Modeling and Simulation of Modular Complex System: Application to Air-jet Conveyor

Jean-Marie Gauthier[1], Dominique Gendreau[2], Ahmed Hammad[1] and Fabrice Bouquet[1]

Abstract— **In this paper, we propose a SysML model and a Modelica simulation of an air-jet conveyor for micro objects. Indeed, modeling and simulation are part of verification & validation activities, which are important tasks during the conception of a complex system. This article focuses mainly on the modeling and simulation of air-jet nozzles and on their influence on a millimeter size object. The obtained results are discussed and analyzed to obtain information on the conveyor system. This work is part of the ANR Smart Blocks project.**

I. Introduction

Sorting and conveying objects are two main tasks necessary for production lines. Lots of different methods exist to perform these tasks but most of them require a contact between the conveyor and the transported objects. For that reason, small and fragile objects can be damaged by manipulations whereas clean products can be contaminated by the contact with the conveyor (especially in the pharmaceutical industries, microelectronic and food). To solve these problems, we are developing a self-reconfigurable modular conveyor based on a contact-free technology (air-jets technology). This project is in the frame of the Project ANR-2011-BS03-005, named Smart Blocks.

This conveyor is composed of 2.5 centimeters-size blocks which will be linked together to form the conveying surface. Each block includes a MEMS (Micro Electro Mechanical Systems) actuator array in the upper face in order to move the objects, sensors able to detect the object's position, a micro-controller, and some communication ports which link it with its neighbours in order to plan global transport policies.

The main functional requirement of the smart block system is to convey small object with air-jets technology. Therefore, we need to predict and to master the behavior of a small object subjected to an high speed air flow. Modeling, simulation and testing are ways to validate such requirement.

Modeling and simulation of air-jets are important tasks to understand how one small object can move from one point to another. Moreover, simulation could give us some clues on how to arrange the blocks to provide non-linear trajectories.

The first goal of this paper is to propose a mathematical representation of the influence of air-jets to a small object. The second goal is to present a SysML [1] (System Modeling Language) model of this system, based on the mathematical model previously defined. Finally, the SysML model of the smart block conveyor is automatically translated into Modelica [2] simulation code to perform numerical experiments.

This paper is structured as follows: in Section II, we introduce the position of the paper and some related works. Section III, describes the study of air-jets from the mathematical representation to the modeling and shows the results of simulations. Finally, Section IV concludes and outlines our future work.

II. Position of the paper

A. Context, positioning and objectives

A contact-free conveyor solves most issues for the transport and the sorting by avoiding the contact with the conveyed objects.

Conveyors are usually designed as monolithic entities solving one problem at one time. A standard conveying belt is able to convey jumbled objects but it can neither precisely position nor sort them. This lack of functionality and modularity multiplies the different kinds of systems to be installed around the conveying belt, thus raising problems of maintenance, complexity of the production line, etc.

A conveyor made of small modules allows the integration of different kinds of functionalities inside one single conveyor and reduces the design complexity. Besides, having specialized blocks and modularity decreases the complexity of each block which makes them easier and cheaper to produce.

Monolithic design fits the need of fixed types of environments and/or objects. If, for some external reasons, the environment of the conveyor changes, it has to be adapted or even replaced by another model. Flexibility is a key issue in the development of future production lines.

Finally, all these problems have to be solved with a new type of conveyor while keeping the most important feature, that is speed of conveyance. For instance, existing production units of tablets manufacture 1000 to 5000 tablets per minute. They therefore must be conveyed at speeds from 0.2 to 1 m/s. This speed of conveyance has an important impact on the whole architecture as conveying is not the only task to be performed.

B. State of the art

There have been numerous projects of MEMS actuator arrays in the past and more precisely in the 1990's. This pioneering research has developed different types of MEMS actuator arrays, based on actuators either pneumatic and

[1]FEMTO-ST Institute DISC Department, Université de Franche-Comté/CNRS/ENSMM/UTBM, 16 route de Gray 25000 Besançon, France jmgauthi@femto-st.fr, ahmed.hammad@femto-st.fr, fabrice.bouquet@femto-st.fr

[2]FEMTO-ST Institute AS2M Department, Université de Franche-Comté/CNRS/ENSMM, 24 rue Savary 25000 Besançon, France dominique.gendreau@femto-st.fr

electrostatic [3], magnetic [4] or thermobimorph and electrostatic actuators [5]. Some of these preliminary studies use a sensor-less manipulation scheme based on the Goldbergs algorithm [6] for parallel jaw grippers. Bohringer et al. [7] have extended these works by opposing the directions of MEMS ciliary actuators but the absence of a command law can lead to uncertain behaviors [8] and MEMS actuator arrays has to be programmed for each different kind of parts. Furthermore, this kind of control is applicable only to ciliary actuators and cannot be used with pneumatic actuators [9].

