POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Executive Summary of the Thesis

Real-time validation and online input parameters calibration within a digital twin framework: an approach based on Bootstrap particle filtering

Laurea Magistrale in Mechanical Engineering - Ingegneria Meccanica

Author: Abdirahman Said Dhore

Advisor: Prof. Andrea Matta

Co-advisor: Prof. Giovanni Lugaresi

Academic year: 2024-2025

1. Introduction

The main objective of the presented executive summary is to provide a critical overview of the thesis covering the development of a real-time validation and input calibration framework for Discrete Event Simulation (DES) models within Digital Twin architectures for manufacturing systems in Industry 4.0 environments. The inspiration for this thesis stems from the growing reliance on Digital Twin (DT) technology in modern manufacturing, which enables the optimisation of production processes and facilitates real-time decision-making capabilities. However, maintaining the accuracy of Discrete Event Simulation-based twins is difficult due to "model drift". This phenomenon presents significant challenges for manufacturing organisations seeking to leverage Digital Twin technology for operational excellence and competitive advantage. Current validation practices detect mismatches between digital models and physical systems but rarely diagnose whether discrepancies arise from drifting parameters, structural modelling errors, or data quality issues. Furthermore, validation and calibration processes are often disconnected, with diagnostic insights failing to guide parameter adjustments effectively. This limitation leads to generic recalibration approaches that treat symptoms rather than addressing underlying causes.

This thesis addresses these critical gaps by developing an online framework that integrates real-time validation, systematic root cause analysis, and automated parameter calibration. The proposed methodology applies sequence-based validation metrics adopted from [1, 3], including Dynamic Time Warping (DTW), Longest Common Subsequence (LCSS), and modified Longest Common Subsequence (mLCSS) for detecting discrepancies between physical and digital systems. For diagnostic purposes, the framework employs Theil's Inequality Coefficient (TIC) to identify the sources of validation failures [4], while Bootstrap Particle Filter (BPF) algorithms provide automated calibration capabilities [2].

The integration of these components forms a closed-loop system that sustains Digital Twin fidelity without requiring manual intervention. The framework's effectiveness has been demonstrated through pilot studies involving simplified *queuing systems and multi-station manufacturing scenarios, as well as laboratory-scale implementation on a five-station production line. These validation efforts demonstrate the system's ability to detect parameter shifts, trace faulty input parameters, and restore model accuracy in real-time operational environments.*

2. Proposed Methodology

To maintain alignment between simulation and reality, an online validation and calibration framework shown in Figure 1 was purposed. The framework forms a closed-loop cycle with three components: sequence-based validation, root-cause identification, and online calibration. the main contribution of the thesis is highlighted.

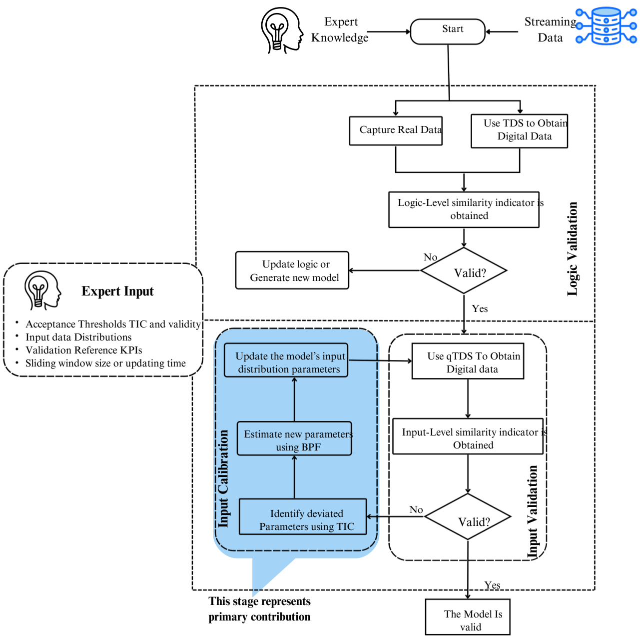


Figure 1: Proposed online validation and calibration procedure (TIC: Theil's Inequality Coefficient; BPF: Bootstrap Particle Filter; TDS: Trace-Driven Simulation; qTDS: quasi-Trace-Driven Simulation).

2.1. Sequence-Based Validation

At regular intervals, the DT's outputs are compared with real-system outputs using DTW, LCSS, or mLCSS. These metrics accommodate temporal misalignment and are normalized to $[0, 1]$. If similarity falls below a set threshold, the model is flagged as invalid for that interval.

