A machine state-based Digital Twin development methodology

Negri E.*, Assiro G.*, Caioli L.*, Fumagalli L.*

** Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Piazza L. Da Vinci 32, 20133 Milan – Italy (elisa.negri@polimi, giammarco.assiro@polimi.it, lorenzo.caioli@polimi.it, luca1.fumagalli@polimi.it)*

Abstract: Industrial automation, supported by the introduction of the concept of Industry 4.0 (I4.0) has become a driving force in production systems. The introduction of I4.0 technologies, services and architectures offer new perspectives for automated and flexible production systems. Within this new industrial revolution, the concept of smart manufacturing represents a new way of managing and controlling production systems. The main enabling technologies belonging to this new trend, such as Big Data, Cloud Computing, Internet of Things, find application in the Cyber Physical Systems (CPS), intelligent systems with communication and computing capabilities that connect virtual and real world. A way to exploit this interaction is advanced simulation in the Digital Twin (DT) paradigm. The DT is able to communicate with a real system in real-time, continuously collecting data directly from the shop floor. This continuous interaction with the real system allows to build a simulation model through which it is possible to monitor what happens in real manufacturing systems, enabling a direct interaction between the decision-making process and the real environment. Within the Industry 4.0 Laboratory of the School of Management of Politecnico di Milano these improvements are evaluated. This work proposes a general methodology for the development of DT simulations. This methodology is applied showing its use in monitor the functional behavior of the production system and at the same time evaluating its energy consumption in real-time.

Keywords: Digital Twin, Cyber Physical Systems, Industry 4.0

1. Introduction

The introduction of Information and Communication Technologies (ICT) and the advent of the Industry 4.0 era have deeply changed the interpretation of simulation and its related role in manufacturing. Simulation is seen as the imitation of the real-world process or system over the time (Canedo, 2016). From this perspective, simulation is used as an engineering tool for different purposes, from the early design phases to run time phases of complex systems resulting in higher efficiency, accuracy and economic benefits for companies (Baheti and Gill, 2011; Gabor *et al.*, 2016; Polenghi, Fumagalli and Roda, 2018). Generally speaking, Industry 4.0 provides with its tools new ways of thinking and doing business (Cattaneo *et al.*, 2017). Its related technologies such as Big Data, Cloud Computing, Internet of Things (IoT), Autonomous Robot and sensors (Rüßmann *et al.*, 2015) introduce a new generation of systems with integrated computational and physical capabilities called Cyber Physical Systems (CPSs) that can interact with humans through many new modalities (Baheti and Gill, 2011). Consequently, the introduction of CPSs in manufacturing systems reflects in the generation of a big amount of data (Heymans *et al.*, 2008; Lee, Bagheri and Kao, 2015). The correct management of this data potentially represent a big advantage for companies (Pagoropoulos, Pigosso and McAloone, 2017; Cattaneo *et al.*, 2018) and will reflect in a faster decision-making process and can improve the productivity of systems (Lee, Kao and Yang, 2014; Cao *et al.*, 2019). The continuous improvements of this kind of systems enable the real-time monitoring of physical assets and the synchronization with virtual environments, introducing a new simulation concept: the Digital Twin (DT) (Negri, Fumagalli and Macchi, 2017). The DT can be identified as a simulation technique which differs from the traditional ones as it uses synchronized real data from the shop-floor, in order to get information, such as machine reliability and availability and to elaborate more accurate diagnostics and prognostics that use up-to-date field data, instead of leveraging on estimations by equipment suppliers (Glaessgen and Stargel, 2012). This innovation allows to simultaneously monitor what is happening in the real manufacturing system and permits a synchronization between system and decisions (Macchi *et al.*, 2018). In this new concept of simulation, the model is no longer embedded inside the software system and controlled by a plant agent, but it communicates directly with the real system thanks to the presence of CPS (Gabor *et al.*, 2016). This new vision of the concept of simulation introduced by the DT leads to new production concepts in order to achieve increased competitiveness with respect to the newest trends in production (energy and resource efficiency, shorten time-to-market, enhanced flexibility) (Lee *et al.*, 2013; Negri *et al.*, 2019).

