II International Conference on Simulation for Additive Manufacturing - Sim-AM 2019
F. Auricchio, E. Rank, P. Steinmann, S. Kollmannsberger and S. Morganti (Eds)

SIMULATION OF A ROBOTIC ARM FOR MULTI-DIRECTIONAL 3D PRINTING

Kevin Castelli[*], **Hermes Giberti** [†]

[*] Dipartimento di Ingegneria Industriale e dell'Informazione
Università di Pavia
via Ferrata 5, 27100, Pavia, Italy
email: kevin.castelli@unipv.it

[†] Dipartimento di Ingegneria Industriale e dell'Informazione
Università di Pavia
via Ferrata 5, 27100, Pavia, Italy
email: hermes.giberti@unipv.it

Key words: Serial Manipulator, Simulation, Multi-Direction 3D Printing, FDM

Abstract. The need to investigate new solutions and novel 3D building strategies not only requires the development of new slicing algorithms and the exploitation of machines with more than 3 Dofs, but also a safe and reliable test-bench to optimize all the phases of the process. The following article describes the assessment by simulation of suitable control architectures for the realization of a Fused Deposition Modeling printer based on a 6 Dofs serial manipulator. The focus is put on the obtained position and speed profiles for unidirectional and multi-directional 3D printing to determine the week points associated with each control strategy.

1 Introduction

Additive Manufacturing (AM), in the last decades, has aroused great interest among people and researchers thanks to its flexibility and versatility, making it the most widespread solution for prototyping and one-off production. One of the advantages of AM is that it can build complex shapes that with Subtractive Manufacturing would not have been feasible. AM technologies, on the other hand, are a direct evolution of subtractive ones and this can be seen from the exploitation of the same machines, but also of the same machining strategy (layer based approach). This represents a constraint to the potentiality of AM to build 3D object in three dimensional space (multi-directional printing) and not just an approximation by 2D features (unidirectional printing). Therefore, new solutions and novel building strategies are being researched (examples can be found in [1],[2],[3]). Industrial robots have been used to this end. Integration is required to perform 3D printing task, from the outline of the working cell to the definition of the control strategy. The flexibility given by systems with more than 3 Degrees of Freedom (Dofs) increases the complexity of the design of the working cell, of the tool and of the trajectories that prevent the collision with the piece ([4] shows this by adopting service robots instead). Simulation is a safe and indispensable tool to carry out this task.

In [5], the authors report their experience with an analogous machine focusing on a visual feedback of the material deposition in the off-line environment. [6] highlights issues arising when the printing process does not reflect the one prescribed, especially related to the machine itself. The authors of [7], [8] and [9] describe their involvement with conventional and multi-directional printing with industrial robots. [10] is an example of three dimensional tool-path generation.

In this context, the following article describes the assessment of suitable control architectures for the realization of a Fused Deposition Modeling (FDM) printer based on a 6 Dofs industrial serial manipulator. Off-line programming (using the software released by the manufacturer for better compliance with the physical robot system) is preferred to validate the robot behavior in order to avoid wastes and damage to the machine. One additional advantage of the simulation is the possibility to easily exchange virtual subsystem with the real ones to asses them individually.

For each identified control strategy, that easily integrate the extruder with the robotic system, the results obtained are shown for paths generated for conventional as well as for multi-directional printing. The focus is put on the obtained position and speed profiles, two of the several aspects that yield a minimization of the error between designed and manufactured object, but strictly concern the robotic system in itself. Simulation results are then compared to the acquisitions on the physical system.

The following article recalls in Section 2 a core aspect of the AM technology here addressed, that is extrusion flow rate; in Section 3, path planning is deepened to highlight the robotic system requirements and constraints; the extrusion subsystems and robot are then briefly introduced as well as its virtual counterpart (Section 4). In the following section, the different control strategies are discussed and simulated (Section 5-6). In Section 7, the results are summarized.

