



**HAL**  
open science

## Real-time Validation of Digital Models for Manufacturing Systems: a Novel Signal-processing-based Approach

Giovanni Lugaresi, Gianluca Aglio, Federico Folgheraiter, Andrea Matta

► **To cite this version:**

Giovanni Lugaresi, Gianluca Aglio, Federico Folgheraiter, Andrea Matta. Real-time Validation of Digital Models for Manufacturing Systems: a Novel Signal-processing-based Approach. 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Aug 2019, Vancouver, Canada. pp.450-455, 10.1109/COASE.2019.8843082 . hal-03880585

**HAL Id: hal-03880585**

**<https://hal.science/hal-03880585>**

Submitted on 1 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Real-time Validation of Digital Models for Manufacturing Systems: a Novel Signal-processing-based Approach

Giovanni Lugaresi<sup>\*1</sup>, Gianluca Aglio<sup>1</sup>, Federico Folgheraiter<sup>1</sup>, Andrea Matta<sup>1</sup>

**Abstract**—Recently, the connection between manufacturing systems and their digital counterparts has become of great significance for planning and control activities in a short-term scope. However, the alignment of a digital model with a very dynamic system is not always guaranteed, and traditional validation techniques cannot be used since they are designed for off-line simulators and rely on the availability of a large amount of data. This work develops a novel validation procedure inspired by signal-processing theory and a novel approach called *quasi Trace Driven Simulation*. The procedure is coherent with a Real-Time Simulation framework since it does not require large datasets to provide a good solution. The approach has been tried on test cases which demonstrated its applicability to a manufacturing environment.

## I. INTRODUCTION

Recent economic pressure has driven manufacturing systems to become much more dynamic than before. Besides, electronic devices such as sensors and data acquisition systems reached higher reliability along with affordability, and in manufacturing environments it is possible to obtain information about the shop floor status anytime [1]. These developments paved the way for the deployment of *Cyber-Physical Systems* (CPSs) [2]. CPSs are based on the symbiotic coexistence of the real system with its digital counterpart, often called digital twin or digital model. With a production planning and control scope, the digital model of a manufacturing system can be represented by a Discrete Event Simulation (DES). The recent dynamism of production systems represents a challenge since it is not always granted that digital models are able to accurately follow the behavior of the real system. A recent survey by Skoogh et al. [3] points out how a great percentage of manufacturing companies do not reuse a simulation model after a decision is made: these models have also been labeled as *throw-away models* [4]. Indeed, if the model is not a close representation of the actual system, any conclusion derived from the simulation is likely to be erroneous and may result in erroneous decisions and unjustified costs [5]. As a result, since traditional models are typically not directly linked to the real system, they are often used only for system design and long-term decision-making [6].

In order to see the application of discrete event simulators as digital models for short-term decision-making in the practice, two main requisites are needed: (1) the possibility of generating and updating the models with a higher frequency

and (2) the ability of identifying when a model is not correctly representing its real counterpart, in other words when it is not valid. In this work, we develop a new input validation procedure that is based on signal-processing theory and we propose a new method to temporally correlate the DES model to its real counterpart through their inputs called *quasi Trace Driven Simulation (QTDS)*, which allows for the effective comparison of real and simulated data for validating digital models in a Real-Time Simulation framework.

## II. STATE OF THE ART

### A. Real-Time Simulation in Industry

Real-Time Simulation (RTS) is a recently coined term that describes the deployment of simulation technology for short-term decision-making. RTS can save considerable time and resources by allowing the same simulation model to be used and corrected when needed [7]. Manivannan and Banks [8] firstly defined the concept of RTS, also called *on-line* DES, and some of its applications can be found in the literature. Harmonosky [9] described a procedure for the selective rerouting of the parts within a shop floor. Spedding et al. [5] delineated an adaptive simulation model for a keyboard assembly cell. Framinan et al. [10] studied how data collected in real-time can be used to rearrange the sequence of jobs in a flow-shop with stochastic processing times with the goal to minimize the makespan. Xie et al. [11] developed a meta-model to calibrate and guide real-time decision-making for complex systems. These works show that simulation can be used as a tool for on-line decision support. According to Hanisch et al. [12], RTS reaches its maximum efficiency with an on-line connection between the simulation model and the real system along with a fast execution time and a valid DES model. Lugaresi and Matta [7] discussed over several features of RTS frameworks found in the literature, highlighting the most important challenges for the future research in this field: data management, model adaptability and model generation, model validation, and reactivity.

