facilitates the surface manufacturing.

Algorithm 2 Algorithm executed asynchronously by each block.

- 1: **if** the object is above the block **then**
 - 2: execute modified Dijkstra's algorithm with itself as starting block, thus finding out the next block
 - 3: send the object to the next block
 - 4: consequently, the degradation of the used MEMS changes
 - 5: update its matrix by changing the values (RUL and transfer time) of its own cell
 - 6: **end if**
 - 7: send its matrix to its four neighbors, so that the next block have always the updated matrix
-

Table 1 Numerical values of the exponential models parameters.

Param.	MEMS 1	MEMS 2	MEMS 3	MEMS 4
a	$-1.025 \cdot 10^4$	$-8.47 \cdot 10^4$	$-3.727 \cdot 10^5$	$4.041 \cdot 10^6$
b	0.0168	0.0157	0.0073	0.0116
c	$1.029 \cdot 10^4$	$8.48 \cdot 10^4$	$3.727 \cdot 10^5$	$-4.041 \cdot 10^6$
d	0.0167	0.0157	0.0073	0.0116

When the object is in the destination, it leaves the conveyor. Meanwhile, the updated matrix spreads to the other blocks. We assume that the information exchange is much faster than the movement of the object, so the source receives the updated matrix before the object completely leaves the conveyor.

5 Simulation and results

Before explaining how values in the surface are generated and the different scenarios of simulation, it is necessary to set the framework of the simulations. This framework is defined by the following assumptions:

- an object covers one block.
- multiple objects can be on the surface at the same time, but the time between sending two consecutive objects is sufficiently high to avoid collision of objects.
- the transfer time is constant and is the same for all MEMS:

$$T(k, i) = \text{const} \quad (4)$$
- as the MEMS valve can perform more than 10 million cycles, the RUL is expressed in days to get the results more quickly (1 day = 85 000 cycles).

5.1 Data generation

In the previous work [36], four MEMS valves were tested. The obtained experimental results allowed to define a generic degradation model (Eq. 1) for all MEMS valves, but with different numerical values of the model parameters (a , b , c and d). These values are given in Table 1.

In practice, the RUL is estimated using the degradation model with a state estimation tool. The implementation of a such tool in the simulator is of interest only when the input data are acquired online. In this work, we propose to use prognostics results obtained in a previous work [36]. For that, a relation between the evolution of the HI and the RUL is defined rather than implementing the state estimation tool:

$$RUL(k, i) = \alpha \cdot \exp(\beta \cdot HI(k, i)) \quad (5)$$

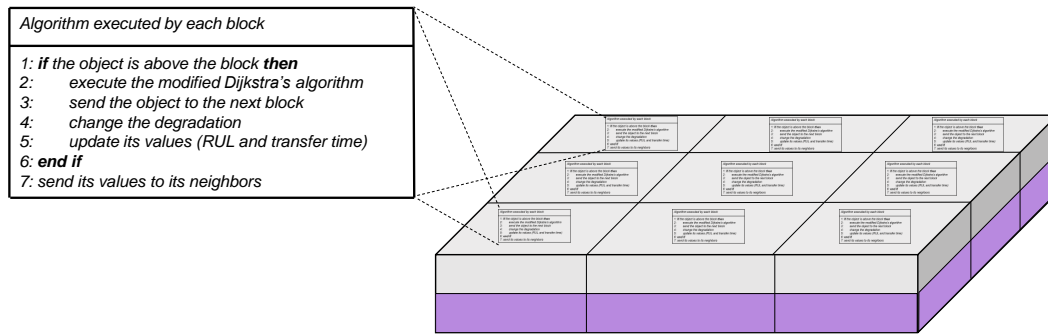


Fig. 6 Distributed operation of the surface.

