Optimal Path Evolution in a Dynamic Distributed MEMS-based Conveyor

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

Univ. Bourgogne Franche-Comté (UBFC)
Institut FEMTO-ST, UMR CNRS 6174 – UFC / ENSMM
15B av. des Montboucons, Besançon, France
firstname.lastname@femto-st.fr

Abstract. We consider a surface designed to convey fragile and tiny micro-objects. It is composed of an array of decentralized blocks that contain MEMS valves. We are interested in the dynamics of the optimal path between two blocks in the surface. The criteria used for optimal paths are related to the degradation of the MEMS, namely its remaining useful life and its transfer time. We study and analyze the evolution of the optimal path in dynamic conditions in order to maintain as long as possible a good performance of the conveying surface. Simulations show that during usage the number of optimal paths increases, and that position of sources greatly influences surface lifetime.

1 Introduction

Conveyors have a widespread use across numerous industries where objects should be moved, including the automotive, computer, agricultural, etc. Using conveyors is much safer than using a forklift or other machines to move objects. They enable safe transportation of objects from a start point to a given destination, which if done by human labor would be strenuous and expensive. Most of the existing solutions to convey objects in production lines rely on contact-based technologies. However, they 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 contactless air-jet technology, which avoid contact with conveyed objects, can be a solution in this case [11, 3, 5].

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 be not 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 [12, 16].

A Micro-Electro-Mechanical System (MEMS) is a *microsystem* that integrates *mechanical* components using *electricity* as source of energy in order to perform measurement functions and/or operating in structure having micrometric 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, biomedical and communication technologies. Classical MEMS include accelerometers, gyroscopes, pressure sensors and micromirror arrays.

The work presented in this paper is a part of a project which aims at increasing the efficiency of future production lines. Our ongoing project consists in designing a contactless distributed MEMS-based conveying surface for safe and fast conveying of fragile and tiny micro-objects. The conveying principle consists in sending micro-objects from a start block to a final destination using controlled air flow coming from MEMS valves. To do so, all MEMS valves involved in conveying the micro-object have to be in a good health state and able to accomplish the mission. However, and according to the literature, the reliability is one of the major concerns of MEMS [14]. They suffer from various failure mechanisms [19, 17] which impact their performance, reduce their lifetime, and the availability of systems in which they are utilized. This highlights the need to monitor their behavior, assess their health state, estimate their remaining time before failure (RUL, Remaining Useful Life) and take appropriate decisions accordingly such as control reconfiguration and maintenance [18]. These tasks can be done by using Prognostics and Health Management (PHM) approaches [8, 7, 15]. Therefore, in our project, PHM is applied to monitor the degradation of the MEMS valves and estimate their remaining time before failure with the aim at predicting failures in blocks.

In this paper, and in order to maintain as long as possible a good performance of the conveying surface, we study the evolution of the optimal path in dynamic conditions to find out how to optimize the use of the conveying surface (lifespan and conveying speed). In order to find the optimal path that avoids degraded blocks, we use the well known Dijkstra's algorithm. It optimizes a criterion related to the degradation of the MEMS valves, which is the RUL, and in case of equality the transfer time, which is also related to degradation. The values of these two criteria can be obtained by applying PHM.

To the best of our knowledge, *no article in the literature* deals with increasing conveyor lifespan by avoiding fatigued blocks. However, a similar problem appearing in the literature is the lifetime extension of multi-hop wireless networks from energy point of view [2, 4, 9]. During packet routing, without special precaution, some nodes are used more often than others, and their energy may be exhausted faster the others. To avoid this, packets need to be routed through the optimal path which optimizes the energy of nodes in the network. As in our case, oscillation between optimal paths has been noticed, without further analysis. Moreover, there are differences compared to our problem. In network case, sources and destinations are spread over the whole surface, whereas in conveyors only one or a few nodes (usually on the border of the surface) can act as source or destination, leading to specific usage patterns. Also, in network case oscillations can reduce the lifetime of the network, which is not the case for us. Finally, in network case the lifetime is defined as the duration of time before the first node fails due to the battery exhaustion, whereas in our case the algorithm allows the conveying surface to still operate by avoiding failed blocks.