2.2. Root Cause Identification and Online Calibration

When a validation failure is detected, the framework performs diagnostic analysis to identify the input causing the deviation. A sliding window examines recent histories of key inputs of the model and compares them with corresponding real-system data. For each input, Theil's Inequality Coefficient (TIC) is computed as a normalised measure (0–1) of discrepancy between the two sequences. Formally, for real inputs $a = \{a_i\}_{i=1}^n$ and simulated inputs $b = \{b_i\}_{i=1}^n$, TIC is defined as:

$$C(a, b) = \frac{\sqrt{\sum_{i=1}^n (a[i] - b[i])^2}}{\sqrt{\sum_{i=1}^n a[i]^2} + \sqrt{\sum_{i=1}^n b[i]^2}} \quad (1)$$

TIC values range from 0 (perfect match) to 1 (maximum divergence). Inputs exceeding a threshold θ_{TIC} are flagged as root causes and sent for calibration.

Calibration is performed using a **Bootstrap Particle Filter (BPF)**, which represents the drifted parameter as a dynamic state using N particles. With each data window, the BPF updates candidate parameter values through prediction, weighting based on TIC-derived likelihoods, and resampling. Over successive iterations, the particles converge toward values that best explain the observed data, incrementally realigning the model with the physical system.

The BPF is particularly suited for this task because it allows **real-time, online parameter updates** while maintaining a **flexible probabilistic representation** of uncertainty. Unlike batch-based methods such as MCMC or reinforcement learning, which require extended convergence periods, the BPF naturally handles streaming data, making it ideal for dynamic manufacturing systems. Algorithm 1 summarizes this windowed bootstrap particle filter–based calibration procedure.

Notation:

- $\mathbf{a} = \{a_{[i]}\}_{i=1}^n$: Real input sequence of the diverged parameter during sliding window W.
- θ_k: Parameter set of the input distribution in particle k
- $\mathbf{b}^{(k)} = \{b_{[i]}^{(k)}\}_{i=1}^n$: Simulated input sequence for the digital model using particle θ_k

- $f_{\theta_k}^{-1}(u_{[i]})$: Inverse CDF of distribution with parameters θ_k
- $C(\mathbf{a}, \mathbf{b}^{(k)})$: TIC between real and simulated input

Algorithm 1 Windowed Bootstrap Particle Filter Calibration

Require: Real input sequence $\mathbf{a} = \{a[i]\}_{i=1}^{n}$, window size W, particle count N, iterations T, TIC threshold θ_{TIC}, distribution D

for $w = 1$ **to** $n - W + 1$ **step** W **do**
  Define window $\mathbf{a}_{[w:w+W-1]}$ and initialize particles $\theta_k \sim \text{Prior}(\mathbf{a}_{[w:w+W-1]})$
  $C^{\text{best}} \leftarrow \infty$, $\theta^{\text{best}} \leftarrow \emptyset$
  for $t = 1$ **to** T **do**
    for each particle θ_k **do**
      Generate trace $\mathbf{b}^{(k)}$ via inverse-CDF sampling
      Compute TIC: $C_k = C(\mathbf{a}_{[w:w+W-1]}, \mathbf{b}^{(k)})$
      if $C_k < C^{\text{best}}$ **then**
        $C^{\text{best}} \leftarrow C_k$, $\theta^{\text{best}} \leftarrow \theta_k$
      end if
      Assign weight $w_k \leftarrow \exp(-C_k)$
    end for
    Normalize weights $\tilde{w}_k \leftarrow w_k / \sum_j w_j$ and resample N particles
    Add small random noise to particles
  end for
  Generate correlated trace $\mathbf{b}^*$ from θ^{best} and compute $C^* = C(\mathbf{a}_{[w:w+W-1]}, \mathbf{b}^*)$
  if $C^* < \theta_{\text{TIC}}$ **then**
    Accept θ^{best} and update model
  else
    repeat calibration for this window
  end if
end for
return Final parameter estimates

These steps shown in figure 2 form a closed-loop validation–calibration cycle. At each time step, the DT validates itself against the real system. If valid, it operates normally; if not, it diagnoses the cause and re-calibrates accordingly. This cycle repeats, enabling the DT to adapt continuously to system changes. Sequence-based detection catches minor drifts early, TIC analysis isolates the responsible parameter, and calibration corrects it before. As a result, the DT remains a reliable mirror of the production process, supporting trustworthy decision-making and control.

