

ІНТЕГРОВАНА АРХІТЕКТУРА DIGITAL TWIN ТА HARDWARE-IN-THE-LOOP ДЛЯ ПРОМИСЛОВИХ КІБЕРФІЗИЧНИХ СИСТЕМ

Олександр Тимашов

Інститут кібернетики імені В.М. Глушкова
НАН України, м. Київ, Україна
ORCID 0000-0003-0902-487X

Олена Єліссєва

Інститут кібернетики імені В.М. Глушкова
НАН України, м. Київ, Україна
ORCID 0000-0002-6502-9192

Лариса Курзанцева

Інститут кібернетики імені В.М. Глушкова
НАН України, м. Київ, Україна
ORCID 0000-0001-5937-1699

Анотація: У роботі розглянуто проблему інтеграції технологій Digital Twin (DT) та Hardware-in-the-Loop (HIL) у промислових кіберфізичних системах. Показано, що HIL і Virtual Commissioning традиційно застосовуються переважно на етапах розроблення та введення системи в експлуатацію, тоді як Digital Twin орієнтований на синхронізацію з фізичним об'єктом, моніторинг, діагностику, прогнозування та оптимізацію під час експлуатації. Сучасні дослідження демонструють перехід до повторного використання моделей Virtual Commissioning у складі операційних цифрових двійників, однак недостатньо формалізованим залишається керування співвідношенням фізичних і віртуальних компонентів упродовж життєвого циклу системи. Запропоновано lifecycle-adaptive архітектуру DT–HIL, у якій єдине модельне ядро повторно використовується в режимах Model-in-the-Loop, Software-in-the-Loop, Hardware-in-the-Loop, Equipment-in-the-Loop та Operational Digital Twin. Архітектура містить фізичний і керувальний рівні, детермінований Real-Time HIL Plane, Unified Model Core, Synchronization and Calibration Layer, Digital Twin Information Plane, сервісний та lifecycle-рівні. На основі попередньо напрацьованих технічних матеріалів деталізовано моделі сенсорів та актуаторів, механізми часової синхронізації, буферизації, контролю цілісності даних і адаптації параметрів. Формалізовано конфігураційний стан DT–HIL системи та багатокритеріальну задачу вибору раціональної конфігурації за точністю, затримкою, обчислювальною вартістю, ризиком і часом реконфігурації. Запропоновано методику Verification, Validation and Uncertainty Quantification та дизайн порівняльного експерименту для перевірки архітектури без підміни майбутніх експериментальних результатів припущеними числовими значеннями. Підхід узгоджується з розвитком ISO 23247, зокрема з положеннями щодо digital thread та композиції цифрових двійників.

Ключові слова: цифровий двійник; Hardware-in-the-Loop; промислова кіберфізична система; віртуальне введення в експлуатацію; моделювання реального часу; керування життєвим циклом; повторне використання моделей; часова синхронізація; адаптивна точність моделі.

1. Вступ

Промислові кіберфізичні системи (Cyber-Physical Production Systems, CPPS) формуються як інтегровані середовища, у яких фізичне обладнання, програмно-логічні контролери, сенсорні мережі, математичні моделі, інформаційні сервіси та аналітичні засоби взаємодіють у замкнених контурах. Зростання гнучкості виробництва одночасно підвищує складність верифікації керувального програмного забезпечення, реконфігурації, модернізації та введення

в експлуатацію. У цих умовах особливого значення набувають технології, здатні переносити частину випробувань із фізичного середовища до контрольованого цифрового простору.

Концепція Digital Twin розвивалася від цифрового представлення окремого об'єкта до системного засобу конвергенції фізичного й віртуального виробництва. Tao і Zhang запропонували Digital Twin Shop-Floor як поєднання фізичного цеху, його віртуального представлення, сервісної системи та даних цифрового двійника [1]. Kritzinger та співавтори показали принципову відмінність між Digital Model, Digital Shadow і Digital Twin за рівнем автоматизованого обміну даними між фізичним і цифровим просторами [2]. Узагальнення Liu та співавторів підтверджує, що DT охоплює різні стадії життєвого циклу, однак інженерне моделювання і runtime-використання моделей досі часто залишаються розірваними [3].

Паралельно HIL і Virtual Commissioning стали базовими засобами ранньої перевірки систем керування. Їхня перевага полягає в тому, що реальний PLC, MCU, DSP або інший controller може взаємодіяти з executable model об'єкта до появи повної фізичної установки. Таким чином, software defects, timing errors, некоректні interlocks і небезпечні сценарії можуть бути виявлені раніше, ніж під час фізичного commissioning.

Останні публікації демонструють поступове зближення двох напрямів. Ferle та співавтори формалізували перетворення Virtual Commissioning models на operational Digital Twins і прямо вказали на відсутність структурованого способу повторного використання цих моделей після engineering phase [4]. Djeulezeck Tanegue та співавтори інтегрують SIL/HIL і DT із двонапрямним обміном через OPC UA та RTDE та розглядають lifecycle-wide reuse simulation models [5]. Отже, наукова проблема вже полягає не у доведенні можливості технічного з'єднання HIL і DT, а у формуванні цілісної архітектури, яка керовано змінює співвідношення фізичних та віртуальних складових залежно від стадії життєвого циклу й поточного Context of Use.