The structure of this paper is the following: Section 2 presents the research objectives; Section 3 introduces the methodology used in order to develop the DT; Section 4 introduces the I4.0Lab environment and shows how the DT has been developed inside it; Section 5 reports the results obtained and Section 6 proposes some conclusions.

2. Research objectives

Starting from the concept of DT presented in the introduction, the aim of this paper is to present a general methodology for the development of a machine states-based DT, through its application on a laboratory assembly line. The proposed DT development allows to replicate the operations of a production system and at the same time to compute its energy consumption over time.

3. Methodology

The main idea at the basis of the proposed DT development is the construction of the virtual copy of the real system, which is able to reproduce different behaviours (machine state, energy consumption, availability etc.) in real-time, in line with previous works as reported by (Negri *et al.*, 2019). From a conceptual point of view, the methodology to construct a machine states-based DT can be decomposed in six different steps (Fig. 1):

1. **Identification of the possible machine states:** the first step is necessary to identify all the possible states that can be assumed by the machines of a given facility. By looking to the sequence of operations performed on a single machine and considering which are the feasible different consumption patterns, it is possible to establish different states. Literature suggests a set of possible machine states, which are: idle, working, failure, setup, slow down, off (Taisch *et al.*, 2013), (Mousavi *et al.*, 2016).
2. **Identification of variables and data sources needed in order to reproduce the states of the stations**: the second step is the identification of the variables and the respective data sources on the production system, such as sensors, actuators and PLCs etc., where the variable values can be obtained from the field, in order to reproduce the machine states identified in the previous step. In fact, IoT technologies applied to production systems open the possibility of collecting real-time data about the system behaviours, extracted directly from the shop floor (Sattar, Anwaruddin and Ali, 2017).
3. **Single Equipment Simulation Model Development:** the third step provides the basis for the real-time connection between real and digital environments. A simulation software has to be selected for the construction of the model of a single Equipment that is than connected to the real system. Once these steps are done for all the equipment of a given facility, is possible to build the final simulation model of the overall system.
4. **Connection of the data sources to the DT simulator**, the synchronization between real world and digital model is granted by a direct connection of the single data sources on the field (i.e. sensors, actuators, PLCs…) and the single equipment simulation model elaborated in step 3: the simulation model is in this way able to replicate the most updated machine states that are actually present in all equipment pieces of the production system (Reifsnider and Majumdar, 2013).
5. **Analysis of the signals in real-time:** in the fourth step, the field data are aggregated, elaborated and analysed in order to reproduce the real system behaviour in the virtual environment, following the modelling assumptions made in the simulation model.
6. **Simulation model development**, the final step creates a Discrete Event Simulation model (DES) which will be used as a basis for the DT model (Trigueiro de Sousa Junior *et al.*, 2019). Including the connected model developed in the previous step in the DES model, the final DT application is created.

The steps 2 to 5 are iterative since they are applied to each single equipment in a production system. Only in the 6th step all the single models are assembled together in order to create the DT of the whole production system.

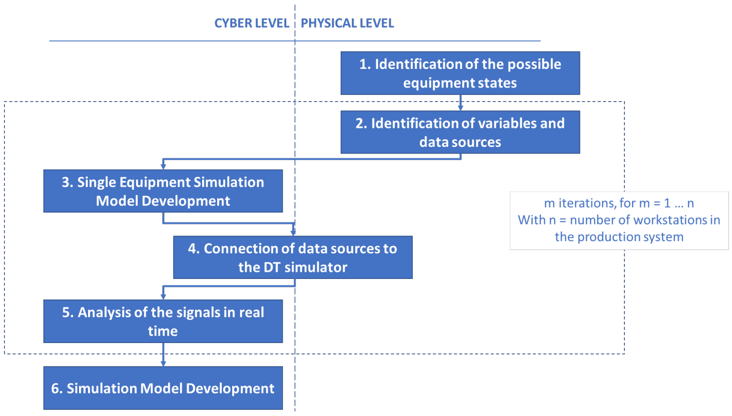

Figure 1: Methodology for the development of DT applications

Next sections introduce an application of the proposed methodology to create a DT in a laboratory assembly line.