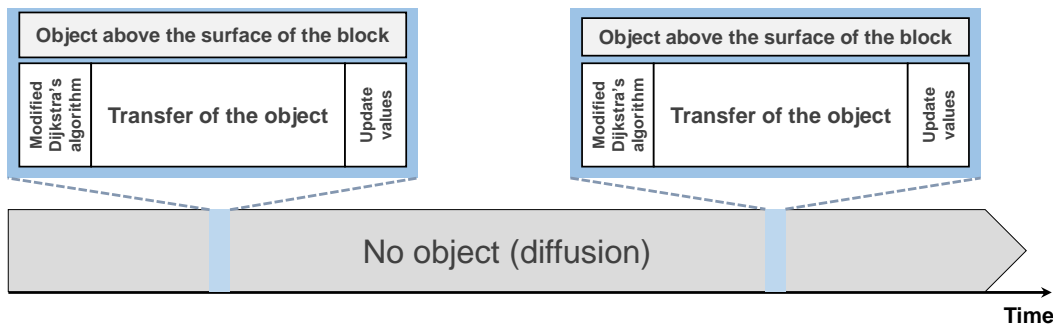


Fig. 7 Steps performed by each block.

where $\alpha = 0.2489$ and $\beta = 0.1575$.

When the surface is generated, all the parameters (a , b , c , d , α and β) are multiplied by a random value in order to obtain different MEMS characterizations.

5.2 Scenarios of simulation

Table 2 resumes the various simulation scenarios. Details are in the following.

A homogeneous initial surface means that all MEMS are not degraded and have almost the same values (differing by a small random value). A heterogeneous surface means that MEMS have various (random) initial degradations.

Scenario one source one MEMS: simulations consist in sending objects from a given source (one block in the left side of the surface) to a given destination (one block in the right side of the surface) (Fig. 4). We assume that the destination is known by all the smart blocks.

Four simulations are performed: 1) RUL as a principal criterion without decision (without decision means that all objects take the path found by the source block and taken by the first object), 2) transfer time as a principal criterion without decision, 3) RUL as a principal criterion with decision (each block executes the decision algorithm once it detects an object) and 4) transfer time

as a principal criterion with decision. Note that, for comparison purposes, the same initial surface is used.

Scenario several sources one MEMS: two types of simulation are performed. The first type consists in alternating sending objects from the sources (all blocks in the left side of the surface) to a given destination. Several simulations are performed and at each one we change the destination. The second type consists in alternating sending objects from the sources to the best destination. This means that each block executes the decision algorithm and sends the object to the destination that allows having the best path RUL or the best transfer time. Thus, the destination can change during the conveyance of the object.

Scenario several sources several MEMS: one type of simulation is performed. It consists in alternating sending objects from the sources to the best destination. In this scenario, the transport of objects continues even with some failed MEMS in the surface, but stops however when there is no possible path. Blocks containing these MEMS are avoided thanks to the decision algorithm.

For each scenario, several simulations were performed with different surface dimensions. It was deduced that the same observations are made regardless of the surface dimension. The following section presents the results obtained with the dimension 4×9 (36 blocks, 144 MEMS).

Table 2 Scenarios of simulation.

Scenario	Initial surface	Simulation
1 s. 1 MEMS	homogeneous	Simulation stops when one MEMS fails
		1. RUL as principal criterion without decision
		2. transfer time as principal criterion without decision
		3. RUL as principal criterion with decision
Sev. s 1 MEMS	homogeneous	Simulation stops when one MEMS fails
		1. alternating sending objects from the sources to a given destination 2. alternating sending objects from the sources to the best destination
Sev. s. sev. MEMS	heterogeneous	Simulation stops when there is no possible path to convey the object – alternating sending objects from the sources to the best destination