2 Extrusion flow rate

FDM technology can be described as the controlled deposition of molten material on a substrate. In order to determine how much filament needs to be forward through the nozzle, it's important to understand how the extruded material will be added to the existing substrate. The nozzle will be spaced of a quantity h from the interested surface and some material will be squirted out taking on a certain shape, usually approximated by a rounded rectangle (see Fig.1 where the convention used to determine the transferred section are indicated). This shape can be deterministically obtained (under the assumption that w is smaller than the surface curvature) by solving equation (1) that represents the relationship among the volumetric flow rates at the filament, nozzle and substrate.

$$A_{wire}V_{robot} = \frac{\pi d_n^2}{4}V_{extrusion} = \frac{\pi d_f^2}{4}V_{feed} \tag{1}$$

The extruded material could be not only constrained frontally, but also on one or both sides. The distance between two adjacent wires is a parameter used moreover to control the density of the build.

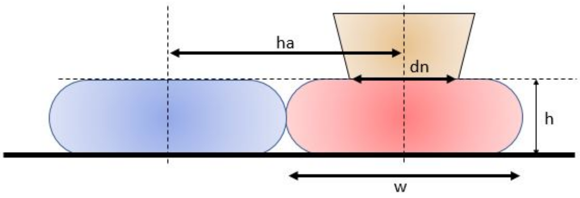


Figure 1: Material deposition. Nomenclature: nozzle diameter (*dn*), layer height (*h*), width (*w*) and hatch distance (*ha*)

Thus, a FDM system has to guarantee the designed robot-filament speed ratio as well as the relative positioning.

3 Path planning

In the previous section, some tool path constrains have been derived related to the process itself. An additional requirement, self-explanatory but not trivial, is that the resulting trajectory reconstructs the 3D CAD model the operator needs to build, not only under a geometrical, but also mechanical point of view. The trace that the TCP (Tool Center Point) has to follow can be determined in several ways.

Conventional 3D printing strategies only require to evaluate the position of the nozzle outlet, approximated as a point, in $\mathbb{R}^3$. Available software return the intersection (contours) of the mesh (stored in the STL file) with planes parallel to the printing bed (transformation to $\mathbb{R}^2$) accounting for the infill or the need for support material.

In the case of multi-direction building strategy, this approach undergoes some modification to compute the generic plane-plane intersection or the recognition of unidirectional blocks to recycle the standard algorithms. Fig.2 shows a robot arm during the building process of a circular rib on a cylindrical surface.

A third approach takes advantage of functions that map a complete 3D problem (thus including the orientation) to a mono-directional one in order to use commercial slicing software before reverting to the initial space ([11]). In Fig.3 an example is reported where geometrical consideration (*A*) are used to determine the dimension of the CAD model that later undergoes slicing (*B*). The tool path is mapped in the required domain and without loss of information (*C*).

Figure 2: Circular rib

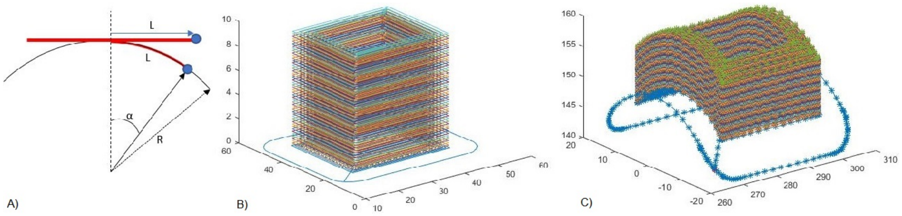


Figure 3: Multi-direction printing by slicing after a $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ map

4 Physical and offline systems

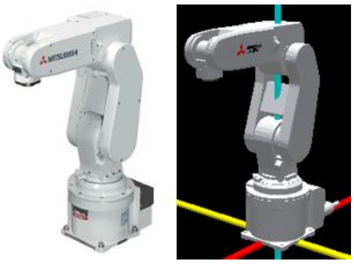


Figure 4: Real system vs virtual twin (*RT ToolBox3*)

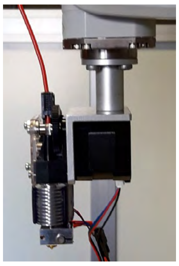


Figure 5: Extrusion system

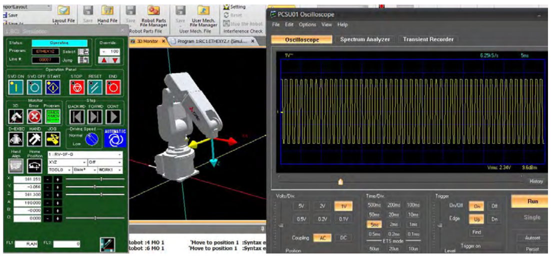


Figure 6: *Virtual robot - Real Extruder*: the PWM is read by means of an oscilloscope

