Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

A LEGO® MANUFACTURING SYSTEM AS DEMONSTRATOR FOR A REAL-TIME SIMULATION PROOF OF CONCEPT

Giovanni Lugaresi
Davide Travaglini
Andrea Matta

Department of Mechanical Engineering
Politecnico di Milano
Via La Masa, 1
Milano, 20156, ITALY

ABSTRACT

Digital models for planning and control of production systems are a key asset for manufacturers to gain or maintain their leadership. However, these models are often based on frameworks that do not take into account the real-time dimension, hence it is arduous to exploit them for taking short-term decisions over complex systems. The goal of this work is to prove the applicability of a Real-Time Simulation (RTS) framework that prescribes to exchange current-status data from a manufacturing system and to run alternative simulation models to decide the next moves. For this scope, we exploit a LEGO® Manufacturing System (LMS) together with a discrete event simulator that plays the role of its digital model. The results of this proof of concept show that the proposed framework can effectively be used to find better production rules for a manufacturing system in real-time manner.

1 INTRODUCTION

In the current industrial scenario, manufacturing systems are changing drastically and the related research is receiving the greatest attention. Thanks to innovations within the Industry 4.0 hype, several application scenarios can be envisioned for new real-time planning and control methodologies that make use of advanced data-driven tools and digital models. These models have to face a continuously changing environment (Qi and Tao 2018), yet it is not very clear how the huge amount of collected information can be exploited to evaluate alternative reaction scenarios following deviant behaviors. With a planning and control scope, we may consider a discrete event simulator as digital model of a manufacturing system. Real-Time Simulation (RTS) is the research direction that aims at exploiting Simulation for short-term decision making, and examples of its applications for production planning and control can be found in the literature.

This paper proposes an RTS framework and presents a proof of concept for its application on a LEGO®-based physical model. The aim is to prove that, following a change in the system state, alternative production policies can be tested through simulation with the goal to select the best performing one depending on the real system status at the time of the disruption and to apply it on the real system.

The paper is organized as follows. Section 2 lists the main contributions on the Digital Twin literature. Section 3 shows the proposed framework for RTS. Section 4 presents the LEGO®-based model that has been used in this work. Section 5 explains the test-case and section 6 provides the numerical results. Conclusions are drawn in section 7.

978-1-7281-3283-9/19/$31.00 ©2019 IEEE

2 STATE OF THE ART

One of the first definitions of Digital Twin (DT) has been introduced in the aerospace field (Shafto et al. 2012; Boschert and Rosen 2016) and cites:

> *"an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. It is ultra-realistic and may consider one or more important and interdependent vehicle systems."*

The original scope of the DT was to mirror the state of vehicles with a series of integrated sub-models that reflect different aspects of the vehicle system, by taking into consideration stochastic factors, historical data and sensor data, including interactions of the vehicle with the real world.

2.1 Digital Twin in Manufacturing

The DT concept has been rapidly conjugated in the manufacturing context. Grieves (2014) considered the DT a virtual representation of what has been manufactured, by promoting the idea of exploiting it to compare what was produced versus what was designed. The DT is described as a digital informational construct about a physical system, including all the information about the physical system that could be potentially obtained from its inspection in the real world (Grieves and Vickers 2017). Rhodes and Reid (2016) defined a DT as an integrated model of an as-built product including physics, fatigue, life-cycle, sensor information and performance simulations, which is intended to reflect all manufacturing defects and be continuously updated to include wear-and-tear information while in use. Several definitions have been attributed to the Digital Twin. In manufacturing, the DT is usually identified as a virtual representation of a machine tool or a production system which is able to deal with various simulation disciplines characterized by the synchronization between the virtual and real system (Grieves and Vickers 2017).

Kritzinger et al. (2018) proposed a classification of the DT concept based on the different levels of integration between the physical and digital counterpart. The integration level is defined as the degree of connection between the virtual and the real system. The authors introduce three integration levels: Digital Model (DM), Digital Shadow (DS) and Digital Twin (DT). The first level corresponds to have two distinct and independent systems, with no connection between them. The second level of integration (DS) is represented by an automated one-way real-time data exchange between the physical and the virtual systems. The DT level is characterized by a complete real-time data exchange between the physical and the real system in both directions and – in such setting – the digital system might also act as controller of the real system.

2.2 Applications

Several applications of the DT in manufacturing can be found in the literature. Bottani et al. (2017) developed a prototype of a DT for an Automated Guided Vehicle (AGV) for solving the material handling problem of a job-shop manufacturing system in a laboratory environment. The authors aim at developing the concepts of decision-making autonomy and self-adaptation of machines operating in an industrial plant. Results have shown that the combined architecture of a Cyber Physical System (CPS) and a DT allows a strategic optimization of the plant resources. The authors used a discrete event simulation software for simulating several environments, and implemented different policies to optimize the auto-adaptive behavior of the CPS-AGV. Gonzalez (2013) presented an application of a simulation software algorithm applied on a miniaturized flexible manufacturing system. The author was able to control the system and, after the initialization of the simulation model, to perform simulation experiments for predicting the future performances of the model. Leng et al. (2018) developed a demonstrator of DT-driven control for a smart

