Research on the Deep Integration and Application of Digital Twins and Virtual Debugging Technology in Teaching Experimental Platform

Yuwen Yang[1], Song Wang[2,*], Bo Chen[3]

[1]Guizhou Polytechnic of Construction, Guiyang, Guizhou, China

[2]Guizhou Huilian Technology Co., Ltd., Guiyang, Guizhou, China

[3]Guizhou Polytechnic of Construction, Guiyang, Guizhou, China

[*] Corresponding author

Abstract

Digital Twin and Virtual Commissioning are the core enablement technology of Industry 4.0, which is deeply penetrated from the industry to the field of education, providing a new paradigm for innovating the traditional experimental teaching model. This paper proposes a teaching experimental platform construction solution that deeply integrates digital twins and virtual debugging technology. This solution takes Siemens TIA Portal and UGNX/MCD as the technical base. By establishing a high-fidelity mechatronic digital twin model and using PLCSIM Advanced to achieve real-time data interaction with virtual PLCs, it builds a complete "virtual and real mapping, rehearsal optimization, and linkage verification" closed-loop system. This article not only explains the overall architecture and key technologies of the system, but also theoretically analyzes the mathematical model of the virtual debugging process driven by digital twins. Through design comparison experiments, the improvement of the platform's teaching effectiveness was quantitatively evaluated. The data shows that after adopting this plan, the pass rate of student program debugging increased by about 40%, the time of experimental equipment occupied by 60%, and the depth of understanding of complex systems was significantly enhanced. This study provides replicable technical paths and theoretical support for the new generation of practical teaching of engineering science.

Keywords

Digital Twin, virtual debugging, mechatronics conceptual Design, teaching experiment, PLCSIM, TIA Portal.

1. Introduction (Heading 1)

Traditional programmable logic controllers (PLC) experimental teaching highly relies on the physics laboratory bench and faces bottleneck problems such as expensive equipment, limited quantity, easy damage, and time and space limitations. Students lack practical opportunities and usually learn in a "observation" rather than "practice" way, which seriously restricts their ability to solve complex engineering problems. The advent of virtual debugging technology provides the possibility to solve this problem. It allows engineers to conduct comprehensive testing and verification of control systems in a virtual environment, thereby greatly shortening on-site debugging time and reducing costs[1].

However, traditional virtual debugging mostly focuses on geometric animation simulation, lacks deep portrayal of physical laws, behavioral logic and data closed loops, and there is a risk of "disconnection between virtual and real"[2]. Digital twin technology, as an advanced form to achieve deep integration of information physics systems (CPS), lays the foundation for realizing high-reliability virtual debugging by building full-factor digital mapping that is highly consistent with physical entities[3].

In recent years, the integration of digital twins and virtual debugging has become a research hotspot, but its applications are mostly focused on the planning and optimization of industrial production lines[4]. The application in the field of education, especially teaching experiments, is still in the exploration stage, and lacks a systematic theoretical framework and detailed validation of effectiveness data.

This study aims to fill this gap, with the core contributions as follows:

(1) A teaching experimental platform architecture that deeply integrates digital twins and virtual debugging technology is proposed and implemented, and the entire process from three-dimensional modeling, physical attribute assignment, control logic mapping to virtual and real linkage is elaborated in detail.

(2) A theoretical model of this fusion system was constructed, and the key processes were mathematically abstracted and described, which improved the theoretical depth of the plan.

(3) A rigorous comparative experiment was designed, and through multiple sets of quantitative and qualitative data, the platform's significant improvement in teaching efficiency, learning effect and equipment utilization rate was empirically analyzed, providing strong evidence for the digital transformation of education..

2. Overall System Architecture

The deep fusion architecture of the teaching experimental platform built in this paper is shown in Figure 1. Its core lies in building a digital twin that completely corresponds to the physical experimental platform, and real-life interaction through a high-speed data bus.

