Clausen *et al. Energy Informatics* (2024) 7:12
https://doi.org/10.1186/s42162-024-00312-8

Energy Informatics

REVIEW

Open Access

A scoping review of In-the-loop paradigms in the energy sector focusing on software-in-the-loop

Christian Skafte Beck Clausen[1], Bo Nørregaard Jørgensen[1] and Zheng Grace Ma[1*]

*Correspondence: zma@mmmi.sdu.dk

[1] SDU Center for Energy Informatics, The Maersk Mc Kinney Moller Institute, University of Southern Denmark, Campusvej 55, 5230 Odense M, Odense, Denmark

Abstract

Software-in-the-Loop (SIL) testing is an approach used for verification and validation in the energy sector. However, there is no comprehensive overview of the application, potential, and challenges of SIL within this sector. Therefore, this paper conducts a thorough scoping review of the existing literature within the scope of SIL and related in-the-loop approaches in the energy sector. A total of 88 full-text articles from four significant databases ACM, IEEE Xplore, Scopus, and Web of Science are analyzed and categorized to map the purpose, methods, architecture, interoperability and protocols, technologies, challenges, and limitations. The results present a grand perspective of in-the-loop across several domains followed by an analysis of SIL in the energy sector. The application domains carry characteristics from complex systems, systems-of-systems, cyber-physical systems, critical systems, real-time systems, and sociotechnical systems. The energy sector and the automotive industry are amongst the most applied domains. Within energy- and electricity systems, hardware-based in-the-loop paradigms are mostly applied for testing low-level signaling, and SIL is used for control strategy testing, optimization, dispatching, and experimentation. The examined SIL architectures have distributed-, real-time, and closed-loop properties, and are constrained by specialized simulation power hardware. Future research should address how to systematically develop SIL testing environments with guiding principles to support application development for the future digitalized energy system.

Keywords: In-the-loop, Software-in-the-loop, Scoping review, Energy sector, Energy systems, Electricity systems

Introduction

The energy systems are undertaking a digital transformation to combat the climate crisis, provide accessible and affordable energy, and minimize dependency (https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52022DC0552). Many challenges arise when extending the applications of the energy system with smart capabilities e.g., smart grid, demand-response (DR), and distributed energy resources (DER) (Dekeyrel and Fessler 2023). The digital transformation exposes the energy system to challenges similar to complex systems (Hanel et al. 2018a), system-of-systems (Sommerville 2016), cyber-physical systems (Lee 2008, 2010; Derler et al. 2012), critical systems, and sociotechnical systems. Interoperability is highlighted as one of the key challenges to developing applications that utilize information across the entire energy system (Gopstein et al. 2020; Papaioannou, et al. 2018; Directorate-General for Energy 2018). Furthermore, an interconnected, decentralized, resilient, and self-adaptable energy system is characterized by complex dynamics and emergent behavior. The complex and emergent behavior is not only hard to model, but stakeholders will have difficulty in verifying and validating system scenarios with traditional methods.

Verification & validation (V&V) processes can be performed for every integration level in the development lifecycle to ensure integrity and to enable early feedback. The resulting feedback can be used to validate and modify the product in a timely and cost-efficient manner. V&V takes up more than 50% of the total development costs in critical systems (https://software-engineering-book.com/web/critical-systems/). Critical systems exist in many sectors e.g., energy, automotive, medical, aviation, military, etc. where failures may result in tremendous costs including loss of infrastructure capability, injuries, death, and economic or environmental damages. Therefore, critical systems need to be reliable, and the integrity must be thoroughly verified and validated at all levels (system, hardware, and software) before operation.

V&V can be achieved through various approaches e.g., model-based design (MBD), in-the-loop testing, formal methods, model checking, property-based testing, co-simulation, multi-agent simulation, (virtual) prototyping, static code analysis, and debugging. MBD is an engineering method applied in critical control systems where V&V activities are prioritized. The development steps involve (i) modeling the plant or process to be controlled, (ii) modeling the controller, (iii) establishing interoperability between the plant and controller, (iv) simulating the symbiotic behavior, and (v) collecting metrics for V&V analysis. Sophisticated development tools like MATLAB/Simulink and Modelica/Dymola exist to implement the plant and controller inside an integrated environment using block elements. The MBD lifecycle follows the V-model where system integrity and rapid feedback are ensured across all levels (system, hardware, and software) to reduce faults, risks, and costs before operation.

However, traditional testing methods are unsuitable for holistically verifying and validating the energy system. Therefore, there is a need to develop integrated methods and tools that can assist stakeholders to verify and validate the behavior and interoperability of the future energy sector. Efforts on this matter are documented in the European Guide to Power System Testing (Strasser and Jong 2020). SIL testing holds considerable promise as a potential solution for verification issues in the energy sector. However, there is a notable absence of a comprehensive overview of the application, potential, and challenges of SIL within this sector.

Therefore, the primary objective of this paper is to conduct a thorough scoping review of the existing literature within the scope of SIL in the energy sector. This review will provide a panoramic view of the current state of SIL application in the energy domain, including its methodological approaches, benefits, limitations, and contextual variations. Furthermore, this paper aims to identify significant gaps in the current research landscape regarding SIL in the energy sector. By scrutinizing the literature, this paper will highlight underexplored areas, thereby illuminating possible directions for future research. Through these endeavors, we anticipate this paper will serve as a valuable resource for researchers, industry professionals, and policymakers, equipping them with the insights necessary to navigate and shape the future of SIL in the energy sector.

The paper is organized as follows: section Scoping review method details the scoping review method. Section Results presents the results of the literature analysis in the dimensions of interest. Section Discussion provides a discussion of the results. Section Conclusion concludes the paper with suggestions on future research directions.

Scoping review method

The scoping review process is depicted in Fig. 1. The approach involves the formulation of multiple search strings given the scope. The search strings were transformed into database-specific syntax compatible with the selected databases. The search results were downloaded, and unique references were found across the entire set of references. The unique references were screened based on the inclusion criteria. The screened references were downloaded in full text and analyzed individually. The following subsections describe the review method in detail.

Search string formulation

Table 1 shows the identified search terms based on the scope. The domain includes terms from the energy system that impose integration challenges. The topic focuses

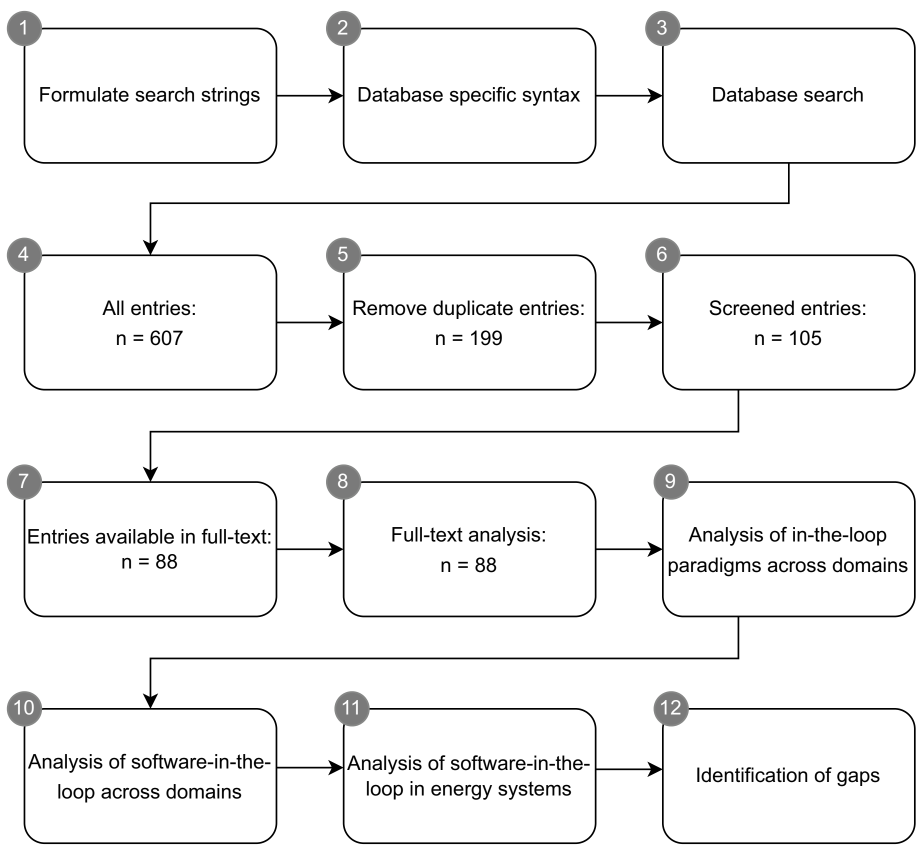

Fig. 1 The scoping review process

Table 1 Search terms in the literature search

Category	Search terms
Domain	electricity *OR* distribution grid *OR* demand side *OR* consumption side *OR* distributed energy resources *OR* DER *OR* solar *OR* photovoltaic* *OR* PV *OR* electric vehicle *OR* EV *OR* heat pump
Topic	architect* *OR* pattern* *OR* communicat* *OR* connect* *OR* protocol* *OR* interop* *OR* interconnect* *OR* *exchange* *OR* feedback* *OR* synchroniz* *OR* distribut* *OR* digital environment *OR* virtual environment *OR* emulator
Methodology	*in*the*loop *OR* model-based systems *OR* model-based software *OR* model-based design *OR* MBD *OR* MBS*

on architecture, communication patterns, and virtual environments because these are fundamental issues for in-the-loop testing. The methodology is MBD, including model-based software, model-based systems, and in-the-loop.

The search terms were used to construct the generic search strings in Table 2. The generic search strings were then translated into the database-specific syntax (see Table 14 appendix A). Multiple search strings were constructed to explore the extent of the literature. G1 is a narrow search string that includes domain terms and all 'in-the-loop' paradigms. G2 focus on SIL and architectures, patterns, and frameworks. G3 is a broad search string that includes all SIL literature. G4 is almost identical to G1, but G4 is domain- and methodology agnostic.

Evidence collection

Four significant and reputable databases within the engineering field were targeted for evidence collection:

- Web of Science (WoS)
- Scopus
- The ACM Guide to Computing Literature (ACM)
- The Institute of Electrical and Electronics Engineers (IEEE) Xplore.

Table 2 Generic search strings

ID	Generic search string
G1	*Title* = (electricity *OR* distribution grid *OR* demand side *OR* consumption side *OR* distributed energy resources *OR* DER *OR* solar *OR* photovoltaic* *OR* PV *OR* electric vehicle *OR* EV *OR* heat pump) *AND* (architect* *OR* pattern* *OR* communicat* *OR* connect* *OR* protocol* *OR* interop* *OR* interconnect* *OR* exchange *OR* feedback* *OR* synchroniz* *OR* distribut* *OR* digital environment *OR* virtual environment *OR* emulator) *AND* (in-the-loop *OR* model-based systems *OR* model-based software *OR* model-based design *OR* MBD OR MBS*)
G2	*Title* = (software-in-the-loop) *AND* (architect* *OR* pattern *OR* framework)
G3	*Title* = (software-in-the-loop)
G4	*Title* = ("software-in-the-loop") *AND* (architect* *OR* framework *OR* platform *OR* pattern *OR* interop* *OR* interconnect* *OR* environment *OR* exchange *OR* communicat* *OR* protocol *OR* distribut* *OR* synchroniz* *OR* emulator *OR* simulat*)

Table 3 Database search hits

Search string ID	Date of search	Hits in database				Hits across databases		
		WoS	Scopus	ACM	IEEE	Total hits	Unique hits	
G1	2022-09-27	25	43	1	21	90	46	199
G2	2022-09-29	6	8	0	6	20	14	
G3	2022-09-29	92	148	4	61	305	166	
G4	2022-10-03	56	92	4	40	192	107	

The search had no date restriction, and the database hits are shown in Table 3. The evidence was managed in the Mendeley reference manager and was organized into hierarchical subfolders corresponding to each search string and database. This enabled aggregation of the results in a parent folder where duplications were removed in a two-step process: (i) With the 'Check for Duplicates' functionality and (ii) by manual inspection of duplicate entries. This left 199 unique hits across all databases and search strings. The evidence content type included conference proceedings, journal articles, short papers, early access, and book chapters in English (see Table 15 in appendix B).

The unique hits of each search string reveal the extent of the literature. The search strings G1 and G2 provide a narrow evidence base for the scoping review. The inclusion of search strings G3 and G4 provides a wider evidence base. Collectively, the search strings enable comprehensive analysis with a total of 199 unique references.

Inclusion criteria and evidence screening

The 199 unique references were screened for relevance based on the following inclusion criteria:

Based on the title or abstract the paper *must*:

1. Apply one of the stated methodologies in combination with in-the-loop
2. AND
3. (Address one of the stated topics *OR*
4. Address one of the targeted domains OR
5. Indicate general applicability, i.e., domain agnostic)

No exclusion criteria were formulated because in this case such criteria would be the inverse of the inclusion criteria. For example, one inclusion criterion is to address one of the stated topics. The inverse of this criterion is that the paper does not address one of the stated topics.

105 references were included based on the criteria by which 88 references were available for full-text download. The unavailable references were either behind a paywall or unavailable for download (see Table 16 in appendix C).

The publication trend of the literature is shown in Fig. 2. Most research was published in 2010–2019 and declined in 2020. The trend increased again in 2021. The year 2022 is not fully covered, but it is expected that the number of references is around the same or slowly declining compared to 2021. The selected references follow the publication trend and span most of the period.

Full-text examination and data extraction

The selected references were carefully and systematically examined, and the evidence was extracted into dimensions with the use of a spreadsheet. Each reference was recorded together with 13 columns corresponding to each dimension as shown in Table 4. These dimensions map to the aim of the scoping review.

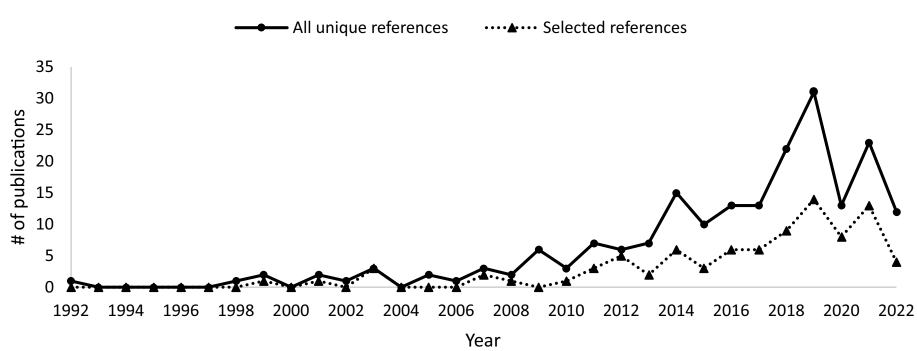

Fig. 2 Publication trend from 1992 to 2022

Table 4 Dimensions extracted from the analyzed material

Dimension	Description
Domains	The domain(s) of the study
Focused topics	The topics that the study address. Usually the keywords
Purpose	The stated purpose of conducting the study
Methods	The applied methods or approaches
The System Under Test (SUT) and plant (virtual model)	The SUT refers to a system or part of a system
Performance metrics	The performance metrics that the study collects of the SUT and plant using the in-the-loop test
Architecture and communication patterns	The core elements and their relationships
Interoperability protocols	The standardized interoperability protocols used for information exchange
Technologies	Concrete technologies applied within the study
Challenges	Stated challenges in the study
Limitations	Stated limitations within the study
Research Gaps and future work	Stated research gaps and future work stated within the study

Results

This section systematically presents the analysis of the selected literature. The SIL application domains are presented first to provide the breadth of SIL. Subsequently, the focus shift to energy- and electricity systems and explores the various in-the-loop paradigms in this domain. Finally, the focus narrows to SIL in energy- and electricity systems and provides a detailed analysis in each dimension; purpose, methods, SUT & plant, performance metrics, architecture, communication patterns, interoperability protocols, and technologies.