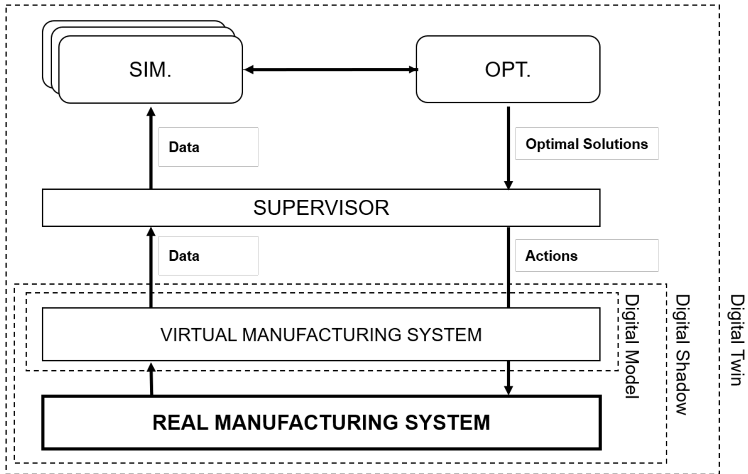


Figure 1: RTS Framework proposed in this paper in comparison with the integration levels defined by Kritzinger et al. (2018): Digital Model, Digital Shadow, and Digital Twin.

manufacturing workshop producing board-type products, in which data coming from the real system feed a virtual model in real time.

Gabor et al. (2016) defined a DT-based simulation architecture in which the CPS communicates with both the physical system and and its simulation model using the same interface. The authors analyzed the different types of information flows between the components of a CPS by classifying different methods for controlling the system behavior. Yang, Tan, Yoshida, and Takakuwa (2017) showed the modeling framework for a simulation-driven Digital Twin and demonstrated through a test case that a simulator can reflect the physical system situation with real-time data feeds. Several other frameworks for RTS have been proposed in the literature and have been reviewed by Lugaresi and Matta (2018), who also outlined the main research directions for the successful application of RTS to the manufacturing context.

3 REAL-TIME SIMULATION FRAMEWORK

3.1 Architecture

In this paper we refer – without loss of generality – to the framework for planning and control in a manufacturing plant based on RTS described in Figure 1. The Real Manufacturing System block represents the production plant of interest. From here, data are continuously collected from several sources (typically, the Manufacturing Execution System or the Advanced Planning and Scheduling System). The datasets contain information such as the layout, processing times and failure rate of each machine, products to be processed, and queue levels. All these data are used to align the digital model of the manufacturing system to its physical counterpart (synchronization). The generation of the digital model of the system in the form of a DES model takes place in the Virtual Manufacturing System block. The model building phase takes place either if a digital model is not existing yet, or if the current model is not considered as a valid representation of the real system. Simulation models are validated, updated and synchronized with the current system states through the Supervisor block. Starting from the digital model developed in the virtual block, several alternative simulation models may be created with the aim to test different alternatives (for instance, several routing options after one parallel branch is down). The simulation models are run and the optimal solutions are identified by the Optimization block. Then, the Supervisor block is used to check the applicability of the solution with the scope of controlling if the prescribed corrective actions can be taken before their implementation on the real system.

As shown in Figure 1, our framework is compliant with the definitions proposed by Kritzinger et al. (2018). Indeed, the DM is the digital counterpart of the production system (in our case, a discrete event

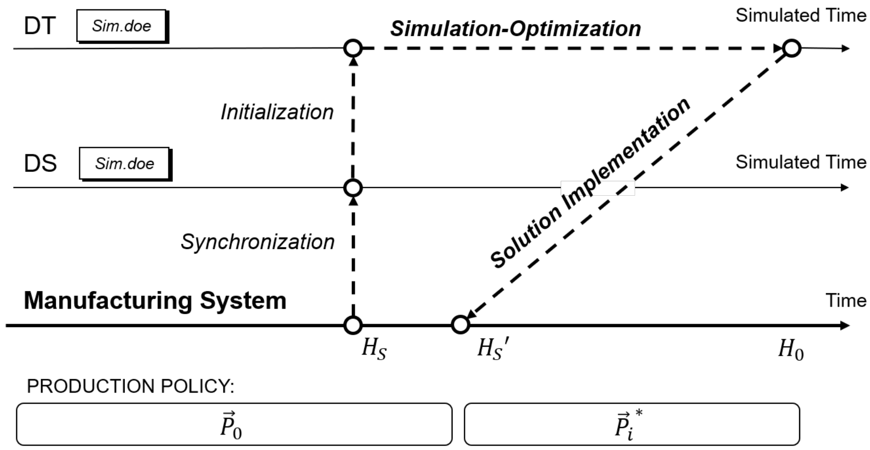


Figure 2: Temporal view of the RTS working procedure.

simulator). The DS integration level is reached if the digital model is correctly linked with the physical system and it reflects any change that is occurring (in other words, if synchronization can be guaranteed). The DS, the Supervisor, the Simulation, and the Optimization blocks are composing the DT. Indeed, all the aforementioned components arranged in the proposed framework allow for searching new solutions (for instance, in response to system changes or disruptions) and for implementing actions to the real system with a short-term horizon.