### B. Validation of Digital Models

Validation is defined as the *process of determining whether a simulation model is an accurate representation of the system, for the particular objectives of study* [14]. It compares the input-output transformation of the simulation model with the real system one by running the model using the same input conditions derived from the real system and analyzing its behavior in terms of output measures [15]. Several validation methods have been developed over the years for the off-line validation of a DES model. These

<sup>1</sup> Department of Mechanical Engineering - Politecnico di Milano - Via La Masa 1, 20156 Milano, Italy. \*Corresponding Author: giovanni.lugaresi@polimi.it

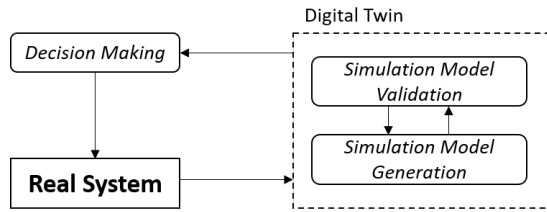


Fig. 1: Real-Time Simulation framework taken as reference in this paper.

methods are mostly based on statistical techniques allowing the comparison between output data from the DES model and the real system [16]. A first-cut evaluation of the model performance is to compute the differences between observed and predicted responses by means of statistical indicators. Some are used very often in the practice (i.e. the mean, the median, the mean squared error) whereas others are less popular although they can be used effectively: for instance, the Coefficient of Determination [17], the Coefficient of Residual Mass [17], the Modeling Efficiency [18]. Usually, a simulation model is then validated by means of statistical tests. Among others, Rebba et al. [19] combined a *t-test* statistic for the mean and *F-test* statistic for the variance. The Two-Sample Kolmogorov-Smirnov Test [20] exploits Empirical Cumulative Distribution Functions (ECDF) of the output data. Kleijnen et al. [21] developed a regression-based test for Trace Driven Simulations (TDS). Khan et al. [23] created an approach based on the comparison between the feedback from a simulation model and specified outputs of each event in the specification model. The approach checks if each event in the DES model is consistent with the event that would happen in the specification model under the same inputs.

The list of validation techniques is very long [13], [16]. However, all of them have been developed for an off-line use. Hence, they are limited when applied to RTS conditions due to their need of a large amount of data to reach a statistical significance of their results. Differently, our work aims at validating a simulation model on-line, hence directly comparing data from the real system with simulation data. To the best of our knowledge, such methods for the validation of digital models in a real-time context are not present in the literature.

### III. SCOPE OF WORK

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 System block represents the manufacturing plant. From here, data are continuously collected by tools such as the Manufacturing Execution System (MES) or the Advanced Planning and Scheduling (APS) System, and sent to the digital counterpart of the system where the generation of the DES model of the system takes place in the Simulation Model Generation block. The model building phase occurs 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.

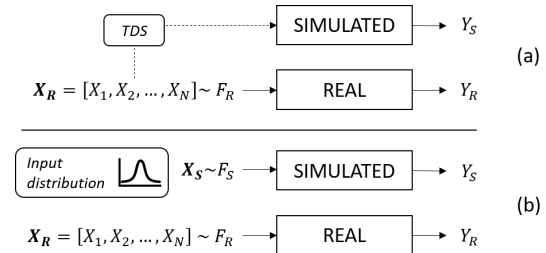


Fig. 2: Validation: (a) of the logic, (b) of the inputs.

Simulation models are validated, updated and synchronized with the current system states through the Simulation Model Validation block. The models are then exploited by the Decision Making block to provide production control actions.

Referring to the framework by Nelson [24], we may separate validation activities in two main parts: validation of the logic (Figure 2a) and validation of the inputs (Figure 2b). The validation of the logic regards, for instance, the arrangement of the machines in a layout, the routing of parts, so on. On the other hand, input validation concentrates on the probability distributions generating the model input data and the model parameters. In this work, we concentrate on the latter problem.