A second similar problem we have found in the literature is about finding optimal paths in transportation systems. [6] and its related work are about algorithms of finding optimal paths in real transportation systems formed by roads and cars, modeled as stochastic time-dependent (STD) networks. In those systems, many cars exist on a road, so that traffic flow propagates both in time and space (a congestion influences its region for some time, and also the region around it), so congestion is an important parameter, whereas in our case only one object at a time exists in the whole system. Also, external conditions, such as accidents and thunderstorms, affect the model (e.g. travel time), whereas our model is not affected by external factors. Travel time of cars on a road depend on time (i.e. more cars at morning and evening than during the night), whereas in our case it depends on its degradation, which depends on its usage. Cars are driven by humans who add uncertainty, whereas in our case all objects follow specified unstochastic algorithms. To conclude, the model used in our work is simpler and also *different*, making these works useless in our case. Furthermore, our paper is not about finding the optimal paths, but about their evolution during usage.

For clarity of presentation, this paper does not address the PHM part, but only the evolution of the optimal path. After the introduction, Section 2 presents the distributed MEMS-based surface. Section 3 introduces the simulator and presents the results. Section 4 concludes the paper.

2 The distributed MEMS-based conveyor

The conveying surface is composed of an array of decentralized blocks, called smart blocks. Fig. 1 shows a design scheme of a smart block. Currently, the blocks are in manufacturing phase. Each block contains a micro-controller, a sensor to detect if an object is above it, a power supplier, a MEMS valve and is able to communicate with its four block neighbors thanks to the blinky block [10].

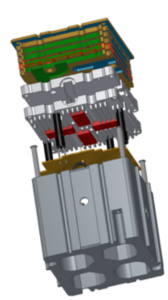


Fig. 1. Design scheme of a smart block.

The MEMS used to generate air flow is an electrothermally-actuated valve designed by DunAn Microstaq, In. (DMQ), company. To predict the remaining time before failure of this micro-system, we have first to define its degradation model. This model can be obtained experimentally through accelerated lifetime tests or given by experts of the MEMS. It is influenced by drifts of the physical parameters of the MEMS (friction coefficient, stiffness, etc.) and can be obtained by analyzing the collected data from tests (evolution of the parameters as a function of time) by using appropriate modeling tools such as regression and curve fitting.

Currently, we are performing accelerated lifetime tests to define the degradation model of the targeted MEMS. The simplest and most useful accelerated lifetime tests to define the degradation model of a MEMS is to stress it by applying a square signal (cycling) [13]. To do so, we have already designed an experimental platform where several MEMS are continuously cycling with a square signal of 8 V magnitude and 1 Hz frequency. We are performing measurements every day and at each measurement we estimate the values of the physical parameters. We will keep repeating this process until the occurrence of a failure.

As in the macro-systems, the degradation of MEMS devices can be modeled by linear or nonlinear functions [14]. In this work, we aim at studying the evolution of the optimal path in a dynamic conveying surface. So, while waiting to have complete experimental data to define the degradation model, we assume that the degradation of the MEMS valve is given by a linear function (see Eq. 1).

Let's suppose that the conveying surface is composed of m blocks, each one containing a MEMS valve denoted as M_k ($k \in [1, m]$) and using the following model:

– *A linear degradation model* $deg(M_k)$:

$$deg(M_k) = 0.01 \times C(M_k) \tag{1}$$

  where $C(M_k)$ is the number of cycles performed by the MEMS M_k.
– *A RUL value* $RUL(M_k) \in [0, 100]$: the remaining time before failure expressed in number of cycles. In practice, this value is calculated using the degradation model and prognostics methods (particle filter [20], Kalman filter [1], etc.). In this work, we assume that the RUL value, which depends on the degradation level, is calculated as follows:

$$RUL(M_k) = (1 - deg(M_k)) \times 100 \tag{2}$$

  We consider that a new MEMS without degradation can perform 100 cycles.
– *A transfer time of the object* $t(M_k) \in [1, 11]$: the time that takes an object to traverse the block and reach the next block. A new MEMS without degradation has a transfer time of 1 second. The transfer time value is calculated as follows:

$$t(M_k) = 1 + deg(M_k) \times 10 \tag{3}$$

To convey an object, a MEMS can communicate only with its four neighbors and can send the object only to one of them. Fig. 2 shows the connection between MEMS M_a and its four neighbors M_b, M_c, M_d and M_e (one MEMS per block).

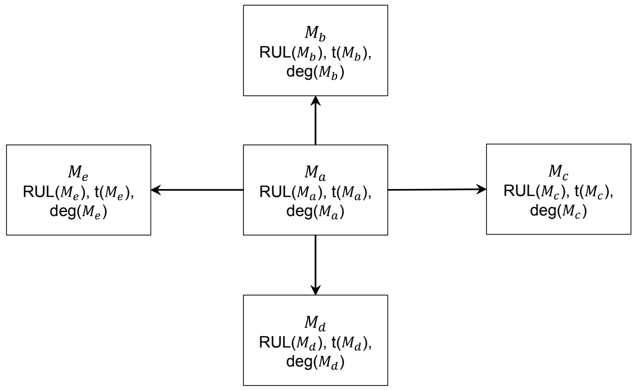


Fig. 2. Representation of the connection between one MEMS and its four neighbors.

Fig. 3 gives an overview of the surface. The block denoted as S represents the source of the object that is going to be moved towards the block denoted as D, which represents the destination of the object. The black blocks represent a path taken by the object.

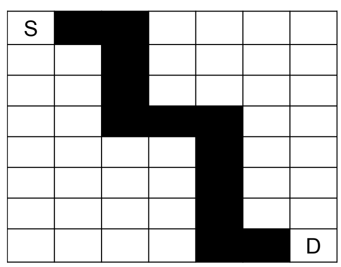


Fig. 3. Representation of a path taken by an object from the source block (S) to the destination block (D).

Let p be an optimal path from S (Source) to D (Destination). The path p is a set of n MEMS that participate at conveying the object, $p = \{S, M_2, ..., M_{n-1}, D\}$, where S corresponds to M_1 and D to M_n. The following equations represent respectively the RUL and the transfer time of the path:

$$RUL(p) = \min_{k=1,\dots,n} RUL(M_k) \tag{4}$$

$$T(p) = \sum_{k=1}^{n} t(M_k) \tag{5}$$

The main objective is to find a path that maximizes the RUL ($RUL(p)$) and minimizes the transfer time of the objects ($T(p)$) in the conveying surface.

3 Simulation results

3.1 Simulator features

We have developed a simulator to analyze the optimal path evolution in dynamic conditions. It is written in Java language and is multi-threaded (each block is a thread). It allows to choose the dimensions of the conveying surface, the number of objects to introduce on the surface, their source(s), and the principal criterion (RUL or time). It creates the surface with random values for both criteria in each block.

Objects are introduced at the given source(s). Each time a MEMS M_k participates at conveying an object, its number of cycles $C(M_k)$ is incremented. As a consequence, its degradation value deg increases, its RUL decreases and its transfer time t increases, cf. (1), (2) and (3) respectively. Hence, RUL and transfer times of blocks change dynamically during the simulation.

Each block has a matrix of the same size as the surface. Each cell of the matrix contains the RUL and transfer time of the corresponding block in the surface. Before starting the simulation, each block communicates with their four neighbors and sends them its matrix; 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 Algo. 1. The big advantage of being asynchronous is that the surface does not need a global clock for all the blocks, which facilitates greatly the surface manufacturing.