4. The application environment

The Industry 4.0 Lab (I4.0Lab) of the Manufacturing Group of the School of Management of Politecnico di Milano is constituted of a fully automated line by FESTO.

Figure 2: Assembly line of the I4.0Lab

The line hosts the assembly process of simplified mobile phones, made of four components: Front Cover, Back Cover, Fuses, Print Circuit Board (PCB). The line, shown in Fig. 2, is composed of seven modular workstations that run different operations and are connected through automated belts.

Each of the seven station is equipped with different CPSs and IoT technologies which directly communicate with two PLCs (one PLC controls the operations of the station, the other monitors the energy consumption of the station).

These PLCs use the OPC UA communication protocol (Open Platform Communication Unified Architecture) (Zezulka *et al.*, 2018), which allows an open and reliable mechanism for transferring information between the PLCs and the application developed (Fumagalli *et al.*, 2016).
From a technical point of view, in order to better understand how the application has been implemented, it is possible to notice that the autonomous belts that connect the line and its embedded sensors are the same for each station. The conceptual difference among the stations is the kind of operation performed.

4.1 Identification of the possible machine states

The first step for the development of the DT of the line is the identification of the possible machine states. From the analysis of the real system and looking to the energy consumption of each station during a generic working cycle of the line, five different machine states have been identified that can be resumed as follow:

- *Idle:* the conveyor of the station is moving but no operation is performed. The machine is waiting for a piece to be processed;
- *Working:* The machine is performing an operation;
- *Error:* For each station of the line, a specific fault has been identified. The machine is blocked due to abnormal behaviour and shows an error on the Human-Machine Interface (HMI);
- *Emergency button:* It is a specific fault state, in which the normal behaviour of the machine is stopped due to the fact that the operator has triggered the emergency button;
- *Energy-saving mode:* The machine is on, but the belt is not moving to save energy when there is no piece to work immediately.

Once identified the possible states that each station can assume, the next step is the identification of the sensors, actuators and variables whose combination is able to reproduce the behaviour of the real system in time.

4.2 Identification of Sensors, actuators and variables useful in order to reproduce the states of the stations

The products flow on a carrier which is transported from a station to the next one by the set of belts that connect the overall system. As mentioned, each station of the I4.0Lab has the same set of embedded sensors and actuators in the belt that are represented in Fig. 3.

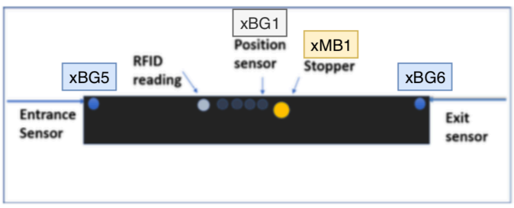

Figure 3: Schematization of the sensors of each belt group

Each of these sensors produce a binary output, meaning that the output value can be 1 or 0, depending on the action performed. The sensors belonging to the belt group used for the identification of the machine states are the following ones:

- *xQA_A1:* sensor used to know if the belt is moving or not; it is equal to 1 if the belt is moving;
- *xBG1:* carrier's sensor; it is equal to 1 when the carrier is ready to be processed;
- *xBG5:* entrance sensor of the machine's belt; it is equal to 1 when the carrier moves over it;
- *xBG6:* exit sensor of the machine's belt; it is equal to 1 when the carrier moves over it;
- *xMB1:* stopper sensor that releases the carrier from working position; it is equal to 1 when activated.

These sensors contain the information required in order to evaluate the presence of the piece in the station (from this, for example it is possible to find the difference between Energy saving state and Idle state) or about the working condition of the machine. Therefore, based on the combination of these sensors, it is possible to identify three of the mentioned states: *Idle, Working* and *Energy Saving.*
Even for the representation of the *Emergency Button* state, the exploited sensors are common for all the stations. Basically, the Emergency state correspond to an error state in which, for an abnormal behaviour of the machine, the emergency button is pushed, and the production is stopped. Also in this case the sensors are binary variables.