Furthermore, since a production system is continuously subjected to changes, only a single data history may be used. Hence, frequent changes of the real system lead to the problem of model validation with small datasets. Indeed, in RTS the time horizon is much shorter than conventional simulation and only one small dataset may be available from the real system. A possible solution is to treat Key Performance Index (KPI) data as signals. The exploitation of signal-processing techniques in simulation output analysis have been encouraged by Schruben and Coglianò [25], who used the Discrete Fourier Transform (DFT) to perform a sensitivity analysis by oscillating the input data fed to the DES model. The authors showed the advantages of such an approach for identifying the terms of a regression model of the simulation response against the inputs. Our approach is based on the same intuition: the output data of a simulation model can be treated as a signal in a frequency domain, hence two signals may be compared: the output signal generated from the simulation model and the one collected from the real system.

### IV. PROPOSED PROCEDURE

The proposed validation procedure consists in the following steps: (A) quasi-Trace Driven Simulation (QTDS) setting, (B) Batch Partitioning, (C) calculation of a validation degree indicator. QTDS is performed on input data, while the latter two points on output data.

#### A. Quasi Trace Driven Simulation (Input Data)

In an RTS framework, a digital model should properly represent the real system in every considerable time-frame. However, if input data of the simulation model (e.g., interarrival times) are generated independently from the real

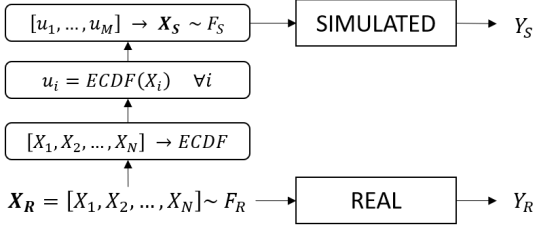


Fig. 3: Schema of the quasi Trace Driven Simulation

system input data trace, the two signals may have very different profiles (see Figure 4) and only a comparison of aggregated measures can be performed (that is, through traditional validation techniques).

Indeed, consider a real system (Figure 2) with its inputs  $\mathbf{X}_R$  following a generic, unknown distribution  $F_R$ . The distribution of the input data of the simulator  $F_S$  is usually assumed by the modeler. If the scope is the validation of the logic (Figure 2a), one way of validating the simulator is to perform a Trace Driven Simulation (TDS). In TDS the same input data of the real system are fed to the simulator; then, a comparison of the output measures  $\mathbf{Y}_R$  and  $\mathbf{Y}_S$  is done. On the other hand, if the scope is the validation of the inputs (Figure 2b), TDS is not suitable since, in this case, the mechanisms of input data generation are investigated. Hence, input data shall be generated through the same probability model that has been assumed ( $F_S$ ). In this case, comparing  $\mathbf{Y}_R$  and  $\mathbf{Y}_S$  directly might be a difficult task due to the innate noise of the Random Number Generation that makes  $\mathbf{Y}_S$  totally uncorrelated from  $\mathbf{Y}_R$  and not useful for any comparison when only a small dataset is available. QTDS is based on the intuition that with correct assumptions on the inputs, the simulation output data will show a behavior very similar to the real system, allowing for the comparison of  $\mathbf{Y}_R$  and  $\mathbf{Y}_S$ . On the other hand, assuming a distribution that is not realistic will yield to significantly different outputs.

QTDS simply generates random variates  $X_S$  starting from  $F_S$  that are highly correlated with the real system input datasets  $X_R$ . The QTDS procedure works as follows. A sample of input data  $[X_1, \dots, X_N]$  is gathered from the real system and the discrete ECDF is obtained. Then, the ECDF is used to extract a stream of numbers between 0 and 1,  $\mathbf{u} = [u_1, \dots, u_N]$ , where  $u_i = ECDF(X_i) \forall i = 1 \dots N$ . The vector  $\mathbf{u}$  constitutes the basis for the generation of the simulation model input data according to the distribution  $F_S$  assumed for the simulator, hence  $\mathbf{X}_S = \Phi_S^{-1}(\mathbf{u})$  with  $\Phi_S$  being the cumulative distribution of  $F_S$ . Notice that this way a temporal correlation is guaranteed since each input variate of the simulator refers to a single input value of the real system, and vice versa. This way each simulated datum can be compared with the related real counterpart.