Сукупність цих тенденцій формує потребу у дослідженні архітектури, в якій HIL і Digital Twin розглядаються не як дві незалежні технології, а як взаємопов'язані режими виконання та використання спільного модельного ядра на різних стадіях життєвого циклу промислової кіберфізичної системи.

2. Об'єкт і предмет дослідження

Об'єктом дослідження є процеси цифрового моделювання, напівнатурної перевірки, синхронізації та супроводу промислових кіберфізичних систем упродовж їхнього життєвого циклу – від раннього проектування і верифікації керувального програмного забезпечення до введення в експлуатацію, реконфігурації та операційного використання цифрового двійника.

Предметом дослідження є архітектурні моделі та методи інтеграції Digital Twin і Hardware-in-the-Loop, що охоплюють повторне використання виконуваних моделей, керування фізично-віртуальною конфігурацією, синхронізацію часових доменів і станів, адаптацію параметрів і точності моделі, розділення детермінованого real-time контуру та інформаційно-сервісного контуру, а також процедури верифікації, валідації та оцінювання невизначеності.

Дослідження обмежено класом industrial CPPS, у яких існує програмований або embedded-контролер, доступний інтерфейс телеметрії та можливість виконання принаймні частини plant model у цифровому середовищі. Конкретні physics-based моделі технологічного об'єкта не вважаються універсальними: універсальним елементом є структура інтеграції, правила конфігурації та процедура оцінювання придатності моделі для визначеного Context of Use.

3. Мета та задачі дослідження

Метою дослідження є розроблення та формалізація lifecycle-adaptive інтегрованої архітектури Digital Twin–Hardware-in-the-Loop для промислових кіберфізичних систем, яка забезпечує повторне використання єдиного модельного ядра між режимами Model-in-the-Loop, Software-in-the-Loop, Hardware-in-the-Loop, Equipment-in-the-Loop та Operational Digital

Twin, а також адаптацію фізично-віртуальної композиції, рівня точності моделі й механізму синхронізації до стадії життєвого циклу та поточного Context of Use.

Для досягнення мети поставлено такі задачі: 1) узагальнити сучасні підходи до DT-driven Virtual Commissioning, HIL, синхронізації та повторного використання моделей; 2) сформулювати функціональні та нефункціональні вимоги до інтегрованої архітектури; 3) визначити структуру Unified Model Core і технічні механізми sensor/actuator emulation, time alignment, buffering, integrity control та parameter adaptation; 4) формалізувати конфігураційний стан і критерії вибору режиму; 5) обґрунтувати розділення real-time та information-service planes; 6) сформулювати VVUQ-oriented протокол і дизайн порівняльної експериментальної перевірки запропонованого рішення.

4. Аналіз літератури

Scheifele, Verl і Riedel розробили real-time co-simulation platform для Virtual Commissioning, яка використовує partitioning, parallelization, synchronization та data exchange між спеціалізованими моделями [6]. Ця робота є важливою для інтегрованого DT-HIL, оскільки складний промисловий двійник практично неможливо розглядати як одну монолітну модель: механіка, електропривод, логіка PLC, робототехніка, технологічний процес і discrete-event production model часто мають різні часові масштаби та чисельні методи.

Leng та співавтори запропонували Digital Twin-based Remote Semi-Physical Commissioning, у якому різні частини production system можуть бути фізичними або віртуальними [7]. Barbieri та співавтори використали Virtual Commissioning не лише для controller validation, а як середовище проєктування, інтеграції та перевірки DT architecture перед фізичним впровадженням [8]. Ці роботи формують передумову для трактування фізичної/віртуальної композиції як змінної конфігурації системи.

Iyer, Sangwan і Dhiraj запропонували DT-based Virtual Commissioning реконфігурованої виробничої лінії; новий asset перевіряється за HIL-принципом у синхронізованому двійнику існуючої лінії [9]. Ionescu та співавтори інтегрували Digital Twin, HIL, SCADA, IoT і cloud control у роботизованій комірці [10]. Ni та співавтори доповнили DT-driven Virtual Commissioning машинним навчанням і consistency-retention mechanism, показавши, що модель має не лише отримувати дані, а й коригувати параметри відповідно до змін фізичної системи [11].

На рівні industrial controllers Reuther та співавтори запропонували DT framework для PLC із IEC 61131-3, OPC UA та Asset Administration Shell, що підтримує bidirectional synchronization, monitoring, control, programming і simulation [12]. Kronová та співавтори у 2026 р. показали laboratory-scale online DT із Siemens S7-1200, KEPServerEX та OPC UA; отримані response times 50–200 ms підтверджують доцільність standardized near-real-time integration, але водночас демонструють, що такий інформаційний контур не можна автоматично прирівнювати до hard-real-time HIL loop [13].

Окрема група праць стосується синхронізації та даних. Zipper запропонував real-time-capable state synchronization на основі dynamic optimization для зменшення розбіжності online simulation з фізичним plant [14]. Hildebrandt та співавтори в systematic literature review показали, що data integration, semantic interoperability та якість джерел даних є центральними проблемами industrial Digital Twins [15]. Cho і Noh реалізували MQTT-based synchronization Digital Twin Factory, що є корисним для telemetry та service traffic, але не скасовує потреби в окремому deterministic communication domain [16]. Park та співавтори демонструють service-oriented DT-based CPPS із Asset Administration Shell [17], а Chen, Neubauer і Verl розглядають Virtual Commissioning models як reusable DT services для testing and optimization під час production phase [18].