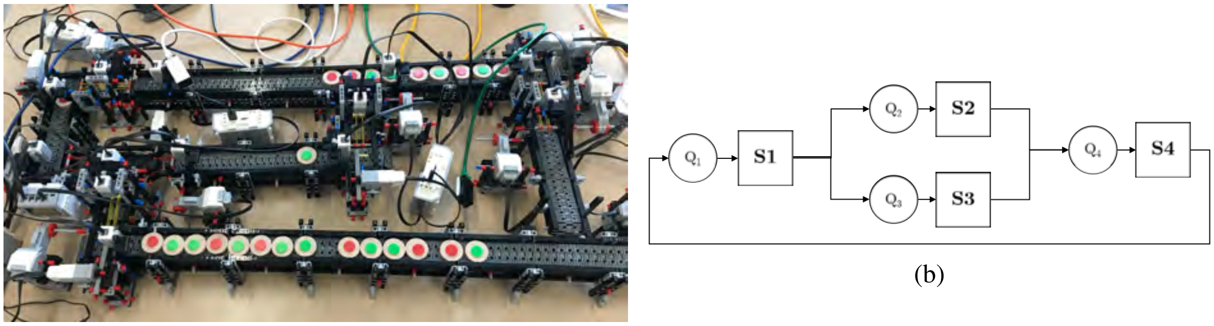
3.2 Working Procedure

A discrete event simulation model of the manufacturing system (for instance, an ARENA® model *Sim.doe*) is available and validated. The same simulation model may be used both as DM of the manufacturing system and as base model to create sub-models representing the simulation of several alternative scenarios. We are interested in the performance of a production system for a specified time horizon H_0, in which at a certain moment H_S an unexpected event or situation occurs. First, the DS is synchronized with the physical system status. Therefore, the system state at H_S will be the starting point for our analysis. A change in the physical system state might provoke a detriment of its performances and the current production policy might not be optimal any further. Therefore, an evaluation has to be made to search for an optimal reaction strategy.

Let us then define $P_i \in \mathbb{P}$ a finite set of $N+1$ policies $\{P_0, P_1, \ldots, P_N\}$ defined a-priori in which P_0 is the original policy run on the system and the remaining N policies define alternative reaction scenarios. The performances obtained by each policy are evaluated with the same simulation model as the DS. Hence, several simulation experiments of the system are generated with the aim to detect a solution that improves the expected performance of the system in terms of a generic Key Performance Index (KPI) Θ for the time interval $[H_S, H_0]$. All the simulation models start from the same time H_S (initialization). The solution corresponds to a new optimal system management policy P_i^* which satisfies:

$$P_i^* = \arg\max_{P_i \in \mathbb{P}} \{\Theta(P_i)\} \tag{1}$$

P_i^* will be implemented in the physical system at the time H_S' and the interval $[H_S, H_S']$ is proportional to the computation effort required to identify the new policy. Figure 2 summarizes the temporal evolution of the working procedure.



(a) (b)

Figure 3: The LEGO® Manufacturing System (LMS) – (a) Picture of the demonstrator, (b) logical scheme of the system with stations (S1, S2, S3, and S4) and buffers (Q_1, Q_2, Q_3 and Q_4).

4 LEGO® MANUFACTURING SYSTEM (LMS)

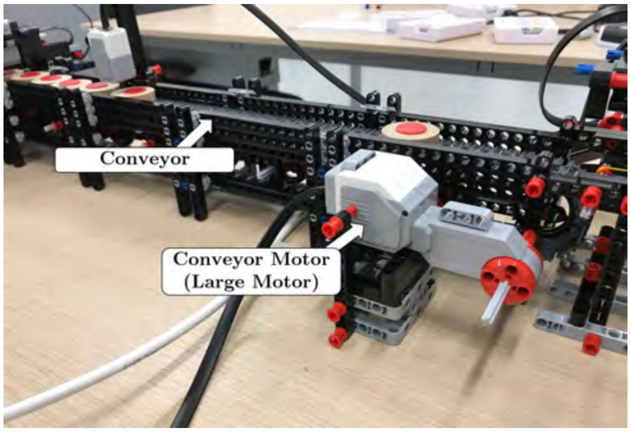
With the aim of testing the RTS framework presented in section 3, we exploited a LEGO® Manufacturing System (LMS), a miniaturized model of a manufacturing system made of LEGO® components. Examples of LMS can be found in the literature. Sanchez and Bucio (2012) built an LMS composed by two workstations, two feeding systems, two dispatchers and a conveyor belt system. Each workstation is composed of a feeder, a processing machine, and an unloading device for finished pallets. The authors used the LMS to teach control systems. The controllers designed by students had to supervise the resource allocation of tasks during production. Syberfeldt (2010) built a LEGO® factory for simulating the refinement of raw materials using three stations, controlled by dedicated LEGO® MINDSTORMS® EV3® bricks. Each station simulates a particular operation on the raw material, hence the whole model simulates a general production plant. Jang and Yosephine (2017) developed an automated manufacturing system using LEGO®. The system consists of one feeder, two machines and buffers with a finite size. Even if not formally LMS, we also refer to the experience of Tan et al. (2018), who developed a IOT-aided model made of Fischer-Technik® components with the aim of demonstrating the alignment of a simulator with a physical system in real-time.