More recent research has been conducted in order to include sensors and to add intelligence to MEMS actuator arrays. Biegelsen et al. have developed a sensor/actuator arrays which can manipulate paper sheets by using air-jets [10]. Although the integration of actuators, sensors and a FPGA to move each sheet has been developed, the actuators are macro-scale ones and generate air-flow in only one direction. As the actuators cannot move, 4 actuators have to be grouped to blow in the 4 directions and they take a surface of 6.45 cm^2. Moreover, only very light objects with a high surface to weight ratio can be moved. Servoed roller wheels have been used in open-loop control in preliminary work [11], [12] and later [13] with distributed control. As the roller wheels are of macro-size, the target of this actuator array is large objects.

Concerning the reconfigurable systems, several laboratories have initiated research programs in this domain [14], [15], [16], [17], [18]. Among the most advanced works and the closest to our project, we can mention the reconfigurable modular robots [19], [20], [21], [22], [23]. These robots consist of a set of modules that are able to organize themselves to accomplish a common task. Beyond conventional actuation, sensing and control typically found in fixed-morphology robots, self-reconfiguring robots are also able to deliberately change their own shape by rearranging the connectivity of their parts, in order to adapt to new circumstances, perform new tasks, or recover from damage.

Concerning the distributed intelligent MEMS, in 1997, following a DARPA ISAT working group on distributed MEMS, Berlin and Gabriel published a paper [24] which defined the new challenges raised by distributed MEMS. They focused on computer science and networking problems and they chose MEMS dust clouds as the possible application. Even though the application of MEMS dust was not successful, the challenges they define have yet to be completely overcome.

Claytronics [25], [26], [27] is a project at Carnegie Mellon University to develop programmable matterbringing the power of programming to physical matter. A Claytronics system consists of millions of tiny computing units called catoms. Each catom is capable of executing code, sensing and communicating with nearby catoms, and moving around its neighbours subject to the laws of physics. The result is an ensemble of particles which can change their physical properties (e.g., shape, color, etc.) under program control. Two programming languages, LDP [28] and Meld [29], have been developed which have shown some simple yet useful behaviours [30]. Each language excelled at some tasks, but not on others like expressivity. The hardware part consists of 1mm diameter cylinders that can move by themselves which requires MEMS fabrication and MEMS/CMOS integration. This is the most advanced research work in this topic.

Programmable matter is also the objective reached by the DRL team in the Pebbles Robot project [31]. The current state lies in the design of 1cm cubes that stick together with electromagnetic force and are capable of executing instructions. Rather than working on building a 3D shape by assembling cubes, they work on the disassembly process much like sculpture, which means starting from an already assembled Pebbles and removing cubes to obtain the final shape. The advantage of this approach is that assembly forces can be obtained by low-power electropermanent magnets. The Pebbles project has shown that intelligent 1 cm-side cubes are a possible target even though they have not yet reached the size of MEMS elements. The programming paradigm has not been defined yet and only low-level functionalities (raw communication capabilities and latching) have been published.

The main goal of modeling and simulation [32] of air-jets is to detect the object and to apply immediately the air-jets forces to have the best answer of push.

The existing environments used for the description and the simulation of this kind of systems are:

- VHDL-AMS [33], which allows the modeling and the simulation of circuits and logical, analog and mixed systems. It permits to model abstracted objects treating signals quantified at discreet time.
- The language MAST [34] is a proprietary language developed in 1984 and completely connected to the simulator SABER. MAST is a hardware description language with signals and mixed technology.
- Modelica is an object-oriented modeling language that allows the modeling of physical complex and heterogeneous systems. It can be considered as a multi-disciplinary modeling language. The Modelica solvers contain very effective algorithms for solving equation systems and allow the handling of complex models that are described by thousands of equations.
- The tool SIMULINK [35] allows to model, to simulate and to analyze dynamic systems multi-domains. This tool takes advantage of the digital computing power offered by the environment MATLAB [36].

III. Modeling and Validation

The target of this work is to define a methodological approach with tools based on the SysML language. This approach will allow us to model, simulate and verify functional and non-functional properties of the smart blocks system before its conception.