Application domains and scope of software-in-the-loop

Table 5 shows an overview of the SIL applications in the analyzed literature, including the domains and subdomains together with the frequency. A reference may span multiple domains—for example (Nguyen et al. 2019a) is located within Grids, Renewables, and Low inertia systems. The application domains carry characteristics from complex systems, systems-of-systems, cyber-physical systems, critical systems, real-time systems, and sociotechnical systems. These types of systems are notoriously difficult to test due to their inherent complexity. However, the applications require significant upfront testing because faults may lead to major losses including death, injury, and economic losses. The SIL testing paradigm focuses on verifying the emergent behavior between the controller and the virtual model. This includes verifying that the control strategy can achieve the intended functionality by manipulating the virtual model (Raghupatruni et al. 2019). The functionality can be verified by observing the response of the virtual model. Non-functional behavior e.g., performance, interoperability, real-time constraints, etc. may also be verified (Raghupatruni et al. 2019).

SIL testing may vary in scope, meaning that the approach can be used to verify different aspects of the overall system (Raghupatruni et al. 2019). The SIL testing

Table 5 Application domains of SIL

Domain	Sub-domains	References	Frequency
Energy- and electricity systems	Grids (power/distribution/smart/low-voltage/medium-voltage/micro)	Nguyen et al. (2019a), Guerrero, et al. (2016), Tuominen et al. (2017), Nguyen et al. (2019b), Osadcuks and Galins (2012), Bonassi 2020; Guerrero et al. (2021)	7 (9.9%)
	Renewables	Nguyen et al. (2019a), Huber et al. (2014), Rossi et al. (2019), Singh and Shubhanga (2017), Frotscher et al. (2019), Bonassi (2020)	6 (8.5%)
	Distributed energy resources	Osadcuks and Galins (2012), Ponchant et al. (2021), Gourisetti, et al. (2018)	3 (4.2%)
	Gas turbines	Rossi et al. (2019), Kwon and Choi (1999)	2 (2.8%)
	Low inertia systems	Nguyen et al. (2019a)	1 (1.4%)
Automotive	Automotive	Raghupatruni (2019), Stevi (2020), Shivanandaswamy et al. (2021), Bailey et al. (2014), Chen et al. (2008), Parthasarathy and Johansson (2021), Russo et al. (2007), Ersdi et al. (2021)	8 (11.3%)
	Electric vehicles	Casolino et al. (2016), Li et al. (2020)	2 (2.8%)
	Autonomous vehicles	Ahamed et al. (2018)	1 (1.4%)
	Automated driving	Strohbeck et al. (2021)	1 (1.4%)
	Battery management in electric vehicles	Ponchant (2021)	1 (1.4%)
	Hybrid vehicles	Vandi et al. (2014)	1 (1.4%)
	Traffic motion	Strohbeck et al. (2021)	1 (1.4%)
	Vehicle power management	Zhang and Mi (2011)	1 (1.4%)
	Vehicle-to-vehicle information sharing	Brambilla et al. (2014)	1 (1.4%)
Embedded systems	Embedded systems	Muresan and Pitica (2012; Werner et al. (2015)	2 (2.8%)
	Electronic circuits	Lai and Lin (2021)	1 (1.4%)
	Distributed real-time systems	Pieper and Obermaisser (2018a)	1 (1.4%)
	Home appliances	Park et al. (2020)	1 (1.4%)
	Image processing	Werner et al. (2015)	1 (1.4%)
Generally applicable	Clock synchronization in software-in-the-loop	Lee et al. (2017)	1 (1.4%)
	Computer network analysis and simulation	Demers et al. (2007)	1 (1.4%)
	Job scheduling in multiprocessor systems	Collins and George (2001)	1 (1.4%)
	Parallel Agent Discrete Event Simulation	Riley and Riley (2003b)	1 (1.4%)
	Pervasive systems	Brambilla et al. (2014)	1 (1.4%)
Manufacturing	Automated Manufacturing Systems	Bonivento et al. (2011)	1 (1.4%)
	Multi-agent-based manufacturing	Scholz et al. (2017)	1 (1.4%)
	Material flow	Nagy et al. (2012)	1 (1.4%)
	Railway Manufacturing	Pieper and Obermaisser (2018b)	1 (1.4%)

Table 5 (continued)

Domain	Sub-domains	References	Frequency
Unmanned aerial vehicles	UAV	Bittar et al. (2014), Bayha et al. (2012)	2 (2.8%)
	Rotary wing	Silva et al. (2019), Silano et al. (2019)	2 (2.8%)
	Fixed wing	Yang et al. (2019)	1 (1.4%)
	UAV Swarms	Mastronarde et al. (2022)	1 (1.4%)
Robots	Industrial robots	Jaensch et al. (2019)	1 (1.4%)
	Lithography	Nagy et al. (2012)	1 (1.4%)
	Robot manipulators	Ayed et al. (2017)	1 (1.4%)
Satellites	Nanosatellites	Pereira et al. (2016), Fatimi et al. (2022)	2 (2.8%)
	Attitude determination	Kiesbye et al. (2019)	1 (1.4%)
	Microsatellites	Hassani and Lee (2013)	1 (1.4%)
	Satellite formation flying	Park et al. (2013)	1 (1.4%)
Railway	Unmanned railway vehicles	Vignati et al. (2021)	1 (1.4%)
	Radio communication for railway systems	Moreno et al. (2019)	1 (1.4%)
DC-motor control	–	Werner et al. (2015), Taut et al. (2019)	2 (2.8%)
Military	Joint Tactical Radio Systems	Martinez et al. (2003)	1 (1.4%)
Educational	Electro-mechanical engineering (Windowlifters)	Beyer et al. (2017)	1 (1.4%)
	Automation and supervisory systems education (SCADA systems)	Calderón and González (2018)	1 (1.4%)

scope is defined by the testing objectives in a given scenario and the objectives in the reviewed literature can be summarized according to the application domains:

- In energy- and electricity systems, SIL has been used for control strategy testing (Nguyen et al. 2019a; Guerrero, et al. 2016; Huber et al. 2014; Rossi et al. 2019; Singh and Shubhanga 2017; Tuominen et al. 2017), optimization (Nguyen et al. 2019a, b; Tuominen et al. 2017; Frotscher, et al. 2019), dispatching/scheduling (Osadcuks and Galins 2012; Bonassi 2020), and experimentation (Kwon and Choi 1999). A detailed analysis of SIL in energy- and electricity systems are provided in later sections.
- In the automotive domain, SIL has been used to test battery management systems Frotscher, et al. 2019 [135], ESP software (Kwon and Choi 1999), geographically distributed subsystems in real-time (Pieper and Obermaisser 2018a), advanced driver assistance systems (Taut et al. 2019), and automated driving functionality (Ersdi et al. 2021), autonomous vehicles [188], power steering control systems [129], brake control systems [159], and inverter control strategies [170].
- For embedded systems, SIL has been used for testing geographically distributed co-simulations (Pieper and Obermaisser 2018a), electronic circuits (Lai and Lin 2021), home appliances with property-based testing of the source code (Park et al. 2020), embedded code for DC motor control (Muresan and Pitica 2012; Werner et al. 2015), and image processing (Werner et al. 2015).

- In manufacturing, SIL has been used for rapid prototyping of automated manufacturing (Bonivento et al. 2011), and material flow simulation (Scholz et al. 2017; Nagy et al. 2012). For these applications, agent-based and discrete event simulations are often used.
- For unmanned aerial vehicles (UAV), SIL is used to test adaptive formation controllers (Yang et al. 2019), altitude controllers using proportional integral derivative (PID) control (Silva et al. 2019), pathfinding (Bittar et al. 2014), and radio frequency communication in swarm UAV applications (Mastronarde et al. 2022).
- In robot systems, SIL has been used for industrial robots (Jaensch et al. 2019), and verification of source code for a robot manipulator (Ayed et al. 2017).
- In satellite applications such as microsatellites and nanosatellites, SIL has been used for attitude control and determination using PID control (Hassani and Lee 2013; Kiesbye, et al. 2019; Pereira et al. 2016), and formation flying (Park et al. 2013).
- For railway systems, SIL has been used to integrate geographically distributed railway components with co-simulation (Pieper and Obermaisser 2018b) and to test the control unit for driverless railway vehicles (Vignati et al. 2021).

Generally applicable SIL methods have been suggested to solve particular challenges such as relative time synchronization (Lee et al. 2017), distributed agent simulation (Riley and Riley 2003a, b), pervasive simulation (Brambilla et al. 2014), and job scheduling in heterogenous cluster computing (Bonivento et al. 2011).

In many of the analyzed application domains the controller and the controlled process are separated and communicate over a network e.g., TCP/IP (Nguyen et al. 2019a), UDP/IP (Bittar et al. 2014), OPC UA (Nguyen et al. 2019a, b; Bonivento et al. 2011), or other domain-specific protocols. The communication topology is either point-to-point, publish/subscribe (Pieper and Obermaisser 2018b; Ahamed et al. 2018), or file sharing (Frotscher, et al. 2019; Werner et al. 2015) and is mostly in real-time (Guerrero, et al. 2016, 2021; Nguyen et al. 2019b; Hassani and Lee 2013; Vignati et al. 2021; Ponchant et al. 2021). The complexity of the controller depends on the applied method within the controller logic. For example, PID control (Nguyen et al. 2019a; Muresan and Pitica 2012; Silva et al. 2019; Hassani and Lee 2013; Pereira et al. 2016; Fatimi et al. 2022; Casolino et al. 2016; Taut et al. 2019), and genetic algorithms (Nguyen et al. 2019b; Newaz et al. 2020).

In-the-loop paradigms in energy- and electricity systems

The in-the-loop approach is concerned with MBD V&V on the emergent behavior between the plant and controller in a virtual environment at different abstraction levels. Early use of in-the-loop can be traced back to military applications (Garcia et al. 1994; Lynch et al. 1989; Organization 1988). The approach aims to objectively collect evidence to confidently ensure that the controller and System Under Test (SUT) conform to the V&V scenarios.

Common in-the-loop paradigms shown in Fig. 3 include model-in-the-loop (MIL), software-in-the-loop (SIL), processor-in-the-loop (PIL), and hardware-in-the-loop (HIL). MIL verifies the design of a controller on a virtual model, often developed and modeled in schematic block diagrams. MIL is used early in the development cycle to test

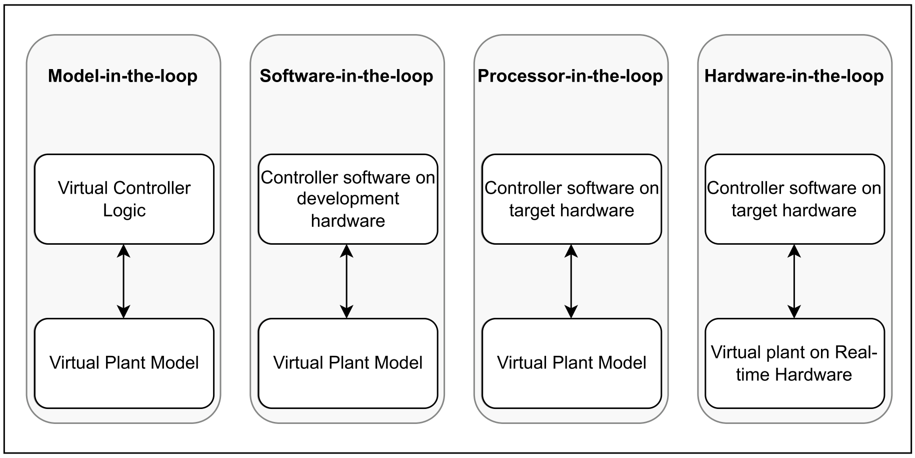

Fig. 3 Common in-the-loop paradigms—MIL, SIL, PIL, and HIL

the controller design. SIL verifies the controller code on virtual models. The output of the controller serves as input to simulated virtual models. These models mimic physical systems/hardware with a high degree of complexity and real-time properties.

The in-the-loop paradigm has similarities with the Process Control paradigm (Shaw 1994; Åström and Wittenmark 2011). The Process Control paradigm separates the process of interest in a virtual model from the control policy i.e., the controller. The interaction between the controller and the virtual model is that the controller receives values from the process and the controller continuously guides the process to achieve a defined objective. The collected evidence depends on the scenario and may address functional or non-functional issues. For example, a scenario may focus on the functionality to verify that the controller can control the SUT to satisfy a functional requirement. Non-functional focus on qualitative issues, for example, reliability, performance, safety, adaptability, resiliency, etc. The advantage of in-the-loop testing lies in the early fault detection which lowers the overall project costs. Furthermore, the controller software can be generated from block diagrams through code generation, reducing development effort. Advantages of SIL include early verification, fault detection, reduced hardware costs, and rapid feedback cycles. PIL verifies the controller code executed on the target hardware against a virtual model. PIL testing identifies faults when the controller code is executed on the targeted hardware platform. HIL verifies the controller output in a simulated environment involving physical equipment that is limited by real-world constraints. Higher costs are involved with HIL testing and are used late in the development process to test real-world behavior or integration issues that cannot be detected by earlier tests. HIL runs in real-time and consequently has a slow feedback cycle.