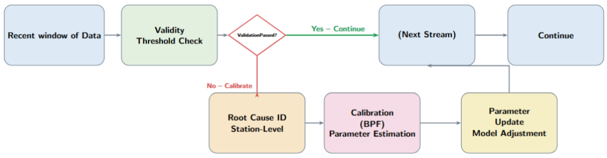


Figure 2: Work flow

2.3. Experimental Case Studies

To evaluate the proposed methodology, two case studies were conducted: (1) a controlled pilot case realized as DES model in Arena software, and (2) a real-world lab-scale production line serving as a physical twin. These studies allow testing the framework in both a simulated environment and a tangible environment.

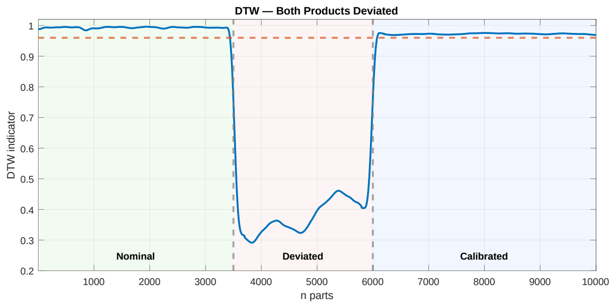
2.3.1 Pilot Case Study

The pilot case study validated the proposed framework through controlled Arena simulation experiments across three phases of increasing complexity:

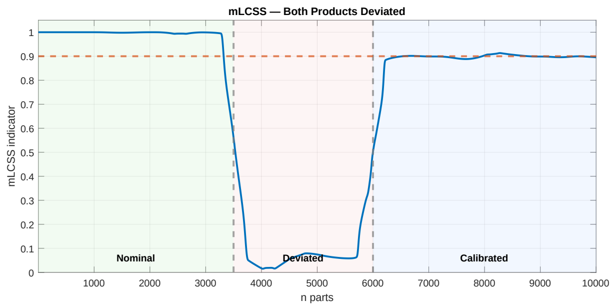
- **Phase 1:** M/M/1 queuing system with single parameter drift introduction.
- **Phase 2:** Single-station, two-product system testing product-specific parameter identification.
- **Phase 3:** Two-station, two-product manufacturing line with bottleneck effects.

Throughout these phases, the validation algorithms (DTW and mLCSS) monitored the simulation vs. "real" output sequences in real time. The TIC-based root cause analysis was performed whenever a validation metric dropped below its threshold, and the Bootstrap Particle Filter then tried to recalibrate the flagged parameter online.

Results: The controlled experiments demonstrated the effectiveness of the proposed approach. In Phase 1, even a simple drift was promptly detected by both DTW and mLCSS metrics, but interestingly mLCSS proved more sensitive to small parameter changes than DTW. Across all phases, it was observed that DTW tends to be conservative whereas mLCSS could flag a deviation sooner. For example, in scenario A a minor 5% increase in service time caused mLCSS to fall below 90% (triggering a validation alert) while the DTW similarity stayed around 97% (just above its 96% threshold). This indicates mLCSS is better at catching small changes, though it may also be more prone to false alarms if not tuned. Theil's Inequality Coefficient (TIC) analysis isolated the correct parameter in every single trial where a drift was introduced. Even in the multi-product Phase 3, the TIC values for the truly drifted parameter spiked significantly higher than for other inputs, clearly pointing to the culprit.



(a)



(b)

Figure 3: (A) DTW — scenario G; (B) mLCSS — scenario G.

After root cause identification, the Bootstrap Particle Filter calibration was invoked. Throughout the experiments, as shown in table 1 the BPF consistently adjusted the model parameters with high accuracy and speed. In quantitative terms, the calibration achieved about 94–98% accuracy in recovering the true parameter values, and it did so within a short window of 15–25 processed parts (on average) after the detection of a drift. This rapid convergence is crucial for real-world applications, as it minimizes the duration for which the model is invalid. Figure 3 compares DTW and mLCSS for scenario G and shows how quickly the calibration brings the similarity back above threshold.

Table 1: Bootstrap Particle Filter Parameter Recovery Performance (10 Replications)