In order to deal with heterogeneity and complexity of smart block, we need tools to model, simulate and verify some properties on the system before its conception. In the former project (Smart Surface), SysML language was used to model the system and VHDL-AMS was used to simulate it [37].

A. Methodology

Modeling is at the core of any design process. A model is a representation of a system to a selected level of abstraction. One purpose of a model is to enable the analysts to predict the behaviours of the system.

On the one hand, a model should be a close approximation to the real system and integrate most of its features. On the other hand, it should not be too much complex so that it can be understood and it can be at the starting point of experiments. Models may describe the behavior and/or the structure of the designed system, its may be used to validate characteristics of some part or of the whole of the designed system (functionality or its performances). An important issue in modeling is model validation. Model validation techniques include simulating the model under known input conditions and comparing model output with system output. Generally, a model intended for a simulation is a mathematical model developed with the help of simulation software.

The combined use of SysML and Modelica is a way to satisfy the requirements of the specification. SysML is a UML (Unified Modeling Language)profile that can be used to specify graphically all aspects of complex systems. SysML is a modeling language that permits to easily obtain a specification of a complex system including structural and behavioural aspects. Modelica is an object-oriented modeling language that allows the modeling of physical systems which can be complex and heterogeneous. It can be considered as a multidisciplinary modeling language. The Modelica models are described mathematically through differential equations, algebraic equations and discrete equations. Thus, the first step is to give a mathematical representation of the system.

B. Mathematical modeling

To model the behaviour of the object subjected to the propulsion of several air-jets, we consider the elementary force of one air-jet in two dimensions. Indeed, we consider the following hypotheses:

- levitation is provided by the air jets below the object but we don't consider their effect, *i.e.* they are not evaluated,
- air-jets are independent, *i.e.* there aren't any interactions between air-jets.

Then, we make the sum of each air-jets with their degree of influence. That influence, depends on the position of the air-jet in regard to the position and the distance with the object. Overall, the object is subjected to two forces: a driving force and an opposition force of displacement (air friction). Their scope is represented in the Figure 1.

$$\sum \overrightarrow{F} = \overrightarrow{F_d} + \overrightarrow{F_v} = m \cdot \ddot{x} \cdot \overrightarrow{u_n} \tag{1}$$

m: weight of the object, x: position of the object.

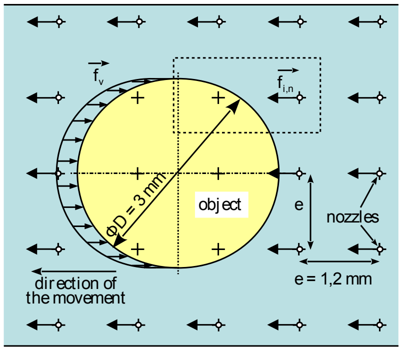


Fig. 1: Object subjected to air-jets

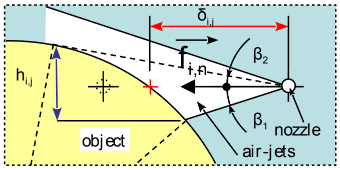


Fig. 2: Model of one air-jet

The driving force $\overrightarrow{F_d}$ results to the sum of air-jet propulsion forces which act near to the object.

$$\overrightarrow{F_d} = \sum_{i=1}^{M} \sum_{n=1}^{N} \Delta i, n \cdot \overrightarrow{f_{i,n}} \tag{2}$$

$\Delta i, n = 1$ if the air-jet reach the object surface.

The elementary force $\overrightarrow{f_{i,n}}$ (Figure 2) induced from each air-jet is determined as follow [38]:

$$\overrightarrow{f_{i,n}} = \frac{1}{2} \rho \cdot C_D \cdot s_{i,n} \cdot v_{i,n}^2 \cdot \overrightarrow{u_n} \tag{3}$$

$\rho = 1,3kg/m^3$ if the air-jet reach the object surface
$C_D = 1,2$ drag coefficient for an half-cylinder
$s_{i,n}$: projected surface in contact of the air-jet
$v_{i,n}$: relative speed of the air-jet, defined as:

$$v_{i,n} = \dot{x} - v_{air}(\delta_{i,n}) \tag{4}$$

v_{air}: absolute speed of the air-jet, at the outlet of the air nozzles, which can take two values.

$$\delta_{i,j} > 0 \Rightarrow v_{air}(\delta_{i,n}) = 5500 \cdot \exp^{-\frac{\delta_{i,j}^2}{4}} \tag{5}$$