### B. Batch Partitioning (Output Data)

Consider  $\mathbf{Y}_R$  and  $\mathbf{Y}_S$  the real and simulated output datasets, respectively, which could represent a certain performance index over a fixed time horizon (for instance, the system throughput over the last 30 minutes). Each of these

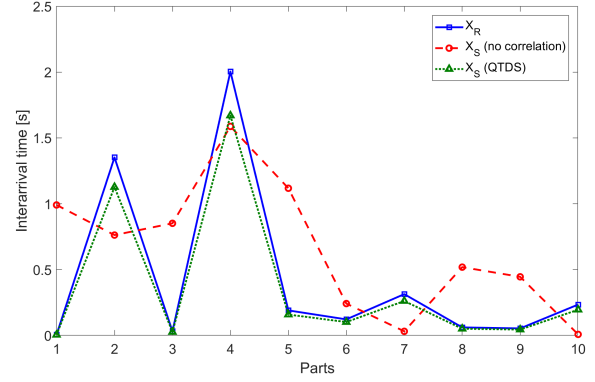


Fig. 4: Example showing datasets of ten values representing interarrival times. The  $\mathbf{X}_R$  curve represents the real system input data. The  $\mathbf{X}_S$  curves are the simulated input data. Even if generated by the same probability model (exponential distribution with mean 2 s), the  $\mathbf{X}_S$  curve with no correlation (red) shows a very different profile with respect to the  $\mathbf{X}_R$  curve, while the  $\mathbf{X}_S$  curve correlated with QTDS (green) has the same profile because it shares the same trace of random numbers.

signals is a time series and it might be affected by auto-correlation (i.e. there are lag values  $l$  that link two values of the series:  $y_i = y_{i+l} \forall i$ ) which may invalidate our analysis. To remove auto-correlation, a common countermeasure is data batching. In this step, the real and simulated input datasets are both split into  $K$  data batches through a procedure named *Batch Partitioning (BP)*:

$$\begin{aligned} \mathbf{Y}_R &= \{\mathbf{Y}_R^{(1)}, \mathbf{Y}_R^{(2)}, \dots, \mathbf{Y}_R^{(K)}\}, \\ \mathbf{Y}_S &= \{\mathbf{Y}_S^{(1)}, \mathbf{Y}_S^{(2)}, \dots, \mathbf{Y}_S^{(K)}\}. \end{aligned}$$

BP may be performed only on the real system data, while the dataset generated by the DES model is divided into batches with the same partition scheme. This practice is very common in the literature. Since the selection of a batching procedure is beyond our scope, we refer to Appendix I for the one we used.

### C. Spectral Indicator (Output Data)

In this step an indicator of the degree of validation is computed as follows:

- 1) The Power Spectral Density (PSD)<sup>1</sup> for both simulated and observed data is computed for each  $k$ -th batch:

$$\begin{aligned} \mathbf{P}_R^{(k)} &= PSD(\mathbf{Y}_R^{(k)}) \quad \forall k, \\ \mathbf{P}_S^{(k)} &= PSD(\mathbf{Y}_S^{(k)}) \quad \forall k. \end{aligned}$$

- 2) A value  $\delta$  is decided and fixed by the user. This parameter represents the size of equally-spaced intervals in which the batch is divided. Hence:

$$\begin{aligned} \mathbf{P}_R^{(k)} &= \{\mathbf{P}_{R,1}^{(k)}, \dots, \mathbf{P}_{R,i}^{(k)}, \dots, \mathbf{P}_{R,n_k}^{(k)}\}, \\ \mathbf{P}_S^{(k)} &= \{\mathbf{P}_{S,1}^{(k)}, \dots, \mathbf{P}_{S,i}^{(k)}, \dots, \mathbf{P}_{S,n_k}^{(k)}\} \end{aligned}$$

<sup>1</sup>The Power Spectral Density of a signal is defined as follows:

$$PSD = \frac{S_{xx}(f)}{\Delta f}$$

Where  $S_{xx}$  is the auto-spectral density function (i.e. the power spectrum) and  $\Delta f$  is the frequency resolution.

where  $n_k$  indicates the number of intervals in the  $k$ -th batch.

- 3) For each  $i$ -th interval in the  $k$ -th batch, a weighted average  $M_i^{(k)}$  based on the absolute difference between the spectral densities is computed as follows:

$$M_i^{(k)} = \frac{\sum_{j=1}^{t_i} \frac{|p_{R,i,j}^{(k)} - p_{S,i,j}^{(k)}|}{t_k}}{f_k} \cdot \delta \quad \forall i, k \quad (1)$$

where  $f_k$  is the sampling frequency for each batch and  $t_i$  is the number of data entries in the  $i$ -th interval.