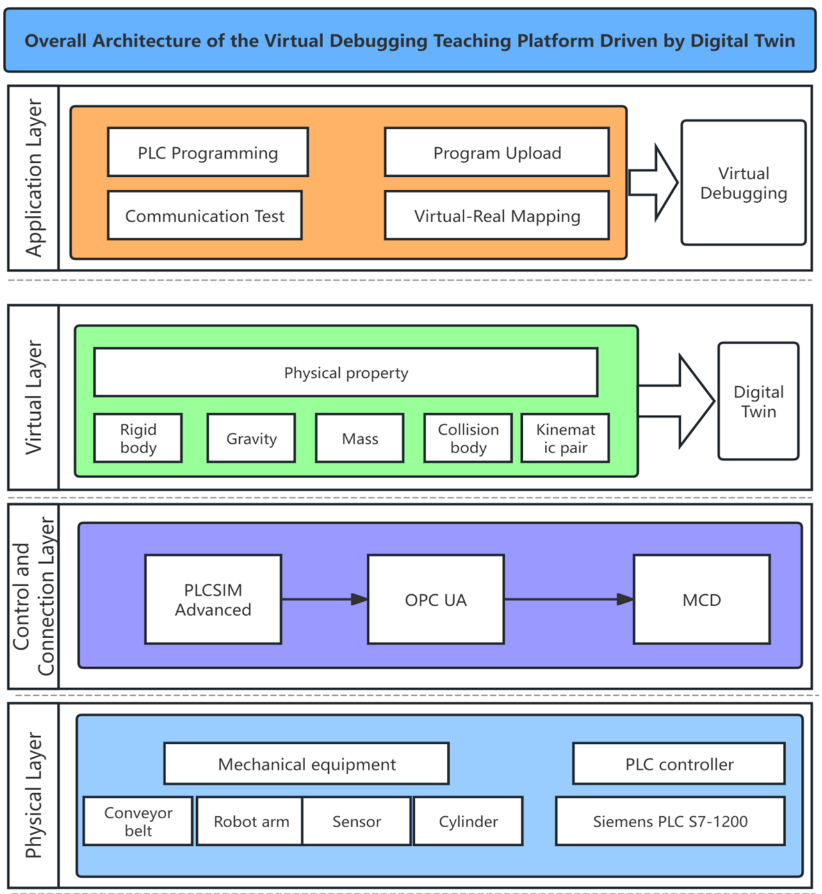

Figure 1. Overall architecture of digital twin-driven virtual debugging teaching platform

The system can be divided into four levels:

Physical layer: Contains real experimental platforms (such as conveyor belts, robots, sensors, etc.) and their Siemens S7-1200/1500 PLC controller.

Control and Connection Layer: Siemens TIA Portal is used for PLC program development, and a virtual PLC instance is created using PLCSIM Advanced. Through the OPC UA or S7 communication protocol, a real-time data exchange channel between the virtual PLC and the MCD digital twin is established to achieve millisecond synchronization of control instructions and status feedback.

Virtual layer: It is the core of the system. It builds a high-precision three-dimensional model based on UGNX, and uses the MCD module to give it physical properties such as gravity, collision, and rigid bodies to define the motion pair and sensor logic, thereby generating a digital twin with behavioral capabilities.

Application layer: Students can complete the writing, downloading and debugging of programs in a purely virtual environment (virtual PLC + digital twin). After the program is fully verified in the virtual environment, it can be downloaded to the real PLC in the physical layer with one click to control the operation of the real device, thereby achieving seamless connection of "virtual preview and real execution".

3. Key Technologies and Theoretical Models

A. High-fidelity digital twin construction

The fidelity of the digital twin is the key to determining the credibility of virtual debugging. In the UGNX/MCD environment, the construction of twins is far more than geometric modeling, but also a multi-level behavioral modeling process.

Physics Engine and Dynamics Modeling: MCD's built-in physics engine is the basis for simulating the physical laws of the real world. For a rigid body component i, its motion can be described by the Newton-Euler equation:

$$m_i \frac{d\boldsymbol{v}_i}{dt} = \sum \boldsymbol{F}_i \quad (1)$$

$$\boldsymbol{I}_i \frac{d\omega_i}{dt} + \omega_i \times \boldsymbol{I}_i \omega_i = \sum \tau_i \quad (2)$$

Among them, m_i is mass, $\mathbf{v}_i$ is the linear velocity, $\mathrm{F_i}$is the combined force being subjected, $\mathbf{I}_i$ is the inertia tensor, ω_i is the angular velocity, and τ_i is the combined moment being subjected. By correctly setting physical properties such as rigid body, collision body and gravity for the model components, the engine can automatically solve the above equations and simulate physical effects such as collision, friction, and fall[5].