Індустріальні інтерв'ю та огляди також підтверджують, що поширенню VC і DT заважають трудомісткість моделей, неоднорідність toolchain, проблеми стандартизації та підтримки моделей після commissioning [19]. Окремим невирішеним питанням є довіра до DT. Systematic review Bitencourt, Wooley і Harris показав, що повні процедури verification і

validation застосовуються в manufacturing DT непослідовно, а уніфікованої процедури V&V немає [20]. Shao, Hightower і Schindel наголошують, що credibility DT має оцінюватися протягом усього lifecycle та включати Verification, Validation and Uncertainty Quantification (VVUQ) для конкретного Context of Use [21].

У сукупності література дозволяє виділити п'ять взаємопов'язаних прогалин. По-перше, HiL/VC і Operational DT часто мають різні моделі та різні життєві цикли. По-друге, переходи MiL→SiL→HiL→EiL→Operational DT недостатньо формалізовані як керовані стани однієї архітектури. По-третє, hard-real-time HiL і asynchronous/near-real-time DT-services мають різні temporal constraints. По-четверте, reality gap вимагає регулярного state/parameter synchronization. По-п'яте, переваги model reuse, adaptive fidelity і lifecycle reconstruction мають бути доведені кількісно, а не лише концептуально.

Таблиця 1. Порівняльний аналіз ключових напрямів Digital Twin–Hardware-in-the-Loop та Virtual Commissioning

Джерело	Основний фокус	HiL/V C	Синхроні зація	Lifecycle reuse	Невирішений аспект
Scheifele et al. [6]	Real-time co-simulation	Так	Так	Ні	Фокус на VC
Leng et al. [7]	Semi-physical commissioning	Так	Так	Частково	Commissioning-oriented
Barbieri et al. [8]	DT integration through VC	Так	Так	Частково	Немає adaptive lifecycle state
Iyer et al. [9]	Reconfigurable line	Так	Так	Частково	Окремий production case
Djeulezeck et al. [5]	SiL/HiL + DT	Так	Так	Так	Не формалізовано global configuration
Ionescu et al. [10]	Robotic cell DT+HiL	Так	Так	Частково	Domain-specific
Ferle et al. [4]	VC→Operational DT	VC	Так	Так	Немає універсального HiL state model
Reuther et al. [12]	PLC Digital Twin	Частково	Так	Так	Фокус на controller asset
Kronová et al. [13]	OPC UA online DT	Ні	Так	Так	Near-real-time monitoring
Chen et al. [18]	Reusable VC services	VC	Так	Так	Не визначено adaptive fidelity

Аналіз показує, що сучасний стан досліджень уже виходить за межі простого поєднання DT і HiL. Найменш формалізованим залишається керування переходами між різними фізично-віртуальними конфігураціями, встановлення temporal contract для різних потоків даних і кількісне обґрунтування повторного використання моделей. Саме ці аспекти покладено в основу запропонованого підходу.

5. Методи дослідження

Нормативною рамкою обрано ISO 23247. ISO 23247-2:2021 визначає reference architecture manufacturing Digital Twin [22], а ISO 23247-4:2021 — технічні вимоги до information exchange між сутностями architecture [23]. Особливо важливими для запропонованої концепції є нові ISO 23247-5:2026, присвячений digital thread протягом design, planning, production і testing [24], та ISO 23247-6:2026, який визначає integrated, unified і federated digital twin composition [25]. Запропонована DT–HiL architecture розглядається як HiL-oriented execution extension до manufacturing DT framework, а не як його заміна.

Методологічною основою дослідження обрано Design Science Research, оскільки очікуваним результатом є інженерний артефакт: архітектура, формальна конфігураційна модель, алгоритм реконфігурації та процедура VVUQ. Nevner та співавтори визначають Design Science як парадигму, в якій знання отримують через побудову й оцінювання інноваційного артефакта [26]. DSRM Peffers та співавторів задає послідовність problem identification, objectives, design/development, demonstration, evaluation та communication [27]. У

межах цієї роботи вона конкретизується як: research gap → requirements → architecture → formal model → prototype → VVUQ → comparative experiment.

Формування технічної структури також спирається на попередньо напрацьовані матеріали з інтеграції DT і HIL. Із них до статті доцільно перенести не заявлені числові результати, а інженерні елементи: моделі sensor noise, drift і latency; actuator saturation, inertia і rate limitation; protocol adaptation; timestamp alignment; delay compensation; FIFO/double/ring buffering; sequence/CRC checks; anomaly detection; online parameter identification і adaptive filtering. Саме ці елементи перетворюють загальну архітектурну схему на відтворювану технічну конструкцію.

Для наступної емпіричної перевірки передбачається контрольований порівняльний експеримент. Базові конфігурації мають бути реалізовані на одному й тому самому physical testbed з однаковим controller software: B1 — conventional HIL; B2 — standalone Operational DT; B3 — fixed integrated DT–HIL; P — proposed lifecycle-adaptive DT–HIL. Це дозволить відокремити ефект архітектури від відмінностей апаратного забезпечення.