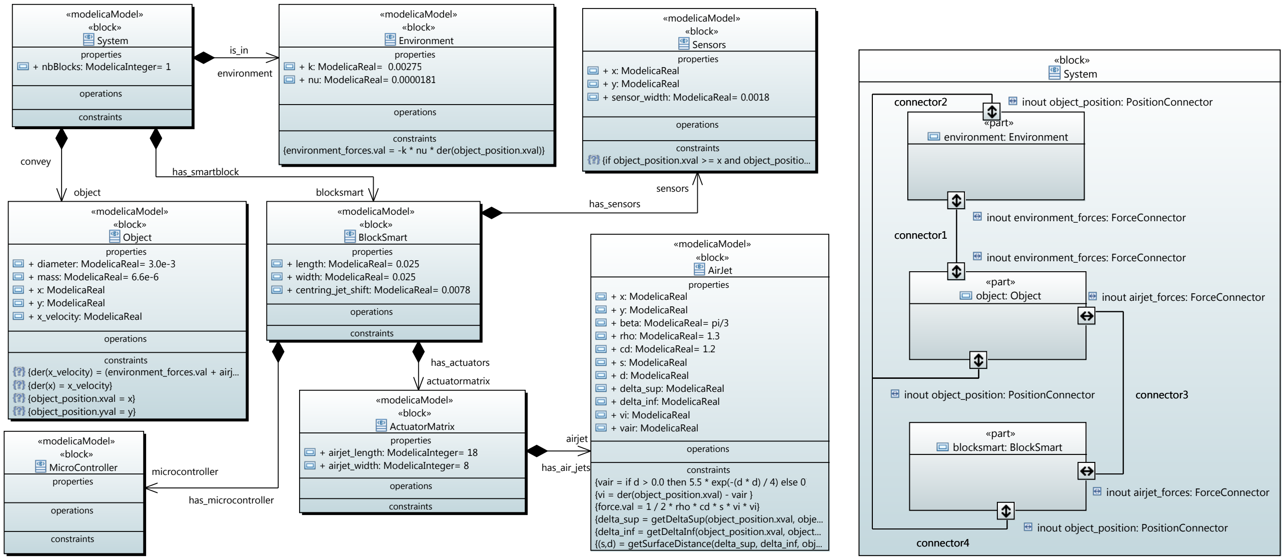
$$\delta_{i,j} = 0 \Rightarrow v_{air}(\delta_{i,n}) = 0 \tag{6}$$

$\delta_{i,j}$: distance between the air nozzle and the contact point of the object.

In opposition to the displacement appear viscous drag forces $\overrightarrow{F_v} = \int_S \overrightarrow{f_v}$ represented as:

$$\overrightarrow{F_v} = -K \cdot \eta \cdot \dot{x} \cdot \overrightarrow{u_n} \tag{7}$$

$K = 2,75mm$: geometric coefficient of the viscosity force
$\eta = 1,81.10^-5$: air viscosity


«modelicaModel»
«block»
System
properties
+ nbBlocks: ModelicaInteger= 1
operations
constraints
is_in
environment
«modelicaModel»
«block»
Environment
properties
+ k: ModelicaReal= 0.00275
+ nu: ModelicaReal= 0.0000181
operations
constraints
{environment_forces.val = -k * nu * der(object_position.xval)}
«modelicaModel»
«block»
Sensors
properties
+ x: ModelicaReal
+ y: ModelicaReal
+ sensor_width: ModelicaReal= 0.0018
operations
constraints
{if object_position.xval >= x and object_positio...
convey
object
has_smartblock
blocksmart
sensors
has_sensors
«modelicaModel»
«block»
Object
properties
+ diameter: ModelicaReal= 3.0e-3
+ mass: ModelicaReal= 6.6e-6
+ x: ModelicaReal
+ y: ModelicaReal
+ x_velocity: ModelicaReal
operations
constraints
{der(x_velocity) = (environment_forces.val + airj...
{der(x) = x_velocity}
{object_position.xval = x}
{object_position.yval = y}
«modelicaModel»
«block»
BlockSmart
properties
+ length: ModelicaReal= 0.025
+ width: ModelicaReal= 0.025
+ centring_jet_shift: ModelicaReal= 0.0078
operations
constraints
has_actuators
actuatormatrix
«modelicaModel»
«block»
AirJet
properties
+ x: ModelicaReal
+ y: ModelicaReal
+ beta: ModelicaReal= pi/3
+ rho: ModelicaReal= 1.3
+ cd: ModelicaReal= 1.2
+ s: ModelicaReal
+ d: ModelicaReal
+ delta_sup: ModelicaReal
+ delta_inf: ModelicaReal
+ vi: ModelicaReal
+ vair: ModelicaReal
operations
constraints
{vair = if d > 0.0 then 5.5 * exp(-(d * d) / 4) else 0
{vi = der(object_position.xval) - vair }
{force.val = 1 / 2 * rho * cd * s * vi * vi}
{delta_sup = getDeltaSup(object_position.xval, obje...
{delta_inf = getDeltaInf(object_position.xval, object...
{(s,d) = getSurfaceDistance(delta_sup, delta_inf, obj...
«modelicaModel»
«block»
MicroController
properties
operations
constraints
microcontroller
has_microcontroller
«modelicaModel»
«block»
ActuatorMatrix
properties
+ airjet_length: ModelicaInteger= 18
+ airjet_width: ModelicaInteger= 8
operations
constraints
airjet
has_air_jets
«block»
System
connector2
inout object_position: PositionConnector
«part»
environment: Environment
inout environment_forces: ForceConnector
connector1
inout environment_forces: ForceConnector
«part»
object: Object
inout airjet_forces: ForceConnector
inout object_position: PositionConnector
connector3
«part»
blocksmart: BlockSmart
inout airjet_forces: ForceConnector
inout object_position: PositionConnector
connector4