Table 6 summarizes the results of various in-the-loop paradigms which have been applied within energy- and electricity systems in the reviewed literature. Some studies mix the paradigms and therefore, a single study may appear multiple times across each paradigm, as is the case with (Kotsampopoulos et al. 2018). Paradigms that rely on hardware-based virtual plants account for 71.1% of the examined studies. The hardware-based paradigms include HIL, PHIL, CHIL, FPGA-IL, and Power and communications hardware-in-the-loop (PCommHIL). Software-based in-the-loop paradigms account for

Table 6 In-the-loop paradigms of energy- and electricity systems

Application domain	Hardware/software-based virtual model	Paradigm	References	Frequency
Energy- and electricity systems	Hardware-based	Hardware-in-the-loop (HIL)	Nguyen et al. (2019a), Tuominen et al. (2017), Ponchant et al. (2021), Gourisetti, et al. (2018), Kotsampopoulos et al. (2018), Musse et al. (2017), Han et al. (2022), Venturi et al. (2015), Ravikumar et al. (2020), Wang et al. (2021b), Caro et al. (2022), Wang, et al. (2020), Cao et al. (2019), Wang et al. (2021a), Rotger-Griful et al. (2016), Kim et al. (2030)	16 (42.1%)
	Hardware-based	Power-hardware-in-the-loop (PHIL)	Guerrero, et al. (2016), Kotsampopoulos et al. (2018), Viehweider et al. (2012), Seo et al. (2011), Zhang et al. (2021), Mo et al. (2014), Huerta et al. (2016), Wang et al. (2021b), Wang et al. (2020), Herdt et al. (2021), Wang et al. (2021a), Hubschneider et al. (2018)	12 (31.6%)
	Software-based	Software-in-the-loop (SIL)	Nguyen et al. (2019a), Guerrero et al. (2016), Huber et al. (2014), Rossi et al. (2019), Singh and Shubhanga (2017), Tuominen et al. (2017), Nguyen et al. (2019b), Frotscher et al. (2019), Osadcuks and Galins (2012), Bonassi (2020), Kwon and Choi (1999), Guerrero et al. (2021)	12 (31.6%)
	Hardware-based	Controller Hardware-in-the-loop (CHIL)	Newaz et al. (2020), Kotsampopoulos et al. (2018), Musse et al. (2017), Zhang et al. (2021; Wang et al. (2020)	5 (13.2%)
	Software-based	Processor-in-the-loop (PIL)	Youcefa et al. (2019), Fekkak et al. (2020)	2 (5.3%)
	Hardware-based	FPGA-in-the-loop (FPGA-IL)	Singh and Shubhanga (2017; Morales-Caporal et al. (2018)	2 (5.3%)
	Hardware-based	Power and communications hardware-in-the-loop (PCommHIL)	Zhang et al. (2021)	1 (2.6%)
	Software-based	Controller software-in-the-loop (CSIL)	Nagy et al. (2012)	1 (2.6%)

39.5% which includes SIL, PIL, and Controller-software-in-the-loop (CSIL). The over-representation of hardware-based paradigms may be explained by the need for verifying real-time aspects of the system. However, studies rarely discuss the reasons for using hardware-based vs. software-based in-the-loop verification approaches. The hardware-based in-the-loop studies involve exploration of the integration of DER into power grids (Newaz et al. 2020; Kotsampopoulos et al. 2018; Musse et al. 2017; Han et al. 2022; Morales-Caporal et al. 2018; Venturi et al. 2015; Viehweider et al. 2012; Seo et al. 2011; Zhang et al. 2021; Mo et al. 2014; Huerta et al. 2016), real-time simulation (Nguyen et al. 2019a; Newaz et al. 2020; Gourisetti, et al. 2018; Kotsampopoulos et al. 2018; Musse et al. 2017; Han et al. 2022; Venturi et al. 2015; Viehweider et al. 2012; Seo et al. 2011; Mo et al. 2014; Huerta et al. 2016; Ravikumar et al. 2020; Wang et al. 2021b, 2020, 2021b; Caro et al. 2022; Herdt et al. 2021; Cao et al. 2019), cybersecurity (Gourisetti, et al. 2018; Zhang et al. 2021; Ravikumar et al. 2020), geographically distributed entities (Gourisetti, et al. 2018), co-simulation (Gourisetti, et al. 2018; Venturi et al. 2015; Wang et al. 2021a; Rotger-Griful et al. 2016; Kim, et al. 2030), closed-loop control (Musse et al. 2017), digital twins testing (Han et al. 2022), demand response (Rotger-Griful et al. 2016), multi-objective optimization (Wang et al. 2021b), distribution management systems (Wang et al. 2020, 2021a,b; Kim, et al. 2030), vehicle-to-grid integration (Caro et al. 2022; Herdt et al. 2021; Cao et al. 2019), and island mode (Kotsampopoulos et al. 2018; Mo et al. 2014).

Power-hardware-in-the-loop (PHIL) is a specialization of HIL that is often used to test applications in the electricity domain (Hubschneider et al. 2018). The paradigm is applied in power systems to study and test the performance and interactions given different control strategies. PHIL uses specialized hardware which enables real-time simulation of virtual power grids and circuits. The virtual circuits are built from virtual electrical components such as amplifiers, rectifiers, photovoltaics (PV), inverters, and more. The virtual electrical circuits are deployed to a real-time power hardware simulator. The input/output of the simulator is low-voltage signals that can be amplified to imitate real-world signals.

Purposes of software-in-the-loop applications in energy- and electricity systems

SIL has been used for four major purposes as summarized in Table 7: Control strategy testing/evaluation, optimization dispatching/scheduling, and experimentation. In power grids, generally, SIL has been used to evaluate a power oscillation damping (POD) controller of a synchronous condenser (Nguyen et al. 2019a). Furthermore, the authors Nguyen et al. (2019a) optimized the parameters for an IEEE standard automatic voltage regulator (AVR). In distribution grids, SIL has been used to test coordinated Volt/Var control strategies by the same authors Guerrero, et al. (2016, 2021). The authors in Tuominen et al. (2017) applied SIL to both evaluate the functionality of a distribution automation architecture and to evaluate a power control algorithm. In microgrids, SIL has been used for both control strategy testing and optimization for microgrids. Osadcuks and Galins (2012), Bonassi (2020) studied dispatch and scheduling algorithms using SIL, while Lai and Lin (2021) used optimization to deal with oscillatory stability issues.

For renewable energy sources, the authors in Huber et al. (2014) used SIL to evaluate solar cooling control strategies in solar thermal systems. For photovoltaics, SIL has been used to test maximum power point tracker (MPPT) control (Singh and Shubhanga 2017). A more complex system consisting of district heating, combined heat and power plant, solar collectors, and thermal energy storage had its operation optimized

Table 7 Purposes of SIL in energy- and electricity systems

Category	Purpose	References
Control strategy testing/evaluation	Testing a Coordinated Volt/VAR Control strategy	Guerrero et al. (2016), Guerrero et al. (2021)
	Evaluation of solar cooling control strategies	Huber et al. (2014)
	Test and verification of control logic and its reliability governing a power plant	Rossi et al. (2019)
	Evaluation of a POD controller of a synchronous condenser	Nguyen et al. (2019a)
	Evaluation of the functionality of a novel decentralized distribution automation architecture	Tuominen et al. (2017)
	Evaluation of a power control algorithm	Tuominen et al. (2017)
	Testing of MPPT control for photovoltaics	Singh and Shubhanga (2017)
Optimization	Parameter optimization to deal with oscillatory stability issues	Nguyen et al. (2019b)
	Parameterization of an IEEE standard automatic voltage regulator (AVR)/excitation system	Nguyen et al. (2019a)
	State estimation to minimize network losses, curtailed production of distributed generation units, minimize the number of On-Load Tap-Changers (OLTC) actions, and maintain as flat a voltage profile as possible	Tuominen et al. (2017)
	To optimize the operation of renewable energies—district heating, combined heat and power, solar collectors, and thermal energy storage	Frotscher et al. (2019)
Dispatching/scheduling	Evaluation of advanced resource dispatch strategies for autonomous hybrid power systems including renewable energy sources	Osadcuks and Galins (2012)
	Test of a distributed day-ahead scheduling algorithm	Bonassi (2020)
Experimentation	Experimentation with distributed control structures	Kwon and Choi (1999)

(Frotscher et al. 2019). Furthermore, SIL was studied to test and verify control logic and its reliability governing a power plant (Rossi et al. 2019) and to experiment with distributed control structures (Kwon and Choi 1999). The distributed control structures, in this case, mean that the control algorithm was separated from the plant.

Experimental validation and reported benefits of SIL testing

The previous Sect. 3.3 accounted for the overall purpose of SIL. This section reports how studies have validated or benefitted from SIL. Table 8 shows the analysis overview. The authors in Guerrero et al. (2016) focus on validating the closed-loop interoperability aspects between the control algorithm and the simulator over a TCP/IP-based network instead of using shared files. The authors in Rossi et al. (2019), Tuominen et al. (2017) reported that SIL was used to validate the control approach prior to field testing. The authors in Singh and Shubhanga (2017), Kwon and Choi (1999) reported that SIL can be useful in classrooms for teaching in control education or for demonstration purposes. The authors in Singh and Shubhanga (2017) also reported that SIL can be useful for both testing and deploying optimal control algorithms. The authors of Osadcuks and Galins (2012), Kwon and Choi (1999) reported that SIL provided design and

Table 8 Experimental validation and report advantages of SIL

Validation and reported benefits of SIL	References
Closed-loop interoperability validation	Guerrero et al. (2016)
Validation of control approach prior to field testing	Rossi et al. (2019), Tuominen et al. (2017)
Useful in classroom teaching, control education or demonstrations	Singh and Shubhanga (2017), Kwon and Choi (1999)
Testing and deploying optimal control algorithms	Singh and Shubhanga (2017)
Design and modification benefits of control algorithms (also in high-level programming languages)	Osadcuks and Galins (2012), Kwon and Choi (1999)
Enables model-driven architecture and software development	Osadcuks and Galins (2012)
Enables transferring the control system to the field with minimal change	Osadcuks and Galins (2012)
Analysis of system behavior under different system operating scenarios	Guerrero et al. (2021)
Analysis of real-time performance of the control algorithm	Guerrero et al. 2021)

modification benefits, and furthermore enabled implementing the control algorithm in high-level programming languages as opposed to low-level programming in e.g. C/C++. The authors in Osadcuks and Galins (2012) also noted that the control system could be transferred to the field with minimal change. The authors in Guerrero et al. (2021) used SIL to (i) validate the system behavior under different system operating scenarios (functional behavior), and (ii) to analyze the real-time performance of the control algorithm (non-functional behavior).

Software-in-the-loop methods and approaches applied in energy- and electricity systems

Table 9 shows the methods and approaches used for SIL in energy- and electricity systems, and they can be divided into eight categories. The method categories are closely related to the purpose categories because each method is used to realize the application purpose.

- Simulation methods

Simulation is inherent in the SIL paradigm. A model of a plant must be simulated over time to estimate the performance and impact of control strategies. Real-time simulation is a dominant simulation approach (Nguyen et al. 2019a, b; Guerrero et al. 2016, 2021; Singh and Shubhanga 2017; Tuominen et al. 2017; Bonassi 2020; Kwon and Choi 1999). This approach advances in sync with "wall clock" time—either because of the computational complexity or to study the system behavior under real-time constraints. Alternative approaches such as Frotscher et al. (2019) use TRNSYS which is also a time-dependent simulation approach. The authors in Huber et al. (2014) use equation-based and Number of Transfer Units (NTU) simulation models in Dymola.

- Optimization methods

Optimization is a frequently used approach for minimizing/maximizing objectives such as control variables or parameters of a control strategy. Biological-inspired algorithms i.e., Artificial Immune Systems Optimization (AISO) (Guerrero et al. 2016, 2021), Genetic Algorithms (GAs) (Nguyen et al. 2019a, b), and Clonal Selection Algorithm

Table 9 Methods and approaches of SIL in energy- and electricity systems

Category	Method/approach	References
Simulation	Real-time simulation	Nguyen et al. (2019a), Guerrero, et al. (2016), Singh and Shubhanga (2017), Tuominen et al. (2017), Nguyen et al. (2019b), Bonassi (2020), Kwon and Choi (1999), Guerrero et al. (2021)
	Unspecified method	Huber et al. (2014), Frotscher et al. (2019)
	Discrete-time simulation	Osadcuks and Galins (2012)
Optimization	Artificial Immune Systems optimization	Guerrero, et al. (2016), Guerrero et al. (2021)
	Clonal Selection Algorithm (CLONALG)	Guerrero et al. (2021)
	Genetic algorithm	Nguyen et al. (2019a), Nguyen et al. (2019b)
	Maximum Power Point Tracking (MPPT)	Singh and Shubhanga (2017)
	Numerical optimization	Frotscher et al. (2019)
	Alternating Direction Method of Multipliers (ADMM) for local microgrid optimization	Bonassi (2020)
	Mixed-integer Linear Programming (MILP)	Bonassi (2020)
Emulation	IEEE-34 bus test feeder	Guerrero et al. (2016), Guerrero et al. (2021)
Dispatch	Cyclic battery charging	Osadcuks and Galins (2012)
	Distributed economic dispatch and reserve provision	Bonassi (2020)
	Load following	Osadcuks and Galins (2012)
Forecasting	Short-term load forecasting	Guerrero, et al. (2016), Tuominen et al. (2017), Frotscher et al. (2019), Bonassi (2020), Guerrero et al. (2021)
	System state estimation	Guerrero, et al. (2016), Tuominen et al. (2017)
	System state forecast	Tuominen et al. (2017)
	Energy price forecast	Bonassi (2020)
	Production forecast	Tuominen et al. (2017)
	Forecasts based on artificial neural networks	Frotscher, et al. (2019)
Real-time prices	Energy price input signals	Bonassi (2020)
Control	Coordinated Volt/Var Control	Guerrero et al. (2016), Guerrero et al. (2021)
	Proportional-Integral-Derivative (PID) control	Nguyen et al. (2019a)
	Model-predictive control	Rossi et al. (2019)
	Real-time power control algorithm	Tuominen et al. (2017)
Analysis	Prony analysis (Prony's method)	Nguyen et al. (2019a, (b)

(CLONALG) (Guerrero et al. 2021) have been studied. AISO was applied in a coordinated Volt/Var control strategy (Guerrero, et al. 2016, 2021). In Nguyen et al. (2019b) a GA was used for parameter optimization for oscillatory stability issues and parameter optimization of an automatic voltage regulator (Nguyen et al. 2019a). In photovoltaic applications, Maximum Power Point Tracking (MPPT) is used to optimize the power output (Singh and Shubhanga 2017). Numerical optimization has been used to optimize the operation of a system consisting of renewable energies (Frotscher et al. 2019). The authors in Bonassi (2020) applied the Alternating Direction Method of Multipliers (ADMM) for local microgrid optimization.

- Emulation approaches

Emulation was used as part of a simulation loop to emulate an IEEE-34 bus test feeder in two studies by the same authors (Guerrero et al. 2016, 2021).

- Dispatching approaches

Dispatching approaches were used to dispatch and schedule the operation of renewable energy sources in two studies (Osadcuks and Galins 2012; Bonassi 2020). Dispatching has a notion of optimization since the objective is to find optimal dispatch strategies to operate a facility.

- Forecasting approaches

Forecasting is a prominent approach in the literature. Forecasting has been used to estimate (1) short-term loads (Guerrero, et al. 2016, 2021; Tuominen et al. 2017; Frotscher, et al. 2019; Bonassi 2020), (2) system state (Guerrero et al. 2016; Tuominen et al. 2017), (3) future energy prices (Tuominen et al. 2017), and production (Tuominen et al. 2017). The authors in Frotscher et al. (2019) used artificial neural networks for weather-, electricity price-, and load forecasting. Other studies do not provide an exact forecasting method, but state to have used real-world measurements.