4.1 Physical Model

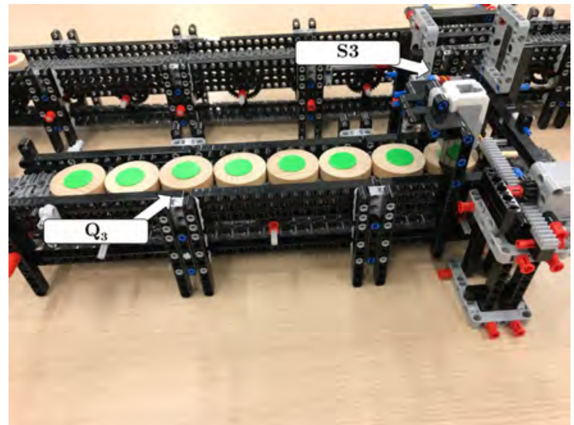
The LMS used for this work is shown in Figure 3a and it has been installed at the Department of Mechanical Engineering of Politecnico di Milano (Milan, Italy). A similar model has been built for didactical purposes and it is exposed in Lugaresi et al. (). The model represents a closed-loop production system with four stations: $S1$, $S2$, $S3$, and $S4$. Each station is controlled by a dedicated LEGO® MINDSTORMS® EV3® brick which is programmed using python scripts (EV3DEV OS). Each station has its own script that is run locally. The stations are also connected through a local network and a PC with SSH connection protocol and different scripts can be sent to be executed on the EV3s. Conveyors are controlled through proprietary LEGO® software that allows switch on/off and speed setting directly from the EV3 bricks.

Pallets are represented by wooden disks that are tagged with green and red plates representing pallets dedicated to two different product types. Figure 3b reports the layout of the system: $S2$ and $S3$ are arranged in parallel and a flow splitter system after S1 is able to recognize the part type coming from S1 and to send the pallet to S2 or S3 depending on the product type.

The conveyor is characterized by an EV3 brick, a *Large Motor*, and several other LEGO® components (e.g., beams, connectors, axles and wheels). The reader is referred to Travaglini (2018) for further details on the LEGO® components we used. Figure 4a shows the model of the conveyor. All the conveyors in the LMS have the same speed. Pallets are lining on the conveyors before entering the stations, hence conveyors represent intermediate buffers and each station has a dedicated input buffer (Q_1, Q_2, Q_3, and Q_4, respectively).



(a)



(b)

Figure 4: Details of the LMS – (a) the conveyor system; (b) the part-entrance system with the upstream conveyor queue (S3).

The model of a station is composed by an EV3 brick, three EV3 optical sensors, a *Large Motor* and a part-entrance system. The part-entrance system (Figure 4b) is placed in front of each station and it is made by several beams and a *Medium Motor*. A beam blocks the pallets in front of the station and avoids the entrance of more than one pallet at a time. Further, it pushes the next pallet to be worked inside the station. The three optical sensors are in the following positions (Figure 5):

- Sensor 1 stays over the part-entrance system, to recognize if a part is waiting to be worked;
- Sensor 2 is placed in the middle of the station structure to see if a pallet has entered the machine and to distinguish the product type;
- Sensor 3 is installed over the conveyor at the station $S(i+1)$ with the aim to identify if the downstream buffer is full.

The processing of a part is modeled by assigning to a certain operation a defined amount of time, that is $T_{w,A}$ for product type A and $T_{w,B}$ for product type B. Pallets are held by stations for this amount of time, after which the *Large Motor* pushes them toward the downstream conveyor, provided that it is not full. Notice that each station can be programmed to have different distributions of the processing times.

Summarizing, the stations scripts provide the following capabilities:

- recognizing the state of the input buffer;
- recognizing the state of the station;
- distinguishing the product type;
- attributing the correct processing time, $T_{w,A}$ or $T_{w,B}$, accordingly to the product type;
- recognizing the state of the output buffer Q_{i+1};
- simulating a failure state by increasing the time parts are held in the station.

4.2 Digital Model and Data Exchange

In our case, the digital model of the LMS is a discrete event simulator model written in ARENA® (*Sim.doe*). The simulator has been built with the same characteristics of the real system and the logical layout as in Figure 3b. The simulation model can evaluate a certain set of KPIs (e.g. throughput, efficiency). The same features of the physical model described in section 4.1 have been modeled in the simulator.

