A Remote and Intelligent Supervision Platform for Cyber-Physical Systems using Digital Twins

Tiago Lopes
University of Coimbra
CISUC/LASI, Dept. of Informatics Engineering
Coimbra, Portugal
tiagolopes@student.dei.uc.pt

Alberto Cardoso
University of Coimbra
CISUC/LASI, Dept. of Informatics Engineering
Coimbra, Portugal
alberto@dei.uc.pt

***Abstract*—This paper presents the development of a modular and scalable platform for the remote and intelligent supervision of Cyber-Physical Systems (CPS). In the context of Industry 5.0, there is an increasing demand for human-centric monitoring solutions. The proposed architecture integrates Digital Twins to mirror physical assets in real-time, utilizing Edge computing for high-frequency data acquisition (up to 10kHz). Key features include a dynamic plugin system for proprietary mathematical models, a synchronized control loop for closed-loop experiments, and a resource management system inspired by WebLab-Deusto. Preliminary results using a thermal-electric plant demonstrate the platform's effectiveness in synchronizing virtual representations with physical hardware.**

***Keywords*— Cyber-Physical Systems, Digital Twins, Industry 5.0, Remote Laboratories, Intelligent Supervision.**

I. Introduction

The industrial landscape is undergoing a significant transformation. While Industry 4.0 focused on efficiency and automation, Industry 5.0 introduces a human-centric approach, emphasizing sustainability and resilience [1]. At the core of this evolution are Cyber-Physical Systems (CPS), which require advanced monitoring capabilities to ensure safe and efficient human-machine collaboration.

Traditional remote laboratories have evolved significantly from simple web-interfaces to complex interactive environments. Several reference architectures have established the standard for hardware sharing. Platforms such as WebLab-Deusto [2] have pioneered federated access and resource management, while the iLab Shared Architecture (ISA) provides a robust framework for distributed experiments. However, many of these systems still suffer from high latency or rigid configurations that limit the researcher's ability to test custom control laws.

This work distinguishes itself from these established platforms by introducing a highly flexible "Lab-as-a-Service" (LaaS) model. Unlike the rigid monitoring logic of traditional systems, the proposed architecture allows researchers to dynamically "hot-swap" Python-based control laws and Digital Twin models via a plugin system. This enables real-time comparison between centralized Cloud control and distributed Edge control, a feature often restricted in proprietary solutions

Recent trends suggest a shift towards the 'Lab-as-a-Service' (LaaS) model, where the integration of Digital Twins and Edge Computing allows for a more flexible and intelligent supervision [3]. The integration of Digital Twins (DT) emerges as a solution to provide a 'living' virtual representation that mirrors the physical asset's performance and behaviour in real-time [4]. Recent studies also highlight the importance of low-cost devices to democratize access to these systems [5].

II. Methodology

The development of this platform followed an Action-Research and Incremental Development methodology. This approach was divided into two main phases: 1) Investigation and Analysis: consisted of a systematic review of the state-of-the-art regarding Digital Twins and remote experimentation, followed by the requirement specification (functional and non-functional); 2) Development and Validation: an iterative process where modules (Acquisition, Orchestration, Intelligence) were built and tested. The validation used a "Spiral" model, starting with virtual simulations and progressing to real hardware interaction with a thermal-electric plant.

III. System Architecture

The platform is built on a modular, data-centric architecture designed to minimize latency and maximize scalability. It is divided into four main layers:

A. Acquisition and Edge Layer

The hardware interface (Arduino/ESP8266) operates in a "Burst Mode", collecting samples at frequencies up to 10kHz. This raw data is processed at the edge to reduce network congestion. Communication with the server is performed via aRESTful API and WebSockets (WSS).

B. Orchestration and Multithreading

The backend, developed in Flask, implements a multithreaded motor where each laboratory scenario runs in an independent thread. This ensures that a failure or high computational load in one experiment does not affect the others.

C. Digital Twin and Intelligence Module

Instead of static monitoring, the system implements dynamic Digital Twins. The platform supports: a) Standard Models: first-order and linear equations; b) Custom Models: a plugin-based system using importlib that allows users to upload .py files. These models have access to a circular buffer of the last N historical values (e.g., 1200 points at 20Hz), enabling the implementation of complex control laws.

The architecture is inherently scalable (N-Channels), meaning the Intelligence Module and the Human-Machine Interface (HMI) dynamically adapt to the number of sensors and actuators defined in the scenario's JSON configuration. This modularity ensures that the computational overhead scales linearly with the complexity of the experiment, maintaining system stability.

D. Security and Management

Security is ensured via TLS/SSL encryption for all data traffic. A resource management system (Booking) prevents hardware conflicts by ensuring that only one user has control during a specific time slot, with an automated "kickout" mechanism when the session expires.