The overall machine is constituted by two subsystems, an industrial 6 DoFs robot (Mitsubishi RV-2F-Q, Fig.4) with its control unit and a commercial FDM extruder controlled by means of an Arduino Mega with Ethernet connection and custom made firmware (Fig.5). The robot moves and orients the nozzle to comply with the requirements introduced heretofore.
The extruder accepts instructions to set the heat block temperature and the motor/fan speeds and to inquire the current temperature. The firmware is coded to be able to comply any new instruction as soon as it is read and processed.
The robot is replicated in the manufacturer software (*RT Tool-Box3*) that offers offline all the functionalities that the real system posses (the one that are going to be extensively used are TCP/IP communication, motion planner and oscillograph) because when recreating the digital twin it is not only important to match the mechanical aspects, but especially those concerning movements and communication.
Here the possible combination between physical and simulated systems are listed pointing out some remarks. The dissertation will afterwards vert solely on the robot.

- *Virtual robot - Virtual Extruder:* Consider each system on its own. Used solely during the early stage of the assessment of the robotic system. The offline extruder requires a code capable of reproducing the reading and the generation of the output signals (pulse-width modulation, PWM, of the stepper motor driver and of the MOSFET).

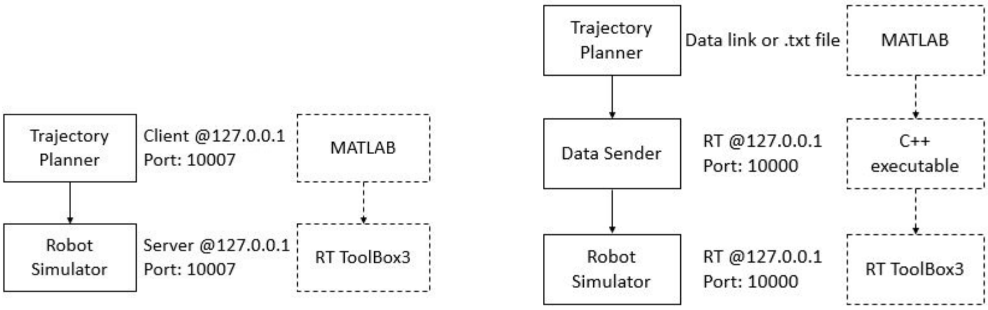
- *Real robot - Virtual Extruder:* This case could be used when the robot is reading the G-code and mastering the extruder. It's advised during the first robot tests to avoid that the heated element fortuitously collides with any object present in the cell.

- *Virtual robot - Real Extruder:* This combination can be used to validate the extrusion system during the print while keeping it still (see Fig.6).

- *Real robot - Real Extruder:* the system is printing.

5 Control architectures

Industrial robots offer several means to communicate with the external world: I/O modules, serial and Ethernet ports. It has been decided for the two subsystems to talk via Ethernet to simplify future addition of devices in the network. Thus, a LAN is designed with the robot as server and the extruder as client (with the possibility to be converted into a server).
Two possible scenarios therefore arise. The first (*DL*, *Data Link*) sees the robotic system as any CNC machine, to which is feed a series of target positions leaving the motion law assignment to the built-in functions. The second (*RT*, *Real Time*) relies on real-time protocols to master the actual motion profile, therefore it should make sure that the synchronization of the extruder to the TCP speed is obtained as designed.
Both solutions requires an extra device (although the *DL* case might be adjusted to read a G-code file on the hard disk of the controller) able to guarantee the time interval between two instructions (down to milliseconds) and to process the trajectory. Therefore, a PC is set up to belong to the LAN. On the computer side all the planning is performed ahead (using *MATLAB*) and, once checked and post-processed, the data is send to the robot controller where the control loops are closed.
In order to mimic these configurations offline, the only alterations required concern the LAN setup. The IP addresses identifying the robot ad the computer become one, that is, since all the talking has to happen internally to the PC, *127.0.0.1* (i.e. localhost). The communication is finally established by indicating the port, in both software, according to the scenario selected. This resolves into the following cases.