5.3 Results

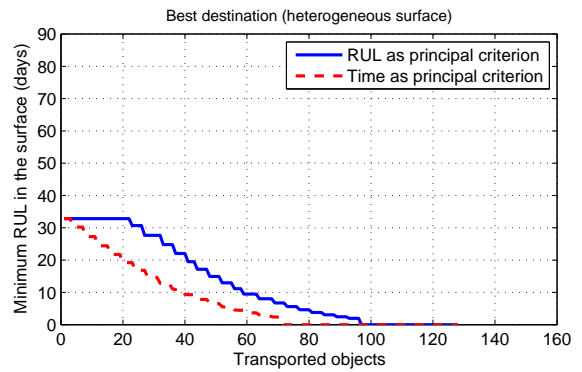
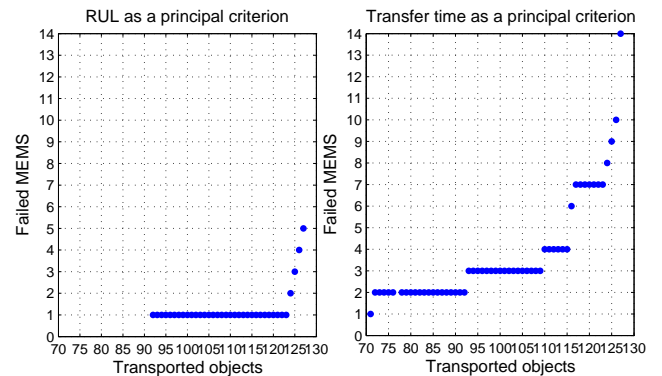
In the first scenario, Fig. 8, more objects are transported with decision than without decision. Also, with decision better optimizes the utilization of the surface. The minimum RUL value in the surface is greater than without decision for the same number of transported objects. In this scenario, the same observations are made regardless of the optimized criterion (RUL or transfer time).

Even with decision, only 69 objects are transported. This can be explained by the fact that the source and the destination are the most used and then their MEMS fail quickly since they are more solicited.

In the second scenario, Fig. 9, the first type of simulation shows that even if sources alternate, the same number of objects is transported as in the first scenario. This is explained by the fact that the destination is the most used. For that, the second type of simulation is performed. It allows to transport more objects and to better optimize the utilization of the surface.

The conclusion of these first two scenarios, in which the surface is homogeneous, is that there is not a big difference between optimizing the RUL as principal criterion or the transfer time. Also, it is better to alternate sources and choose the best destination in order to optimize the utilization of the surface.

The third scenario clearly shows the advantages of optimizing the RUL as a principal criterion rather than the transfer time. The minimum RUL value in the surface is more important when optimizing the RUL (Fig. 10). Also, the first MEMS fails after 92 objects. However, when the transfer time is used as first criterion, the first MEMS fails after only 71 objects. In addition, with the transfer time as a principal criterion, we have more failed MEMS at the end of the simulation (14 MEMS) rather than with the RUL (5 MEMS) after transporting the same number of objects (Fig. 11).


Fig. 10 Scenario several sources several MEMS.

Fig. 11 Number of failed MEMS as a function of the number of transported objects.

To conclude, in order to maintain a good performance of the surface, optimize its usage, transport more objects and have less failed MEMS, it is much better to use the third scenario with the RUL as a principal criterion. The transfer time is still optimized in the case of several paths with the same RUL value to take the fastest path.