(a) BDD of the system

(b) IBD of the system

Fig. 3: SysML model of the system

The dynamic combination of these functions is not easy because the object is always in movement and the speed is not continuous. The trajectory control depends on the position of the object onto each blocks.

This subsection has described the analytical model of an object subjected to air-jets. The next step is to model the system in SysML in order to perform Modelica simulation code generation.

C. System modeling

The first level of SysML modeling is the Block Definition Diagram (BDD). The Figure 3a shows the structural view of the system. It has been made with the modeling software named Papyrus.

The block named *System* is decomposed into three sub-blocks (*Environment*, *Object* and *BlockSmart*). The block named *Object* represents an object to convey. The block named *BlockSmart* represents conveying blocks that are composed of an actuator matrix, a sensor and a micro-controller. The block *ActuatorMatrix* is composed of air-jets. These air-jets are at the core of our study. Finally, the block *Environment* represents the ambient air which imposes a frictional force $\overrightarrow{F_v}$ to the object. In addition, each block of the BDD contains the equations from the mathematical model. These equations are Modelica formed to allow their interpretation and resolution at each time step.

The second level of SysML modeling is the Internal Block Diagram (IBD). The Figure 3b shows the internal view of the system. This view permits to represent the interactions between components which compose the system, to say forces and position of the object.

As the object is subjected to two main forces $\overrightarrow{F_v}$ and $\overrightarrow{F_d}$, it's block is linked with the block *Environment* and *BlockSmart*. Moreover, each conveyor block has to know the position of the object in order to calculate the force of the air-jets. This is why there is a link typed with *PositionConnector* between the object and the conveyor.

This subsection has described the model of the Smart Block at a system level. The last step of our work is to simulate the model in order to validate it and to study the influence of the air-jets on the object. We have developed a plugin for Papyrus which can automatically translate the SysML model into Modelica code. The process of code generation is out of the scope of this paper. Therefore, we present simulation results directly from the execution of the model in the OpenModelica environment.

D. Simulation

Simulations were performed under the following initial conditions:

- mass of the object : 6.6e-6 kg
- size of the object: 0.003 m in diameter
- initial velocity of the object: 0.0 m/s