- Real-time price approaches

Bonassi 2020) uses real-time day-ahead energy price signals as part of its SIL approach to improving the overall control strategy.

- Control methods

The control category contains methods and approaches from the perspective of the actuation. A limited number of studies conduct control. The authors of Guerrero, et al. (2016), Guerrero et al. (2021) apply Volt/VAR control for feeder automation. The specific functions are (1) feeder voltage control, (2) feeder reactive power control, and (3) substation voltage control. The authors of Nguyen et al. (2019a) apply Proportional-Integral-Derivative (PID) control for automatic voltage regulation and they optimize the three PID coefficients with a GA. The authors of Rossi et al. (2019) apply a two-level hierarchical control. The first level covers a 15-min horizon for plant configuration i.e., the control variables, and the second level governs the power plant with real-time resolution using model-predictive control (MPC). In Tuominen et al. (2017), the authors describe the architecture for three-level hierarchical control where each level operates at different time scales. The first level operates in seconds and is responsible for controlling locally connected devices e.g., voltage control or power control of a DER. The second level operates in minutes and is responsible for controlling larger sections of the distribution network. The third level operates for tens of minutes and is responsible for coordinating secondary controllers and incorporating financial aspects into the network control.

- Analysis methods

Prony's method has been used in two studies by the same author (Nguyen et al. 2019a, b) to analyze oscillation and to find frequency and damping ratios.

Systems under test, virtual models, and performance metrics of software-in-the-loop in energy- and electricity systems

Table 10 shows the SUT, the virtual model, and metrics used for SIL in energy- and electricity systems. The SUT is the system being evaluated by enacting the virtual model.

Table 10 System under test, virtual plant, and metrics

References	System under test (SUT)							Plant (virtual model)								
	Coordinated Volt/VAr Control	**Power oscillation damping controller**	**Solar controller**	**Control software**	**Control algorithm**	**Maximum power point tracker (MPPT)**	**Day-ahead scheduling algorithm**	**Distribution network**	**Low-voltage network**	**Medium-voltage network**	**Power grid**	**Model of solar cooling**	**Model of photovoltaic system**	**Models of renewable energy systems**	**Power plant**	**Turbine generator plant**
Guerrero et al. (2016)	x							x								
Guerrero et al. (2021)	x							x								
Tuominen et al. (2017)	x								x	x						
Nguyen et al. (2019b)		x									x					
Nguyen et al. (2019a)		x									x					
Huber et al. (2014)			x									x				
Rossi et al. (2019)				x											x	
Kwon and Choi (1999)					x											x
Osadcuks and Galins (2012)					x									x		
Singh and Shubhanga (2017)						x							x			
Bonassi (2020)							x									

Table 10 (continued)

References	Metrics													
	Active Power	**Reactive power**	**Voltage**	**Power output**	**Ampere**	**System frequency**	**Rate of change of frequency (ROCOF)**	**Algorithmic performance**	**Solar energy**	**Cooling energy**	**Energy demand**	**Coefficient of Performance (COP)**	**State of Charge (SoC)**	**P–V curves**
Guerrero et al. (2016)	x	x	x											
Guerrero et al. (2021)			x					x						
Tuominen et al. (2017)	x	x	x											
Nguyen et al. (2019b)	x	x	x			x								
Nguyen et al. (2019a)	x	x	x			x	x							
Huber et al. (2014)									x	x	x	x		
Rossi et al. (2019)				x										
Kwon and Choi (1999)			x	x										
Osadcuks and Galins (2012)				x									x	
Singh and Shubhanga (2017)				x										x
Bonassi (2020)			x	x	x									

The model may vary in scale and represent e.g., a distribution network, a power plant, or a PV system. The metrics are either the direct output of the virtual model or performance indicators of the SUT. For electricity systems, the most common metrics include active power, reactive power, voltage, and power output. For power grids, the performance metric is system frequency. For PV systems, the performance metrics include solar energy, cooling energy, COP factor, and P–V curves. For battery control systems, the performance metric is the State of Charge (SoC).

Software-in-the-loop technologies used in energy- and electricity systems

This section presents a detailed analysis of the implementing technologies used for SIL in energy- and electricity systems. The results are shown in Table 11 and are organized into categories. Each category is described in detail.

- Simulation technology

The most used technology is the Real-time Digital Simulator (RTDS) (Nguyen et al. 2019a, b; Guerrero, et al. 2016, 2021) which is a specialized simulator for power grids. The authors of Guerrero et al. (2016, 2021) use RTDS to simulate an IEEE 34-bus distribution network. The network has three phases, 23 busses, 19 distribution lines, and 19 loads. The authors in Nguyen et al. (2019b) use RTDS to simulate microgrids and renewable energy sources. The authors in Nguyen et al. (2019a) simulate the future Western Danish power system in RTDS. RSCAD is a platform (or ecosystem) that

Table 11 SIL technologies in energy- and electricity systems

Category	Technology	References
Simulation technology	Real-time digital simulator (RTDS)	Nguyen et al. (2019a), Guerrero et al. (2016), Tuominen et al. (2017), Nguyen et al. (2019b), Guerrero et al. (2021)
	HOMER	Osadcuks and Galins (2012)
	FPGA XCS100E	Singh and Shubhanga (2017)
	TRNSYS	Frotscher et al. (2019)
Environment software	MATLAB/Simulink	Guerrero, et al. (2016), Rossi et al. (2019), Tuominen et al. (2017), Nguyen et al. (2019b), Osadcuks and Galins (2012), Bonassi (2020), Guerrero et al. (2021)
	Modelica/Dymolink	Huber et al. (2014)
	RSCAD/Draft	Nguyen et al. (2019a), Nguyen et al. (2019b)
	RSCAD/Runtime	Nguyen et al. (2019a), Guerrero et al. (2016), Guerrero et al. (2021)
	Siemens AMESIM	Rossi et al. (2019)
Interoperability protocols	MaktrikonOPC	Nguyen et al. (2019a), Nguyen et al. (2019b)
	RSCAD/MATLAB	Guerrero et al. (2016)
Optimization tools	ABB Power Grids e-mesh™ EMS	Bonassi (2020)
Programming language/framework	Python	Tuominen et al. (2017)
	C#	Osadcuks and Galins (2012)
	C	Huber et al. (2014)
	.NET framework	Osadcuks and Galins (2012)
Metering hardware	Smart meters	Tuominen et al. (2017)

interfaces with RTDS. For example, RSCAD/Draft is used for circuit assembly and parameter entry (Nguyen et al. 2019a, b), and RSCAD/Runtime is used to control simulation scenarios performed on the RTDS hardware (Nguyen et al. 2019a; Guerrero, et al. 2016, 2021). HOMER has been used to provide load profiles and available resources in hybrid power systems (Osadcuks and Galins 2012). In Singh and Shubhanga (2017), the authors used an FPGA XCS100E for real-time simulation where they used a hardware description language (VERILOG) to realize a photovoltaic module. In Frotscher et al. (2019), TRNSYS has been used as simulation software to create a complex reference model (CRM) of a district heating system, including solar thermal plants and short-term thermal energy storage.

- Integrated environment software technology

Integrated environments are software that provides a substantial and integrated toolbox to develop plant models, simulations, controllers, and graphical user interfaces and to execute test scenarios. MATLAB/Simulink was used the most (Guerrero et al. 2016, 2021; Rossi et al. 2019; Tuominen et al. 2017; Nguyen et al. 2019b; Osadcuks and Galins 2012; Bonassi 2020). In Guerrero et al. (2016, 2021), the short-term load forecast, system state estimation, and coordinated volt/var control were implemented in MATLAB. In Nguyen et al. (2019b), MATLAB was used for both optimization and control of an RTDS in a closed loop. In Rossi et al. (2019), MATLAB was used to implement an MPC controller that controlled a model developed in Siemens AMESIM. In Osadcuks and Galins (2012), MATLAB was used to design a simulation model library of hybrid power system equipment. In Bonassi (2020), MATLAB was used to implement microgrid models and optimizations. Modelica/Dymolink was used by Huber et al. (2014) to implement the model of a solar cooling system.

- Interoperability protocols

MaktrikonOPC is an implementation of Open Platform Communication (OPC) which was used for data collection (Nguyen et al. 2019a, b). In Guerrero et al. (2016), the authors used a plugin for the RSCAD platform, to establish communication between RTDS and MATLAB. Interoperability has also been established through sockets (TCP/IP). These instances are elaborated on in the architecture and communication pattern section.

- Optimization tools

MATLAB and Siemens AMESIM have been used as optimization technologies in Nguyen et al. (2019a), Guerrero et al. (2016), Nguyen et al. (2019b), Bonassi (2020), Guerrero et al. (2021) and Rossi et al. (2019), respectively. Other studies that use optimization are not explicit about the technology (Singh and Shubhanga 2017; Frotscher et al. 2019).

- Programming languages/frameworks

The use of programming languages was sparse and for different purposes. Python has been used for load- and production forecasts (Tuominen et al. 2017). C# and.NET have been used to implement the controller and user interface (UI) for the dispatch of resources. C has been used to implement an adapter between a controller and Modelica (Huber et al. 2014). The number of references that use conventional programming languages to implement the controller is unexpected when considering the purpose of SIL. Namely, to test or verify control software written in a conventional language against a virtual plant. It should be noted that MATLAB can generate code based on the controller models. The analyzed studies do not report the use of this feature.

- Metering hardware technology

A single study uses smart meters for power measurements including load and reactive power (Tuominen et al. 2017).

Software-in-the-loop architectures and communication patterns used in energy- and electricity systems

Various SIL architectures have been applied in energy- and electricity systems. Architecture refers to a system consisting of a set of structures, the relations among them, and the corresponding properties of both. This section first describes the structures and relations and then describes the architectural properties.

Structures and relations

The structure and relations are analyzed in this section. Generally, the controller is separated from the virtual model and interconnected through networking.

Guerrero et al. (2016, 2021) use a structure as depicted in Fig. 4. The physical devices consist of a developer machine that hosts the controller and SCADA system and a real-time digital simulator that hosts the virtual model of a distribution grid. The subsystems are interconnected by TCP/IP for data collection and control.

Nguyen et al. (2019a, b) use a structure as depicted in Fig. 5. This architecture contains three physical devices. The developer machine hosts the controller in MATLAB. The real-time digital simulator hosts the virtual model of a distribution grid. The middleware server hosts an OPC broker for flexible interoperability. Data from the virtual model are collected by the OPC. The controller is connected to the OPC to get the data from the

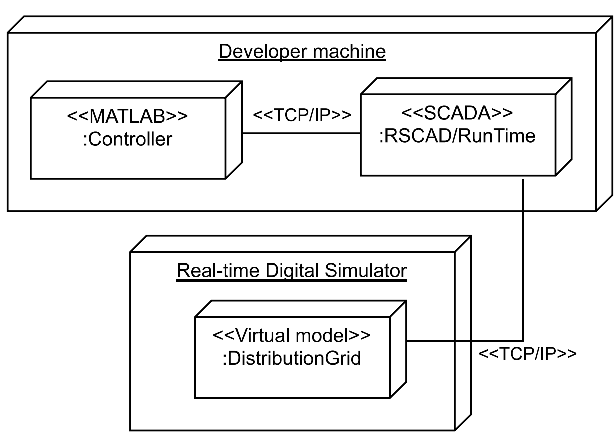

Fig. 4 SCADA architecture used in Guerrero et al. (2016, 2021)

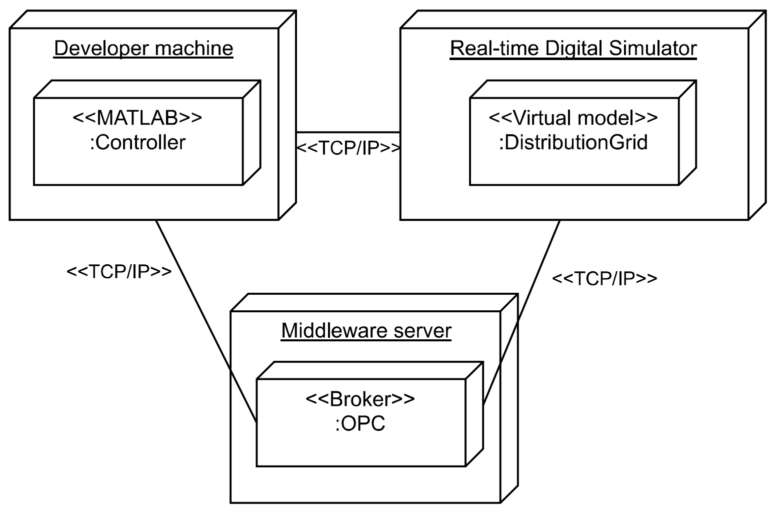

Fig. 5 Broker architecture used in Nguyen et al. (2019a, b)

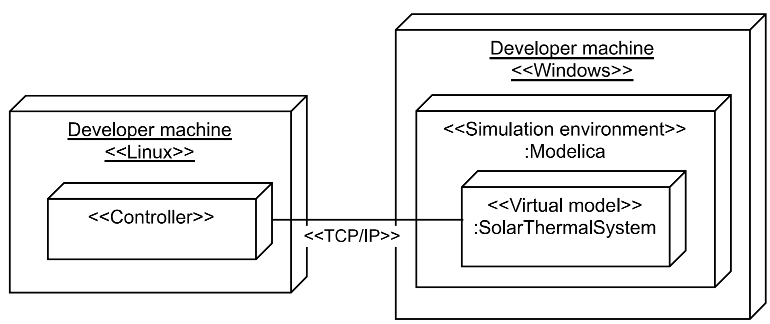

Fig. 6 Peer-to-peer architecture used in Huber et al. (2014)

virtual model. The connection between the controller and the virtual model is used for control.

Huber et al. 2014) uses a structure as depicted in Fig. 6. The controller is connected via TCP/IP to the Modelica simulation environment which contains the virtual model of a solar thermal system. This architecture uses peer-to-peer communication and resembles the Process Control paradigm closely.

Kwon and Choi (1999) uses a master–slave architecture as depicted in Fig. 7 where a coordinator controls a distributed SIL of multiple controllers and simulators. Each controller controls separate simulators. The elements are connected through UDP/IP to improve performance. This architecture has similarities with co-simulation which is a distributed simulation of subsystems.