Algorithm 1 The algorithm executed asynchronously by each block.

if the object is above the block **then**
    execute Dijkstra's algorithm with itself as start block, thus finding out the next block
    send the object to the next block
    consequently, its degradation increments
    update its matrix by changing the values (RUL and transfer time) of its own cell
end if
send its matrix to its four neighbors, so that the next block have always the updated matrix

When the object is in the destination, it gets out of the surface. During this time, the updated matrix is propagating to the other blocks. We assume that information exchange is much faster than object moving, hence the source receives the updated matrix before the object gets out. A new object enters the system as soon as the previous one got out of the surface, so that only one object exists on the surface.

The simulation consists in sending the given number of objects from a given source to the destination. Fig. 4 shows the initial surface used in all simulations.

Blocks are characterized by RUL values of MEMS (left), transfer time values (right) and position.

95 – 1.5 (0,0)	98 – 1.2 (0,1)	57 – 5.3 (0,2)	23 – 8.7 (0,3)	64 – 4.6 (0,4)
14 – 9.6 (1,0)	44 – 6.6 (1,1)	16 – 9.4 (1,2)	88 – 2.2 (1,3)	58 – 5.2 (1,4)
79 – 3.1 (2,0)	83 – 2.7 (2,1)	27 – 8.3 (2,2)	83 – 2.7 (2,3)	22 – 8.8 (2,4)
44 – 6.6 (3,0)	98 – 1.2 (3,1)	72 – 3.8 (3,2)	96 – 1.4 (3,3)	99 – 1.1 (3,4)

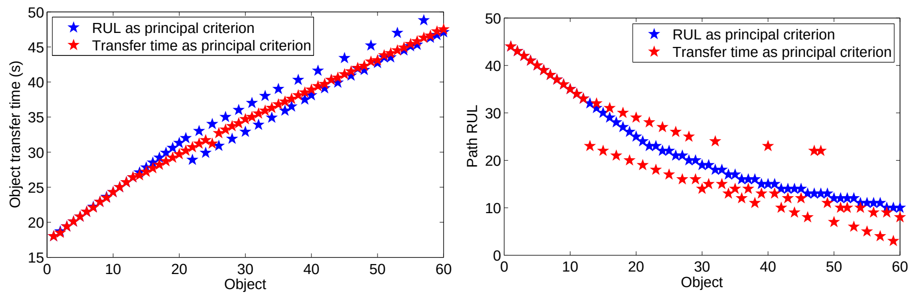
Fig. 4. Initial surface used in simulations. It contains 20 MEMS and each one is characterized by RUL (left), transfer time values (right) and position.

Note that the multi-threading does not change the results, so the program is still deterministic. The results in our case mean the path along which objects go. We prove in the following that the path is the same, no matter if there is multi-threading or not. So we need to prove that when an object enters a block, the next block will be the same, no matter if multi-threading or not. As written in Algo. 1, when an object arrives in a block, the block executes the lines 2–5, i.e. it executes Dijkstra's algorithm using its matrix. During object movement, the only modification in the surface is that the values (RUL and transfer time) have been modified for the previous block. Its matrix is received from its neighbours (previous block included) much faster than object move, as written above, hence the matrix has up to date values. So, since the matrix is updated, the next block is always the same (i.e. deterministic).

3.2 Choice of principal criterion

As mentioned before, two criteria need to be optimized and one of them has to be set as principal one. Since we aim to improve at maximum the lifetime of the conveying surface, we set the RUL as principal criterion, and the transfer time of the object is used only if multiple paths have the same value of RUL.

Fig. 5 (a) and (b) show the evolution of the transfer time and the path RUL value as a function of the number of objects, using each time a different principal criterion and otherwise the same conditions. We notice that we have greater path RUL values and almost the same transfer time if the RUL is used as principal criterion comparing using transfer time as principal criterion.



(a) (b)

Fig. 5. (a) Object transfer time and (b) path RUL as a function of number of objects when principal criterion changes.