Table 1: Definition of the Emergency state

xSF5	xPF1	xPF3	Action	Em. Button
0	0	0/1	Button triggered	1
1	0	0/1	Button released	1
1	0/1	1	Wait for start	0
1	1	0	Start triggered	0

The ones used to define this machine state are resumed as follow:

- *xSF5:* it is set to 0 when the emergency button is pushed; it is set to 1 when the emergency button of the related station is released;
- *xPF1:* when the emergency button is pushed, it is set to 0 until the operator checks the error on the HMI, eliminating the error alarm that appears on it;
- *xPF3:* when is set to 1, the green light of the 'Start' button is turned on to show that the operator must press it to resume the work.

Therefore, based on the sensors identified, MATLAB functions are implemented in order to combine these sensors and reproduce the machine state. In the case of the *Emergency state,* the combination of its related sensors, needed to define the above-mentioned state, is described in Table 1.
As said before, the sensors and actuators identified up to now are common to all the stations. This implies that the schematisation of the Idle, Working, Energy saving, and Emergency states is the same for all the machines. What

differs from one station to another one is the definition of the error state, since each station of the line performs a specific operation on the product to which are linked different failure modes.
For sake of simplicity only one of the seven station of the line is presented here. The methodology steps 2 to 5 are replicated for all the other stations evaluating first of all the possible type of error that can occur in the station and then identifying the related sensors useful to check if all the operations are performed in the correct way. Once completed this evaluation, also in this case MATLAB functions will be implemented, and the definition of the error state will be integrated with the other ones in order to create the digital model of the station.
The model presented represents the Front Cover station reported in Fig. 4. Here the carrier loaded with the product to be assembled stops and the front cover is placed on it. Three different kinds of error can occur in this station when the carrier stops in the working position:

- There is no pallet on the carrier, so the assembly operation of the final product cannot have place;
- The cover is already on the pallet;
- The cover storage close to the station is empty.

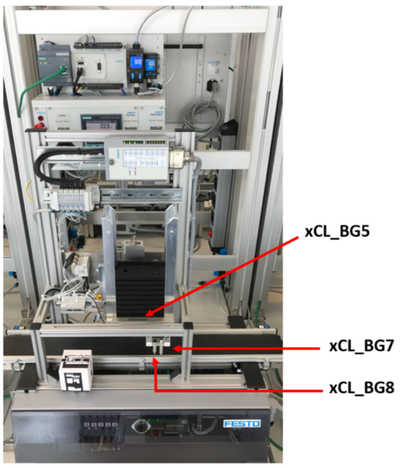

Figure 4: Front Cover station

Due to these errors, the operation cannot be performed. They can be identified through three different sensors that are available on the line, which are:

- *xCL_BG5:* detects if there is at least one front cover in the front cover storage, and it is set to 1 if this condition happens, 0 otherwise;
- *xCL_BG7:* detects if there is the pallet on the carrier, and it is equal to 0 if no pallet is detected;
- *xCL_BG8:* detects if there is already a front cover on the pallet, and it is equal to 1 if this condition happens.

By using these sensors, it is possible to associate the presence of an error in the station to one of the causes reported before.
Since each time the operation is performed, until the stopper releases the carrier from the working position, the sensor xCL BG8 is perceiving the presence of the front cover, allowing possible mistakes in the evaluation of the state, another variable is used to take into account this inconvenient. This variable is the iRetCode. It is able to provide different values based on different states of the machine. More in detail, it is equal to 2 if an error message appears on the HMI, 0 in any other case. In this way, this variable is able to identify the error state of the machine, but it is not able to associate the error to a specific cause. The iRetCode is not updated until a new carrier moves through the machine. For these reasons a combination of iRetCode, xCL_BG5, iRetCode, xCL_BG8 is used for the definition of the error state. Table 2 resumes how they are combined in order to identify the error state of the Front Cover station.