(a) Data link (b) Real time

Figure 7: Off-line control scenario: data link and real time

5.1 *Data Link*

The data returned by the slicer (G-code) is post-processed and send over Ethernet with TCP/IP protocol. The data packet is constituted by the X-Y-Z coordinates followed by the A-B-C to impose the orientation of the tool and the commanded velocity (feed).
In *MATLAB* the communication is opened using the function *tcpip(IP,port)* (*tcpip('127.0.0.1', 10007);*) and by setting the terminator and the timeout. Analogously, the extruder could be

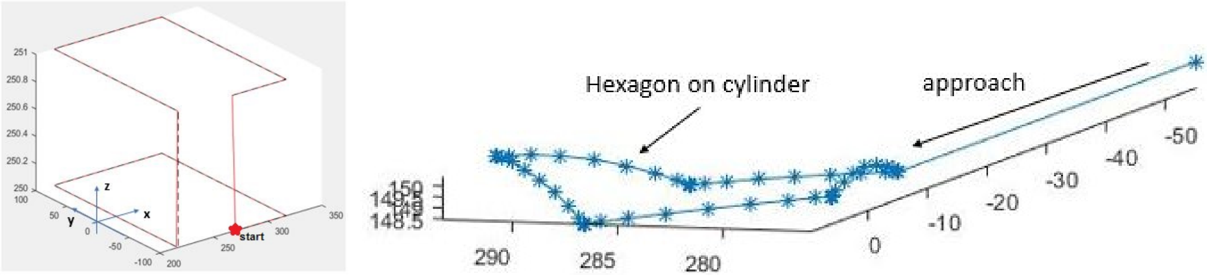
included in the simulation.
In *RT ToolBox3*, the *IP, Mode, Protocol* and *Packet type* parameters need to be customized to match the ones demanded by the network.

5.2 *Real Time*

The robot controller can retrieve the reference position at real-time (in cycle units of 7.11 [ms]) and execute it. To operate in this mode, it is required the adoption of a specific UDP packet, determined by manufacturer. The size of the data package is fixed to 196 byte, either for monitoring or instructing and ordering method is little-endian.
In this configuration an executable (*.exe*, written in c++) is used to generate the data packet according to the prescribed structure and to deal with the transmission and reception of the information. The trajectory, sampled every 7.11[ms] and stored in a text file (although it could be streamed via localhost as well), is read one line at a time at every cycle.
In *RT ToolBox3*, the default port for real time is the *10000*, so only the network IP has to be set.

6 Simulation

In this section, the simulation conducted are going to be deepened for both a unidirectional case as well for a multi-direction build. The *DL* is configured to transit from one instruction to the other using a 1[mm] radius circumference.



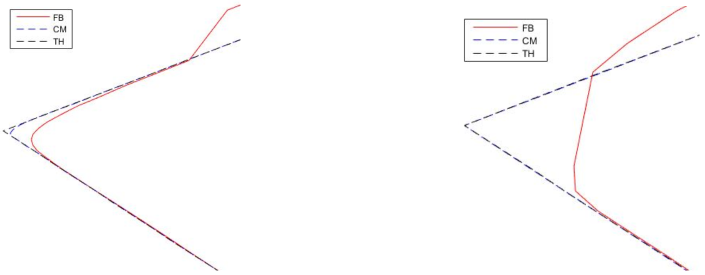
(a) Unidirectional trajectory (b) Multi-directional trajectory