3.3 Optimal path evolution

To study the evolution of the path according to the change of the RUL and transfer time, three types of simulations are done: one source, two sources and several sources. For all simulations only one destination is used, at position (3, 4).

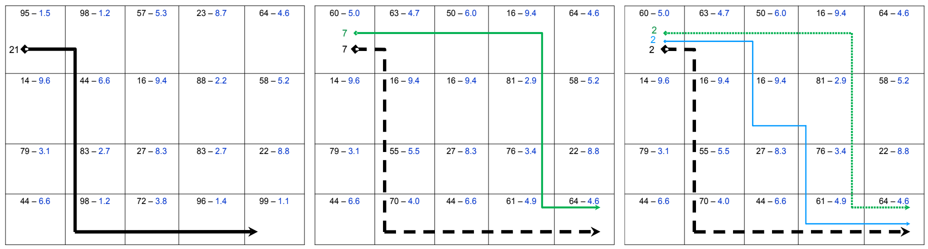
For **one source**, the simulation consists in sending objects from the source (0, 0) to the destination (3, 4). In Fig. 6, arrows indicate some paths taken by objects. The thickness of arrows is proportional to the number of times the path is used. The first object takes the path indicated in Fig. 6 (a). This path has a RUL value of 44 cycles and a transfer time value of 18 seconds (the object takes 18 seconds to reach the destination). The next 20 objects take the same path.

For the 22^{th} object, a new optimal path indicated in Fig. 6 (b) is taken (the dashed arrow indicates the previous path). This is explained by the fact that the block (1,1) is used 21 times, the RUL value of its MEMS decreases to 23 cycles and the transfer time of the path increases to 32 seconds. The new optimal path has the same RUL value of 23 cycles but a lower path transfer time of 28 seconds.

Then, paths oscillate seven times between the new optimal path and the previous one. Fig. 6 (b) shows the updated RUL and transfer time values in the surface after sending 21 objects. The 36^{th} object takes a new optimal path indicated in Fig. 6 (c) and which oscillates 2 times with the two previous optimal paths. Fig. 6 (c) shows the updated value after sending 35 objects. The same thing for the other objects, once a new optimal path is found, it oscillates with the ancient paths.

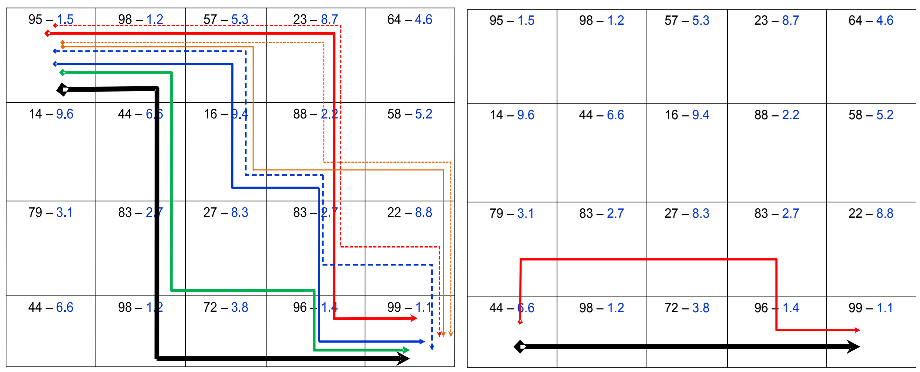
For **two sources**, objects enter through two sources (0, 0) and (3, 0) alternatively (Round Robin). Unlike the previous simulation, optimal paths oscillate more or less randomly because the blocks are used by multiple objects. Fig. 7 shows all optimal paths taken by the different objects.

For **several sources** (all the blocks on the left side of the surface are sources), optimal paths change randomly due to multiple objects that enter through the



(a) Oscillation 1 (b) Oscillation 2 (c) Oscillation 3

Fig. 6. Optimal path evolution for one source: objects enter through (0,0) (the number on left of arrows indicates the number of times the path is used in the oscillation).