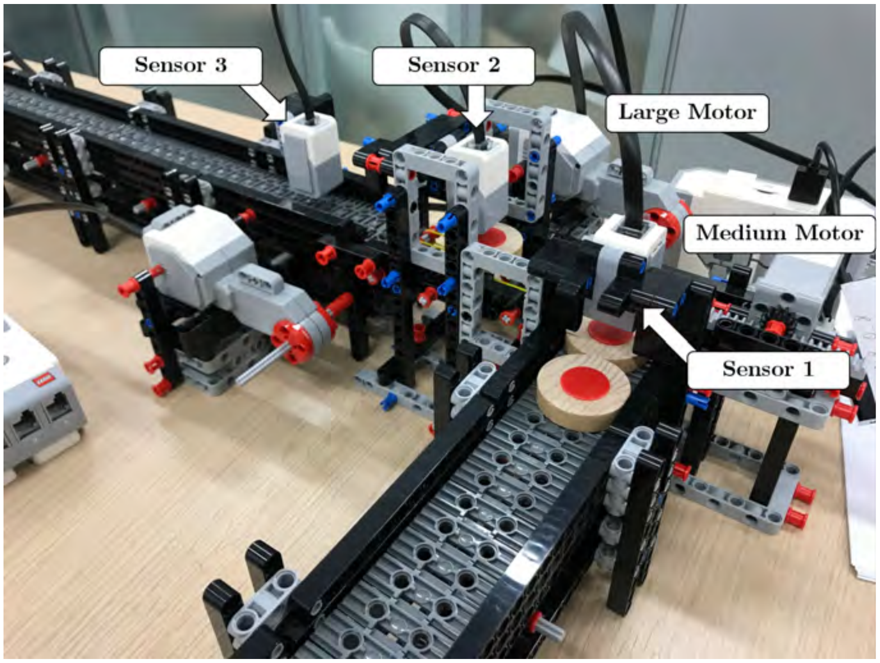


Figure 5: The model of a station in the LMS.

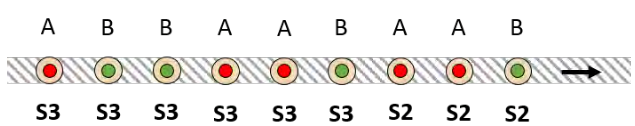


Figure 6: Example of the policy sequence $P_i := \{\Phi_A^{(i)} = (2,3), \Phi_B^{(i)} = (1,3)\}$.

The simulation model can be updated anytime with data coming both from the LMS sensors infrastructure and from observations. In this work, the data-exchange phase between the LMS and its DM is done by stopping the LMS, collecting data from the EV3s through data-exchange files (**.txt* files), and feeding the simulation model with the same files. Data include the buffer levels, the parts produced so far, the processing times assigned to each part, the remaining time to complete the production schedule. For instance, the pallets waiting at the entrance of each station are counted and buffer levels in the simulator are updated. Hence, the simulator can be synchronized to reflect the state of the LMS at any moment. Notice that as long as it is possible to update the simulation model with the real system status anytime, the DS integration level is guaranteed.

5 CASE STUDY

5.1 Production System Description

Consider LMS shown in Figure 3a. Two product types (A and B) can be worked by all the stations. The products require different processing times depending on the station and are modeled as stochastic and sampled from Weibull distributions. The parameters of the distributions are listed in Table 1 together with the buffer sizes. Station S1 does not require any setup time for switching neither from A to B nor vice versa. Stations S2, S3, and S4 are characterized by setup times as in Table 2. Notice that the processing times on S2 are lower than on S3, but S2 requires higher setup times. Moreover, the setup time to switch

Buffer	**Size**
Q_1	9
Q_2	5
Q_3	13
Q_4	8

Station	$\mathbf{T_{w,A}}$ **[s]**	$\mathbf{T_{w,B}}$ **[s]**
S1	WEIB(10,3)	WEIB(12,7)
S2	WEIB(16,5)	WEIB(10,8)
S3	WEIB(18,4)	WEIB(25,5)
S4	WEIB(19,3)	WEIB(20,4)

Table 1: Test Case – Buffers sizes Q_i and Processing Times $T_{w,A}$ and $T_{w,B}$.

S2	*from*	A	B
to	A		35
	B	15 + $f(x)$	

S3	*from*	A	B
to	A		20
	B	15	

S4	*from*	A	B
to	A		18
	B	22	

Table 2: Test Case – Input setup times [s] for stations S2, S3, and S4.

from product type A to type B on station S2 is $15+f(x)$ seconds, and $f(x)$ is:

$$f(x) = \begin{cases} 0 & if \quad x \leq 3 \\ e^{3+x} & if \quad x > 3 \end{cases}$$

where x is the number of consecutive pallets of type A arriving at the station. Hence, after 3 consecutive pallets of product type A worked by the station, the setup time required for switching to the production of product type B increases according to (2). This could represent a situation in which, for instance, the setup has not been forecasted in the production planning phase and – along a production day – less operators can dedicate time to it. Additionally to setup times, station S4 is characterized by maintenance operations that are done once every three pallets of product type B produced. The duration of the maintenance is stochastic and its value is sampled from a Weibull distribution with parameters 50 and 10 seconds.