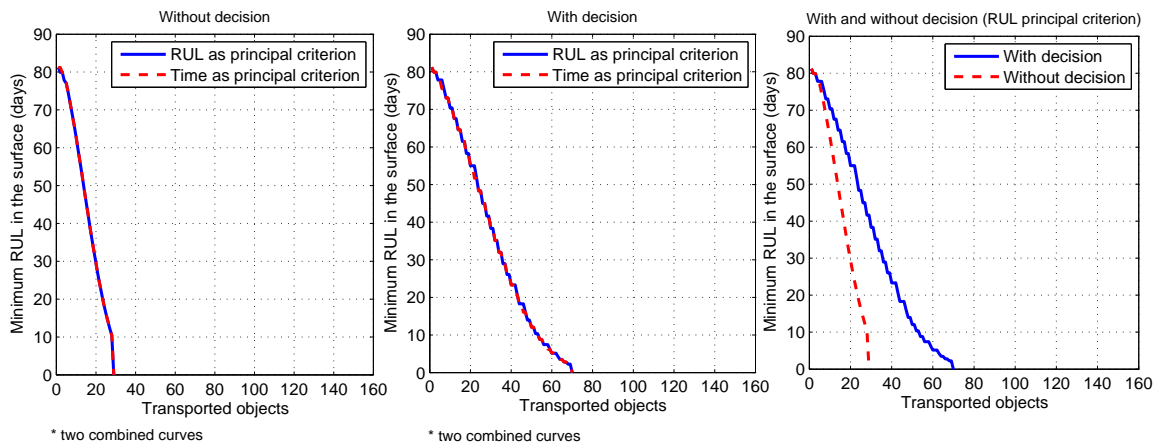


Fig. 8 Scenario 1 source 1 MEMS.

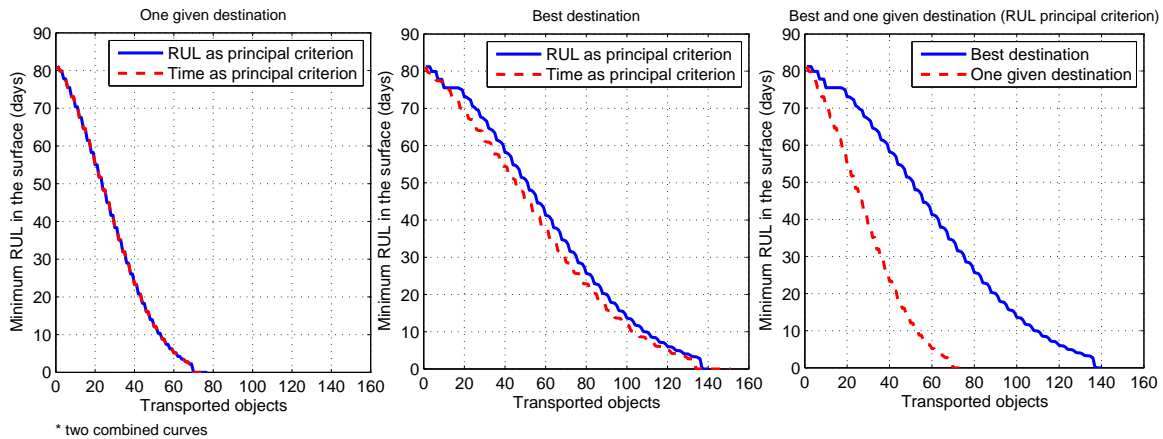


Fig. 9 Scenario several sources 1 MEMS.

6 Conclusion

In this paper, the problem of post-prognostics decision in distributed MEMS-based systems is addressed. First, a brief literature review related to the decision in the field of PHM is provided. After that, a strategy to make post-prognostics decision in distributed MEMS-based systems is proposed.

The proposed strategy is then applied to a new conveying surface to convey fragile and tiny micro-objects. This surface is composed of an array of decentralized smart blocks containing MEMS valves. The main aim is to use the prognostics information related to the MEMS valves to optimize the usage of this surface and maintain as long as possible a good performance. For this purpose, the Dijkstra's algorithm is modified and adapted in order to optimize criteria related to the health state of the used MEMS valves (RUL and transfer time). The decision consists in finding which path an object should take to optimize the usage of the conveying surface. To simulate this decision, a simulator writ-

ten in Java programming language has been developed. Simulation results show the significance of the proposed strategy and the importance of the post-prognostics decision to maintain the operation of the system and optimize its usage.

As future work, other simulation contexts will be performed such as objects sorting and block maintenance. Also, the prognostics and the decision algorithms will be implemented in the blinky blocks to confirm the results in a real physical system.

Acknowledgements This work has been supported by the *Région Franche-Comté* and the Labex ACTION project (contract ANR-11-LABX-0001-01).