Table 2: Definition of the Error state for the Front Cover station

xQA1_A1	xCL_BG5	xCL_BG8	iRetCode	Error
0	0/1	1	0	0
1	0	0	2	0
0	0/1	1	2	1
1	1	0	2	1
0	0	0	2	1
		Else		0

Once the combination of the sensors to reproduce the error state have been identified, it is possible to schematize the five possible machine states of the first station as reported in Table 3.

Table 3: Definition of the machine state for the Front Cover station

xQA1_A1	xBG1	Error	Em. Button	Machine state
1	0/1	0	0	Idle
0	1	0	0	Working
0/1	1	1	0/1	Error
0/1	0/1	0	1	Em. Button triggered
		Else		Energy saving

4.3 Single Equipment Simulation Model Development

The third step of the presented methodology is based on the development of a simulation model for each single equipment piece. For this purpose, Simulink and MATLAB have been used as simulation software, as a result of the analysis conducted by (Fumagalli *et al.*, 2019). The idea at the basis of this step is the development of a model on Simulink that is able to replicate the steps done previously on a simulation environment. It is important to underline

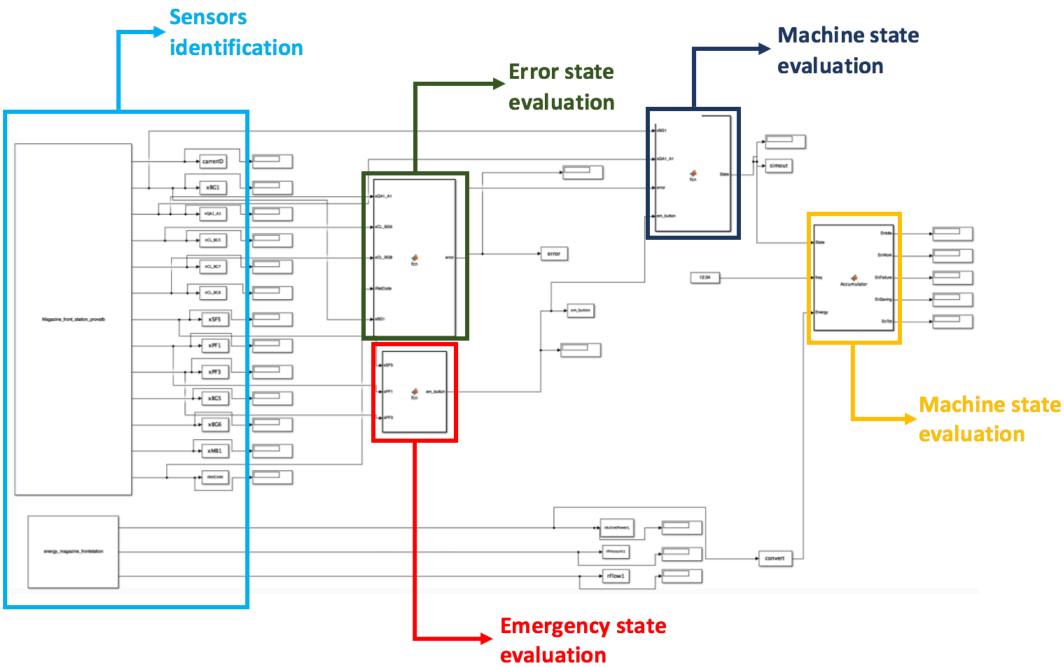

Figure 5: Simulink model of the Front Cover station

the fact that the at this point of the methodology, the equipment model is still not connected to the real system. Fig. 5 shows the model of the first station of the line in the I4.0Lab. As it is possible to see, all the data sources identified in the previous steps are present in the model. These sources are then used as inputs in other blocks for the evaluation of the states and the energy consumption of the line during the simulation.

4.4 Connection of the data sources to the DT simulator

Once the model of the single equipment has been developed on the Simulink environment, the next step consists in establishing a direct connection between the data sources and the single equipment simulation model. MATLAB was used for this step, thanks to its OPC UA communication protocol installed in the laboratory environment. In fact, thanks to MATLAB-OPCUAtoolbox, it is possible to establish a real-time connection between the software and the facility.
Therefore, MATLAB is able to read the value of sensors and actuators identified in the previous step at fixed time point by setting a sample time. In this way, the model developed in the previous step is connected in real-time with the equipment. It means that the value of each data source is uploaded continuously during the simulation. At this point it is required the combination of the data sources for the evaluation of the different behaviours and energy consumption of the equipment.