IV. Implementation Details

A. Synchronized Control Loop

Following the requirement for real-time control, the system operates on a synchronized cycle. In each time step (Ts = 50ms), the Python motor performs three atomic actions: a) Preparation: calculates the next control action (if in closed-loop); b) Execution: sends the command (u) and receives the sensor data (y) in a single HTTPS request to avoid "bottlenecks"; c) Update: updates the Digital Twin and broadcasts data to the HMI via Socket.IO.

To ensure deterministic behavior and avoid network congestion, the system separates the high-frequency sampling rate (Ts=100µs at the Edge) from the transmission rate (Ttx=50ms to the Cloud). A soft real-time scheduler measures the execution time of each cycle; and if the sum of computation and network latency exceeds the 50ms target, the system issues a QoS alert to the user.

B. Periodic Signal Generator

The platform includes a periodic signal builder. Users can define vectors for Time (T) and Value (U or Ref). Using modulo arithmetic (t (mod *Tmax*)), the system repeats the defined sequence, allowing for continuous identification or stress tests.

V. Experimental Validation

The platform was validated using a thermal-electric plant scenario.

A. Phase 1: System Identification

Using the automated step generation, data was collected in CSV format. The "post-facto timestamping" technique allowed the reconstruction of the time axis with 100µs precision, despite the network jitter. This data enabled the calibration of the system's Gain (K) and Time Constant (τ).

Quantitative analysis of the collected data shows that the "post-facto timestamping" achieves a synchronization accuracy of 100 μs between samples. During trials, the average end-to-end latency (from sensor to HMI) remained stable at 65ms, with a negligible computational overhead on the server (< 2% CPU usage per lab thread).

B. Phase 2: Closed-Loop Control

This phase involved the use of a custom controller. Fig. 1 shows the configuration interface where the user defines the T and Ref vectors. The preview graph allows the user to validate the planned profile (including ramps) before execution.

Fig. 2 displays the real-time execution. The green dashed line represents the Target (Set-point), while the blue and orange lines show the Real and Twin responses, respectively. The system successfully demonstrated the ability to follow the reference with a minimal error margin, even during transition ramps.

In closed-loop mode, the system achieved a steady-state error of less than 1.5% relative to the set-point. Even when subjected to periodic repeating signal tests, the Digital Twin maintained a temporal correlation higher than 98% with the physical plant, validating the effectiveness of the synchronized control loop.

Figure 1 - Dynamic experiment configuration interface, allowing the definition of Time (T) and Reference (Ref) vectors with a real-time preview of the planned signal profile.

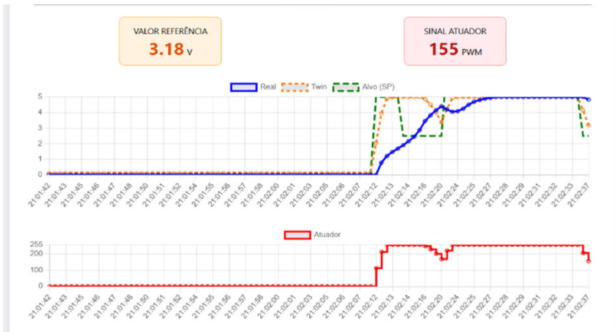

Figure 2 - Real-time supervision dashboard during a closed-loop trial. The plot displays the synchronization between the physical system (Real) and the Digital Twin (Twin) as they track the defined Set-point (green dashed line), response with an average latency of 65ms and steady-state error < 1.5%).

VI. Conclusion And Future Work

This paper describes the design and initial implementation of a modular supervision platform for CPS. The current stage of development has successfully validated the core orchestration engine, the multithreaded motor, and the basic synchronization between physical assets and their Digital Twins. As this is an ongoing work, the next phase will focus on the full deployment of the Intelligent Analysis Module. The final goal of this research is to move beyond first-order models and implement deep-learning-based Autoencoders for multi-variate anomaly detection.

Acknowledgment

This work is partially funded by national funds through FCT – Foundation for Science and Technology, I.P., within the scope of the research unit UID/00326 - Centre for Informatics and Systems of the University of Coimbra, https://doi.org/10.54499/UID/00326/2025.

References

[1] Weiss, A., et al., "Cobots in Industry 4.0: A roadmap for future practice," IEEE Trans. on Hum.-Mach. Syst., 2021.

[2] P. Orduña et al., "WebLab-Deusto: a new visual tool for remote experimentation," IFAC Proceedings Volumes, vol. 44, no. 1, 2011.

[3] M. J. Z. Abedin et al., "Digital Twin-enabled Smart Control for Remote Laboratories," IEEE Transactions on Industrial Informatics, 2023.

[4] Rasheed, A., and San, O., "Digital twin: Values, challenges and enablers from a modeling perspective," IEEE Access, vol. 8, 2020.

[5] Sousa, V., "Platform for the supervision of remote systems using low-cost devices," Master's Thesis, Univ. of Coimbra, 2015.