Figure 8: Representation of the test cases trajectories

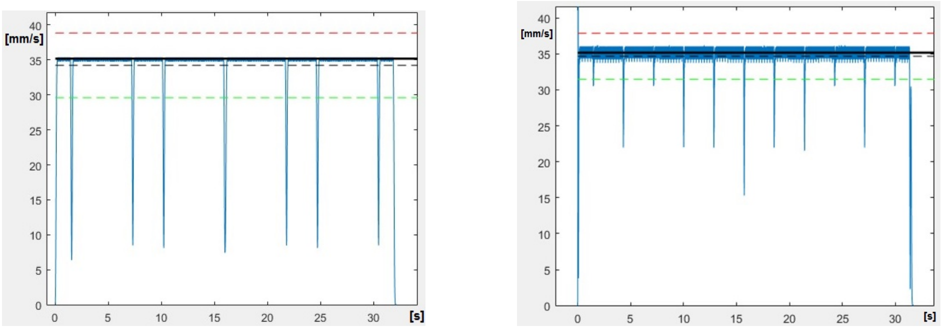
6.1 Unidirectional printing (in $\mathbb{R}^2$)

To simulate the behavior of the AM machine with this particular built philosophy, a simple two layer square is chosen (see Fig.8(a)) with 35.16 [mm/s] feed.
In Fig.9(a), it's possible to see a corner using the *DL* approach; in Fig.9(b), the same feature, but with the *RT* strategy. What can be noticed is that even RT give rise to a smooth corner despite the fact that was not instructed, thus a motion planner is still applied to the data via UDP. In both cases a positioning error is present.
In Fig.9(c) and 9(d), the respective speed module profiles (evaluated along the path), obtained by numerical derivation can be seen. The *DL* shows less bounded fluctuation ($\pm 1[mm/s]$) around

the feed then the *RT*, although it tends to approach zero speed during the path more markedly.



(a) *DL*: position detail (b) *RT*: position detail



(c) *DL*: speed (d) *RT*: speed

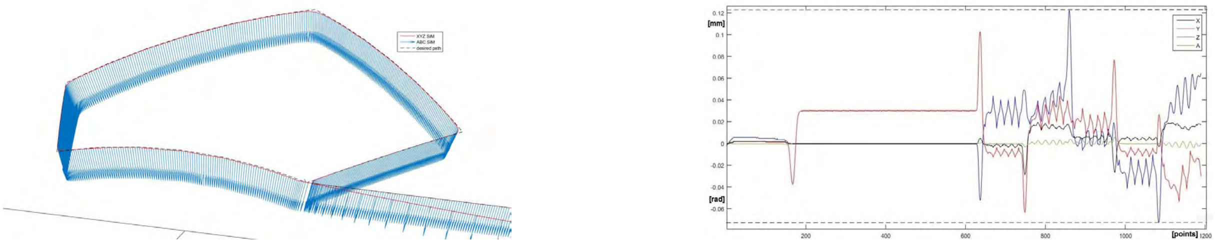
Figure 9: Unidirectional printing simulation: *DL* vs *RT*

6.2 Multi-directional printing (in $\mathbb{R}^3$)

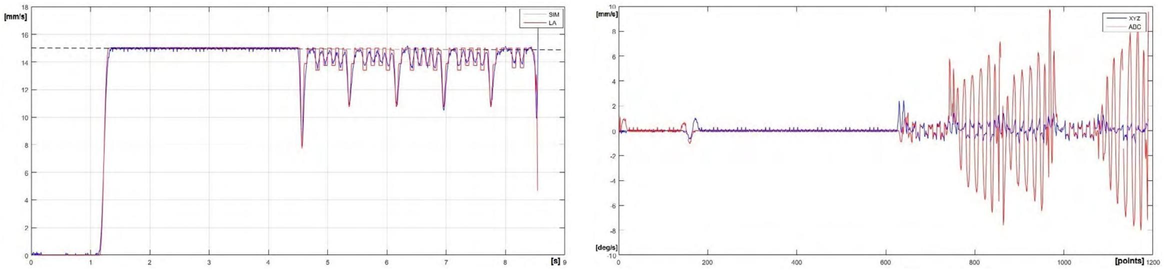
The trajectory, that can be seen in Fig.8(b), has to be followed at 15 [mm/s]. The path is obtained by slicing an hexagon by a cylindrical surface (2nd approach of the section on path planning).