4.5 Analysis of the signal in real-time

In order to combine the data sources, different MATLAB functions have been implemented in order to analyse the signals in real-time directly from the field and thanks to them, the simulation becomes a synchronized virtual copy of the chosen station simulated through its machine states. At the same time, the coding on MATLAB allows the calculation of the energy consumption of each station of the line real-time. Taking as input the machine state, the value of the instantaneous power and the time of the simulation, the DT simulator is able to give as output the value of the energy consumed in each state while the simulation is running. Fig. 5 shows the Simulink model developed for the Front Cover station. As it is possible to notice from the figure, the green block corresponds to the definition of the error state, it takes as input the value of the sensors presented in Section 4.2 and gives 1 as output if the error state is detected and 0 in the other case. In the same way the red block represents the definition of the emergency state while the blue block the final evaluation of the state of the machine taking based on the output of the two previous presented blocks. The yellow one is responsible of the evaluation of the energy consumption of the station. At the end, different values can be extracted from this block: the energy consumption of the line in each state assumed while the simulation was running and the overall consumption of the line. These values can be combined in order to introduce some Key Performance Indicators able to evaluate the energetic performances of the line. Following the schematization of the methodology presented in Fig. 1, the same kind of model developed for the first station of the line is created for the other six stations of the line. This iterative process gives the basis for the construction of the DT simulation of the overall assembly line of the I4.0Lab.

4.6 Simulation model development

The final step for the development of the DT application is the development of a DES simulation that is able to represent the real system (initially this is not directly connected with the shop floor, it will be connected only in a second time). In order to create it, the library of Simulink is used, which allows to find standard blocks that represent the station, tools and component of the line. In particular, a specific kind of standard block called *entity server*

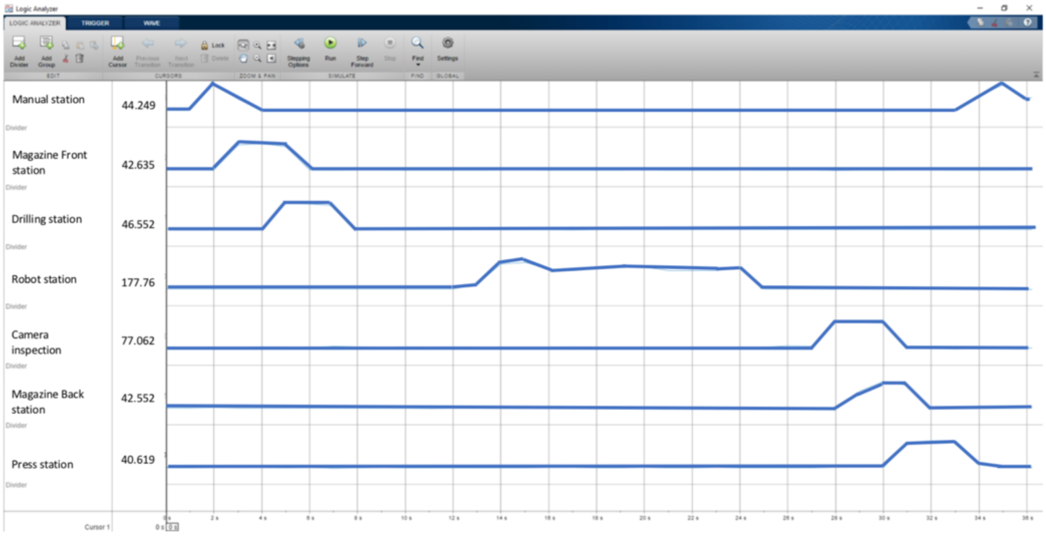

Figure 6 - Energy consumed during one-piece assembly in all stations of the I4.0LAB line

represents the machine of the line. In this way it is possible to model and simulate the production process of the real system on the simulation environment. Once this model has been created, the entity server blocks have to be connected to the real-time model of the stations presented before. This model sends a real-time message to the DES simulation model that identifies the state of the production system as a whole in real-time. Therefore, the final DT is built that collects data from each single equipment and is able to replicate the production in real-time, giving the evaluation of the energy consumption of the line during the process.