Незалежними факторами експерименту доцільно визначити architectural mode, model fidelity, synchronization period, communication load, computational load, fault severity та кількість взаємодіючих model components. Залежними змінними є model error, end-to-end latency, jitter, deadline-miss ratio, CPU/RAM usage, reconfiguration time, model reuse coefficient та якість diagnostic/predictive services. Для виявлення взаємодій факторів доцільним є fractional factorial Design of Experiments із подальшим response-surface або Pareto analysis у найбільш значущих областях параметрів.

Ключова методологічна вимога полягає в тому, що будь-які числові характеристики proposed architecture мають походити з вимірювань або чітко позначеного simulation experiment. Проектні значення, що містяться в попередніх технічних матеріалах, не використовуються як фактичні результати без доступних журналів вимірювань, dataset і опису experimental setup.

На етапі синтезу архітектури застосовано структурно-функціональний аналіз: функції системи розподіляються між physical/control, real-time simulation, model core, synchronization, information integration, service та lifecycle-management domains. Для часових характеристик використовується decomposition end-to-end latency та контроль worst-case execution time. Для оцінювання відповідності цифрового двійника фізичному об'єкту передбачено RMSE, MAE, event timing error і trajectory error залежно від Context of Use. Для оцінювання повторного використання вводиться коефіцієнт частки повторно використаних модельних та інтерфейсних компонентів.

Емпіричну перевірку доцільно виконувати за Design of Experiments із порівнянням чотирьох конфігурацій: conventional HIL, standalone Operational Digital Twin, fixed integrated DT–HIL та proposed lifecycle-adaptive DT–HIL. Факторами є fidelity, synchronization period, communication load, computational load, fault severity і number of active model components. Відповідями експерименту є model error, P95/P99 latency, jitter, deadline-miss ratio, CPU/RAM utilization, time-to-reconfiguration та model reuse coefficient.

Credibility моделі оцінюється відповідно до VVUQ-принципу. Verification охоплює коректність equations, state transitions, I/O mapping, software interfaces і timing implementation. Validation визначає достатність відповідності фізичному об'єкту для заданого Context of Use. Uncertainty Quantification враховує measurement, parameter, model-form, communication та numerical uncertainties. Жодне проектне числове значення з попередніх технічних матеріалів не трактується як експериментальний результат без журналів вимірювань і відтворюваного опису стенду.

6. Результати дослідження

Запропонована архітектура складається з семи логічних підсистем: Physical Asset and Control Layer, Real-Time HIL Plane, Unified Model Core, Synchronization and Calibration Layer, Digital Twin Information Plane, Digital Twin Service Layer та Lifecycle Management Layer. Їхнє

розділення є функціональним: у конкретній реалізації окремі модулі можуть розміщуватися на одному edge-computer, PLC/HIL platform або розподілятися між edge і cloud.

Physical Asset and Control Layer охоплює технологічне обладнання, сенсори, актуатори, drives, PLC, MCU, CNC/robot controllers та safety controllers. Ключовим принципом є збереження controller application logic під час перемикання між реальним plant і HIL model настільки, наскільки це дозволяють interfaces. Це забезпечує повторюваність тестів і зменшує ризик того, що у simulation environment тестується інша версія control logic, ніж у production.

Real-Time HIL Plane відповідає за детерміноване виконання plant model, sensor/actuator emulation, I/O mapping, fault injection і temporal verification. Для кожного simulation step повинна виконуватися умова $T_{\text{exec}} < T_s$, де T_{exec} — worst-case execution time активної HIL-моделі, а T_s — допустимий sampling period. На відміну від DT services, цей контур не повинен залежати від cloud latency або непередбачуваного asynchronous messaging.

Unified Model Core складається з взаємозамінних модулів: physics-based model M_{phys} , behavioural/discrete-event model M_{beh} , control-interface model M_{logic} , data-driven model M_{data} та fault/degradation model M_{fault} . Концептуально його можна подати як $M = \{M_{\text{phys}}, M_{\text{beh}}, M_{\text{logic}}, M_{\text{data}}, M_{\text{fault}}\}$. Не всі модулі повинні виконуватися одночасно: lifecycle stage і Context of Use визначають активний subset і потрібний fidelity.

Моделі sensor і actuator channels включено до модельного ядра як окремі компоненти, оскільки саме їхня неідеальність значною мірою формує reality gap у HIL. Для сенсорного каналу використовується узагальнена модель

$$y_s(t) = h[x(t - \tau_s)] + b_s(t) + v_s(t) \quad (1)$$

де $h(\cdot)$ – measurement function; τ_s – затримка сенсора; $b_s(t)$ – systematic bias або drift; $v_s(t)$ – stochastic noise. Формула (1) відображає наявні в технічних матеріалах вимоги до моделювання шуму, затримки, похибки калібрування, нелінійності та обмежень діапазону. Такий підхід дає змогу використовувати однакову sensor abstraction у HIL та Operational Digital Twin, змінюючи лише параметри і джерело даних.

Для actuator channel мінімально достатня модель повинна враховувати saturation, inertia, rate limitation і transport delay. Для first-order approximation застосовується

$$\tau_a \frac{du_a(t)}{dt} + u_a(t) = \text{sat}[u_{\text{cmd}}(t - \tau_d)] \quad (2)$$

де τ_a – інерційна стала; u_{cmd} – команда контролера; τ_d – транспортна затримка; $\text{sat}(\cdot)$ – функція насичення. Формула (2) не претендує на універсальність для всіх приводів, але задає уніфікований interface-level template, який у конкретній системі замінюється електромеханічною, гідравлічною або іншою domain model.