Tuominen et al. (2017) uses a hybrid architecture as depicted in Fig. 8 of SIL and HIL where physical smart meters and automatic voltage controllers/regulators are part of the loop. The substation automation unit is a form of supervisor that exists on two levels: Primary and secondary depending on the control level. The depicted architecture is a simplification of the two levels that encapsulates the abstract idea. The virtual models of low- and medium-voltage networks run on the real-time digital simulator. MATLAB is used both as a controller and a measurement collector. The smart meter is connected

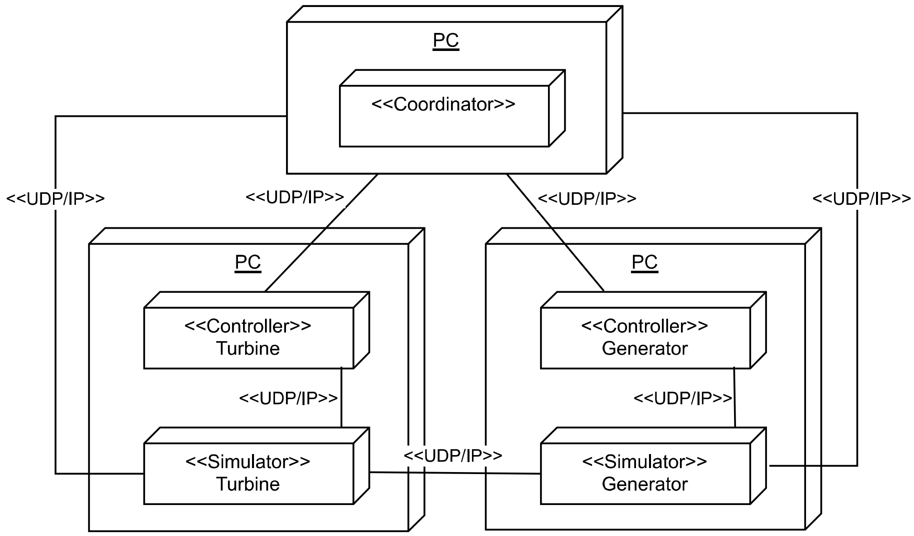

Fig. 7 The master–slave architecture is used in Kwon and Choi (1999)

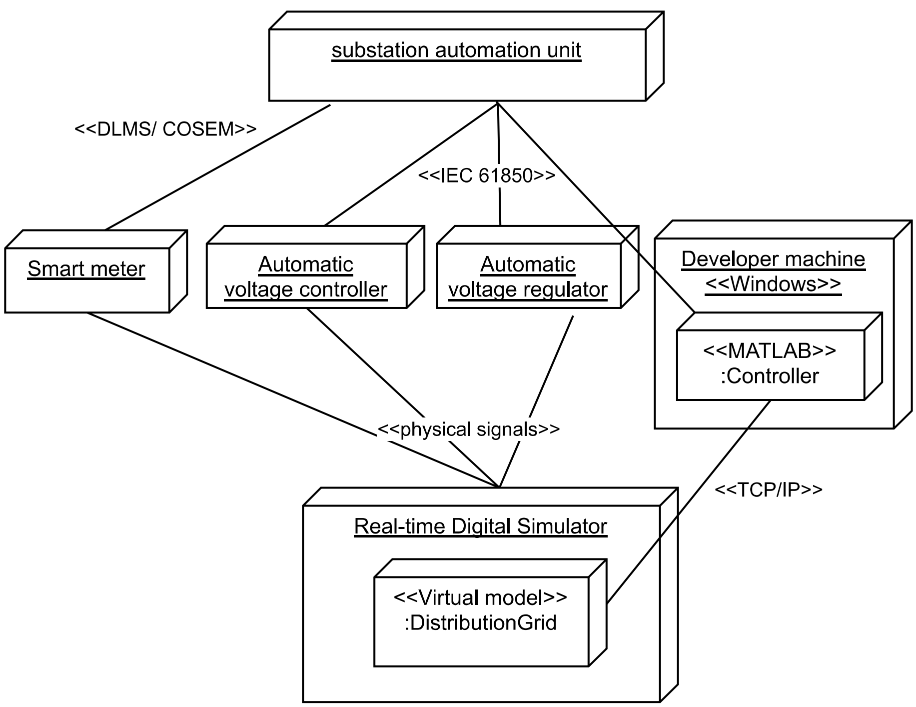

Fig. 8 The hybrid architecture used in Tuominen et al. (2017)

through the DLMS/COSEM protocol. The automatic voltage controller and regulator are connected through the IEC 61850 protocol.

Furthermore Rossi et al. (2019) uses an architecture in which the virtual power plant model is developed and executed in the Siemens AMESIM environment, and the controller is developed in MATLAB. The deployment configuration is not further specified. Osadcuks and Galins (2012) uses MATLAB to model the virtual model, its sensors, and actuators. The controller is running in a separate program written in C#.NET. The simulation runs in discrete time steps in contrast to real-time. This boosts the simulation speed on average 150:1. Singh and Shubhanga (2017) uses a Field-Programmable Gate Array (FPGA) board to conduct real-time simulation and evaluation of an MPPT algorithm for photovoltaics. Thus, the architecture includes an element of specialized hardware. The output of the simulator can be used to control DC-DC converters. The authors separate the photovoltaic module (virtual model) and the MPPT module (controller) to enable the testing of various MPPT algorithms with few modifications. Both modules are deployed in a single FPGA board and communicate via electric signaling i.e., voltage and current.

Moreover, Bonassi (2020) uses a proprietary platform e-mesh™ to optimize the dispatching of DER in microgrids. Their simulation and control architecture is sparsely described. They use MODBUS over TCP/IP to communicate with field controllers and HTTP/REST to communicate with a SCADA system. Frotscher et al. (2019) uses TRNSYS to simulate a virtual district heating system in discrete time. The overall architecture is a closed loop where a separate controller optimizes the district heating system. The communication between the simulator and controller is established through files that are exchanged in a shared folder.

Architectural properties

The architectural properties are presented in Table 12. Most studies have similar properties i.e., closed loop, real-time simulation, and distributed architecture. A closed loop is a common type of control loop where process variables are fed back into the control algorithm as a point of reference. The control algorithm then decides the values of the manipulated variables of the controlled process. The real-time property is evident in the cases where the virtual model and controller are executed at the same rate as "wall clock" time. Consequently, if a two-week scenario is tested in a real-time simulator, it takes two weeks to obtain the results. Each application operates at application-specific time scales. Thus, a real-time application may operate within short intervals (micro-seconds or seconds) e.g., grid-stabilization, or long intervals (e.g., minutely, hourly, daily, yearly, decades) for long-term decision-making. In

Table 12 Architectural properties of SIL in energy- and electricity systems

Architectural property	References	Frequency
Closed loop	Nguyen et al. (2019a), Guerrero et al. (2016), Huber et al. (2014), Singh and Shubhanga (2017), Tuominen et al. (2017), Nguyen et al. (2019b), Frotscher, et al. (2019), Osadcuks and Galins (2012), Kwon and Choi (1999), Guerrero et al. (2021)	10 (76.9%)
Real-time simulation	Nguyen et al. (2019a), Guerrero et al. (2016), Huber et al. (2014), Rossi et al. (2019), Singh and Shubhanga (2017), Tuominen et al. (2017), Nguyen et al. (2019b), Bonassi (2020), Kwon and Choi (1999), Guerrero et al. (2021)	10 (76.9%)
Distributed architecture	Nguyen et al. (2019a), Guerrero et al. (2016) Huber et al. (2014), Rossi et al. (2019) Tuominen et al. (2017), Nguyen et al. (2019b), Bonassi (2020), Kwon and Choi (1999), Pieper and Obermaisser (2018a), Guerrero et al. (2021)	10 (76.9%)
Discrete-time simulation	Frotscher et al. (2019), Osadcuks and Galins (2012)	2 (15.4%)
Hybrid SIL/HIL	Tuominen et al. (2017)	1 (7.7%)

contrast, two studies (Frotscher, et al. 2019; Osadcuks and Galins 2012) use discrete-time simulation to progress faster than "wall clock" time. A distributed architecture is commonly used to separate the controller and virtual model over a network. These two components may be deployed to physical nodes or executed in separate processes on the same developer machine. A single study (Tuominen et al. 2017) combines SIL and HIL to study interfacing issues before field demonstration.

Interoperability in the context, of the OSI model

The analyzed studies use a variety of protocols for realizing interoperability. The interconnected subsystems are usually the controller, virtual model, middleware, and physical devices. Table 13 lists the applied protocols. TCP/IP is a connection-oriented used for reliable socket communication (Nguyen et al. 2019a, b; Guerrero, et al. 2016, 2021; Tuominen et al. 2017). On the contrary Kwon and Choi (1999) uses the connectionless UDP/IP for performance reasons. These protocols operate at the transport layer of the OSI model. Similarly, Ethernet is also used, but this protocol operates at the data link layer. The DNP3 protocol is used within utilities e.g., power grids to control physical equipment over a network. Nguyen et al. (2019a, b) states its use between the real-time digital simulator and local equipment. MODBUS is a competing protocol for DNP3 and is used in two studies (Tuominen et al. 2017; Bonassi 2020). OPC is the middleware used for secure interoperability and uniform access to industrial automation equipment. Nguyen et al. (2019a, b) uses OPC as a middleware technology to interconnect the controller and real-time digital simulator. One study (Tuominen et al. 2017) uses application standards i.e., IEC 62056–5-3:2017 for electricity metering data exchange, and IEC 61850 which is a communication protocol for intelligent electronic devices. Two studies state that they transfer files over the network to exchange data (Huber et al. 2014; Frotscher et al. 2019).

Table 13 Protocols for interoperability and their mapping to the ISO model

Protocol	Protocol layer (OSI model layer)	References
TCP/IP socket communication	Transport (4)	Nguyen et al. (2019a), Guerrero, et al. (2016), Tuominen et al. (2017), Nguyen et al. (2019b), Guerrero et al. (2021)
Ethernet	Data link (2)	Huber et al. (2014), Kwon and Choi (1999)
File-based data exchange	Application (7)	Huber et al. (2014), Frotscher et al. (2019)
Distributed network protocol (DNP3)	Application, (7) Transport (4), Data link (2), Physical (1)	Nguyen et al. (2019a, b)
Open Communications Platform (OPC)	Application (7)	Nguyen et al. (2019a, b)
MODBUS	Application (7)	Bonassi 2020)
UDP/IP socket communication	Transport (4)	Kwon and Choi (1999)
HTTP/REST	Application (7)	Bonassi (2020)
IEC 61850/Modbus RTU protocol	Application (7)	Tuominen et al. (2017)
IEC 62056–5-3:2017 DLMS/COSEM	Application (7)	Tuominen et al. (2017)

Discussion

The digitalized energy system of the future operates with heterogenous, interconnected, and concurrent entities on a large scale. Extending the energy system capabilities with advanced applications, e.g., DR, and DER, requires that entities can interoperate to achieve a greater goal. The local control systems must be capable of responding to external events to operate optimally. It is no longer sufficient to operate in siloes. These capabilities call for advancement in V&V, ontology, and interoperability development for testing the future energy system. Hardware-based testing approaches such as PHIL require specialized equipment that is difficult to scale. Most of the analyzed literature rely on hardware-based plants for in-the-loop testing of energy- and electricity systems. Relying on hardware testbeds in a lab environment involves upfront investment for equipment, installation, and training. The use of a hardware-based real-time simulator (e.g., RTDS) for V&V constrains the lab environment to a proprietary hardware platform.

Future potential, emerging trends, technologies, and related fields

SIL has the potential to reduce testing costs if the architectural properties required for testing (e.g., real-time, closed-loop, distributed computation, heterogenous models, protocol simulation, etc.) can be realized with software-based approaches on commodity hardware.

A virtual lab environment for SIL testing would enable earlier V&V feedback and potentially reduce hardware costs by executing the simulation on commodity hardware. It may also be possible to speed up the simulation execution on commodity hardware beyond real-time as shown in Osadcuks and Galins (2012).

Another potential of SIL testing is experiment reproducibility and benchmarking. SIL testing enables to rerun hypothetical scenarios and configurations to analyze edge cases that may be difficult to test in the real-world. Reproducibility can also be used as an enabler for benchmarking testing. For example, many SIL applications involve optimization where the objective is to find an optimal control strategy for a particular system. If the same simulation environment can be used to test various control strategies, it is possible to make benchmarking analysis.

The emerging field of digital twins enable high-fidelity modeling of energy equipment to be used in closed-loop testing. This area of research is called twin-in-the-loop (Park et al. 2019; Dettù et al. 2023; Eyring et al. 2022) and exhibits promising prospects such as (i) improved integrability testing of renewable energy source and electric vehicles, and (ii) improved reliability and resilience testing of the energy system. An example could be in testing and comparing electric vehicle charging strategies on a simulated model of the infrastructure. Here the 'software' part of 'software-in-the-loop' would constitute a component that accurately resembles software that can be deployed in the real-world. The 'in-the-loop' part would resemble a system-level digital twin simulation of the infrastructure and would provide responses of enacted control. A concrete case could be testing real control software using the Open Charge

Point Protocol (OCPP) 2.0.1 as a basis of protocol and topology. The control software in this case would be deployed as a charging station management system and a digital twin would simulate the infrastructure with individual charging stations and other distribution grid entities.

Game engines commonly serve as a technology for creating digital twins (Clausen et al. 2022; Sørensen et al. 2022; Negrin et al. 2021; Eyre et al. 2018). For example, Universal Scene Description (USD) can be used to create simulation-ready assets and to create ecosystems to describe, compose, simulate, and collaborate within 3D worlds. Imagine if vendors provide simulation-ready assets that can be seamlessly integrated into a simulation to evaluate scenarios. Another game engine example is the object model used in Unreal Engine 5. It separates the actor from the behavior like the process control paradigm. A major limitation of game engines in the perspective of energy- and electricity systems is that game engine technology focus on real-time body simulation (collision, fluids, destruction, etc.) in contrast to electromagnetism, realistic wind patterns, and information flow analysis. Nevertheless, the exploration of game engine technology for in-the-loop testing of energy- and electricity systems is an open issue.

Agent-based modeling and simulation have also been proposed to model digital twins (Schimeczek et al. 2023a, b). Each agent reflects a certain entity in the energy- and electricity system, encapsulating its control logic. Furthermore, the agent-based approach facilitates the examination of higher-level economics and interactions among roles, actors, and objects in different ecosystem configurations (Ma et al. 2019; Fatras et al. 2022; Værbak et al. 2021). In the context of SIL, agent-based modeling can be used to model the individual plants in one simulation and then have externalizes control interacting with each agent in the model. This approach may be useful for SIL testing on a system-level scale.

Energy metaverses (Ma 2023) can enable analysis of the emerging behavior between stakeholders, infrastructure, environment, business models, regulations, and policies. The energy metaverse connects tangible and intangible assets in the energy system through digital twins. In this case, it would be interesting to assess the emergent behavior of SIL in a grand perspective.

Cooperative simulation (co-simulation) is an approach that enables global simulation of a coupled system through composition of independent simulators (Gomes et al. 2017). Each simulation may differ internally in terms of solving and model paradigm (discrete, state machine, ordinary differential equations, etc.), but they agree upon a well-defined interface to operate in concert through an orchestrator. The orchestrator is responsible for data exchange and synchronization between the models (Steinbrink et al. 2017). This approach is modular, and offers flexibility, reusability, parallel development, and protection of intellectual property (model internals are hidden). The result of the co-simulation is the effect of emergent behavior. Co-simulation and SIL could be combined to analyze real-time properties of a system under test.