Kinematic modeling and signal mapping: For movements driven by actuators (such as cylinders, motors), their motion pairs (joints), such as translation pairs (prismatic joins) and rotation pairs (revolute joins). The motion relationship can be expressed as:

$$q = f(u(t)) \quad (3)$$

Where q is the generalized coordinates of the system (such as position, angle), and $u(t)$ controls the input signal (such as the Boolean quantity or analog quantity output by the PLC). In MCD, through the Signal Mapping table, a strict correspondence between the PLC output point and the twin motion sub-control port is established to realize the conversion of "control signal –>virtual action".

Sensor modeling and information feedback: The trigger logic of virtual sensors (such as photoelectric sensors, limit switches) is the basis of closed-loop control. Its state change function can be simplified to:

$$s_{out} = g(\boldsymbol{q}(t), \boldsymbol{o}) \tag{4}$$

Among them, s_{out} is the sensor output signal (0 or 1), s_{out} is the position of the target object, and **o** is the sensor's own parameters (such as position and detection range). When the target object enters the detection area, s_{out} is set and fed back to the PLC input point in real time through the signal mapping table, thereby driving the execution of the control program.[6]

B. Closed-loop control model for virtual debugging

Virtual debugging is introduced in PLC training teaching, and its essence is to build a closed-loop control system that integrates information physics. The system allows students to conduct comprehensive, safe and efficient testing and optimization of the control logic they write in a purely virtual environment. Its theoretical model is shown in Figure 2.

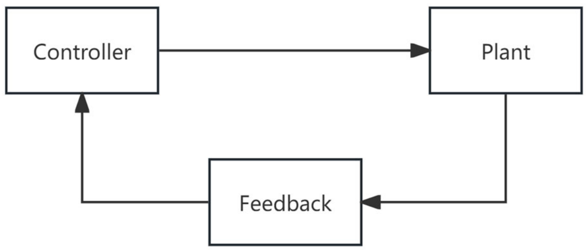

Figure 2. Virtual debugging closed-loop control theory model

Controller: A PLC control program written by students in integrated development environments such as TIA Portal. The program is the "brain" of the system, which executes preset logic or algorithms based on the received feedback signals and generates control instructions.

The controlled object (Plant): that is, a digital twin that carries the geometric, physical and behavioral properties of the device. It accurately reproduces the dynamic behavior of physical devices and is a virtual mapping of the real world.[7-9]

Feedback: Implemented by virtual sensors deployed inside the digital twin. These sensors "sensor" the state of the virtual device and convert it into electrical signals that the PLC can recognize.

The entire virtual debugging process can be abstracted as a simulation process of a discrete event dynamic system (DEDS). DEDS is characterized by changes in system state driven by asynchronous, discrete events (such as sensor triggering, motor in place, cylinder protrusion), rather than by continuous passage of time. This is highly consistent with the PLC's operation mode based on scan cycles and the event-driven characteristics of industrial automation scenarios. The student’s goal is to design the controller so that the dynamic response of this closed-loop DEDS can meet the preset design goals, such as avoiding component collisions, ensuring the correct sequence of logic, optimizing the production beat (Cycle Time), etc.

i. Input layer model: Mathematical characterization and dynamics of virtual sensors

The input layer is the bridge connecting the virtual world with the PLC controller, and its core is the modeling of virtual sensors. The function of the virtual sensor is to simulate a real sensor, converting the continuous or discrete state x of the digital twin (the controlled object) into an input signal y that the PLC can process through the function y = h(x).

(1) Classification and mathematical representation of sensor models

In a virtual debugging environment, the mathematical model of the sensor is not the same, and its complexity depends on the teaching objectives and simulation fidelity requirements. At present, there is no unified "standard mathematical formula" but it can be summarized into the following categories:

1) Logic/Switching Model

This is the most commonly used and basic model, used to simulate proximity switches, photoelectric sensors, limit switches, etc. Its dynamic behavior can be represented by a simple segmentation function or conditional statement. For example, a proximity sensor P_sensor used to detect the workpiece in place, whose output y (corresponding to the PLC input point I0.0) can be determined based on the distance d between the workpiece's coordinate pos_workpiece in the digital twin model and the target position pos_target:

$$y(t) = h\big(x(t)\big) = \begin{cases} 1 & if\,||pos_{workpiece}(t) - pos_{target}|| \le \delta \\ 0 & otherwise \end{cases}$$

where δ is the trigger tolerance range of the sensor. Under the DEDS framework, the state change of this model (0 becomes 1 or 1 becomes 0) is an "event" that triggers the migration of the system state.