- 4) For each batch  $k$ , the average of all the differences  $M_i^{(k)}$ , is computed as follows:

$$\bar{M}_k = \sum_{i=1}^{n_k} \frac{M_i^{(k)}}{n} \quad \forall k. \quad (2)$$

- 5) A synthetic value  $\Delta$  describes the degree of similarity between the two PSD functions. It is computed through a single mean, considering all the  $k$  batches:

$$\Delta = \frac{\sum_{k=1}^N \bar{M}_k}{N} \quad (3)$$

where  $N$  indicates the total number of batches. The indicator is then normalized between 0 and 1 by means of a negative exponential function as follows:

$$I = e^{-\Psi\Delta} \quad (4)$$

where we call  $I$  Spectral Indicator.

Finally, we have obtained a function which is equal to 1 in the origin (if the signals are equal,  $\Delta = 0$ ) and asymptotic to 0 for  $\Delta \rightarrow +\infty$ .

## V. NUMERICAL RESULTS

### A. Calibration

A Design of Experiments has been carried out to investigate the effect of the parameters of the procedure over the final value of the proposed indicator. The Spectral Indicator has been computed to compare two datasets referred to the lead time of parts flowing through an M/M/2 queue and its DES model. The M/M/2 queue has interarrival and processing times following exponential distributions with parameters  $\lambda = 0.4$  and  $\mu = 0.5$ , respectively. A full factorial study has been done on the following factors:

- the initial batch length  $L_1$ , two levels (20, 30);
- the size of the interval  $\delta$ , two levels (0.01, 0.03);
- the assumptions on the simulation input data distribution ( $F_S$ ), four levels:
  - 1) exponential, with the same parameters of the real system inputs;
  - 2) exponential, with parameters fitted to the real system inputs;
  - 3) exponential, with different parameters;
  - 4) normal, with the same mean and standard deviation of the real system inputs;

The Spectral Indicator has been computed for each different combination of factors (in total, 16 combinations), and the

Source of Variation	F <sub>0</sub> Statistic	P-Value
$\delta$	250.54	< 0.05
$L_1$	0.77	0.384
$F_S$	33.48	< 0.05
$\delta * L_1$	0	0.997
$\delta * F_S$	1.85	0.159
$L_1 * F_S$	0.97	0.418
$\delta * L_1 * F_S$	0.38	0.768

TABLE I: ANOVA table obtained during the calibration phase

experiments have been repeated three times, for a total of 48 runs. The results of the ANOVA are summarized in Table I. Residuals have been verified to be independent and identically distributed as a normal distribution. The size of the interval  $\delta$  and the type of distribution  $F_S$  used in the DES model both present a low P-Value, indicating that these two parameters significantly affect the final result; on the other hand, the choice of the initial batch length  $L_1$  does not impact on the final result. Moreover, no significant interactions among the three parameters are visible. Therefore, we may state that: (1) the batch decomposition manners are impacting on the comparison results since the value of  $\delta$  influences the amount of data that is included in each interval, (2) the choice of the initial batch length  $L_1$  does not impact the results of the analysis, and (3) the Spectral Indicator is sensitive with respect to the input data distribution  $F_S$ .

Further, with the aim to test the effectiveness of the proposed approach, we have applied it to two test cases: (1) an M/M/2 queue and (2) a flow line made of seven stations. Given the aforementioned results, in these cases the initial batch length and the size of the intervals have been set equal to  $L_1 = 20$  and  $\delta = 0.01$ , respectively. The test cases will be presented in the following subsections.

### B. Test Case 1 - M/M/2 Queue

Let us take as reference the same M/M/2 queue analyzed in Section V-A. An experimental plan has been generated by fixing real system parameters (exponential distributions with parameters  $\lambda_{real} = 0.4, \mu_{real} = 0.5$ ), the number of parts ( $\eta = [100, 200, 500, 1000, 5000, 10000]$ ), and varying the simulated system parameters in three settings:

- 1) the  $F_S$  distribution is exponential and the interarrival and process parameters are fitted to the real system input data;
- 2) the  $F_S$  distribution is exponential and the parameters are the same as in the real system;
- 3) the DES inputs follow a normal distribution  $F_S$ .