(a) (b)

Fig. 7. Optimal path evolution for two sources: (a) paths used when objects enter through (0, 0) and (b) paths used when objects enter through (3, 0).

different sources which decreases the RUL and so the change of the optimal path (no oscillation between paths). Fig. 8 shows the results of this simulation.

These simulations show that:

– *Number of optimal paths:* During usage, a first optimal path is used for a number of times, afterwards a new optimal path appears and the two optimal paths are used for a number of times, afterwards another new optimal path appears and the three optimal paths are used etc.
– *Oscillation among optimal paths:* We define an oscillation as the interval of time where the optimal path oscillates (alternates) among several paths, for example between paths p_1 and p_2 from time t_1 to t_2. In case of one source, once a new optimal path appears, it oscillates with the previous optimal paths: in oscillation 1, there is only one path and no oscillation, in oscillation 2 the optimal path oscillates between this one and a new one, in oscillation 3 the optimal path alternates among the two paths and a new one etc. This phenomenon is seen less as the number of sources increases. For example, no oscillation of the optimal path is seen in the case of 4 sources.

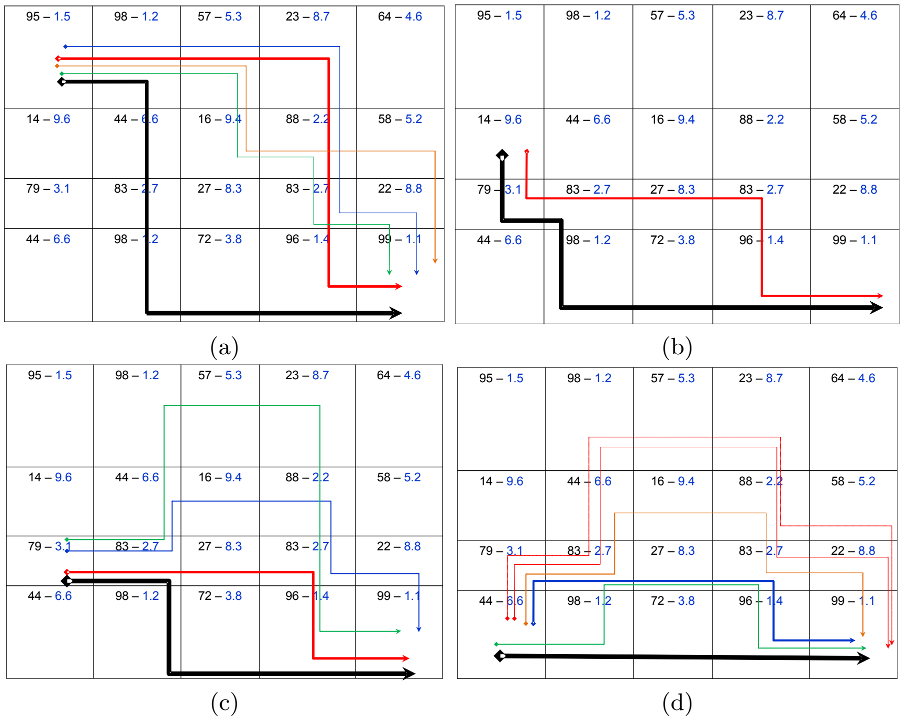


Fig. 8. Optimal path evolution for several sources: (a)–(d) present all the paths used for the four sources.

The reason is that optimal paths used for a given source change when blocks are used by objects entering through other sources.

– *Duration of usage of each optimal path during one oscillation:* In case of one source, in the oscillation 1, the first optimal path is used a number of times, in oscillation 2, the two optimal paths are used fewer times each, in oscillation 3 the three optimal paths are used even fewer times each, etc. Formally:

$$u(i) > u(i+1), \quad i > 0 \tag{6}$$

where $u(i)$ is the number of times *each* optimal path is used during oscillation number i.