Phase	Scen.	Param	True	Est	95% CI	Rel Err (%)
I	A	μ	4.0	4.02	[3.98, 4.06]	0.5
		σ	1.0	0.98	[0.94, 1.02]	2.0
	B	μ	4.0	4.10	[4.05, 4.15]	2.5
		σ	1.0	0.94	[0.92, 0.96]	6.0
	C	μ	4.0	4.10	[4.05, 4.15]	2.5
		σ	1.0	0.94	[0.92, 0.96]	6.0
	D	μ	4.0	4.10	[4.05, 4.15]	2.5
		σ	1.0	0.93	[0.90, 0.96]	7.0
II	E	μ_A	4.0	4.10	[4.04, 4.20]	2.5
		σ_A	1.0	0.83	[0.79, 0.87]	17.0
	F	μ_B	5.0	5.08	[5.02, 5.14]	1.6
		σ_B	1.0	0.96	[0.94, 0.98]	4.0
	G	μ_A	4.0	4.07	[4.02, 4.10]	1.75
		μ_B	5.0	5.08	[5.00, 5.16]	1.6
		σ_A	1.0	0.94	[0.91, 0.97]	6.0
		σ_B	1.0	0.94	[0.91, 0.97]	6.0
	H	Shape	3.0	3.20	[3.13, 3.30]	6.67
		Scale	4.2	4.25	[4.20, 4.30]	1.19
III	I	μ_A	4.0	4.07	[4.02, 4.12]	1.75
		μ_B	5.0	5.08	[5.03, 5.13]	1.6
		σ_A	1.0	0.95	[0.93, 0.97]	5.0
		σ_B	1.0	0.97	[0.95, 0.99]	3.0
	J	μ_A	3.0	3.11	[3.07, 3.16]	3.67
		μ_B	4.0	4.04	[4.00, 4.09]	1.0
		σ_A	1.0	0.95	[0.93, 0.98]	5.0
		σ_B	1.0	0.94	[0.92, 0.97]	6.0

Overall, the digital simulation study established baseline performance benchmarks for the framework. It confirmed that: (a) the validation metrics can detect a wide range of deviation magnitudes, (b) TIC reliably diagnoses the right parameter among several, and (c) the particle filter can correct the model efficiently. It also highlighted that bottleneck parameters cause more severe validation failures and are thus even easier to detect. These findings give confidence that the methodology can handle real-world scenarios, and they provide guidelines.

3. Lab-Scale Production Line Case Study

The second case study involved a physical lab-scale manufacturing system to evaluate the framework in a real-world setting, figure 4 shows Schema of the system. It is a five-station closed-loop production line built with LEGO® Mindstorms components, figure 5 shows the system. Each station (S1–S5) has its own sensors and actuators and performs a processing step; conveyors link the stations, circulating pallets in a loop and stations S3 and S4 are parallel. The Digital Twin of this line was implemented as fol-

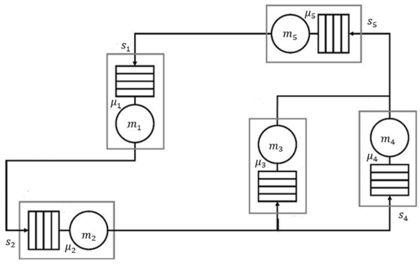


Figure 4: Schema of the system

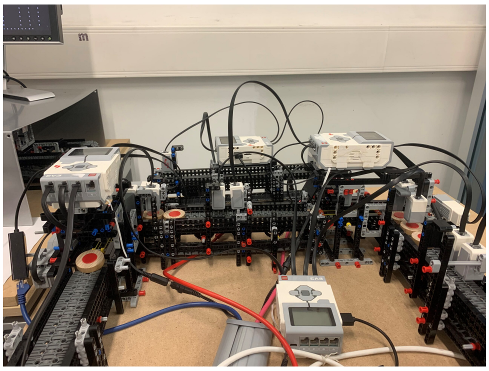


Figure 5: Lab-scale manufacturing system available at the Department of Mechanical Engineering

lows: an Arena DES model serves as the model interface representing the line's logic and timing; a system controller orchestrates the simulation runs and data flow; a validation engine continuously computes sequence similarity metrics between real and simulated event logs; a calibration engine performs TIC analysis and particle filter updates; and a dashboard provides real-time visualization of system status, validation indicators, and calibration actions. All these components communicate via a networked interface to guarantee the real system and its digital twin exchange data in real time.

For our experiments, the lab line was operated under nominal conditions to establish a baseline, and then an induced degradation was introduced to test the DT's response. Specifically, we focused on Station S1's processing time as the point of degradation. Initially, each station's processing time followed a normal distribution with parameters given in Table 2. Then S1's mean processing time was increased by 25% (to 15 seconds) and doubled its variance, simulating a scenario of a machine slowing down. The validation engine was configured with an mLCSS threshold of 90% for similarity (based on prior

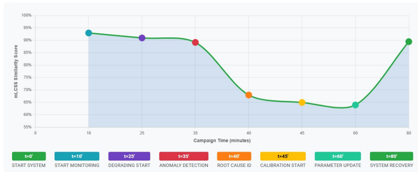


Figure 6: Validation trends across the experimental campaign

tuning). The TIC threshold for flagging a drifted input was set to 0.1.

Table 2: Nominal system parameter values.