Let us define the i-th alternative *production policy* as the rule that governs the flow splitter before S2 and S3 as a tuple $P_i := \{\Phi_A^{(i)} = (\phi_{A,2}, \phi_{A,3}), \Phi_B^{(i)} = (\phi_{B,2}, \phi_{B,3})\}$, where $\phi_{A,2}$ is the number of pieces of product type A sent to S2 in a sequence. For instance, if $P_i := \{\Phi_A^{(i)} = (2,3), \Phi_B^{(i)} = (1,3)\}$, it means that 2 pieces of A are sent to S2, then 3 pieces to S3, while 1 piece of B is sent to S2 and 3 to S3, independently from the sequence of products of type A that has arrived in the meantime (Figure 6).

5.2 Problem Description

Now, we represent a situation in which the LMS is subject to an unexpected event and we exploit the proposed RTS framework to decide how to react. In general, we assume that every time an improvement scenario has been identified, the corresponding corrective actions are to be selected and implemented on the LMS.

Let us consider the case in which $H_0 = 20\,min$ and the system is running with 22 pallets (11 pallets of product type A and 11 of product type B). We represent a situation in which at the time $H_S = 2\,min$, 15 new pallets have to be added to the system (this could happen for instance after a maintenance or for meeting a sudden demand increase); specifically, 10 pallets of type A and 5 of type B with a predefined sequence. The addition of extra pallets causes a decrease in the system performances due to the high setup time. Indeed, the added batch is mostly composed by part A and setups on S2 mainly depend on the number of pallets of product type A. Therefore, the current production policy P_0 might not be optimal any more. New policies must be evaluated as they could enhance the setup times and the system performances. We analyze the possible alternative policies that govern the flow splitter system. The identified sequence is written inside a **.txt* file which can be read both by the LMS and its digital model.

Production Policies ($\mathbb{P}$)	Φ_A	Φ_B
P_0 – *Base Policy*	(2,1)	(2,1)
P_1 – *Only mix A*	(1,2)	(2,1)
P_2 – *Only mix B*	(2,1)	(1,2)
P_3 – *Mixed Policy*	(1,7)	(7,1)
P_4 – $\sigma_{min,2}$	(1,5)	(2,1)

Table 3: Test Case – Alternative production mix policies that have been tested.

6 RTS SOLUTION AND NUMERICAL RESULTS

In order to face the situation described in section 5.2 we have implemented the RTS-based procedure introduced in section 3.2. Therefore, we simulated a finite set of scenarios defined by tentative production policies in order to increase the productivity of the system in the rest of the production horizon $[H_S, H_0]$. The KPIs of interest are the system efficiency θ and the total setup times at stations $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. For brevity, we show only the results for σ_2.

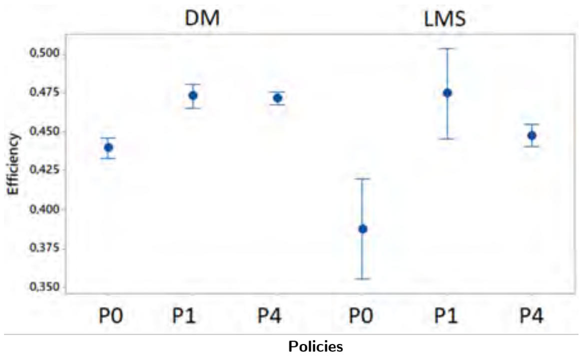
For this case we have analysed $N = 4$ alternative policies. Further, we assumed that dedicated production is not allowed, hence it is not possible to produce a product type on either S2 or S3 alone. As a result, we defined the set of policies to be explored $\mathbb{P} = \{P_0, P_1, P_2, P_3, P_4\}$ composed by:

- P_0 *(Base Policy)* is the current production policy.
- P_1 *(Only mix A)* modifies the product mix of product A. The number of pallets of type A sent to S3 is increased.
- P_2 *(Only mix B)* modifies the product mix by increasing the number of pallets sent to S3.
- P_3 *(Mixed Policy)* modifies contemporarily the mix of product type A and product type B.
- P_4 *(*$\sigma_{min,2}$*)* is the policy which minimizes the setup times on S2.

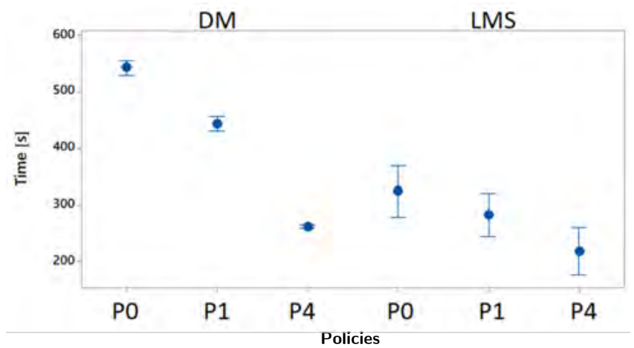
Table 3 lists the details of the aforementioned policies. Each policy has been implemented and simulated in the ARENA® model (*Sim.doe*). Each experiment is a simulation of $H_0 - H_S = 18\,min$ of production and 20 replications have been done.