– *Duration of usage of each optimal path during the whole simulation:* The first optimal path is the most used, afterwards the second optimal path etc. This is because once an optimal path appears, it is used until the end of simulation. For example, in one source the first optimal path is used in all the oscillations, the second path is used for oscillations 2, 3, ..., the third path is used for oscillations 3, 4, ..., and so on.

3.4 Optimal number and position of sources

The same three previous simulations are used.

Fig. 9 (a) shows the evolution of the total transfer time as a function of object number reaching the destination, which measures the conveying speed of the surface. It is found that the transfer time is greater with one source than with two or four sources. Intuitively, the reason is that source (0, 0) is at distance of 7 blocks from destination, whereas for two or four sources, source (3, 0) is sometimes used, and it is nearer to destination than (0, 0), at distance of 4 blocks.

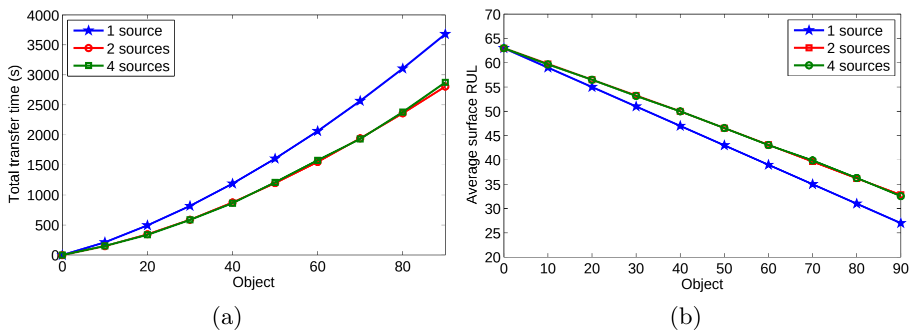


Fig. 9. (a) Total transfer time and (b) average surface RUL as a function of number of objects for different number of sources.

Fig. 9 (b) shows the evolution in time of the average surface RUL (counting all the blocks) which measures the degradation, i.e. the health state, of the surface: the smaller the average surface RUL, the more degraded the surface. We note that the average surface RUL is smaller when objects enter only through (0, 0). The reason is the same as previously: the source (0, 0) is farther than other sources.

To confirm that the reason is not the number of sources, but their position, a forth simulation is done which uses only the source (3, 0). This simulation takes less time and the average RUL on the surface is greater than all the three simulations, which confirms that the important parameter is the position of the source. However, in this fourth simulation only 44 objects can be sent because the RUL of the source is 44 cycles. More importantly, as a general fact, the source (and the destination) are the most used blocks and their degradation increases greater than other blocks: the nearer to the source or destination, the greater the degradation of a block.

As a conclusion, in order to increase the lifetime (allow to send more objects) and the conveying speed of the surface, we propose to place sources at convenient positions and use them alternatively. In a future work, we will study source positions and object scheduling among these sources.

4 Conclusion and future works

In this paper, we have considered a surface for conveying fragile and tiny micro-objects based on air-jet technology. The surface is composed of an array of decentralized blocks, each one containing an electro-thermally actuated MEMS valve.

We took into account the optimal path, using criteria related to MEMS degradation, which avoids degraded blocks on the surface. We analyzed the evolution of the optimal path in dynamic conditions in order to maintain as long as possible a good performance of the conveying surface. We noticed that optimal paths alternate or simply change during usage depending on the number of sources, some paths are used much more than others, and the greater the number of sources, the greater the number of optimal paths.

Future works include the scheduling of objects through the different sources, the extension of the algorithm to allow several concurrent objects on the surface, and the use of 4 MEMS per block, one block for each direction.

This work is only a step towards the realization of a contactless distributed MEMS-based conveyor. The ongoing results of the PHM part will be implemented and carried out on an experimental centimeter scale self-reconfigurable smart blocks conveyor which is being manufactured.