Challenges of SIL

SIL testing is no silver-bullet and there are limitations and challenges towards this approach that are discussed below.

- Development process. The applied SIL development process is not systematically reported in the literature. The focus is devoted to functional- and behavioral aspects, simulation modeling, optimal control modeling, results, and structure (components, and their mutual relation). Thus, it remains a challenge to adapt SIL based on a set of guiding principles.
- Real-world operation aspects. SIL is insufficient in testing the physical aspects of a system. It is challenging to accurately model all aspects of real-world operating conditions and hardware interactions. PIL, PHIL, and HIL testing must still be used for testing physical aspects that belong in that domain i.e. low-level signaling concerns, physical connections, physical component failure, processor capabilities, etc. Furthermore, critical conditions of security and safety may not be appropriate to test using SIL and it is still necessary to test in the real world.
- Simulation model validity and fidelity. SIL relies on a simulation model of the environment. The experiments conducted on these models are therefore dependent on (i) the validity of the model, i.e. to which degree and under which circumstances do the simulation model represent the real-world. (ii) The fidelity of the simulation model which is the degree/accuracy to which the simulation model represents its real-world counterpart. Fidelity involves a tradeoff between accuracy and computation time. A higher accuracy demands higher computational resources. Sacrificing accuracy yields faster computation, but the simulation must still provide usable results. Determining validity and fidelity of the SIL approach is a challenge.
- Interoperability aspects. The software under test and the simulation model must be interoperable which requires well-defined interfaces, protocol and data exchange formats starting already in the early stages of development. These issues are still fundamental to solve in SIL.
- Real-time and concurrency aspects. Separating the control logic from the simulation model into separate components implies that these execute concurrently and possibly in real-time. This fosters challenges of synchronization, timeliness, traceability, and other properties of real-time systems.
- Scaling. Scaling is not magically solved through SIL testing. However, scaled test scenarios may be easier to conduct because the controller and simulation model may be cheaper and easier to instantiate on scale.
- Controller emulator acquisition. A premise for SIL testing is that the actual control software is tested. Therefore, it is implied that the software must be able to execute on a development platform. If the control software is written in Java a Java Virtual Machine (JVM) must be available. If the control software is written in C/C++ for embedded devices, then an embedded emulator must be available.
- Cost/benefit analysis. It is a challenge to decide whether SIL testing is worth pursuing based on cost and potential benefits. Even estimating the costs for SIL test-

ing alone is challenging. Costs are challenging because it depends on the domain, scale, and analytic metrics to provide. The benefits also vary and are therefore hard to quantify. However, in many of the domains such as the energy sector, faults cannot be tolerated during experimental endeavors on a large scale. Such experiments jeopardize the business, well-being or can cause serious harm. But experimentation is needed to evaluate innovative solutions and scenarios prior to real-world deployment. These benefits are purely functional (i.e., testing the controller against the system), but there must also be added benefits from the project management perspective e.g., rapid development, rapid feedback cycles, reduced risks, and overall costs. Claims on this matter can only be validated through rigid project monitoring and evidence analysis.

Study validity threats

This section accounts for validity threats within this scoping review. The identified threats are enumerated and addressed below.

- Evidence collection. The evidence collection does not include grey literature in the extent of doctorial theses, patents, or standards. Furthermore, white literature such as industry reports, non-academic publications, technical blogs, software manuals, are not included. Therefore, this study does not explore literature beyond white literature. One such example of a white paper is found in Reyes (2024).
- Selection bias. The literature search is subject to selection bias to the extent that the inclusion criteria may be narrow in the domain dimension of search string G1. This search string may not uncover the entire energy domain with regards to all in-the-loop paradigms. However, G1 contains 46 hits and has a reasonable size for evidence for a scoping review study. This problem is not inherent when the focus is narrowed to software-in-the-loop across domains, because search string G3 is domain agnostic.
- Language bias. Depending on the field (domain), scholars may use different terms for addressing in-the-loop semantics. (i) Papers may address in-the-loop in the main body text but does not entitle to address it in the title. Examples of this can be found in Čech et al. (2017) and (Steinbrink, et al. 2017). (ii) Papers may use abbreviations such as MIL, SIL, PIL and HIL directly in the title. Such examples can be found in the papers (Nibert et al. 2012).

Conclusion

The study applies a scoping review methodology to find, screen, analyze and synthesize the literature, including conference and journal papers from the significant engineering databases ACM, IEEE Xplore, Scopus, and Web of Science. A thorough literature processing is conducted including screening for relevance based on inclusion criteria. The included 88 articles were full-text analyzed and categorized to map the purpose, methods, architecture, interoperability and protocols, technologies, challenges, and limitations.

The analysis result shows that hardware-based in-the-loop paradigms are commonly applied for testing low-level signaling issues. Power-hardware-in-the-loop is most frequently used to test control applications in the energy sector. Power-hardware-in-the-loop focuses on testing the physical phenomenon at the hardware level. With the transition towards a digitalized energy system and increasingly advanced interconnected applications, hardware-based in-the-loop testing approaches fall short. Software-defined in-the-loop testing received increased attention over the past decade but is less frequently applied in the energy sector.

There is a gap in the literature on how to systematically develop software-based in-the-loop lab environments to verify and validate new control strategies, technologies, regulations, and policies in energy- and electricity systems. This includes functional and non-functional requirements, conceptual design, architecture, interfaces, and compliance with interoperability standards. There was found no previous work that address these issues.

Based on the scoping review, there is limited research on:

- Modeling of energy- and electricity systems using software-in-the-loop approaches, game engine technology, agent-based simulation, and co-simulation.
- Applying software-in-the-loop for energy- and electricity systems to test emergent behavior in an ecosystem context in applications such as EV charging, Power-to-X (PtX), demand-response, DER, and renewable energy.
- Software-in-the-loop testing at scale in environments that are highly heterogeneous and geographically distributed. This includes the problem of achieving interoperability across sectors.
- Using high-fidelity models, for example, digital twins for twin-in-the-loop testing.
- Reproducibility and benchmarking of SIL testing.

Therefore, the following future research directions are proposed:

- Development of a conceptual software-in-the-loop testing framework with reference architectures, guiding design principles, use cases, non-functional properties in energy- and electricity systems.
- Explore and apply game engine technology, digital twins, agent-based simulation, and co-simulation to construct an energy metaverse that can be used for software-in-the-loop testing.
- Identify interoperability issues and propose solutions that can be used for sector coupling in the energy ecosystem.

Appendices

Appendix A: Database-specific syntax

The generic search string was mapped into database specific syntax as shown in Table 14.

Table 14 Database specific syntax

ID	Database	Database-specific syntax
G1	Web of Science (WoS)	TI = ((electricity OR "distribution grid" OR "demand side" OR "consumption side" OR "distributed energy resources" OR DER OR solar OR photovoltaic* OR PV OR "electric vehicle" OR EV OR "heat pump") AND (architect* OR pattern OR communicat* OR connect* OR protocol* OR interop* OR interconnect* OR exchange OR synchroniz* OR distribut* OR "digital environment" OR "virtual environment" OR emulator) AND (in-the-loop OR "model-based systems" OR "model-based software" OR "model-based design" OR MBD OR MBS*))
G1	Scopus	TITLE ((electricity OR "distribution grid" OR "demand side" OR "consumption side" OR "distributed energy resources" OR DER OR solar OR photovoltaic* OR PV OR "electric vehicle" OR EV or "heat pump") AND (architect* OR pattern* OR communicat* OR connect* OR protocol* OR interop* OR interconnect* OR exchange OR feedback* OR synchroniz* OR distribut* OR "digital environment" OR "virtual environment" OR emulator) AND (in-the-loop OR "model-based systems" OR "model-based software" OR "model-based design" OR MBD OR MBS*))
G1	ACM	((Title: electricity) OR (Title: "distribution grid") OR (Title: "demand side") OR (Title: "consumption side") OR (Title: "distributed energy resources") OR (Title: der) OR (Title: solar) OR (Title: photovoltaic*) OR (Title: pv) OR (Title: "electric vehicle") OR (Title: ev) OR (Title: "heat pump")) AND ((Title: architect*) OR (Title: pattern*) OR (Title: communicat*) OR (Title: protocol*) OR (Title: interop*) OR (Title: interconnect*) OR (Title: exchange) OR (Title: feedback*) OR (Title: synchroniz*) OR (Title: distribut*) OR (Title: "digital environment") OR (Title: "virtual environment") OR (Title: emulator)) AND ((Title: in-the-loop) OR (Title: "model-based system") OR (Title: "model-based software") OR (Title: "model-based design") OR (Title: mbd) OR (Title: mbs*))
G1	IEEE Xplore	(("Document Title": electricity) OR ("Document Title": "distribution grid") OR ("Document Title": "demand side") OR ("Document Title": "consumption side") OR ("Document Title": "distributed energy resources") OR ("Document Title": DER) OR ("Document Title": solar) OR ("Document Title": photovoltaic) OR ("Document Title": PV) OR ("Document Title": "electric vehicle") OR ("Document Title": EV) OR ("Document Title": "heat pump")) AND (("Document Title": architectur*) OR ("Document Title": pattern) OR ("Document Title": communicat*) OR ("Document Title": protocol) OR ("Document Title": interop*) OR ("Document Title": interconnect*) OR ("Document Title": exchange) OR ("Document Title": feedback*) OR ("Document Title": synchroniz*) OR ("Document Title": distribut*) OR ("Document Title": "digital environment") OR ("Document Title": "virtual environment") OR ("Document Title": emulator)) AND (("Document Title": in-the-loop) OR ("Document Title": "model-based systems") OR ("Document Title": "model-based software") OR ("Document Title": "model-based design") OR ("Document Title": MBD) OR ("Document Title": MBS*))
G2	Web of Science (WoS)	TI = ("software-in-the-loop") AND (TI = (architect*) OR TI = (framework) OR TI = (pattern))
G2	Scopus	TITLE(software-in-the-loop) AND (TITLE(architect*) OR TITLE(framework) OR TITLE(pattern))
G2	ACM	(Title: "software-in-the-loop") AND ((Title: architect*) OR (Title: framework) OR (Title: pattern))
G2	IEEE Xplore	("Document Title": "software-in-the-loop") AND (("Document Title": architect*) OR ("Document Title": framework) OR ("Document Title": pattern))
G3	Web of Science (WoS)	TI = ("software-in-the-loop")
G3	Scopus	TITLE(software-in-the-loop)
G3	ACM	(Title: "software-in-the-loop")
G3	IEEE Xplore	("Document Title": "software-in-the-loop")
G4	Web of Science (WoS)	TI = ("software-in-the-loop") AND (TI = (architect*) OR TI = (framework) OR TI = (platform) OR TI = (pattern) OR TI = (interop*) OR TI = (interconnect*) OR TI = (environment) OR TI = (exchange) OR TI = (communicat*) OR TI = (protocol) OR TI = (distribut*) OR TI = (synchroniz*) OR TI = (emulator) OR TI = (simulat*))

Table 14 (continued)

ID	Database	Database-specific syntax
G4	Scopus	TITLE("software-in-the-loop") AND (TITLE(architect*) OR TITLE(framework) OR TITLE(platform) OR TITLE(pattern) OR TITLE(interop*) OR TITLE(interconnect*) OR TITLE(environment) OR TITLE(exchange) OR TITLE(communicat*) OR TITLE(protocol) OR TITLE(distribut*) OR TITLE(synchroniz*) OR TITLE(emulator) OR TITLE(simulat*))
G4	ACM	(Title: "software-in-the-loop") AND ((Title: architect*) OR (Title: framework) OR (Title: platform) OR (Title: pattern) OR (Title: interop*) OR (Title: interconnect*) OR (Title: environment) OR (Title: exchange) OR (Title: communicat*) OR (Title: protocol) OR (Title: distribut*) OR (Title: synchroniz*) OR (Title: emulator) OR (Title: simulat*))
G4	IEEE Xplore	("Document Title": "software-in-the-loop") AND (("Document Title": architect*) OR ("Document Title": framework) OR ("Document Title": platform) OR ("Document Title": pattern) OR ("Document Title": interop*) OR ("Document Title": interconnect*) OR ("Document Title": environment) OR ("Document Title": exchange) OR ("Document Title": communicat*) OR ("Document Title": protocol) OR ("Document Title": distribut*) OR ("Document Title": synchroniz*) OR ("Document Title": emulator) OR ("Document Title": simulat*))

Appendix B: Literature content types

The search results content type is shown in Table 15. The 'Other' column covers magazines, book chapters, and reviews. One of the articles from G1 Web of Science is also marked early access and consequently it counts double in the total.

Table 15 Search results content type

Search ID	Database	Conference proceeding	Journal article	Short paper	Early Access	Other
G1	Web of Science	16	9	–	1	–
	Scopus	27	16	–	–	–
	ACM	–	–	1	–	–
	IEEE Xplore	16	3	–	1	1
	Total	59	28	1	2	1
G2	Web of Science	4	2	–	–	–
	Scopus	6	2	–	–	–
	ACM	–	–	–	–	–
	IEEE Xplore	4	–	–	1	1
	Total	14	4	–	1	1
G3	Web of Science	61	32	–	–	–
	Scopus	94	49	–	–	5
	ACM	–	2	–	–	2
	IEEE Xplore	54	4	–	1	2
	Total	209	87	–	1	9
G4	Web of Science	38	19	–	–	–
	Scopus	61	27	–	–	4
	ACM	–	2	–	–	2
	IEEE Xplore	36	2	–	1	1
	Total	135	50	–	1	7

Appendix C: Unavailable full-text references

Table 16 shows the unavailable full-text papers that were excluded from the analysis.