2) State space and transfer function model

For sensors that need to consider dynamic response characteristics, such as sensors with delay or inertia, more complex models can be used. Although the canonical state space or transfer function representations directly applied to sensors are not common in the literature, their theory is complete. For example, a first-order inertia link can be used to simulate the response delay of the sensor:

Transfer function represents: $G(s) = \frac{Y(s)}{X(s)} = \frac{K}{\tau s+1}$

State space represents: $\dot{z} = -\frac{1}{\tau}z + \frac{K}{\tau}x; y = z$

where x is the measured physical quantity, y is the sensor output, K is the gain, and τ is the time constant. Under the simulation step length Δt of discrete time, it can be discrete into a differential equation for calculation. This model is crucial for training students to handle advanced control tasks such as signal delay and filtering. For example, a random forest sensor model may include state space and measurement space describing the different states of the sensor and their probabilities. The behavioral model of digital twins can itself be constructed through a state space method to describe the real-time response of physical entities.

3) Data-Driven and Noise Model:

To improve the authenticity of the simulation, noise can be superimposed on the basis of an ideal sensor model. This helps to train students to write more robust control programs. Common noise models include:

Gaussian white noise: $y_{noisy}(t) = y_{ideal}(t) + N(0, \sigma^2)$, where N is a normal distributed random number with mean 0 and variance σ^2.

Signal jitter and drift: Simulates the measurement deviation caused by temperature or aging of the sensor.

The virtual sensor framework can combine physical space and virtual space data to synchronize physical systems with virtual models through digital twin technology, thereby enhancing monitoring capabilities. After data acquisition, pre-processing such as denoising and cleaning are usually required.

ii. Control model: Discretization and implementation of PLC control law

The controller model u = f(y) is the core of student practice, i.e. the PLC program itself. With its inherent cyclic scanning working mechanism, the PLC is essentially a discrete time controller. Whether it is simple logic control or complex PID regulation, its control law must be implemented in discrete form.

(1) Logical control mapping

For logic such as sequential control and chain protection, the PLC program implements Boolean algebraic operations through languages such as ladder diagrams and SCL. This can be seen as a complex logical mapping from the input state vector y to the output state vector u. The entire PLC program is a complex function composed of hundreds of such basic logical maps.

(2) Discreteization of continuous control: Taking PID as an example

In many industrial scenarios, closed-loop control is required for continuous quantities such as temperature, pressure, and speed. The most classic control law is PID control. The ideal formula for continuous time is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

In PLC, it must be converted into a differential equation of discrete time. The commonly used discretization method is the positional PID algorithm:

First, the error is calculated: $\mathrm{e(k) = SP(k) - PV(k)}$, where SP is the set value and PV is the process quantity (i.e., sensor feedback value y).

Then, the discrete expression of the control law is:

$$u(k) = K_p \cdot e(k) + K_i \cdot T \sum_{j=0}^{k} e(j) + \frac{K_d}{T}[e(k) - e(k-1)]$$

where:

u(k): The controller output at the kth sampling time.

e(k): Error at the kth sampling time.

T: The sampling period, usually equals the scanning period of the PLC.

The u integral term $\int e(t)dt$ is approximate to the accumulation and $T\sum e(j)$ . .

The u differential term $\frac{de(t)}{dt}$ is approximated as a backward differential $\frac{e(k)-e(k-1)}{T}$.

In a PLC program block (such as FB), it is necessary to define static variables to store the error e(k-1) and error accumulation sum ∑e(j) of the previous moment to achieve state update. This transformation from continuous equations to discrete differential equations is the key theoretical and practical skills that students need to master in virtual debugging.

iii. Controlled object model: Dynamics of digital twins and DEDS abstraction

The controlled object x_dot = g(x, u) is a digital twin, which is the core of virtual debugging. A high-fidelity digital twin should contain a model of four dimensions: geometry, physics, behavior and rules.

(1) Bottom Continuous Variable Dynamics Model (CVDS)

The underlying behavior of a digital twin is usually described by a Continuous Variable Dynamic System (CVDS), following physical laws such as Newtonian mechanics and motors, and can be accurately described by differential equations or partial differential equations.