6.2.1 Real Time case

For the *RT* case, the motion law is designed offline by means of a look-ahead algorithm and discretized with a 7.11 [ms] interval. The robot is moved imposing the reference at the TCP. In Fig.10(a) the position is depicted while in Fig.10(b) the positioning errors (only the rotation around the X-axis (*A*) are shown since the cylindrical surface is oriented in that direction, therefore any normal vector on the surface will intersect the X-axis). In Fig.10(c)-10(d), the speed profile simulated is superimposed to the theoretical one and the error profiles are shown, as previously the robot controller performed a smoothing operation on the commanded points. Running the *RT* example on the real system, outputted similar results to the one simulated, but with the introduction of terms unmanageable by the simulator, strictly related to the physical system considered (Fig.11).

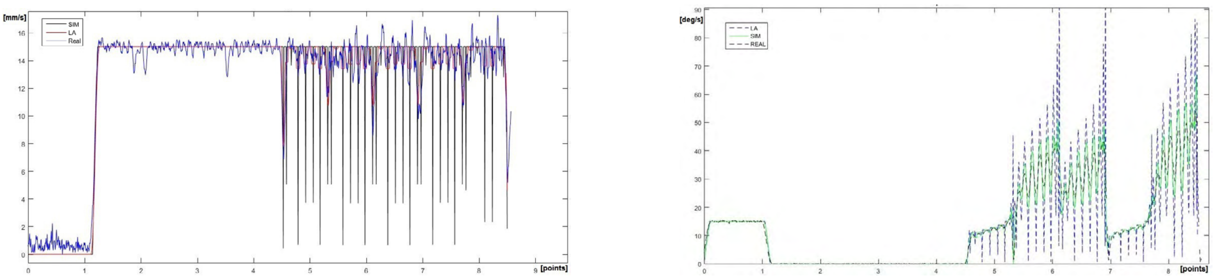


(a) Simulated curve (b) Coordinates error



(c) XYZ resulting speed (d) Speed error

Figure 10: RT: simulation

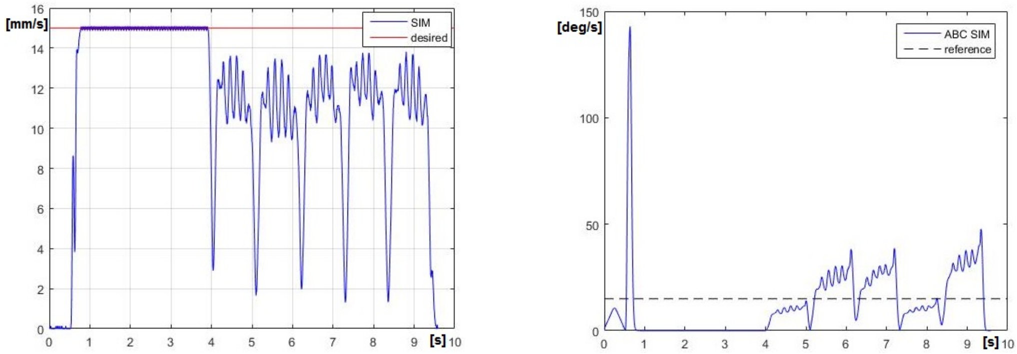


(a) XYZ resulting speed (b) ABC speed

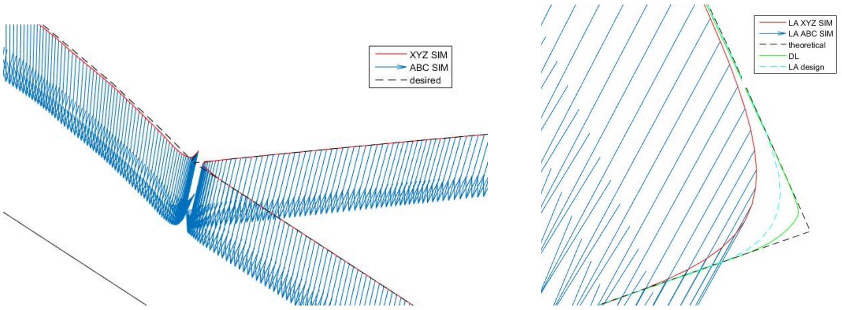
Figure 11: RT: real

6.2.2 Data Link case

Fig.12(a)-12(b) report the speed for the simulation of this typology. What can be inferred is that the system is not capable of guaranteeing the feed speed during the build on the cylinder giving the sampling distance chosen. One possible solution could be to let the system know more than one point at a time, therefore developing a more sophisticated mean to synchronize the speeds. An other possible fix could be to use the simulation itself to predict the correct extrusion speed. At the end in Fig.12(c), it possible to inspect the deviation from the one desired.