Table 16 Unavailable full-text references

Title	URL	Reason
Development of a toolbox in matlab for designing discrete and continuous-time linear controllers with system control application using software in the loop	https://www.praiseworthyprize.org/jsm/index.php?journal=ireaco&page=article&op=view&path[]=17832	Paywall
Towards Establishing Continuous-X Pipeline Using Modular Software-in-the-Loop Test Environments	https://www.sae.org/publications/technical-papers/content/2021-26-0412/	Paywall
Development of a vehicle model for FCHV control and functional specification development within a Software-in-the-Loop Simulation environment	https://www.sae.org/publications/technical-papers/content/2010-01-0939/	Paywall
Software-in-the-Loop Simulation environment realization using Matlab/Simulink	https://www.sae.org/publications/technical-papers/content/2006-01-1470/?src=1999-01-2783	Paywall
Software-in-the loop based end to end validation methodology for aerospace software development	http://iafastro.directory/iac/paper/id/3660/summary/	Only abstract available
Chassis control system development using simulation: Software in the loop, rapid prototyping, and hardware in the loop	https://www.sae.org/publications/technical-papers/content/2002-01-1565/	Paywall
The Facility for Aerospace Systems and Technology Simulation Toolkit: FASTKit—An Open-Source Configurable Software in the Loop Simulation Environment	https://arc.aiaa.org/doi/10.2514/6.2022-3943	Paywall
Engine ECU function development using software-in-the-loop methodology	https://www.sae.org/publications/technical-papers/content/2005-01-0049/	Paywall
Integrated software-in-the-loop simulation of an autonomously acting rescue boat	https://www.researchgate.net/publication/276899972_Integrated_Software-in-the-Loop_Simulation_of_an_Autonomously_Acting_Rescue_Boat	Not available as download
Development of a plug-in hybrid electric vehicle control strategy employing software-in-the-loop techniques	https://www.sae.org/publications/technical-papers/content/2013-01-0160/	Paywall
Desktop simulation and calibration of diesel engine ECU software using software-in-the-loop methodology	https://www.sae.org/publications/technical-papers/content/2014-01-0189/	Paywall
Software-in-the-loop development and experimental testing of a semi-active magnetorheological coupling for 4WD on demand vehicles	https://www.scopus.com/inward/record.uri?eid=2-s2.0-79959823119&partnerID=40&md5=103be04c9b5386d2132a0a545f34c01e	Not available as download
Real-time software-in-the-loop simulation for control education	https://www.researchgate.net/publication/268351829_Real-time_software-in-the-loop_simulation_for_control_education	Not available as download
Using a Co-simulation framework to enable software-in-the-loop powertrain system development	https://www.sae.org/publications/technical-papers/content/2009-01-0520/	Paywall
Evaluation of virtual controller interface device and software-in-the-loop simulation	https://www.icevirtuallibrary.com/doi/epdf/https://doi.org/10.1680/jtran.17.00149	Paywall
Development of a Software-In-The-Loop Model for a Parallel Plug-In Hybrid Electric Vehicle	https://saemobilus.sae.org/content/2016-01-1255/	Paywall
Integration of Autonomous Vehicle Frameworks for Software-in-the-Loop Testing	https://www.sae.org/publications/technical-papers/content/2020-01-0709/	Paywall

Abbreviations

ACM	The ACM Guide to Computing Literature
ADMM	Alternating Direction Method of Multipliers
AISO	Artificial Immune Systems Optimization
AVR	Automatic voltage regulator
CLONALG	Clonal Selection Algorithm
COP	Coefficient of Performance
CSIL	Controller-software-in-the-loop
DER	Distributed energy resources
DNP	Distributed network protocol
DR	Demand response
FPGA	Field-Programmable Gate Array
FPGA-IL	FPGA-in-the-loop
GA	Genetic algorithm
HIL	Hardware-in-the-loop
IEEE	The Institute of Electrical and Electronics Engineers
IP	Internet Protocol
MBD	Model-based design
MIL	Model-in-the-loop
MILP	Mixed-integer Linear Programming
MPC	Model-predictive control
MPPT	Maximum power point tracker
NTU	Number of Transfer Units
OPC	Open Platform Communication
OSI	Open Systems Interconnection
PID	Proportional integral derivative
PCommHIL	Power and communications hardware-in-the-loop
PHIL	Power-hardware-in-the-loop
PIL	Processor-in-the-loop
POD	Power oscillation damping
PtX	Power-to-X
PV	Photovoltaics
ROCOF	Rate of change of frequency
RTDS	Real-time digital simulator
SCADA	Supervisory control and data acquisition
SIL	Software-in-the-loop
SoC	State of Charge
SUT	System under test
TCP	Transmission Control Protocol
UAV	Unmanned aerial vehicles
UI	User interface
USD	Universal Scene Description
VAR	Volt-Amps Reactive
V&V	Verification & validation
WoS	Web of Science

Acknowledgements
Not applicable.

Author contributions
Based on the CRediT (Contributor Roles Taxonomy). CSBC: Conceptualization, methodology, formal analysis, investigation, writing—original draft, visualization. BNJ: Supervision, funding acquisition. ZGM: Methodology, writing—review and editing, supervision, project administration, funding acquisition. All authors read and approved the final manuscript.

Funding
Open access funding provided by University of Southern Denmark The "Digital Energy Hub" project, funded by the Danish Industry Foundation. The "IEA EBC Annex 81 Data-Driven Smart Buildings" project, funded by EUDP (case number: 64019–0539). The "ClusterSouthH2 - Designing a PTX Ecosystem in Southern Denmark" project, funded by the European Regional Development Fund.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
Bo Nørregaard Jørgensen is a section editor and a member of the editorial board of Energy Informatics. Zheng Grace Ma is a section editor and a member of the editorial board of Energy Informatics.

Received: 3 January 2024 Accepted: 9 February 2024
Published online: 27 February 2024

References

Addaim A, Guennoun Z (2022) Real-time software in the loop simulation for pid control of the nanosatellite reaction wheel, p 61–66, https://doi.org/10.1109/IRASET52964.2022.9737929
Ahamed MFS, Tewolde G, Kwon J (2018) Software-in-the-loop modeling and simulation framework for autonomous vehicles. International conference on electro information technology, p 305–310
Åström KJ, Wittenmark B (2011) Computer-controlled systems: theory and design, third edition (Dover books on electrical engineering). Dover Publications, Mineola
Ayed MB, Zouari L, Abid M (2017) Software in the loop simulation for robot manipulators. Eng Technol Appl Sci Res 7(5):2017–2021
Bailey D, Francois G, Nice G (2014) Putting real production software in the loop, methodologies enabling SW co-development between OEMs and tier 1s. In: CEUR Workshop Proceedings, vol 1234. CEUR-WS, pp 37–47. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84922170241&partnerID=40&md5=4d1897df5d4d742d02afb5d30f98216c
Bayha A, Grueneis F, Schaetz B (2012) Model-based software in-the-loop-test of autonomous systems. Simul Ser 44(4):226–231
Beyer C, Emmerich J, Werner U (2017) Software in the loop - a window lifter model to guide students through the software development process. IEEE global engineering education conference, p 1488–1493
Bittar A, Figuereido HV, Guimaraes PA, Mendes AC (2014) Guidance software-in-the-loop simulation using X-Plane and Simulink for UAVs. International conference on unmanned aircraft systems, p 993–1002
Bonassi F (2020) Software-in-the-loop testing of a distributed optimal scheduling strategy for microgrids' aggregators. IEEE PES Innovative Smart Grid Technologies Conference Europe, 985–989
Bonivento C, Cacciari M, Paoli A, Sartini M (2011) Rapid prototyping of automated manufacturing systems by software-in-the-loop simulation. Chinese Control and Decision Conference, 3968–3973
Brambilla G, Grazioli A, Picone M, Zanichelli F, Amoretti M (2014) A cost-effective approach to software-in-the-loop simulation of pervasive systems and applications. International conference on pervasive computing and communications, p 207–210
Calderón AJ, González I (2018) Software-in-the-loop approach for automation and supervisory systems education. INTED Proceedings, p 4241–4245
Cao C, Chen B, Harper JD (2019) The hardware-in-the-loop simulation of vehicle-grid integration in a distribution grid. Vol. 2019-Augus, https://doi.org/10.1109/PESGM40551.2019.8973838
Caro FD, Mbarek NB, Fredj F, Vaccaro A (2022) Impact analysis of electric vehicles on distribution grid by hardware-in-the-loop simulations, p 30–35, https://doi.org/10.1109/MetroAutomotive54295.2022.9855029
Casolino GM, AlizadehTir M, Andreoli A, Albanesi M, Marignetti F (2016) Software-in-the-loop simulation of a test system for automotive electric drives. IEEE industrial electronics society, p 1882–1887
Čech M, Königsmarková J, Reitinger J, Balda P (2017) Novel tools for model-based control system design based on FMI/FMU standard with application in energetics. In 2017 21st International conference on process control (PC), 6–9 June 2017, p 416–421, https://doi.org/10.1109/PC.2017.7976250.
Chen X, Salem M, Das T, Chen X (2008) Real time software-in-the-loop simulation for control performance validation. Simul Trans Soc Model Simul Int 84(8–9):457–471. https://doi.org/10.1177/0037549708097420
Clausen CSB, Ma ZG, Jørgensen BN (2022) Can we benefit from game engines to develop digital twins for planning the deployment of photovoltaics? Energy Informatics 5(4):42. https://doi.org/10.1186/s42162-022-00222-7
Collins DE, George AD (2001) Parallel and sequential job scheduling in heterogeneous clusters: a simulation study using software in the loop. SIMULATION 77(5–6):169–184. https://doi.org/10.1177/003754970107700503
Dekeyrel S, Fessler M (2023) Digitalisation: an enabler for the clean energy transition. https://www.epc.eu/content/PDF/2023/Vodafone_DP_FINAL.pdf. Accessed 8 June 2023
Demers S, Gopalakrishnan P, Kant L (2007) A generic solution to software-in-the-loop. IEEE Military Communications Conference, p 1427–1432
Derler P, Lee EA, Sangiovanni Vincentelli A (2012) Modeling cyber-physical systems. Proc IEEE 100(1):13–28. https://doi.org/10.1109/JPROC.2011.2160929
Dettù F, Formentin S, Sergio Matteo S, The Twin-in-the-Loop approach for vehicle dynamics control. Ithaca: Cornell University Library, 2023.
Directorate-General for Energy. CEN-CENELEC-ETSI Smart Grid Coordination Group Smart Grid Reference Architecture. 2018. shttps://energy.ec.europa.eu/publications/smart-grid-reference-architecture_en
Ersdi Z, Bri G, Sipos G (2021) Trajectory tracking controller testing in software in the loop environment. Lecture Notes Mech Eng 22:30–37. https://doi.org/10.1007/978-981-15-9529-5_3
European Commission - Directorate-General for Energy. Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions digitalising the energy system - EU action plan. COM/2022/552 final. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52022DC0552.

Eyre JM, Lanyon-Hogg MR, Dodd TJ, Lockwood AJ, Freeman C, Scott RW (2018) Demonstration of an industrial framework for an implementation of a process digital twin. In ASME international mechanical engineering congress and exposition, Proceedings (IMECE), 2018, vol. 2, https://doi.org/10.1115/IMECE2018-87361. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060387090&doi=10.1115%2fIMECE2018-87361&partnerID=40&md5=2b85070ecb0a124349f22707f89b9046

Eyring A, Hoyt N, Tenny J, Domike R, Hovanski Y (2022) Analysis of a closed-loop digital twin using discrete event simulation. Int J Adv Manuf Technol 123(1):245–258. https://doi.org/10.1007/s00170-022-10176-5

Fatras N, Ma Z, Jørgensen BN (2022) An agent-based modelling framework for the simulation of large-scale consumer participation in electricity market ecosystems. Energy Inform 5(4):47. https://doi.org/10.1186/s42162-022-00229-0

Fekkak B, Menaa M, Boussahoua B, Rekioua D (2020) Processor in the loop test for algorithms designed to control power electronics converters used in grid-connected photovoltaic system. Int Trans Electr Energy Syst. https://doi.org/10.1002/2050-7038.12227

Frotscher O (2019) Software-in-the-loop-simulation of a district heating system as test environment for a sophisticated operating software. 223–230, https://doi.org/10.5220/0007809602230230

Garcia A, Gocke R, Johnson N (1994) Virtual prototyping: Concept to production. Defense Systems Management College Press, Fort Belvoir

Gomes C, Thule C, Broman D, Larsen PG, Vangheluwe H (2017) Co-simulation: state of the art. http://arxiv.org/abs/1702.00686.

Gopstein A, Nguyen C, O'Fallon C, Hastings N, Wollman D (2020) DRAFT NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0. National Institute of Standards and Technology. https://www.nist.gov/system/files/documents/2020/07/24/Smart%20Grid%20Draft%20Framework.pdf. Accessed 8 June 2023.

Gourisetti SN et al (2018) A cyber secure communication architecture for multi-site hardware-in-the-loop co-simulation of der Control, p 55–62, https://doi.org/10.1109/RWEEK.2018.8473506

Guerrero CAV et al. A new software-in-the-loop strategy for real-time testing of a coordinated Volt/Var Control, p 6–10, 2016

Guerrero CAV, Silveira PM, Filho JMC (2021) Adaptation of the clonal selection algorithm to the real-time coordinated Volt/VAr control through a software-in-the-loop strategy. Electric Power Syst Res. https://doi.org/10.1016/j.epsr.2021.107092

Han J, Hong Q, Feng Z, Booth CD, Syed MH, Burt GM (2022) Design and implementation of a real-time hardware-in-the-loop platform for prototyping and testing digital twins of distributed energy resources. Energies. https://doi.org/10.3390/en15186629

Hanel R, Thurner S, Klimek P (2018a) Introduction to the Theory of Complex Systems. Oxford Scholarship Online, Oxford

Hassani K, Lee WS (2013) A software-in-the-loop simulation of an intelligent micro satellite within a virtual environment. IEEE international conference on computational intelligence and virtual environments for measurement systems and applications, p 31–36

Herdt LD, Shekhar A, Yu Y, Mouli GRC, Dong J, Bauer P (2021) Power hardware-in-the-loop demonstrator for electric vehicle charging in distribution grids, p 679–683, https://doi.org/10.1109/ITEC51675.2021.9490098.

Huber M, Bons C, Mueller D (2014) Exergetic evaluation of solar controller using Software-In-The-Loop method. Energy Procedia 48:850–857. https://doi.org/10.1016/j.egypro.2014.02.098

Hubschneider S et al (2018) Requirements for power hardware-in-the-loop emulation of distribution grid challenges. https://doi.org/10.1109/UPEC.2018.8541851

Huerta F, Gruber JK, Prodanovic M, Matatagui P (2016) Power-hardware-in-the-loop test beds: evaluation tools for grid integration of distributed energy resources. IEEE Ind Appl Mag 22(2):18–26. https://doi.org/10.1109/MIAS.2015.2459091

Jaensch F, Csiszar A, Sarbandi J, Verl A (2019) Reinforcement learning of a robot cell control logic using a software-in-the-loop simulation as environment, p 79–84, https://doi.org/10.1109/AI4I46381.2019.00027.

Kiesbye J et al (2019) Hardware-in-the-loop and software-in-the-loop testing of the MOVE-II CubeSat. Aerospace 6(12):130. https://doi.org/10.3390/aerospace6120130

Kim J et al (2121) Real-time hardware-in-the-loop distributed energy resources system Testbed using IEEE 2030.5 Standard. https://doi.org/10.1109/ISGTASIA49270.2021.9715595

Kotsampopoulos P et al (2018) A Benchmark system for hardware-in-the-loop testing of distributed energy resources. IEEE Power Energy Technol Syst J 5(3):94–103. https://doi.org/10.1109/JPETS.2018.2861559

Kwon WH, Choi SG (1999) Real-time distributed software-in-the-loop simulation for distributed control systems. Proceedings of the IEEE international symposium on computer-aided control system design, 115-–119, https://doi.org/10.1109/cacsd.1999.808634

Lai C-M, Lin Y-J (2021) Novel offline software-in-the-loop simulation technique for modular single-phase flyback current source grid-tie inverter system. IEEE Access 9:100814–100826. https://doi.org/10.1109/ACCESS.2021.3097204

Lee EA (2008) Cyber physical systems: design challenges. Proceedings - 11th IEEE symposium on object/component/service-oriented real-time distributed computing, ISORC 2008, pp 363–369: https://doi.org/10.1109/ISORC.2008.25

Lee EA (2010) CPS foundations. Proceedings - Design Automation Conference. pp 737–742. https://doi.org/10.1145/1837274.1837462.