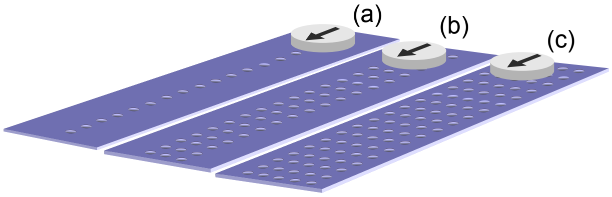
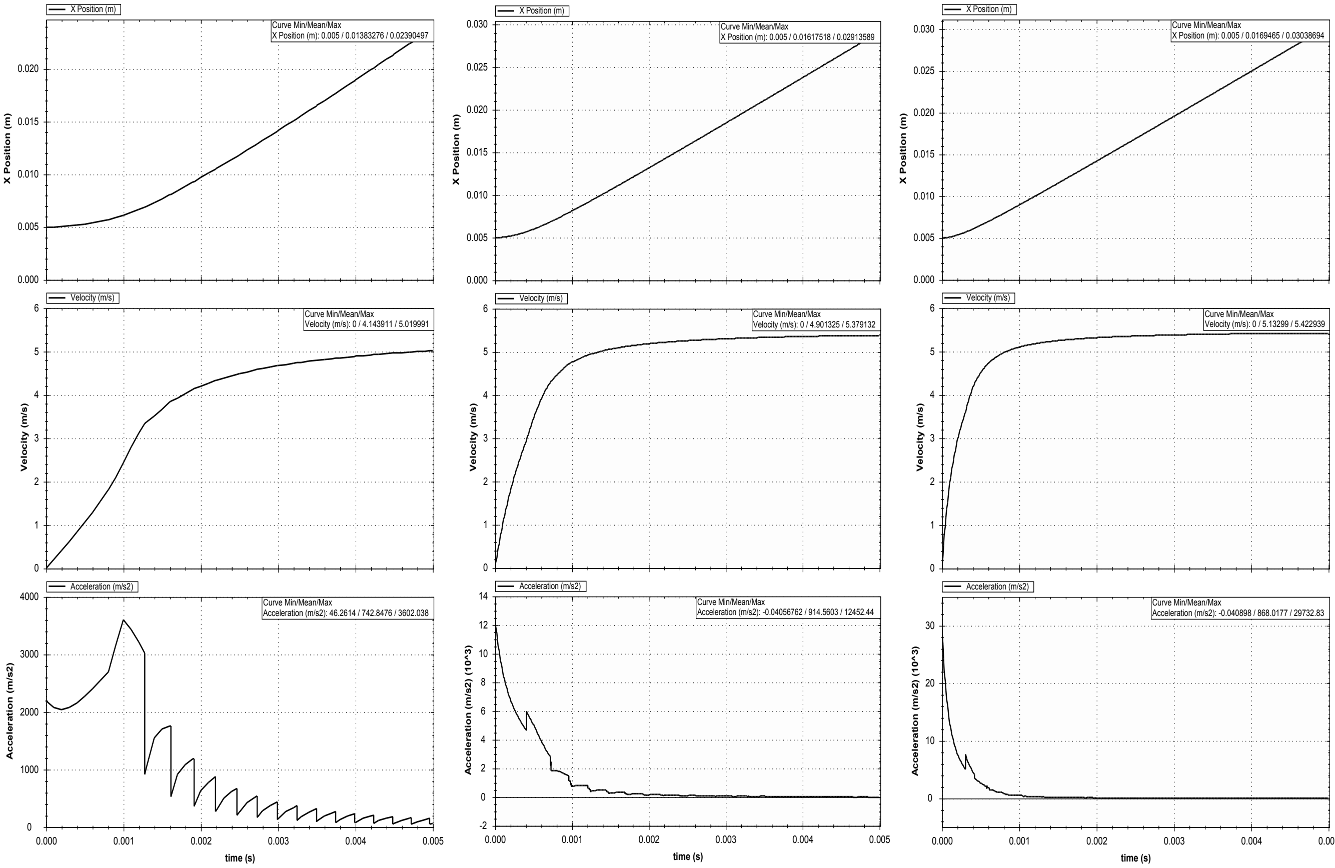


Fig. 4: Simulation scenarios

As a reminder, a block is composed of a conveying surface (named *ActuatorMatrix* in the SysML model). This surface is a matrix of 18 air-jets long and 8 air-jets width. We have simulated three scenarios to understand the air-jets influence on the object. First, the object is subjected to one line of



(a) One line of 18 air-jets

(b) Three lines of 18 air-jets

(c) Five lines of 18 air-jets

Fig. 5: Modelica simulation results

air-jets, then to three lines and finally to five lines of air-jets. For each simulations, the object is centered on the surface. These scenarios, which are illustrated on the Figure 4, allow to validate the model in regards of the expected behaviors of the object. Therefore, there is no need right now to simulate bigger scenarios because we need a correct air-jet model that could be a good starting point for more complicated scenarios.

E. Results and discussion

As shown on the Figure 5, the simulation results focus on the position, the velocity and the acceleration of the object. While the object is moving, it successively losts and gains driving forces from the air-jets.

On simulation results 5a, only one or two air-jets are influential at a time. This results to a 100% variation of forces. That's why we can observe jerks on the acceleration curve. On simulation results 5b, a curve smoothing is observed. Indeed, there are successively 3 to 4 and 4 to 6 influential air-jets at a time, which implies a 50% variation of forces.