Synchronization and Calibration Layer виконує timestamp alignment, resampling, delay compensation, state estimation, parameter identification і reality-gap monitoring. Нехай $x_{\text{phys},j}(t)$ та $x_{\text{DT},j}(t)$ — узгоджені physical і digital states. Тоді інтегральну похибку для визначеного Context of Use можна оцінити як

$$E_{\text{DT}}(t) = \sqrt{\frac{1}{n} \sum_{j=1}^n w_j (x_{\text{phys},j}(t) - x_{\text{DT},j}(t))^2} \quad (3)$$

За перевищення граничного значення E_{DT} , визначеного для поточного Context of Use, модель переходить до state correction, parameter recalibration або до режиму з нижчим рівнем довіри. Отже, механізм синхронізації виконує не лише пересилання телеметрії, а й контроль придатності цифрової моделі для прогностичних і діагностичних сервісів.

Часова синхронізація повинна враховувати різні temporal domains DT і HIL. У мінімальній реалізації пакет або sample містить sequence number, timestamp і integrity field; на приймальній стороні виконуються reorder/duplicate detection, timeout check та time alignment. Для нерівномірних потоків використовуються FIFO, double buffering або ring buffer; predictive buffering допустимий лише тоді, коли його похибка врахована у VVUQ. Контроль integrity

доцільно доповнювати range/physics checks, оскільки правильний CRC не гарантує фізичної правдоподібності signal value.

Комунікаційний механізм розвиває напрацювання вихідних матеріалів щодо protocol adaptation, scaling/normalization, time alignment, FIFO/double/ring buffering, sequence and CRC checks, anomaly detection та fail-safe handling. В інтегрованій архітектурі ці механізми розподіляються між real-time interface і information plane відповідно до temporal criticality потоку. Буферизація використовується не як універсальний спосіб приховування latency, а як керований засіб узгодження різних sample rates і timestamp domains; її глибина повинна входити до конфігураційного стану та V&V protocol.

Digital Twin Information Plane відповідає за OPC UA, MQTT, historian/time-series database, event processing, semantic model, AAS integration, edge/cloud analytics і user interfaces. Принциповим є співвідношення

Принциповим результатом є функціональне розділення Real-Time HIL Plane та Digital Twin Information Plane. Воно не обов'язково означає фізичне рознесення обчислювальних вузлів, але встановлює різні temporal contracts: HIL plane орієнтований на bounded execution, jitter і deadlines, тоді як information plane – на interoperability, semantics, telemetry, analytics і scalable services.

Digital Twin Service Layer містить monitoring, fault reconstruction, what-if analysis, predictive maintenance, process optimization, operator training та pre-deployment testing нового controller software. Окремо виділяється DT-assisted HIL sandbox: поточний synchronized state Operational DT копіюється до isolated HIL environment, де перевіряється майбутня версія PLC/embedded software без впливу на production asset. Після успішної validation зміна може бути допущена до deployment.

Таблиця 2. Функціональні шари запропонованої архітектури

Шар	Основні функції	Критичні вимоги
Physical/Control	Plant, sensors, actuators, PLC/MCU/CNC	Traceability, safety
Real-Time HIL Plane	Plant execution, I/O emulation, fault injection	WCET, jitter, deadline
Unified Model Core	Physics, behavioural, data-driven, fault models	Modularity, reuse, fidelity
Synchronization & Calibration	Time alignment, state/parameter correction	Bounded error, observability
DT Information Plane	OPC UA, MQTT, historian, semantics	Interoperability, scalability
DT Services	Monitoring, prediction, optimization, testing	Context-of-Use validity
Lifecycle Management	MiL/SiL/HIL/EiL/Operational DT transitions	Configuration/version control

Lifecycle Management Layer визначає активну фізично-віртуальну композицію. Послідовність MiL → SiL → HIL → Equipment-in-the-Loop → Operational DT не трактується як незворотна. Під час експлуатації система може повертатися до HIL-mode для testing software updates, fault reproduction або operator training. Отже, HIL у запропонованій концепції є одним зі станів виконання lifecycle Digital Twin environment.

Таблиця 3. Lifecycle-режими Digital Twin–Hardware-in-the-Loop

Режим	Controller	Plant	Дані	Основна задача
MiL	Model	Model	Synthetic	Algorithm design
SiL	Production software	Model	Synthetic/recorded	Software verification
HIL	Physical controller	Real-time model	Emulated/recorded	Controller validation
EiL	Physical controller	Partly physical	Live + simulated	Integration/commissioning
Operational DT	Physical system	Physical + digital representation	Live	Monitoring/prediction

Режим	Controller	Plant	Дані	Основна задача
DT-assisted HIL	Isolated physical controller/HIL	Synchronized model snapshot	Live snapshot + simulated	Safe pre-deployment test

Для переходу від статичної architecture diagram до керованого методу конфігураційний стан системи визначено кортежем

$$C_k = \{P_k, H_k, M_k, D_k, S_k, F_k\} \quad (4)$$

де P_k – активні physical components; H_k – HIL hardware та I/O interfaces; M_k – активні model modules; D_k – data sources; S_k – synchronization policy; F_k – fidelity configuration. Формула (4) дозволяє однаково описувати MiL, SiL, HIL, EiL, Operational DT і DT-assisted HIL sandbox як стани одного lifecycle environment.