Lee S, Hwang BI, Seo KB, Lee WJ (2017) Relative time synchronization of distributed applications for software-in-the-loop simulation, p 753--756, https://doi.org/10.1109/CSE-EUC-DCABES.2016.273.

Li A, Ponchant M, Sturm J, Jossen A (2020) Reduced-order electro-thermal battery model ready for software-in-the-loop and hardware-in-the-loop bms evaluation for an electric vehicle. World Electric Vehicle J 11(4):1–16. https://doi.org/10.3390/wevj11040075

Lynch TJ, Fortmann TE, Briscoe H, Fidell S (1989) Multiprocessor-based data acquisition and analysis. https://repository.arizona.edu/handle/10150/614478

Ma Z (2023) Energy Metaverse: a virtual living lab of the energy ecosystem. Energy Inform 6(1):3. https://doi.org/10.1186/s42162-023-00258-3
Ma Z, Schultz MJ, Christensen K, Værbak M, Demazeau Y, Jørgensen BN (2019) The application of ontologies in multi-agent systems in the energy sector: a scoping review. Energies 12(16):3200. https://doi.org/10.3390/en12163200
Martinez R, Wu WJ, McNeill K, Deal J, Haynes T, Bradford D (2003) Hardware and software-in-the-loop techniques using the OPNET modeling tool for JTRS developmental testing. IEEE military communications conference, p 469–474
Mastronarde N et al (2022) RF-SITL: a software-in-the-loop channel emulator for UAV swarm networks, p 489–494, https://doi.org/10.1109/WoWMoM54355.2022.00076
Mo R, Yang Y, Li H (2014) Power hardware-in-the-loop simulation of integrated voltage regulation and islanding detection for distributed PV systems on GRU model, p 4004–4007, https://doi.org/10.1109/ECCE.2014.6953946
Morales-Caporal M, Rangel-Magdaleno J, Peregrina-Barreto H, Morales-Caporal R (2018) FPGA-in-the-loop simulation of a grid-connected photovoltaic system by using a predictive control. Electr Eng 100(3):1327–1337. https://doi.org/10.1007/s00202-017-0596-1
Moreno J et al (2019) Hardware-in-the-Loop and Software-in-the-Loop platform for testing and validation of adaptable radio communications systems for railways at IP layer. Lecture Notes in Computer Science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol. 11461 LNCS, pp. 111--118, https://doi.org/10.1007/978-3-030-25529-9_10.
Muresan M, Pitica D (2012) Software in the Loop environment reliability for testing embedded code, p 325–328, https://doi.org/10.1109/SIITME.2012.6384402
Musse BF, Silva NL, Silva DC (2017) Controller-hardware-in-the-loop simulation of a distribution system with PV penetration using RTDS and dSPACE. p 1–6, https://doi.org/10.1109/COBEP.2017.8257399
Nagy I, Cleophas L, van den Brand M, Engelen L, Raulea L, Mithun EXL (2012), VPDSL: a DSL for software in the loop simulations covering material flow. IEEE international conference on engineering of complex computer systems-ICECCS, p 318--327, https://doi.org/10.1109/ICECCS.2012.43.
Negrin DAM, Cleophas L, Brand MVD (2021) Using Ptolemy II as a framework for virtual entity integration and orchestration in digital twins
Newaz A, Ospina J, Faruque MO (2020) Controller hardware-in-the-loop validation of a graph search based energy management strategy for grid-connected distributed energy resources. IEEE Trans Energy Convers 35(1):520–528. https://doi.org/10.1109/TEC.2019.2952575
Nguyen HT, Yang G, Nielsen AH, Jensen PH (2019a) Hardware- and software-in-the-loop simulation for parameterizing the model and control of synchronous condensers. IEEE Trans Sustain Energy 10(3):1593–1602. https://doi.org/10.1109/TSTE.2019.2913471
Nguyen HT, Yang G, Nielsen AH, Jensen PH, Coimbra CFM (2019b) Control parameterisation for POD via software-in-the-loop simulation. J Eng 18:4864–4868. https://doi.org/10.1049/joe.2018.9331
Nibert J, Herniter ME, Chambers Z (2012) Model-based system design for MIL, SIL, and HIL. World Electric Vehicle J 5(4):1121–1130
Osadcuks V, Galins A (2012) Software in the loop simulation of autonomous hybrid power system of an agricultural facility. Eng Rural Dev. 500–505
Papaioannou et al (2018) Smart grid interoperability testing methodology
Park HE, Park SY, Park C, Kim SW (2013) Development of integrated orbit and attitude software-in-the-loop simulator for satellite formation flying. J Astron Space Sci 30(1):1–10. https://doi.org/10.5140/JASS.2013.30.1.001
Park H, Easwaran A, Andalam S (2019) TiLA: twin-in-the-loop architecture for cyber-physical production systems. In 2019 IEEE 37th international conference on computer design (ICCD), 17–20 Nov. 2019, pp. 82–90, https://doi.org/10.1109/ICCD46524.2019.00019
Park M, Jang M, Byun T, Choi Y (2020) Property-based testing for LG home appliances using accelerated software-in-the-loop simulation. pp 120–129, https://doi.org/10.1145/3377813.3381346
Parthasarathy D, Johansson A (2021) SilGAN: generating driving maneuvers for scenario-based software-in-the-loop testing, p 65–72, https://doi.org/10.1109/AITEST52744.2021.00022.
Pereira C, Saxena N, Ravula R, Kashyap V, Kamal S, Rakesh R (2016) Software in the loop test set-up of a tethered nano-satellite. IEEE Aerospace Conference Proceedings
Pieper T, Obermaisser R (2018a) Network-centric co-simulation framework for software-in-the-loop testing of geographically distributed simulation components. Proceedings - 21st IEEE international conference on computational science and engineering, CSE 2018, pp. 104–111, https://doi.org/10.1109/CSE.2018.00021.
Pieper T, Obermaisser R (2018b) Distributed co-simulation for software-in-the-loop testing of networked railway systems. Mediterranean conference on embedded computing, p 24–28
Ponchant M, Li A, Beckers C, Paroha M (2021) Battery management system evaluation within a complete electric vehicle model with software-in-the-loop and hardware-in-the-loop approaches. European conference power electronic applications
Raghupatruni I, Goeppel T, Atak M, Bou J, Huber T (2019) Empirical testing of automotive cyber-physical systems with credible software-in-the-loop environments. International Conference on Connected Vehicles and Expo.
Ravikumar G, Hyder B, Govindarasu M (2020) Hardware-in-the-loop CPS security architecture for der monitoring and control applications, https://doi.org/10.1109/TPEC48276.2020.9042578
Reyes V. Virtual hardware "In-the-loop": earlier testing for automotive applications. synopsys. Accessed 5 Feb 2024
Riley PF, Riley GF (2003) Next generation modeling III - agents: spades—a distributed agent simulation environment with software-in-the-loop execution, Wsc '03, pp. 817–825
Riley PF, Riley GF (2003) Spades - A distributed agent simulation environment with software-in-the-loop execution, p 817--825, https://doi.org/10.1109/WSC.2003.1261500.
Rossi I, Reveillere A, Traverso A (2019) Flexibilization of gas turbine combined cycle via heat pump: development of control logics via Software-in-the-loop application. E3S Web Conf. 113:01004. https://doi.org/10.1051/e3sconf/201911301004

Rotger-Griful S, Chatzivasileiadis S, Jacobsen RH, Stewart EM, Domingo JM, Wetter M (2016) Hardware-in-the-Loop co-simulation of distribution Grid for demand response. https://doi.org/10.1109/PSCC.2016.7540828
Russo R, Terzo M, Timpone F (2007) Software-in-the-loop development and validation of a Cornering Brake Control logic. Veh Syst Dyn 45(2):149–163. https://doi.org/10.1080/00423110600866491
Schimeczek C et al (2023a) FAME-Core: an open framework for distributed agent-based modelling of energy systems. J Open Sour Softw 8(84):5087–5087. https://doi.org/10.21105/joss.05087
Schimeczek C et al (2023b) AMIRIS: agent-based market model for the investigation of renewable and integrated energy systems. J Open Sourc Softw 8(84):5041–5041. https://doi.org/10.21105/joss.05041
Scholz M, Oberschachtsiek S, Donhauser T, Franke J (2017) Software-in-the-loop testbed for multi-agent-systems in a discrete event simulation: Integration of the Java Agent Development Framework into Plant Simulation. 2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings https://doi.org/10.1109/SysEng.2017.8088320.
Seo HR, Park M, Yu IK, Song BM (2011) Performance analysis and evaluation of a multifunctional grid-connected PV system using power hardware-in-the-loop simulation, p 1945–1948, https://doi.org/10.1109/APEC.2011.5744862
M. Shaw, "A software design paradigm based on process control," ed, 1994.
Shivanandaswamy KK, Sarkar C, Rajagopal S, Ramachandra LP, Chayakumar CM (2021) Emphasis on evaluative prerequisites for decisive software-in-the-loop (SiL) environments, p 450–457, https://doi.org/10.1109/INDIACom51348.2021.00080.
Silano G, Oppido P, Iannelli L (2019) Software-in-the-loop simulation for improving flight control system design: a quadrotor case study. IEEE international conference on systems man and cybernetics conference proceedings, p 466–471
Silva DHC, Santos MF, Silva MF, Neto MFS, Mercorelli P (2019) Design of controllers applied to autonomous unmanned aerial vehicles using software in the Loop, p 714–719, 2019
Singh SC, Shubhanga KN (2017) Software-in-the-loop based MPPT enabled realtime solar photovoltaics simulator in FPGA platform for academic appreciation.
Sommerville I (2016) Software engineering, global edition. Pearson Education, London
Sommerville I. Web sections - critical systems. https://software-engineering-book.com/web/critical-systems/. Accessed 19 June 2023
Sørensen JV, Ma Z, Jørgensen BN (2022) Potentials of game engines for wind power digital twin development: an investigation of the Unreal Engine. Energy Inform 5(4):39. https://doi.org/10.1186/s42162-022-00227-2
Steinbrink C et al. (2017) Simulation-based validation of smart grids – Status quo and future research trends. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol. 10444 LNAI, p 171–185, https://doi.org/10.1007/978-3-319-64635-0_13
Stevi S (2020) Development of ADAS perception applications in ROS and "Software-In-the-Loop" validation with CARLA simulator. Telfor J 12(1):40–45. https://doi.org/10.5937/TELFOR2001040S
Strasser TI, de Jong ECW (2020) European guide to power system testing.
Strohbeck J, Muller J, Holzbock A, Buchholz M (2021) DeepSIL: a software-in-the-loop framework for evaluating motion planning schemes using multiple trajectory prediction networks. IEEE international conference on intelligent robots and systems, p 7075–7081, https://doi.org/10.1109/IROS51168.2021.9636423.
Taut MA, Chindris G, Taut AC (2019) Software-in-the-loop system for motor control algorithms. International Symposium for Design and Technology in Electronic Packaging, p 419–426
The Strategic Defense Initiative Organization (1988) Report to the congress on the strategic defense initiative. https://apps.dtic.mil/sti/pdfs/ADA321530.pdf
Tuominen V, Reponen H, Kulmala A, Lu S, Repo S (2017) Real-time hardware- and software-in-the-loop simulation of decentralised distribution network control architecture. IET Gener Transm Distrib 11(12):3057–3064. https://doi.org/10.1049/iet-gtd.2016.1570
Værbak M, Ma Z, Demazeau Y, Jørgensen BN (2021) A generic agent-based framework for modeling business ecosystems: a case study of electric vehicle home charging. Energy Inform 4(2):28. https://doi.org/10.1186/s42162-021-00158-4
Vandi G et al (2014) Development of a software in the loop environment for automotive powertrain systems. Energy Procedia 45:789–798. https://doi.org/10.1016/j.egypro.2014.01.084
Venturi C, Lima N, Barroso D, Machado PA, Oliveira JG (2015) Hardware in the loop simulation of DG integration to the distribution grid using RTDS and dSPACE, https://doi.org/10.1109/COBEP.2015.7420137
Viehweider A, Lauss G, Lehfuss F (2012) Interface and stability issues for SISO and MIMO power hardware in the loop simulation of distribution networks with photovoltaic generation. Int J Renew Energy Res 2(4):631–639
Vignati M, Debattisti N, Bacci ML, Tarsitano D (2021) A software-in-the-loop simulation of vehicle control unit algorithms for a driverless railway vehicle. Appl Sci 11(15):6730. https://doi.org/10.3390/app11156730
Wang J et al (2020) Performance evaluation of distributed energy resource management via advanced hardware-in-the-loop simulation. https://doi.org/10.1109/ISGT45199.2020.9087667.
Wang J, Padullaparti H, Ding F, Baggu M, Symko-Davies M (2021a) Voltage regulation performance evaluation of distributed energy resource management via advanced hardware-in-the-loop simulation. Energies 14(20):6734. https://doi.org/10.3390/en14206734
Wang J, Simpson J, Yang R, Palmintier B, Tiwari S, Zhang Y (2021b) Hardware-in-the-loop evaluation of an advanced distributed energy resource management algorithm, https://doi.org/10.1109/ISGT49243.2021.9372182
Werner S, Masing L, Lesniak F, Becker J (2015) Software-in-the-loop simulation of embedded control applications based on virtual platforms, https://doi.org/10.1109/FPL.2015.7294020.
Yang J, Wang X, Baldi S, Singh S, Fari S (2019) A software-in-the-loop implementation of adaptive formation control for fixed-wing UAVs. IEEE-CAA J Automatica Sinica 6(5):1230–1239. https://doi.org/10.1109/JAS.2019.1911702
Youcefa BE, Massoum S, Barkat S, Bella S, Wira P (2019) DPC method for grid connected photovoltaic system acts as a shunt active power filter implemented with processor in the loop. No. 3rd International Conference on Electrical

Sciences and Technologies in Maghreb (CISTEM), pp. 144--150 WE -- Conference Proceedings Citation Inde, doi: https://doi.org/10.1109/CISTEM.2018.8613370.
Zhang X, Mi C (2011) Hardware-in-the-loop and software-in-the-loop testing for vehicle power management. Power Syst 51:303–329. https://doi.org/10.1007/978-0-85729-736-5_10
Zhang J, Prabakar K, Hasandka A (2021) Power and communications hardware-in-the-loop CPS architecture and platform for der monitoring and control applications, p 170–174, https://doi.org/10.1109/SmartGridComm51999.2021.9631992

Publisher's Note