References

1. Asmai SA, Hussin B, Yusof MM (2010) A framework of an intelligent maintenance prognosis tool. In: Computer Research and Development, 2010

- Second International Conference on, IEEE, pp 241–245
2. Balaban E, Alonso JJ (2012) An approach to prognostic decision making in the aerospace domain. In: Annual Conference of the Prognostics and Health Management Society, pp 396–415
 3. Barros A, Bérenguer C, Grall A (2003) Optimization of replacement times using imperfect monitoring information. *Reliability, IEEE Transactions on* 52(4):523–533
 4. Benkedjough T, Medjaher K, Zerhouni N, Rechak S (2015) Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing* 26(2):213–223
 5. Besnard F, Bertling L (2010) An approach for condition-based maintenance optimization applied to wind turbine blades. *Sustainable Energy, IEEE Transactions on* 1(2):77–83
 6. Bogdanov A, Chiu S, Gökdere LU, Vian J (2006) Stochastic optimal control of a servo motor with a lifetime constraint. In: Decision and Control, 2006 45th IEEE Conference on, IEEE, pp 4182–4187
 7. Bole B, Tang L, Goebel K, Vachtsevanos G (2011) Adaptive load-allocation for prognosis-based risk management. In: Annual conference of the prognostics and health management society, pp 1–10
 8. Boutoustous K, Laurent GJ, Dedu E, Matignon L, Bourgeois J, Fort-Piat NL (2010) Distributed control architecture for smart surfaces. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Taipei, Taiwan, 23, pp 2018–2024
 9. Brezak D, Majetic D, Udiljak T, Kasac J (2012) Tool wear estimation using an analytic fuzzy classifier and support vector machines. *Journal of Intelligent Manufacturing* 23(3):797–809
 10. Brown DW, Georgoulas G, Bole B, Pei HL, Orchard M, Tang L, Saha B, Saxena A, Goebel K, Vachtsevanos G (2009) Prognostics enhanced reconfigurable control of electro-mechanical actuators. In: Annual conference of the prognostics and health management society, PHM Society Rochester, NY, pp 1–17
 11. Camci F, Valentine GS, Navarra K (2007) Methodologies for integration of PHM systems with maintenance data. Tech. rep., DTIC Document
 12. Chretien S, Herr N, Nicod JM, Varnier C (2015) A post-prognostics decision approach to optimize the commitment of fuel cell systems in stationary applications. In: Prognostics and Health Management (PHM), 2015 IEEE Conference on, IEEE, pp 1–7
 13. Dahroug B, Laurent GJ, Guelpa V, Fort-Piat L, et al (2015) Design, modeling and control of a modular contactless wafer handling system. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, pp 976–981
 14. Dijkstra EW (1959) Communication with an automatic computer. PhD thesis, University of Amsterdam
 15. Edmonds J, Karp RM (1972) Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)* 19(2):248–264
 16. Elghazel W, Medjaher K, Zerhouni N, Bahi J, Farhat A, Guyeux C, Hakem M (2015) Random forests for industrial device functioning diagnostics using wireless sensor networks. In: Aerospace Conference, 2015 IEEE, IEEE, pp 1–9
 17. Ford LR, Fulkerson DR (1956) Maximal flow through a network. *Canadian journal of mathematics* 8(3):399–404
 18. Fukuta Y, Chapuis YA, Mita Y, Fujita H (2006) Design, fabrication, and control of mems-based actuator arrays for air-flow distributed micromanipulation. *Microelectromechanical Systems, Journal of* 15(4):912–926
 19. Haddad G, Sandborn P, Pecht M (2011) A real options optimization model to meet availability requirements for offshore wind turbines. In: MFPT: The Applied Systems Health Management Conference, Virginia Beach, Virginia
 20. He D, Li R, Bechhoefer E (2012) Stochastic modeling of damage physics for mechanical component prognostics using condition indicators. *Journal of Intelligent Manufacturing* 23(2)
 21. Iyer N, Goebel K, Bonissone P (2006) Framework for post-prognostic decision support. In: Aerospace Conference, 2006 IEEE, IEEE, pp 10–pp
 22. Kirby BT, Ashley-Rollman M, Goldstein SC (2011) Blinky blocks: a physical ensemble programming platform. In: CHI'11 Extended Abstracts on Human Factors in Computing Systems, ACM, Vancouver, Canada, pp 1111–1116
 23. Konishi S, Fujita H (1994) A conveyance system using air flow based on the concept of distributed micro motion systems. *Microelectromechanical Systems, Journal of* 3(2):54–58
 24. Kurokawa H, Tomita K, Kamimura A, Kokaji S, Hasuo T, Murata S (2008) Distributed self-reconfiguration of M-TRAN III modular robotic system. *The International Journal of Robotics Research* 27(3–4):373–386
 25. Lakhlef H, Mabed H, Bourgeois J (2014) An energy and memory-efficient distributed self-reconfiguration for modular sensor/robot networks. *The Journal of Supercomputing* 69(2):908–929