Station	Processing Time (s)	Capacity	Transfer Time (s)
S_1	N(12, 0.7)	5	4.4 to S_2
S_2	N(5, 0.7)	6	14.0 to S_3, 14.2 to S_4
S_3	N(7, 0.7)	5	2.4 to S_5
S_4	N(12, 0.7)	6	3.6 to S_5
S_5	N(12, 0.7)	5	5.0 to S_1

Results: The experiment demonstrated autonomous Digital Twin model maintenance, as illustrated in figure 6. Initial monitoring at t=10 minutes established 93.0% mLCSS baseline similarity, declining to 91.0% at t=25 minutes with controlled degradation introduction and reaching validation failure at 89.2% at t=35 minutes. notably, the validity indicator continued declining while the system diagnosed the root cause, dropping to 68.0% at t=40 minutes during TIC analysis, and further declining to 65.0% at t=45 minutes while Bootstrap Particle Filter calibration initiated. The indicator reached minimum 64.0% at t=60 minutes while parameters were being updated with convergence, achieving "Updated S1 parameters: [15.0, 1.4] → [12.023, 0.510]", before systematic recovery restored performance to 89.5% at t=80 minutes. The experiment events timeline in (min) is shown in figure 7

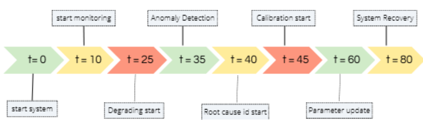


Figure 7: Experiment time tine (min)

The observed time delays occur because validity similarity is measured at output KPIs while degradation is introduced at input parameters, requiring time for effects to propagate through the system until enough parts are produced. Similarly, recovery takes time as the system must finish processing affected parts already in the system after parameter updates. This results in gradual rather than sudden changes during both degradation detection and recovery phases. The complete autonomous cycle from degradation detection to recovery (45 minutes) validates real-time model maintenance capabilities, establishing feasibility for industrial Digital Twin applications requiring continuous fidelity assurance.

4. Key Findings and Contributions

This research contributes both theoretical and practical advancements to the field of Digital Twin modeling for manufacturing. The key findings and contributions can be summarized as follows:

- Integrated Validation–Calibration Framework: A combination of sequence similarity validation with automated parameter correction, creating closed-loop model maintenance.
- Root Cause Identification via TIC: Systematic diagnostic approach enabling targeted parameter identification rather than system-wide recalibration.
- Online Calibration with Particle Filter: Real-time parameter estimation specifically adapted for DES models, overcoming limitations of offline calibration methods.Achieved 94-98% accuracy with rapid model recovery for manufacturing applications.
- Operational Dashboard System: Web-based interface providing real-time monitoring, validation results, and calibration activities for practical industrial deployment.

In conclusion, the thesis contributes a methodology that ensures persistent alignment between the digital model and the physical system, thereby reducing one of the main barriers to deploying Digital Twins in dynamic manufacturing environments.

Limitations

The framework presents several limitations that may influence its broader applicability. It requires empirical tuning of threshold parameters. It assumes primarily parametric discrepancies, meaning that structural model errors continue to demand human intervention. It is associated with computational overhead, which can constrain use in high-frequency or large-scale systems. The detection-to-recovery cycle of approximately 45 minutes may be insufficient for manufacturing environments that require rapid adaptive responses. Finally, industrial-scale deployment will necessitate additional engineering effort.

Future Research Directions

Future work may focus on adaptive thresholding through machine learning for intelligent adjustment of validation criteria. It should also address multi-parameter calibration using advanced data assimilation approaches. The integration of reinforcement learning could enable agents to develop optimal calibration policies through iterative adjustments. Finally, industrial deployment in real production environments is needed to evaluate robustness under noise, varying modes, and unforeseen events.

References

[1] Sofia Gangemi and Giulia Gazzoni. Real - time validation of discrete event simulation models in a digital twin framework: an approach based on sequence comparison techniques, 1 2021.

[2] Jiaming Hu and Hongyong Yan. Data assimilation for online model calibration in discrete event simulation. *Simulation Modelling Practice and Theory*, 135:102840, 2024.

[3] Giovanni Lugaresi, Sofia Gangemi, Giulia Gazzoni, and Andrea Matta. Online validation of digital twins for manufacturing systems. *Computers in Industry*, 150, 6 2023.

[4] Jiao Song, Li Wei, and Yang Ming. A method for simulation model validation based on theil's inequality coefficient and principal component analysis. In Gary Tan, Gee Kin Yeo, Stephen John Turner, and Yong Meng Teo, editors, *AsiaSim 2013*, pages 126–135, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.