5. Results

The proposed application of the DT development methodology in the I4.0 LAB resulted in a DES simulator that was connected in real-time with the laboratory assembly line. The developed DT simulator is capable of replicating the line, by following the machine-states updates, and of computing the energy consumption of the single equipment and of the line as a whole, for different time intervals and for the assembly of one or more pieces. As an example, Fig. 6 reports the energy consumption in time of the seven single stations to assemble one product. As it is possible to see from Fig. 6, the x-axis is the time axis, while y-axis reports the energy consumption of each station of the line. The developed DT, is able to follow the product during each operation, reporting an increase of the energy consumed when the machines are in working state. More in detail, focusing on one single station of the line, it is possible to compare the trend of the energy consumption obtained with the DT and the one obtained through the data directly extracted from the servers of the line, without any communication with the model. As an example, Fig. 7 shows the Magazine Front Station energy consumption taken from the energy sensors (the same of the second station of Fig. 6). It can be easily seen that the second station in Fig. 6 and the values of Fig. 7 follow the same shape, demonstrating that the DT replicates the physical system.

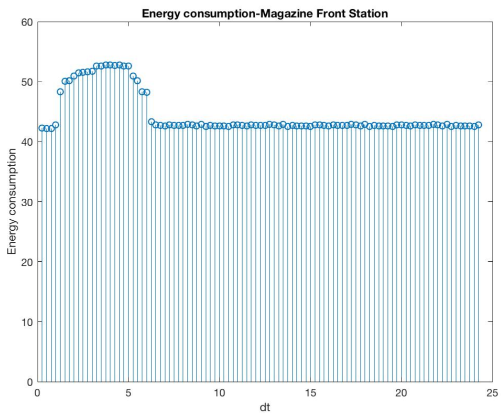

Figure 7 - Energy consumed by Magazine Front Station obtained with data coming from the station sensors

6. Conclusions

According to this new kind of simulation introduced by the I4.0 paradigm, real production systems equipment pieces are synchronized with the virtual model with real-time data directly coming from Internet-connected sensors into the shop floor. Thanks to CPS, the simulation model provides the virtual representation of a given system along its lifecycle, with an effort in reproducing different behaviours in real-time. Research on DT is still an open challenge, this work represents a contribution to it, by proposing a methodology to develop a DT that follows the machine-states of the equipment, in order to replicate the real production system running. The methodology has been applied to the I4.0Lab environment. It leveraged on smart sensors to monitor machine states and energy consumption for better management of resources. The OPC UA connection allows to open a gateway to the available data which is the concept of IoT integration described by the Industry 4.0. In this way, the DT allows the evaluation of the behaviour of a real system in real-time, enabling the collection of data. In the example shown all the data available with the implementation of the DT application (line behaviour and energy consumption) are acquired and collected in order to evaluate the energetic performance of the I4.0Lab. Under the perspective of Industry 4.0, the application of this kind of simulation tools, results in a new way of doing and thinking business that opens the gate to the concept of *Process Effectiveness* focused on the minimization of the volume and consumption of energy inside production systems, with the aim of creating more sustainable and efficient business model.

References

Baheti, R. and Gill, H. (2011). Cyber-physical systems. The Impact of Control Technology. IEEE Control Systems Society, pp. 12(1):161–166.

Canedo, A. (2016). Industrial IoT lifecycle via digital twins. 2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS). ACM, pp. 1–1.

Cao, Q. et al. (2019). Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach. Cybernetics and Systems. Taylor & Francis, 50(2), pp. 82–96.

Cattaneo, L., Rossi, M., Negri, E., Powell, D., and Terzi, S. (2017). Lean thinking in the digital Era. IFIP Advances in Information and Communication Technology.