26. Lebold M, Thurston M (2001) Open standards for condition-based maintenance and prognostic systems. In: Maintenance and Reliability Conference (MARCON), May, pp 6–9
27. Lei X, Sandborn P, Bakhshi R, Kashani-Pour A, Goudarzi N (2015) PHM based predictive maintenance optimization for offshore wind farms. In: Prognostics and Health Management (PHM), 2015 IEEE Conference on, IEEE, pp 1–8
28. Lewandowski S (2010) Shortest paths and negative cycle detection in graphs with negative weights. Tech. rep., University of Stuttgart
29. Montgomery N, Banjevic D, Jardine AK (2012) Minor maintenance actions and their impact on diagnostic and prognostic CBM models. *Journal of Intelligent Manufacturing* 23(2):303–311
30. Mosallam A, Medjaher K, Zerhouni N (2014) Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing* pp 1–12
31. Saha B, Goebel K, Poll S, Christophersen J (2009) Prognostics methods for battery health monitoring using a bayesian framework. *Instrumentation and Measurement, IEEE Transactions on* 58(2):291–296
32. Saha B, Koshimoto E, Quach CC, Hogge EF, Strom TH, Hill BL, Vazquez SL, Goebel K (2011) Battery health management system for electric uavs. In: Aerospace Conference, 2011 IEEE, IEEE, pp 1–9
33. Salemi B, Moll M, Shen WM (2006) SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp 3636–3641
34. Sandborn P (2005) A decision support model for determining the applicability of prognostic health management (PHM) approaches to electronic systems. In: Proc. Reliability and Maintainability Symposium (RAMS), Arlington, VA
35. Skima H, Dedu E, Bourgeois J, Varnier C, Medjaher K (2016) Optimal path evolution in a dynamic distributed MEMS-based conveyor. In: International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, Springer, AISC 470, Brunw, Poland, 11, pp 395–408
36. Skima H, Medjaher K, Varnier C, Dedu E, Bourgeois J (2016) A hybrid prognostics approach for mems: From real measurements to remaining useful life estimation. *Microelectronics Reliability* 65:79 – 88
37. Tang L, Hettler E, Zhang B, DeCastro J (2011) A testbed for real-time autonomous vehicle PHM and contingency management applications. In: Annual conference of the prognostics and health management society, pp 1–11
38. Vieira RJdA, Lietti M, Sanz-Bobi MA (2012) New variable health threshold based on the life observed for improving the scheduled maintenance of a wind turbine. In: Advanced Maintenance Engineering, vol 2, pp 187–192
39. Zhang Z, Wang Y, Wang K (2013) Fault diagnosis and prognosis using wavelet packet decomposition, fourier transform and artificial neural network. *Journal of Intelligent Manufacturing* 24(6):1213–1227