Вибір конфігурації сформульовано як constrained multi-objective problem. Для компактної decision function може використовуватися

$$J(C_i) = w_E E_i + w_L L_i + w_C C_i^{comp} + w_R R_i + w_T T_i^{reconf} \quad (5)$$

де E_i – model error; L_i – latency; C_i^{comp} – computational cost; R_i – operational або safety risk; T_i^{reconf} – reconfiguration effort. Формула (5) є коректною за наявності обґрунтованих ваг. За конфліктних критеріїв доцільнішим є формування Pareto front, після чого конкретна configuration обирається згідно з обмеженнями Context of Use.

Adaptive fidelity реалізується як вибір найпростішої моделі, що одночасно задовольняє вимоги точності й temporal execution:

$$E(M_i) \leq E_{req} \wedge T_{exec}(M_i) \leq T_s \quad (6)$$

Умова (6) принципово відрізняє запропонований підхід від постійного використання максимально деталізованої моделі. Для scheduling, supervisory analytics і fast control потрібні різні моделі або різні execution configurations одного model core. Тому fidelity розглядається як керований параметр життєвого циклу.

Credibility proposed architecture слід оцінювати окремо для software implementation, digital model fidelity та integrated system behavior. Verification відповідає на питання, чи правильно реалізовано equations, state machines, I/O mapping, protocol conversion, units, buffer policy та timing logic. Validation визначає, чи відтворює model фізичний об'єкт із точністю, достатньою для визначеного Context of Use. Uncertainty Quantification охоплює measurement uncertainty, parameter uncertainty, model-form uncertainty, numerical uncertainty та communication/synchronization uncertainty.

Для continuous variables використовуються RMSE, MAE і normalized error; для event-driven CPPS – event mismatch rate, sequence correctness і transition timing error; для motion systems – position, velocity та trajectory errors. Коефіцієнт повторного використання моделі визначається як

$$K_{reuse} = \frac{N_{reused}}{N_{total}} \quad (7)$$

де N_{reused} – кількість model/interface components, перенесених між lifecycle stages без повної повторної розробки; N_{total} – загальна кількість компонентів, необхідних у наступній конфігурації. Поряд із K_{reuse} доцільно вимірювати adaptation time, number of modified components, reused interfaces і time-to-validation.

Експериментальний дизайн повинен перевіряти не лише nominal operation, а й parameter drift, sensor noise/failure, actuator degradation, communication delay, packet loss, computational overload, controller software update, reconfiguration та temporary loss of DT synchronization. Такі fault-injection scenarios є особливо цінними для DT–HIL, оскільки небезпечні режими можна спочатку відтворити в isolated digital environment.

Таблиця 4. Рекомендований план порівняльної експериментальної оцінки

Група	Фактори/сценарії	Основні метрики
Baseline comparison	HIL / Standalone DT / Fixed DT–HIL / Adaptive DT–HIL	RMSE, latency, jitter, reuse
Fidelity	Low / medium / high	Error, WCET, CPU/RAM
Synchronization	Period, drift, temporary loss	State error, recovery time
Network	Normal load, delay, packet loss	P95/P99 latency, deadline misses
Fault injection	Sensor/actuator/model faults	Detection/recovery, false alarms
Lifecycle reuse	VC→DT, DT→HIL sandbox	K_{reuse} , adaptation time
Scalability	Increasing number of assets/models	KPI degradation, bandwidth

У концептуальному плані запропонований підхід розвиває кілька ліній сучасних досліджень одночасно. Від Ferle et al. [4] він успадковує ідею VC→DT reuse, але доповнює її формальною конфігураційною моделлю та можливістю зворотного переходу Operational DT→HIL sandbox. Від Djeulezeck et al. [5] — lifecycle continuity SIL/HIL/DT, але з явним separation temporal planes. Від Scheifele et al. [6] — real-time co-simulation mechanisms, які розміщуються саме в HIL plane. Від Zipper [14] — necessity of online synchronization, що піднята до окремого architectural layer. Від Reuther et al. [12] — стандартизоване представлення PLC/automation assets. Така комбінація формує не нову «схему блоків», а lifecycle execution model.

Наукова новизна полягає у формалізації physical/virtual composition як керованого configuration state; у dual-plane separation deterministic HIL та DT service traffic; у повторному використанні Unified Model Core між engineering і operation; у Context-of-Use adaptive fidelity; та у включенні credibility/VVUQ як lifecycle function. При цьому твердження «вперше» доцільно використовувати лише після фінального патентного й бібліографічного пошуку перед поданням рукопису.

Практичне значення підходу полягає в можливості безпечної перевірки controller software до deployment, повторного використання fault models і calibrated parameters, зменшення duplicated modelling, швидшого fault reconstruction та більш контрольованої реконфігурації CPPS. Найбільш очевидними application domains є роботизовані production cells, CNC/machine tools, embedded control systems, транспортні та енергетичні кіберфізичні системи.

Обмеження роботи визначаються тим, що універсальна architecture не гарантує універсальності конкретних physics models; adaptive fidelity вимагає modular model library; communication results залежать від конкретної network stack; cybersecurity і functional safety потребують окремої формальної оцінки; а empirical superiority proposed architecture повинна бути підтверджена на реальному або laboratory-scale CPPS. Саме тому в цій версії не подаються штучно сформовані «результати», а задається відтворюваний protocol їх одержання.