The most promising solutions are obtained by P_1 and P_4 which dominated P_2 and P_3 for both the KPIs of interest. Hence, in the following only the results of P_1 and P_4 will be shown. The results obtained by P_1 and P_4 in comparison with the performance of the system following the policy P_0 are reported in Figures 7a and 7b. From the results it can be noticed that P_1 and P_4 are maximizing the efficiency θ. On the contrary, P_4 minimizes σ_2.

We expected to see the same improvement observed for the virtual model in the LMS and for this scope both policies P_1 and P_4 have been tested on the LMS for the same interval $[H_S, H_0]$. Results obtained on the LMS confirmed the same ranking of solutions as the one obtained in the simulation experiments. The efficiency is maximized by implementing P_1 and P_4, while setup times are minimized by P_4. However, a much higher variability of the results is observable and this is due to the fact that several mechanical interactions within the LMS (e.g. in the conveyor belts and in the gears) have not been modeled in the simulation. As shown in Figures 7a and 7b, we have identified two possible solutions in order to correct the negative trend to which the system would have been subjected to. Further, we have tested if the differences between performances obtained by the selected policies are statistically significant by means of paired t-tests. Specifically:



(a)



(b)

Figure 7: Test Case – Comparison of the (a) efficiency θ and (b) setup time σ_2, obtained for the policies P_0, P_1, and P_4, respectively, both on the LMS and on its DM.

Model	**KPI**	**Comparison**	**99% Confidence Interval**	**P-Value**
DM	θ	P_0 vs. P_1	[0.0216 , 0.0458]	0.000
		P_0 vs. P_4	[0.0228 , 0.0417]	0.000
	σ_2 [s]	P_0 vs. P_1	[-125.75 , -72.660]	0.000
		P_0 vs. P_4	[-299.96 , -260.95]	0.000
LMS	θ	P_0 vs. P_1	[0.0161 , 0.1584]	0.005
		P_0 vs. P_4	[-0.0010 , 0.1214]	0.011
	σ_2 [s]	P_0 vs. P_1	[-149.90 , 66.200]	0.149
		P_0 vs. P_4	[-201.80 , -9.900]	0.007

Table 4: Test Case – Confidence Intervals ($\alpha = 1\%$) and P-Values obtained from the pairwise comparison tests.

$$H_0: \quad \mu_{\Theta(P_i)} - \mu_{\Theta(P_j)} = 0 \quad \forall i \neq j \tag{2}$$

$$vs.$$

$$H_1: \quad \mu_{\Theta(P_i)} - \mu_{\Theta(P_j)} \neq 0 \quad \forall i \neq j \tag{3}$$

The 99% confidence intervals and the P-Values of the pairwise comparisons between KPIs obtained for both the LMS and its DM are reported in Table 4. All the tests allowed for the rejection of the null hypothesis (2) with a confidence level $\alpha = 1\%$. From an application point of view we may state that, after the increase in the number of pallets, a production planner may be suggested to implement the policy P_4, since it both provides an efficiency increase that is comparable with the one obtained by P_1 and minimizes the setup times.

In addition, we have repeated the same comparisons with a long simulation horizon $H \gg H_0$ ($H = 1000\,min$) and the results confirmed the dominance of policies P_1 and P_4. The pairwise comparisons between efficiency θ results are following: for the P_1 policy $CI_{99\%} = [0,0403;0,2325]$ and $P-Value = 0.001$. The 99% confidence interval of the comparisons for the $\sigma_{min,2}$ policy is $[0,0367;0,2438]$ and the $P-Value = 0.001$. Also in these cases it was possible to reject H_0 with a confidence level $\alpha = 1\%$ and this means that the solutions identified can effectively improve the system performances even with a long-term perspective.

7 CONCLUSIONS

In this work, we proposed an RTS framework and demonstrated its applicability with a test case on a LEGO® Manufacturing System (LMS). Particularly, we have accomplished the data exchange and reaction policy identification procedure. In fact, we have been able to take a decision on the real system for correcting a negative trend imposed by a disruption and improving its efficiency with respect to the base production policy. Future works should aim at improving the decision making process. Particular attention should be given to the generation of alternative policies and how to optimally choose between a pool of candidates. Further, we observed a systematic deviation between the LMS and its DM which is likely due to the combination of several aspects (e.g., the variability in the speed of the conveyor, the number of stations, the number of conveyors). In order to remove this systematic deviation an experimental campaign should be performed with the aim of evaluating the most influencing factors. Moreover, in this work we had no time limitations in order to evaluate and implement a decision (referring to Figure 2, $H_S = H'_S$), hence tests without stopping the LMS should be performed with the aim to evaluate and implement a solution while the system is running.

ACKNOWLEDGMENTS

The construction of the LMS has been partially funded by Sme.UP Group. The Authors also thank the Italian LEGO® Users Group (ItLUG) for the contribution in building the physical model.

REFERENCES

Boschert, S., and R. Rosen. 2016. “Digital twin: the simulation aspect”. In *Mechatronic Futures*, edited by P. Hehenberger and D. Bradley, 59–74. Cham, Switzerland: Springer.

Bottani, E., A. Cammardella, T. Murino, and S. Vespoli. 2017. “From the Cyber-Physical System to the Digital Twin: the process development for behaviour modelling of a Cyber Guided Vehicle in M2M logic”. In *Proceedings of the XXII Francesco Turco Summer School of Industrial Systems Engineering*, 1–7.