(a) XYZ resulting speed (b) ABC resulting speed



(c) Simulated curve (d) Trajectory commands vs simulation

Figure 12: DL: simulation

6.2.3 Comparison

In Fig.12(d), the difference among the theoretical curve, the one evaluated with the look ahead algorithm and the results of the simulations is portrayed. A corner is enlarged because it's the most delicate situation since it has to be assured that the robot arm moves smoothly without under or over extrusion. The *DL* method is less affected by speed fluctuations when performing unidirectional printing. The *RT* is able to follow more closely the speed profile designed along the path while reorienting the tool.

7 Conclusions

This paper describes the assessment of two control strategy for a 6 Dofs robotic arm equipped to become a FDM printer. A LAN is created to connect the robot, the extruder and in case a PC that enables to master the system using real time control or via Ethernet communication.
Off-line programming is chosen to inquire the robot behavior in order to avoid wastes or damage to the machine. The simulation accounts for the dynamic behavior of the robot as well as for the communication with the external world.
Obtained position and speed profiles are used to spot strengths and weaknesses of each method, having identified them as principle factors, on the robot side, that affect the final result. From the results reported, it's possible to comprehend how spacial and time discretization affects the

speed profile and how the *DL* method could be preferred over the *RT* in the case of unidirectional print and vice versa.
The implementation of an hybrid solution might lead to an optimal result and moreover it will require to instruct less points during the execution of straight lines, that fall into the *DL* category, than a pure *Real Time* control strategy.

REFERENCES

[1] Stavropoulos, P., Foteinopoulos, P., Papacharalampopoulos, A. Bikas, H., *Addressing the challenges for the industrial application of additive manufacturing: Towards a hybrid solution*. International Journal of Lightweight Materials and Manufacture (2018).

[2] Castelli, K., Giberti, H., *A Preliminary 6 Dofs Robot Based Setup for Fused Deposition Modeling*. Advances in Italian Mechanism Science. IFToMM ITALY 2018. (2019).

[3] Giberti, H., Sbaglia, L., Silvestri, M., *Mechatronic design for an extrusion-based additive manufacturing machine*. Machines 5(4), art. no. 29. (2017).

[4] McPherson, J., and Zhou, W., *A chunk-based slicer for cooperative 3D printing*. Rapid Prototyping Journal, Vol. 24 Issue: 9. (2018)

[5] Zhang, G.Q., Spaak, A., Martinez, C., Lasko, D.T., Zhang, B., Fuhlbrigge, T.A., *Robotic Additive Manufacturing Process Simulation towards Design and Analysis with Building Parameter in Consideration*. IEEE International Conference on Automation Science and Engineering (CASE). (2016).

[6] Evjemo, L.D. et al., *Additive manufacturing by robot manipulator: An overview of the state-of-the-art and proof-of-concept results*. 22nd IEEE International Conference on Emerging Technologies and Factory Automation (2017).

[7] Pollák, M., Török, J., Zajac, J., Kočiško, M. and Telišková, M., *The structural design of 3D print head and execution of printing via the robotic arm ABB IRB 140*. 5th International Conference on Industrial Engineering and Applications (ICIEA). (2018).

[8] Kubalak, J.R., Wicks, A.L., Williams, C.B., *Exploring multi-axis material extrusion additive manufacturing for improving mechanical properties of printed parts*. Rapid Prototyping Journal (2018).

[9] Dine, A., Vosniakos, G.-C., *On the development of a robot-operated 3D-printer*. Procedia Manufacturing, Volume 17. (2018).

[10] Isa, M.A., Lazoglu, I., *Five-axis additive manufacturing of freeform models through buildup of transition layers*. Journal of Manufacturing Systems, Volume 50. (2019).

[11] Zhao, G., Ma, G., Feng, J. et al., *Nonplanar slicing and path generation methods for robotic additive manufacturing*. Int. J. Adv. Manuf. Technol. (2018) 96: 3149.