Concerning the last scenario, which results are illustrated on the Figure 5c, we can observe a more important smoothing effect on the acceleration curve. We have also simulated an object subjected to 8 lines of air-jets. But, as the model takes into account the distance between the object and the air-jets, only the nearest air-jets influence the x position of the object. Therefore, the simulation results are identical for five or more lines of air-jets.

Despite strong starting assumptions, especially the absence of influence of an air-jet on the others, we consider that the model is a good starting point for other scenarios. For instance, we started to explore the braking capabilities of a smart block positioned in the opposite direction to the object's trajectory. We have observed that only few air-jets are sufficient to slow down the object efficiently.

IV. Conclusion and future work

In this paper, the integration of quantitative analysis in a model-based system engineering approach using SysML and Modelica is addressed. We have first defined mathematically the behavior of an object subjected to air-jets forces. Then, we have presented the efforts to combine the SysML modeling and the Modelica simulation. The obtained results shows that the model is consistency in regards of the expected behavior of a small object subjected to air-jets. Therefore, the mathematical representation under the starting hypothesis and the resulting model are realistic enough. These results can be extrapolated for other blocks in order to simulate a complete smart blocks track. Moreover, we have now an idea of the complexity of this kind of systems: an increase of the amount of blocks and of air-jets implies a longer and more complex simulation.

Our future work is to continue the exploration of the braking issue of an object. This issue is important for two main requirements. The first functional requirement is to stop the object to its arrival position. The second functional requirement is to brake the object in order to change its trajectory. The braking issue raises the question of the speed control of the object. We have to take into account the control module which allows to start and stop air-jets in a continuous way.

V. Acknowledgement

This work has been supported by the Labex ACTION project (contract "ANR-11-LABX-01-01") and by the ANR-2011-BS03-005 Smart Blocks project.

References

[1] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann, 2009, iSBN 9780123743794.

[2] "Modelica Association," http://www.modelica.org/.

[3] K. Pister, R. Fearing, and R. Howe, "A planar air levitated electrostatic actuator system," in *Micro Electro Mechanical Systems, 1990. Proceedings, An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots. IEEE*, Feb 1990, pp. 67–71.

[4] C. Liu, T. Tsao, Y.-C. Tai, W. Liu, P. Will, and C.-M. Ho, "A micromachined permalloy magnetic actuator array for micro robotics assembly systems," in *Solid-State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers' 95. The 8th International Conference on*, vol. 1. IEEE, 1995, pp. 328–331.

[5] J. W. Suh, S. F. Glander, R. B. Darling, C. W. Storment, and G. T. Kovacs, "Organic thermal and electrostatic ciliary microactuator array for object manipulation," *Sensors and Actuators A: Physical*, vol. 58, no. 1, pp. 51 – 60, 1997, ¡ce:title¿Micromechanics Sections of Sensors and Actuators¡/ce:title¿.

[6] K. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 10, no. 2-4, pp. 201–225, 1993.

[7] K.-F. Bhringer, V. Bhatt, B. R. Donald, and K. Goldberg, "Algorithms for sensorless manipulation using a vibrating surface," *Algorithmica*, vol. 26, no. 3-4, pp. 389–429, 2000.

[8] M. Coutinho and P. Will, "A general theory for positioning and orienting 2d polygonal or curved parts using intelligent motion surfaces," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 1, May 1998, pp. 856–862 vol.1.

[9] L. Matignon, G. Laurent, N. Fort-Piat, and Y.-A. Chapuis, "Designing decentralized controllers for distributed-air-jet mems-based micromanipulators by reinforcement learning," *Journal of Intelligent and Robotic Systems*, vol. 59, no. 2, pp. 145–166, 2010.

[10] D. K. Biegelsen, A. A. Berlin, P. Cheung, M. P. Fromherz, D. Goldberg, W. B. Jackson, B. Preas, J. Reich, and L. E. Swartz, "Airjet paper mover: an example of mesoscale mems," vol. 4176, 2000, pp. 122–129.

[11] J. Luntz and W. Messner, "A distributed control system for flexible materials handling," *Control Systems, IEEE*, vol. 17, no. 1, pp. 22–28, Feb 1997.

[12] J. Luntz, W. Messner, and H. Choset, "Parcel manipulation and dynamics with a distributed actuator array: the virtual vehicle," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, Apr 1997, pp. 1541–1546 vol.2.

[13] M. Bedillion and W. Messner, "Control for actuator arrays," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 868–882, 2009.