Rigid body dynamics: For robotic arms, conveyor belts, etc., their movements can be solved in real time by assigning physical properties such as mass, moment of inertia, friction coefficients based on the CAD model.

Actuator dynamics: For example, a model of a DC motor can be expressed as:

$$J\frac{d\omega}{dt} = K_t i_a - B\omega - T_L$$

$$L_a\frac{di_a}{dt} = V_a - R_a i_a - K_e\omega$$

These equations describe how the motor speed ω responds to the input voltage V_a (controlled by the PLC's u) and the load T_L. These physical models ensure that the digital twin's response to control instructions is highly consistent with the real device.

(2) Abstraction of Discrete Event Dynamic Systems (DEDS)

Although the underlying layer is CVDS, at the level of PLC control logic, we are more concerned about state changes triggered by discrete events. For example, we are concerned about "whether the motor has rotated to 90 degrees" rather than every instantaneous speed during its rotation. Therefore, the entire controlled object is abstracted as a DEDS model at the level of interaction with the PLC.

Events: Including changes in the PLC output signal (such as Q0.1 from 0 to 1), changes in the virtual sensor state (such as I0.2 from 1 to 0), timer timeout, etc.

Statements: The state of the system is defined by a combination of all key variables (such as cylinder position, robot joint angle, workpiece position, etc.).

State Transitions: The occurrence of an event triggers the system to transfer from one state to another.

Common formal methods for DEDS modeling include Petri nets, automaton theory, and discrete event system specification (DEVS). In a virtual debugging platform, these complex mathematical models are encapsulated inside the simulation engine. Although students do not write Petri Net directly, the PLC sequential functional diagram (SFC) they write is actually a graphical Petri Net, and its accuracy ultimately needs to be verified through interaction with the DEDS model of digital twins.

iv. Implementation mechanism of output layer *and feedback layer*

(1) Output layer: Executor model and instruction mapping

The output signal u of the PLC is a series of discrete Boolean values (ON/OFF) or numerical values (analog quantities). The core task of the output layer model is to map these discrete instructions into the continuous physical response of the executor in the digital twin.

Switching quantity mapping: When the PLC output Q0.0 is 1, the power supply voltage V_a mapped to the corresponding motor in the digital twin is 24V. When Q0.0 is 0, V_a is 0V.

Analog Mapping: When the PLC outputs an integer of 0-27648 to QW0, the output layer model will linearly map it to the 0-10V voltage signal received by the servo driver, or directly map it to the target speed or position.

This mapping process, together with the actuator dynamics model discussed in the previous chapter, forms the complete link from control instructions to physical actions.

(2) Feedback layer: Implementation and coupling mechanism of virtual sensors

The feedback layer is another key link in the closed loop, and its core is the implementation mechanism of virtual sensors.

Data acquisition: Virtual sensors poll or subscribe to the physical quantity they care about directly from the "World State" of the simulation engine. For example, the proximity sensor continuously acquires the three-dimensional coordinates of a specific object, and the visual sensor acquires the image frames rendered by the virtual camera.

Coupling with digital twins: This coupling is real-time and tight. The state update of the digital twin model is the data source, and the virtual sensor is the consumer and converter of the data. Through the data synchronization mechanism, ensure that the state of the virtual model and physical entity remains consistent within each simulation step.

Noise modeling and filtering: As described in Section 2.1, in order to simulate the non-ideality of the real world, noise can be injected through the algorithm after data is collected and sent to the PLC. For example, a Kalman filter model can be introduced to simulate a sensor that contains both process noise and measurement noise, which provides the possibility for teaching advanced control and signal processing.

4. Experimental Verification and Data Analysis

In order to verify the effectiveness of this platform, we selected two parallel classes (30 people each) in an automation major in a certain university as the experimental subjects and conducted a comparative teaching experiment.

Control group: Using the traditional experimental model, students directly perform PLC programming and debugging on the physical laboratory bench.

Experimental group: Using the virtual debugging mode proposed in this article, students must first complete the virtual verification of the program on the digital twin platform, and then they can perform the final test on the physical laboratory bench.

A. Quantitative data analysis

We recorded and analyzed the key data of the two groups of students when completing the same "material sorting unit" control project, as shown in Table 1.

The pass rate of program debugging in the experimental group was significantly higher than that in the control group, and the average number of debugging was significantly reduced. This shows that the virtual debugging platform can effectively help students discover and resolve most logical errors and simulation exceptions in the early virtual environment, avoiding repeated trial and error on physical devices.