7. Перспективи подальшого розвитку дослідження

Подальший етап роботи має бути спрямований на реалізацію proposed architecture на фізичному або напівнатурному testbed та на відтворювану кількісну перевірку сформульованих гіпотез. Першочерговим завданням є порівняння conventional HIL, standalone DT, fixed integrated DT–HIL і lifecycle-adaptive DT–HIL на ідентичному controller software та однакових fault/network scenarios. Це дасть змогу відокремити ефект власне архітектурних механізмів від властивостей конкретного обладнання.

Другий напрям пов'язаний із автоматизацією adaptive fidelity та configuration selection. Перспективним є використання surrogate models, online identification, Bayesian updating, multi-rate co-simulation і алгоритмів multi-objective optimization. Для великих виробничих комплексів актуальними є distributed та federated Digital Twins, composition multiple DTs, semantic interoperability через Asset Administration Shell, а також узгодження з ISO 23247-5 digital thread та ISO 23247-6 digital twin composition.

Третій напрям стосується safety і cybersecurity. Розширення двонапрямної взаємодії physical asset ↔ Digital Twin створює додаткові attack surfaces і вимагає розмежування authority між predictive/analytical services та критичним controller domain. Подальші дослідження повинні поєднати configuration management із formal safety constraints, authenticated data exchange, anomaly-aware synchronization та контрольованим переходом до fail-safe modes.

Окремою перспективою є DT-assisted HIL sandbox, у якому snapshot актуального стану Operational DT автоматично формує початкові умови для ізольованої HIL-перевірки нового controller software. Такий режим може забезпечити безпечний pre-deployment test без втручання у працюючу production system та практично реалізувати безперервний engineering-to-operation digital thread.

8. Висновки

Проведений аналіз показав, що Digital Twin та Hardware-in-the-Loop еволюціонують від двох відносно відокремлених технологій до спільного model-based lifecycle environment. Сучасні роботи вже демонструють Virtual Commissioning models як operational DT services, HIL у середині production Digital Twin, PLC-level Digital Twins, online synchronization і lifecycle reuse. Водночас недостатньо формалізованим залишається питання, як керувати співвідношенням реальних і віртуальних компонентів та fidelity упродовж життєвого циклу.

Запропоновано lifecycle-adaptive DT–HIL architecture, у якій MiL, SiL, HIL, Equipment-in-the-Loop, Operational DT та DT-assisted HIL sandbox описуються як конфігураційні стани єдиного середовища. Центральним компонентом є Unified Model Core, що поєднує physics-based, behavioural, data-driven і fault/degradation models та допускає stage-specific activation і calibration.

Обґрунтовано розділення Real-Time HIL Plane і Digital Twin Information Plane. Перша підсистема орієнтована на bounded execution time, jitter і deadlines, друга — на interoperability, semantics, telemetry, analytics та scalable services. Їх взаємодія здійснюється через Synchronization and Calibration Layer із time alignment, buffering, state estimation, parameter identification та reality-gap control.

Конфігурацію формалізовано кортежем $C_k = \{P_k, H_k, M_k, D_k, S_k, F_k\}$, а вибір configuration подано як multi-objective problem за model error, latency, computational cost, risk і reconfiguration effort. Adaptive fidelity задається як пошук мінімально складної моделі, що одночасно задовольняє accuracy і real-time constraints.

Для подальшого експериментального дослідження сформовано VVUQ-oriented protocol, baseline configurations, independent factors, fault scenarios і performance metrics. Ключовим принципом наукової доброчесності є відокремлення проектних значень від фактичних результатів: numerical claims мають включатися до фінальної статті лише після відтвореного експерименту.

Запропонований підхід узгоджується з ISO 23247 і особливо з актуальними напрямками digital thread та digital twin composition. Подальша робота має бути спрямована на реалізацію prototype, empirical validation, sensitivity analysis, scalability testing і перевірку DT-assisted HIL sandbox для безпечного deployment змін у промисловій CPS.

9. Конфлікт інтересів

Автори заявляють про відсутність конфлікту інтересів щодо цього дослідження.

10. Фінансування

Дослідження виконано без зовнішнього цільового фінансування.

Список літератури:

- 1) Tao, F., & Zhang, M. (2017). Digital Twin Shop-Floor: A new shop-floor paradigm towards smart manufacturing. IEEE Access, 5, 20418–20427. <https://doi.org/10.1109/ACCESS.2017.2756069>