[14] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. V. Brussel, "Reconfigurable manufacturing systems," {*CIRP*} *Annals - Manufacturing Technology*, vol. 48, no. 2, pp. 527 – 540, 1999.

[15] A. Yigit, A. Ulsoy, and A. Allahverdi, "Optimizing modular product design for reconfigurable manufacturing," *Journal of Intelligent Manufacturing*, vol. 13, no. 4, pp. 309–316, 2002.

[16] *Reconfigurable Manufacturing Systems and Transformable Factories*. Springer, 2006.

[17] "Advances in the tut microfactory concept development," *International Journal of Automation Technology*, vol. 4, no. 2, pp. 117 – 126, 2010.

[18] T. Gaugel, J. Kegeler, and G. Munz, "Miniaturized reconfigurable micro-assembly system," in *Microfactory for Precision Assembly Applications, International Precision Assembly Seminar, Bad Hofgasfeln, Austria*, 2004.

[19] J. S. Dai, M. Zoppi, and X. Kong, *Advances in Reconfigurable Mechanisms and Robots I*. Springer, 2012.

[20] S. Murata and H. Kurokawa, *Self-organizing robots*. Springer, 2012.

[21] D. Duff, M. Yim, and K. Roufas, "Evolution of polybot: A modular reconfigurable robot," in *Proc. of the Harmonic Drive Intl. Symposium, Nagano, Japan*. Citeseer, 2001.

[22] M. Eng, K. Camarata, E. Y.-L. Do, and M. D. Gross, "Flexm: Designing a physical construction kit for 3d modeling," *International Journal of Architectural Computing*, vol. 4, no. 2, pp. 27–47, 2006.

[23] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 43–52, 2007.

[24] A. A. Berlin and K. J. Gabriel, "Distributed mems: New challenges for computation," *Computational Science & Engineering, IEEE*, vol. 4, no. 1, pp. 12–16, 1997.

[25] S. C. Goldstein, J. D. Campbell, and T. C. Mowry, "Programmable matter," *Computer*, vol. 38, no. 6, pp. 99–101, 2005.

[26] S. C. Goldstein and T. C. Mowry, "Claytronics: A scalable basis for future robots," 2004.

[27] S. C. Goldstein, T. C. Mowry, J. D. Campbell, M. P. Ashley-Rollman, M. De Rosa, S. Funiak, J. F. Hoburg, M. E. Karagozler, B. Kirby, P. Lee, *et al.*, "Beyond audio and video: Using claytronics to enable pario," *AI Magazine*, vol. 30, no. 2, p. 29, 2009.

[28] D. J. Dewey, M. P. Ashley-Rollman, M. De Rosa, S. C. Goldstein, T. C. Mowry, S. S. Srinivasa, P. Pillai, and J. Campbell, "Generalizing metamodules to simplify planning in modular robotic systems," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 1338–1345.

[29] R. Ravichandran, G. Gordon, and S. C. Goldstein, "A scalable distributed algorithm for shape transformation in multi-robot systems," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 4188–4193.

[30] S. Funiak, P. Pillai, M. P. Ashley-Rollman, J. D. Campbell, and S. C. Goldstein, "Distributed localization of modular robot ensembles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 946–961, 2009.

[31] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2485–2492.

[32] L. Zhou, Y. Herve, Y. Chapuis, and H. Fujita, "Vhdl-ams modeling for simulation of mems array-based smart surface applied in microfluid air-flow environment," in *Sensors, 2006. 5th IEEE Conference on*, Oct 2006, pp. 807–810.

[33] E. Christen and K. Bakalar, "Vhdl-ams-a hardware description language for analog and mixed-signal applications," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 46, no. 10, pp. 1263–1272, Oct 1999.

[34] "Saber," 2014, http://www.synopsys.com/products/mixedsignal/saber/saber.html,.

[35] "Simulink," 2014, http://www.mathworks.fr/products/simulink/,.

[36] "Matlab," 2014, http://www.mathworks.fr/products/matlab/,.

[37] J.-M. Gauthier, F. Bouquet, A. Hammad, and F. Peureux, "Transformation of SysML structure diagrams to VHDL-AMS," in *IEEE Workshop on design, control and software implementation for distributed MEMS (dMEMS'12)*. Besançon, France: IEEE CPS, 2012.

[38] L. Matignon, G. J. Laurent, N. Le Fort-Piat, and Y.-A. Chapuis, "Designing decentralized controllers for distributed-air-jet mems-based micromanipulators by reinforcement learning," *Journal of intelligent & robotic systems*, vol. 59, no. 2, pp. 145–166, 2010.