Experiments have been replicated 100 times. Figure 5 shows the mean values of  $I$  obtained for each experimental setting depending on the dataset length  $\eta$  (the maximum variance obtained is of magnitude less than  $10^{-3}$ ). As expected, the first setting is the best case, since a fitting procedure is performed on each dataset coming from the real system. As the dataset dimension increases, the exponential distribution approximates better the real dataset distribution, and as a consequence  $I$  increases. The mean values of  $I$  obtained for the third setting (normal distribution) are much

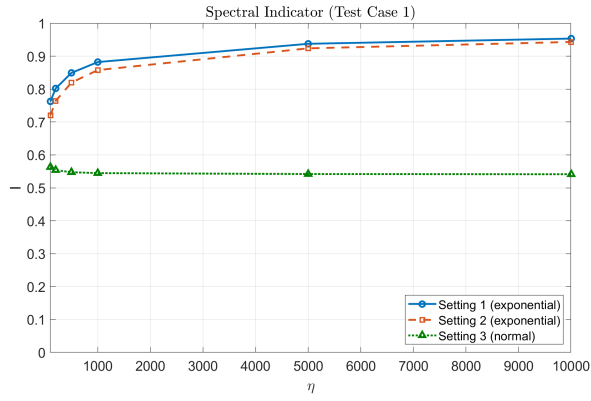


Fig. 5: Test Case 1 - Results obtained for the Spectral Indicator  $I$  (mean values)

smaller. Indeed, this case shows the worst behavior since the DES model input distribution  $F_S$  is incorrect. Notice that the obtained results are valid for any dataset length that we have tested. The same settings have been done performing a Kolmogorov-Smirnov Test [20], which with datasets smaller than  $\eta = 5000$  could never identify the difference between Setting 1 and 3 (P-Value = 0.85).

In addition, a Latin Hypercube Square (LHS) design of experimental points has been created by fixing the real systems parameters at  $\lambda_{real} = 0.4$  and  $\mu_{real} = 0.5$  and varying the parameters of the DES model so that  $\mu_{sim} \in [0.2, 0.6]$  and  $\lambda_{sim} \in [0.3, 0.7]$ . 100 replications have been done for a dataset with  $\eta = 150$  parts. Figure 6 shows the average values of the Spectral Indicator that have been obtained. The maximum variance among the outputs is equal to 0.0084, hence the results are fairly concentrated around the mean. Again, the highest values are achieved when the DES parameters are close to the real system ones.

### C. Test Case 2 - Flow Line

In this case, we considered a multi-product production line composed by 7 workstations whose processing times are each exponentially distributed with mean  $\mu_{real} = 1$  and with parts interarrival times exponentially distributed with mean  $\lambda_{real} = 1$ . Each workstation follows a Blocking Before Service policy. Experiments have been carried out with the Spectral Indicator  $I$  referred to the production lead time. Two settings of experiments have been generated as follows:

- 1) variations of  $\pm 0.1$  in the values of  $\mu_{real}$ ;
- 2) different distributions of the simulation input data  $F_S$ , namely (2a) exponential distribution and (2b) normal distribution;

The experiments have been replicated 100 times. Figure 7 shows the results obtained. Also in this case, if the input datasets are correct the Spectral Indicator shows the highest values. Further, with deviations of  $\pm 0.1$  on the mean value of the processing times, the values of  $I$  drop as lower values correspond to a greater input error, and this is valid for any dataset length. Similarly to the first test case, lower values of  $I$  occur when the assumption on the distribution is erroneous, as it can be noticed from the results of the second setting.