The per capita physical equipment occupation time has been reduced by 60%, which has greatly alleviated the problem of tight equipment resources, enabled more students to obtain practical opportunities, and improved equipment utilization and teaching efficiency.

The improvement in the scores of the experimental report shows that by operating the digital twin, students can more intuitively understand the causal relationship between program execution flow and device actions, and have a deeper understanding of the overall system.

Table 1. Comparison of experimental data of two groups of students

Evaluation Indicators	Traditional Mode	Virtual Debugging Mode	Improvement
One-time Pass Rate of Program Debugging	40%	80%	+40%
Average Number of Program Debugging Attempts	5.2 times	2.1 times	-59.6%
Average Physical Equipment Occupancy Time per Person	115 min	46 min	-60%
Experimental Report Score (Full Score: 100)	78.5	89.2	+13.6%

B. Qualitative feedback

A questionnaire survey of students in the experimental group showed:

(1) 95% of students believe that the virtual debugging platform "greatly enhances learning confidence" because they are not afraid of damage to the device due to programming errors.

(2) 90% of students said that "deeply understand the abstract concepts such as PLC scanning cycles and signal flows" because the simulation process can be slowed down and paused for observation.

(3) 88% of students believe that "feeling the cutting-edge technology of the industry and increasing their interest in learning."

5. Conclusion and Prospect

This article successfully integrates industrial-grade digital twins with virtual debugging technology and applies them to teaching experimental platforms. By building a high-fidelity digital twin and establishing its real-time communication with the virtual PLC, a new "virtual first" experimental teaching model has been created.

At the theoretical level, this paper mathematically models the dynamics, kinematics, and the closed loop formed by the control system of the digital twin, providing a theoretical framework for the in-depth application of this technology. At the practical level, through rigorous comparative experiments, it has been proved that the platform can significantly improve students' learning efficiency, program quality and depth of understanding of complex systems, while significantly reducing the loss and occupation of physical equipment.

Future work will focus on: 1) Introducing artificial intelligence algorithms to realize automatic identification and intelligent guidance for student programming errors; 2) Exploring cloud-native architecture, deploying the platform to the cloud, supporting students to access and experiment through browsers anytime and anywhere, further breaking time and space limitations; 3) Expand multi-disciplinary coupled simulation, and integrate more physics fields such as thermodynamics and fluid mechanics into twins to support a wider range of engineering major experiments.

References

[1] Zhu, Z., Liu, P., & Xu, X. (2022). A review of virtual commissioning technology and its applications in industry and education. Journal of Manufacturing Systems, 64, 562-578. https://doi.org/10.1016/j.jmsy.2022.07.009.

[2] Uhlemann, T. H. J., Lehmann, C., & Steinhilper, R. (2020). The digital twin: Realizing the cyber-physical production system for Industry 4.0. Procedia CIRP, 88, 99-104. https://doi.org/10.1016/j.procir.2020.05.018

[3] Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405-2415. https://doi.org/10.1109/TII.2018.2873186

[4] Schluse, M., Priggemeyer, M., Atorf, L., & Rossmann, J. (2018). Experimentable digital twins—Streamlining simulation-based systems engineering for Industry 4.0. IEEE Transactions on Industrial Informatics, 14(4), 1722-1731. https://doi.org/10.1109/TII.2018.2794183

[5] Cimino, C., Negri, E., & Fumagalli, L. (2019). Review of digital twin applications in manufacturing. Computers in Industry, 113, 103130. https://doi.org/10.1016/j.compind.2019.103130

[6] Lee, C., & Zhou, M. (2021). Modeling and control of discrete-event systems with learning-based approaches: A survey. IEEE/CAA Journal of Automatica Sinica, 8(2), 328-349. https://doi.org/10.1109/JAS.2021.1003864

[7] Hossain, R. B., Ahmed, F., Kobayashi, K., & Sato, T. (2024). Virtual sensing-enabled digital twin framework for real-time monitoring of nuclear systems leveraging deep neural operators.

[8] Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future NASA and U.S. Air Force vehicles. https://doi.org/10.2514/6.2012-1818

[9] Qi, Q. L., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. IEEE Access, 6, 3585-3593. https://doi.org/10.1109/ACCESS.2018.2793265