- 2) Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11), 1016–1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>
- 3) Liu, M., Fang, S., Dong, H., & Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58, 346–361. <https://doi.org/10.1016/j.jmsy.2020.06.017>
- 4) Ferle, F., Chen, S., Kuhn, A. M., Klingel, L., Schenke, C.-C., Tekouo, W., Verl, A., & Ihlenfeldt, S. (2026). Continued use of virtual commissioning models: A novel approach toward digital twins for automated production systems. *The International Journal of Advanced Manufacturing Technology*, 142, 4609–4626. <https://doi.org/10.1007/s00170-025-17195-y>
- 5) Djeulezeck Tanegue, H. B., De Paula Ferreira, W., De Assis, R. F., & Brodeur, D. (2025). Leveraging Virtual Commissioning for Digital Twins: An example case. *IFAC-PapersOnLine*, 59(10), 601–606. <https://doi.org/10.1016/j.ifacol.2025.09.103>
- 6) Scheifele, C., Verl, A., & Riedel, O. (2019). Real-time co-simulation for the virtual commissioning of production systems. *Procedia CIRP*, 79, 397–402. <https://doi.org/10.1016/j.procir.2019.02.104>
- 7) Leng, J., Zhou, M., Xiao, Y., Zhang, H., Liu, Q., Shen, W., Su, Q., & Li, L. (2021). Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. *Journal of Cleaner Production*, 306, 127278. <https://doi.org/10.1016/j.jclepro.2021.127278>
- 8) Barbieri, G., Bertuzzi, A., Capriotti, A., Ragazzini, L., Gutierrez, D., Negri, E., & Fumagalli, L. (2021). A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*, 15, 397–412. <https://doi.org/10.1007/s11740-021-01037-3>
- 9) Iyer, S. V., Sangwan, K. S., & Dhiraj. (2024). Digital twin-based virtual commissioning for evaluation and validation of a reconfigurable process line. *IET Collaborative Intelligent Manufacturing*, 6(3), e12111. <https://doi.org/10.1049/cim2.12111>
- 10) Ionescu, D., Filipescu, A., Simion, G., & Filipescu, A. (2025). Internet of Things-Cloud control of a robotic cell based on inverse kinematics, Hardware-in-the-Loop, Digital Twin, and Industry 4.0/5.0. *Sensors*, 25(6), 1821. <https://doi.org/10.3390/s25061821>
- 11) Ni, H., Hu, T., Deng, J., Chen, B., Luo, S., & Ji, S. (2025). Digital twin-driven virtual commissioning for robotic machining enhanced by machine learning. *Robotics and Computer-Integrated Manufacturing*, 93, 102908. <https://doi.org/10.1016/j.rcim.2024.102908>
- 12) Reuther, A., Plesker, C., Bassauer, D., & Schleich, B. (2026). A Digital Twin Framework for Programmable Logic Controllers in Industrial Automation Systems. *Procedia CIRP*, 139, 446–452. <https://doi.org/10.1016/j.procir.2025.08.193>
- 13) Kronová, J., Pekarčíková, M., Kliment, M., & Trebuňa, P. (2026). Development of an Online Digital Twin for Real-Time Monitoring of Manufacturing Processes Using OPC UA. *Processes*, 14(13), 2030. <https://doi.org/10.3390/pr14132030>
- 14) Zipper, H. (2021). Real-Time-Capable Synchronization of Digital Twins. *IFAC-PapersOnLine*, 54(4), 147–152. <https://doi.org/10.1016/j.ifacol.2021.10.025>
- 15) Hildebrandt, G., Dittler, D., Habiger, P., Drath, R., & Weyrich, M. (2024). Data Integration for Digital Twins in Industrial Automation: A Systematic Literature Review. *IEEE Access*, 12, 139129–139153. <https://doi.org/10.1109/ACCESS.2024.3465632>
- 16) Cho, Y., & Noh, S. D. (2024). Design and Implementation of Digital Twin Factory Synchronized in Real-Time Using MQTT. *Machines*, 12(11), 759. <https://doi.org/10.3390/machines12110759>
- 17) Park, K.-T., Park, Y. H., Park, M.-W., & Noh, S. D. (2023). Architectural framework of digital twin-based cyber-physical production system for resilient rechargeable battery production. *Journal of Computational Design and Engineering*, 10(2), 809–829. <https://doi.org/10.1093/jcde/qwad024>
- 18) Chen, S., Neubauer, M., & Verl, A. (2024). A Digital Twin Platform for service-based Testing and Optimization. *Procedia CIRP*, 126, 260–265. <https://doi.org/10.1016/j.procir.2024.08.336>

- 19) Lidell, A., Ericson, S., & Ng, A. H. C. (2022). The Current and Future Challenges for Virtual Commissioning and Digital Twins of Production Lines. *Advances in Transdisciplinary Engineering*, 21, 508–519. <https://doi.org/10.3233/ATDE220169>
- 20) Bitencourt, J., Wooley, A., & Harris, G. (2025). Verification and validation of digital twins: A systematic literature review for manufacturing applications. *International Journal of Production Research*, 63(1), 342–370. <https://doi.org/10.1080/00207543.2024.2357741>
- 21) Shao, G., Hightower, J., & Schindel, W. (2023). Credibility consideration for digital twins in manufacturing. *Manufacturing Letters*, 35, 24–28. <https://doi.org/10.1016/j.mfglet.2022.11.009>
- 22) International Organization for Standardization. (2021). ISO 23247-2:2021. Automation systems and integration — Digital twin framework for manufacturing — Part 2: Reference architecture.
- 23) International Organization for Standardization. (2021). ISO 23247-4:2021. Automation systems and integration — Digital twin framework for manufacturing — Part 4: Information exchange.
- 24) International Organization for Standardization. (2026). ISO 23247-5:2026. Automation systems and integration — Digital twin framework for manufacturing — Part 5: Digital thread for digital twin.
- 25) International Organization for Standardization. (2026). ISO 23247-6:2026. Automation systems and integration — Digital twin framework for manufacturing — Part 6: Digital twin composition.
- 26) Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–106. <https://doi.org/10.2307/25148625>
- 27) Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>