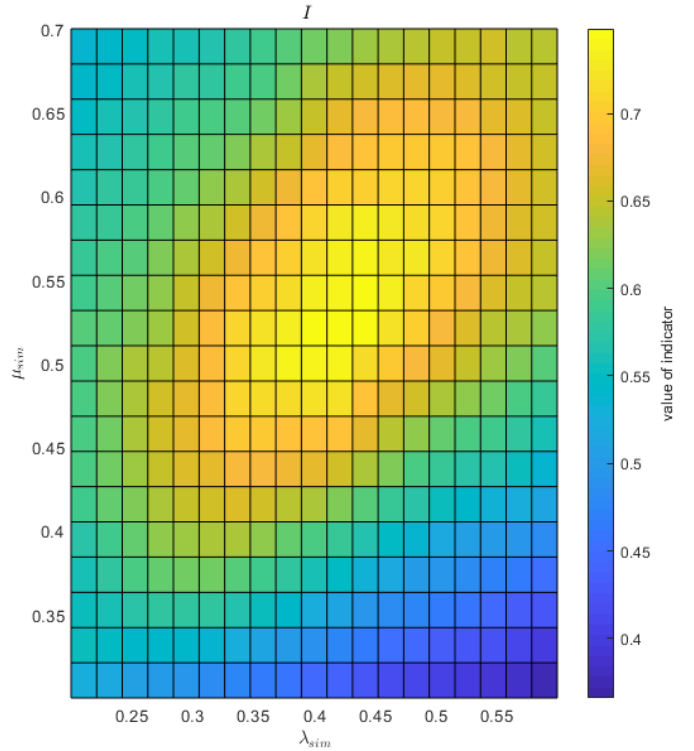


Fig. 6: Test Case 1 - Color map of the mean values of  $I$  obtained: the real system inputs are  $\lambda_{real} = 0.4$  and  $\mu_{real} = 0.5$ . The indicator  $I$  shows its highest peak over the correct values ( $\lambda_{real} = \lambda_{sim}$ ;  $\mu_{real} = \mu_{sim}$ ).

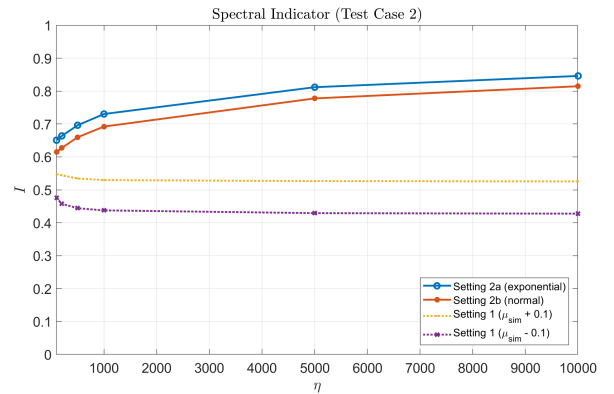


Fig. 7: Test Case 2 - Spectral Indicator computed for different dataset lengths (mean values).

## VI. CONCLUSIONS AND FUTURE RESEARCH

A novel procedure for validating digital models in a real-time context has been presented. The method provides an indicator which can help an industrial user to understand if a digital model is an accurate representation of the real production system. Experiments showed that small errors in the estimation of the input data are recognized even with reduced datasets. The proposed approach is beneficial for Real-time Simulation applications, in which real feeds of data are used to validate a simulation model, thus can positively contribute to the application of discrete event simulation as a short-term decision-making tool. In the future, the proposed procedure should be tested on more complex manufacturing systems.

Further, the approach could also be applied to compare other KPI measures. Last but not least, the means for estimation of the risk of accepting a wrong model shall be investigated.

## APPENDIX I: BATCH PARTITIONING

Following is the batching procedure used in this work:

- 1) An initial batch length  $L_1$  is fixed, together with a minimum length,  $L_{min}$ .
- 2) The sampling frequency  $f_S$  is set.
- 3) The first batch is created, with initial size  $L_2 = L_1$ .
- 4) The period  $\Theta$  is computed by means of the auto-correlation function<sup>2</sup>.
- 5) If the batch size  $L_2$  is an integer multiple of its period, the batch is kept. Otherwise, the batch size is reduced and the algorithm goes back to step 4, until the equality is satisfied. In case this condition is not guaranteed and the minimum batch length  $L_{min}$  is reached,  $L_1$  is used and the dataset is sampled with a Hanning Window [27].
- 6) The partitioning continues until all the data are grouped into batches. If the last batch does not contain enough data, it is discarded.