Acknowledgment

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

References

1. Baraldi, P., Mangili, F., Zio, E.: A kalman filter-based ensemble approach with application to turbine creep prognostics. Reliability, IEEE Transactions on 61(4), 966–977 (2012)
2. Blough, D.M., Santi, P.: Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In: International Conference on Mobile Computing and Networking. pp. 183–192. Atlanta, GA, USA (Sep 2002)
3. Dahroug, B., Laurent, G.J., Guelpa, V., Fort-Piat, L., et al.: Design, modeling and control of a modular contactless wafer handling system. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. pp. 976–981. IEEE (2015)
4. Floréen, P., Kaski, P., Kohonen, J., Orponen, P.: Lifetime maximization for multicasting in energy-constrained wireless networks. Selected Areas in Communications, IEEE Journal on 23(1), 117–126 (2005)
5. Fukuta, Y., Chapuis, Y.A., Mita, Y., Fujita, H.: Design, fabrication, and control of mems-based actuator arrays for air-flow distributed micromanipulation. Microelectromechanical Systems, Journal of 15(4), 912–926 (2006)
6. Huang, H., Gao, S.: Optimal paths in dynamic networks with dependent random link travel times. Transportation Research Part B 46(5), 579–598 (Jun 2012)

7. Javed, K.: A robust and reliable data-driven prognostics approach based on extreme learning machine and fuzzy clustering. Ph.D. thesis, University of Franche-Comté, Besançon, France (2014)
8. Jay, L., Fangji, W., Wenyu, Z., Masoud, G., Linxia, L., David, S.: Prognostics and health management design for rotary machinery systems reviews, methodology and applications. Mechanical Systems and Signal Processing 42(1), 314–334 (2014)
9. Kang, I., Poovendran, R.: On lifetime extension and route stabilization of energy-efficient broadcast routing over MANET. In: International Network Conference. pp. 81–88. London, UK (Jul 2002)
10. Kirby, B.T., Ashley-Rollman, M., Goldstein, S.C.: Blinky blocks: a physical ensemble programming platform. In: CHI'11 Extended Abstracts on Human Factors in Computing Systems. pp. 1111–1116. ACM, Vancouver, Canada (2011)
11. Konishi, S., Fujita, H.: A conveyance system using air flow based on the concept of distributed micro motion systems. Microelectromechanical Systems, Journal of 3(2), 54–58 (1994)
12. Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., Murata, S.: Distributed self-reconfiguration of M-TRAN III modular robotic system. The International Journal of Robotics Research 27(3–4), 373–386 (2008)
13. Matmat, M., Koukos, K., Coccetti, F., Idda, T., Marty, A., Escriba, C., Fourniols, J.Y., Estève, D.: Life expectancy and characterization of capacitive RF MEMS switches. Microelectronics Reliability 50(9), 1692–1696 (2010)
14. Medjaher, K., Skima, H., Zerhouni, N.: Condition assessment and fault prognostics of microelectromechanical systems. Microelectronics Reliability 54(1), 143–151 (2014)
15. Medjaher, K., Zerhouni, N.: Hybrid prognostic method applied to mechatronic systems. The International Journal of Advanced Manufacturing Technology 69(1-4), 823–834 (2013)
16. Salemi, B., Moll, M., Shen, W.M.: SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3636–3641. Beijing, China (Oct 2006)
17. Shea, H.R.: Reliability of MEMS for space applications. In: MOEMS-MEMS 2006 micro and nanofabrication. pp. 61110A–61110A. International Society for Optics and Photonics (2006)
18. Skima, H., Medjaher, K., Varnier, C., Dedu, E., Bourgeois, J.: Hybrid prognostic approach for Micro-Electro-Mechanical Systems. In: IEEE Aerospace Conference. pp. 1–8. 36, Big Sky, Montana, USA (Mar 2015)
19. Tanner, D.M.: MEMS reliability: Where are we now? Microelectronics reliability 49(9), 937–940 (2009)
20. Yin, S., Zhu, X.: Intelligent particle filter and its application on fault detection of nonlinear system. Industrial Electronics, IEEE Transactions on 62(6), 3852–3861 (2015)