Cattaneo, L. et al. (2018). Clarifying Data Analytics Concepts for Industrial Engineering. IFAC-PapersOnLine. Elsevier B.V., 51(11), pp. 820–825. ì

Fumagalli, L., Macchi, M., Pozzetti, A., Taisch, M., Tavola, G., and Terzi, S. (2016). New methodology for smart manufacturing research and education: The lab approach. Proceedings of the Summer School Francesco Turco, 13-15-Sept, pp. 42–47.

Fumagalli, L. et al. (2019). Framework for simulation software selection. Journal of Simulation (In printing), pp. 1–18.

Gabor, T., Belzner, L., Kiermeier, M., Till Beck, M., and Neitz, A. (2016). A simulation-based architecture for smart cyber-physical systems. Proceedings - 2016 IEEE International Conference on Autonomic Computing, ICAC 2016., pp. 374–379.

Glaessgen, E. and Stargel, D. (2012). The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, pp. 1–14.

Heymans, S. et al. (2008). Ontology Reasoning with Large Data Repository', in Ontology Management. Springer US, pp. 89–128.

Lee, J. et al. (2013) Predictive manufacturing system - Trends of next-generation production systems, IFAC Proceedings Volumes (IFAC-PapersOnline). IFAC.

Lee, J., Bagheri, B. and Kao, H. A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters. Society of Manufacturing Engineers (SME), 3, pp. 18–23.

Lee, J., Kao, H. A. and Yang, S. (2014). Service innovation and smart analytics for Industry 4.0 and big data environment. Procedia CIRP. Elsevier B.V., 16, pp. 3–8.

Macchi, M. et al. (2018). Exploring the role of Digital Twin for Asset Lifecycle Management. IFAC-PapersOnLine. Elsevier B.V., 51(11), pp. 790–795.

Mousavi, S., Thiede, S., Li, W., Kara, S., and Herrmann, C., (2016). An integrated approach for improving energy efficiency of manufacturing process chains. International Journal of Sustainable Engineering. Taylor & Francis, 9(1), pp. 11–24.

Negri, E., Fumagalli, L., Cimino, C., and Macchi, M. (2019). FMU-supported simulation for CPS Digital Twin. Procedia Manufacturing. Elsevier B.V., 28, pp. 201–206.

Negri, E., Fumagalli, L. and Macchi, M. (2017). A Review of the Roles of Digital Twin in CPS-based Production Systems. Procedia Manufacturing, pp. 939–948.

Pagoropoulos, A., Pigosso, D. C. A. and McAloone, T. C. (2017). The Emergent Role of Digital Technologies in the Circular Economy: A Review. Procedia CIRP, 64, pp. 19–24.

Polenghi, A., Fumagalli, L. and Roda, I. (2018). Role of simulation in industrial engineering: focus on manufacturing systems. IFAC-PapersOnLine. Elsevier B.V., 51(11), pp. 496–501.

Reifsnider, K. and Majumdar, P. (2013). Multiphysics Stimulated Simulation Digital Twin Methods for Fleet Management. IFIP International Conference on Advances in Production Management Systems, pp. 1–11.

Rüßmann, M. et al. (2015). Indusry 4.0: Future of Productivity and Growth in Manufacturing. Boston Consulting Group (BCG), (April), p. 20.

Sattar, M. A., Anwaruddin, M. and Ali, M. A. (2017). A Review on Internet of Things - Protocols, Issues. IJCST, 5(2), pp. 91–97.

Taisch, M., Stahl, B., Vaccari, F., and Cataldo, A., (2013). A Production-State Based Approach for Energy Flow Simulation in Manufacturing Systems. IFIP International Conference on Advances in Production Management Systems, pp. 227–234.

Trigueiro de Sousa Junior, W. et al. (2019). Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. Computers and Industrial Engineering. Elsevier, 128(December 2018), pp. 526–540.

Zezulka, F., Marcon, P., Bradac, Z., Arm, J., Benesl, T., and I. Vesely. (2018). Communication Systems for Industry 4.0 and the IIoT. IFAC-PapersOnLine, 51(6), pp. 150–155.