## REFERENCES

- [1] C.-C. Chen, C.-L. Chen, C.-Y. Ciou, and J.-X. Liu, "Communication scheduling scheme based on big-data regression analysis and genetic algorithm for cyber-physical factory automation," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, October 9<sup>th</sup>-12<sup>th</sup>, Budapest, Hungary*, pp. 2603–2608, IEEE, 2016.
- [2] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in cps-based production systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017.
- [3] A. Skoogh, T. Perera, and B. Johansson, "Input data management in simulation—industrial practices and future trends," *Simulation Modelling Practice and Theory*, vol. 29, pp. 181–192, 2012.
- [4] S. Tavakoli, A. Mousavi, and A. Komashie, "A generic framework for real-time discrete event simulation (des) modelling," in *Proceedings of the 40th Conference on Winter Simulation*, pp. 1931–1938, Winter Simulation Conference, 2008.
- [5] T. A. Spedding, W. L. Lee, R. de Souza, and S. Lee, "Adaptive simulation of a keyboard assembly cell," *Integrated Manufacturing Systems*, vol. 8, no. 1, pp. 50–58, 1997.
- [6] P. Barlas and C. Heavey, "Automation of input data to discrete event simulation for manufacturing: A review," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 7, no. 01, p. 1630001, 2016.
- [7] G. Lugaresi and A. Matta, "Real-time simulation in manufacturing systems: Challenges and research directions," in *2018 Winter Simulation Conference (WSC)*, pp. 3319–3330, IEEE, 2018.
- [8] Y. Rao, F. He, X. Shao, and C. Zhang, "On-line simulation for shop floor control in manufacturing execution system," in *International Conference on Intelligent Robotics and Applications*, pp. 141–150, Springer, 2008.
- [9] C. M. Harmonosky, R. H. Farr, and M.-C. Ni, "Selective rerouting using simulated steady state system data," in *wsc*, pp. 1293–1298, IEEE, 1997.
- [10] J. M. Framinan, P. Perez-Gonzalez, and V. F.-V. Escudero, "The value of real-time data in stochastic flowshop scheduling: A simulation study for makespan," in *2017 Winter Simulation Conference (WSC)*, pp. 3299–3310, IEEE, 2017.
- [11] W. Xie, P. Zhang, and Q. Zhang, "A stochastic simulation calibration framework for real-time system control," in *Proceedings of the 2017 Winter Simulation Conference*, p. 150, IEEE Press, 2017.
- [12] A. Hanisch, J. Tolujuw, and T. Schulze, "Initialization of online simulation models," in *Proceedings of the Winter Simulation Conference, 2005.*, pp. 9–pp, IEEE, 2005.
- [13] R. G. Sargent, "Verification and validation of simulation models," *Journal of simulation*, vol. 7, no. 1, pp. 12–24, 2013.
- [14] A. M. Law, "How to build valid and credible simulation models," in *2008 Winter Simulation Conference*, pp. 39–47, IEEE, 2008.
- [15] O. Balci, "Principles and techniques of simulation validation, verification, and testing," in *Proceedings of the 27th conference on Winter simulation*, pp. 147–154, IEEE Computer Society, 1995.
- [16] O. Balci, "Validation, verification, and testing techniques throughout the life cycle of a simulation study," *Annals of operations research*, vol. 53, no. 1, pp. 121–173, 1994.
- [17] K. Loague and R. E. Green, "Statistical and graphical methods for evaluating solute transport models: overview and application," *Journal of contaminant hydrology*, vol. 7, no. 1-2, pp. 51–73, 1991.
- [18] D. Mayer and D. Butler, "Statistical validation," *Ecological modelling*, vol. 68, no. 1-2, pp. 21–32, 1993.
- [19] R. Rebba, S. Huang, Y. Liu, and S. Mahadevan, "Statistical validation of simulation models," *International Journal of Materials and Product Technology*, vol. 25, no. 1-3, pp. 164–181, 2006.
- [20] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.
- [21] J. P. Kleijnen, B. Bettonvil, and W. Van Groenendaal, "Validation of trace-driven simulation models: a novel regression test," *Management Science*, vol. 44, no. 6, pp. 812–819, 1998.
- [22] R. B. Whitner and O. Balci, "Guidelines for selecting and using simulation model verification techniques," in *Proceedings of the 21st conference on Winter simulation*, pp. 559–568, ACM, 1989.
- [23] A. Khan, M. Dahl, P. Falkman, and M. Fabian, "Digital twin for legacy systems: Simulation model testing and validation," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 421–426, IEEE, 2018.
- [24] B. Nelson, *Foundations and methods of stochastic simulation: a first course*. Springer Science & Business Media, 2013.
- [25] L. W. Schruben and V. J. Coglianò, "Simulation sensitivity analysis: A frequency domain approach," in *Proceedings of the 13th conference on Winter simulation-Volume 2*, pp. 455–459, IEEE Press, 1981.
- [26] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.
- [27] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

<sup>2</sup>The auto-correlation function is defined as:

$$\hat{R}_{xx} = \frac{1}{T - \tau} \int_0^{T-\tau} x(t) \cdot x(t + \tau) dt$$

where  $x(t)$  is the signal (the dataset),  $T$  is the total length of observation, and  $\tau$  the imposed delay. The distance between each peak in the  $\hat{R}_{xx}$  function represents the period  $\Theta$ .