Gabor, T., L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz. 2016. “A simulation-based architecture for smart cyber-physical systems”. In *Proceedings of the 2016 International Conference on Autonomic Computing (ICAC)*, 374–379. Piscataway, New Jersey: IEEE.

Gonzalez, F. G. 2013. “Real-Time Simulation and Control of Large Scale Distributed Discrete Event Systems”. *Procedia Computer Science* 16:177–186.

Grieves, M. 2014. “Digital twin: Manufacturing excellence through virtual factory replication”.

Grieves, M., and J. Vickers. 2017. “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems”. In *Transdisciplinary perspectives on complex systems*, edited by F. Kahlen, S. Flumerfelt, and A. A., 85–113. Cham, Switzerland: Springer.

Jang, Y. J., and V. Yosephine. 2017. “Teaching Stochastic Systems Modeling using LEGO Robotics Based Manufacturing Systems”. In *Proceedings of the 11th Conference on Stochastic Models of Manufacturing and Service Operations*, edited by T. T. et al., 293–300. Milano, Italy: ITIA-CNR.

Kritzinger, W., M. Karner, G. Traar, J. Henjes, and W. Sihn. 2018. “Digital Twin in manufacturing: A categorical literature review and classification”. *IFAC-PapersOnLine* 51(11):1016–1022.

Leng, J., H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang. 2018. “Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop”. *Journal of Ambient Intelligence and Humanized Computing* 10:1155–1166.

Lugaresi, G., N. Frigerio, M. Zhang, Z. Lin, and A. Matta. “Active Learning Experience In Simulation Class Using A LEGO®-Based Manufacturing System”. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lugaresi, G., and A. Matta. 2018. “Real-time Simulation in Manufacturing Systems: Challenges and Research Directions”. In *Proceedings of the 2018 Winter Simulation Conference (WSC)*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3319–3330. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Qi, Q., and F. Tao. 2018. “Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison”. *IEEE Access* 6:3585–3593.

Rhodes, D. H., and J. B. Reid. 2016. "Digital System Models: An investigation of the non-technical challenges and research needs". In *Proceedings of the 2016 Conference on Systems Engineering Research*, 572. Cambridge, MA: Massachusetts Institute of Technology.

Sanchez, A., and J. Bucio. 2012. "Improving the teaching of discrete-event control systems using a LEGO manufacturing prototype". *IEEE Transactions on Education* 55(3):326–331.

Shafto, M., R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang. 2012. "Modeling, Simulation, Information Technology and Processing Roadmap". Technical Report No. 11, NASA, Technology Area, Washington DC, USA.

Syberfeldt, A. 2010. "A LEGO factory for teaching simulation-based production optimization". In *Proceedings of the 2010 Industrial Simulation Conference*, 89–94. Ostend, Belgium: EUROSIS-ETI.

Tan, Y., W. Yang, K. Yoshida, and S. Takakuwa. 2018. "Application of IoT-aided simulation for a cyber-physical system". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 4086–4087. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Travaglini, D. 2018. *Manufacturing system based on LEGO-robotics: development of physical and digital models*. M.Sc. thesis, Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy.

Yang, W., Y. Tan, K. Yoshida, and S. Takakuwa. 2017. "Digital twin-driven simulation for a cyber-physical system in Industry 4.0". In *DAAAM International Scientific Book*, edited by B. Katalinic, 227–234. Vienna, Austria.

AUTHOR BIOGRAPHIES

GIOVANNI LUGARESI is Ph.D. Candidate in the Department of Mechanical Engineering of Politecnico di Milano. He completed his Master's Degree in Mechanical Engineering in 2016 at Politecnico di Milano. His research interests include simulation-optimization in manufacturing systems, robust optimization for production planning and control, and stochastic programming. His email address is giovanni.lugaresi@polimi.it.

DAVIDE TRAVAGLINI is Mechanical Engineering Graduate from Politecnico di Milano. He graduated in Mechanical Engineering in 2018 at Politecnico di Milano. His research interests include simulation-optimization in manufacturing systems and Industry 4.0. His email address is d.travaglini94@gmail.com.

ANDREA MATTA is Full Professor of Manufacturing at the Department of Mechanical Engineering of Politecnico di Milano and Guest Professor at the Shanghai Jiao Tong University. He graduated in Industrial Engineering at the Politecnico di Milano where he develops his teaching and research activities since 1998. He was Distinguished Professor at the School of Mechanical Engineering of Shanghai Jiao Tong University from 2014 to 2016. He is scientific responsible of the Research Area Design and Management of Manufacturing Systems at MUSP (Laboratory for Machine Tools and Production Systems). His research area includes analysis, design and management of manufacturing and health care systems. He is Editor in Chief of the Flexible Services and Manufacturing Journal, editorial board member of the OR Spectrum journal and the IEEE Robotics and Automation Letters journal. He was awarded with the Shanghai One Thousand Talent and Eastern Scholar in 2013. His email address is andrea.matta@polimi.it.