Exploring the Solution Space for Multi-Paradigm Integration Challenges

Emily Chatziandreou, Pepijn de Haan,
Maryvonne Marang, and Lotte Westrik Broeksma

Graduate students, Delft University of Technology | Faculty of Technology, Policy and Management, Delft, the Netherlands
{a.a.chatziandreou, p.h.m.dehaan, m.n.marang,
l.j.m.westrikbroeksma}@student.tudelft.nl

Abstract. Energy modeling and simulation are essential for enhancing the efficiency of increasingly complex energy systems. As these systems evolve to include communication networks and decentralized energy production, stable integration of different modeling paradigms, such as discrete-event (DE) and continuous-time (CT), becomes vital. This hybrid co-simulation approach presents significant challenges, particularly in areas like semantics, discontinuities, time synchronization, and infinite loops. This paper delves deeper into these challenges using a systematic literature review, highlighting how various researchers address these issues in hybrid co-simulation. A solution space is proposed, detailing potential strategies and their limitations, providing developers with a practical framework for tackling these issues in hybrid co-simulation design.

Keywords: Multi-paradigm · Simulation · Co-simulation · Discrete-Event · Continuous.

1 Introduction

Energy modeling and simulation have become essential tools for addressing and understanding critical issues such as sustainability and efficient energy use. As energy systems grow in complexity, the number of interconnected subsystems playing key roles has expanded. A prominent example is the integration of communication systems, which are now vital for monitoring and controlling modern energy grids. The importance of communication networks in energy systems increases as energy systems have become more distributed. This is highlighted by the installation of over 180 million off-grid solar systems since 2010, including 30 million solar home systems (Nadeem et al., 2023).

However, incorporating communication systems into energy models is not a simple task. Communication networks are typically modelled using a discrete-event (DE) approach that focuses on event-driven changes, whereas energy systems—especially power grids—are modelled using a continuous-time (CT) approach (Mets et al., 2014). In practice, this means that communication networks have to respond immediately to specific events, such as a change in market demand or network failure, allowing for real-time adjustments. In contrast, energy systems operate on a continuous basis, where electricity generation and consumption are managed over time, accommodating gradual fluctuations in supply and demand. Coupling these two distinct paradigms presents a significant challenge.

A similar complexity arises when modeling the supply and demand of energy. With the decentralisation of energy grids, households are no longer just consumers; they are also becoming energy producers, contributing to the supply side (Oberst et al., 2019). This dual role can be modelled using different paradigms, such as agent-based models or system dynamics, both of which would need to be integrated with established power system models to reflect real-world interactions more accurately. For example, an agent-based model could simulate individual households behaviour while integrating with a system dynamics model that reflects overall grid stability.

The challenge lies in effectively bridging these diverse modeling techniques to capture the evolving dynamics of modern energy systems. One way to couple models using different approaches to time is called 'co-simulation', which can be described as follows: "Co-simulation is an emerging enabling technique, where global simulation of a coupled system can be achieved by composing the simulations of its parts" (Gomes et al., 2018). They emphasize that coupled systems are independent (complex) systems that share inputs and outputs, thereby exchanging information. This interdependence introduces additional challenges, such as effective time synchronization between the discrete-event and continuous-time models.

The type of co-simulation this paper addresses is hybrid co-simulation, where discrete and continuous systems are coupled in a simulation. Gomes et al. (2018) mention that there are essentially two ways to implement hybrid co-simulation: either adapt a continuous model to "output" discrete information and use a discrete connector between the models (hybrid DE co-simulation), or vice versa (hybrid CT co-simulation). They also outline several challenges that arise when coupling these systems. In addition to these challenges, Li and Zhang (2014) highlight that side effects such as time delays, packet loss and disconnections in communication networks can significantly impact the performance of power grid operations. To effectively manage the complexities of hybrid co-simulation, it is essential to consider these effects, as they can lead to inaccuracies in simulation results.

The goal of this paper is to explore the solution space for hybrid co-simulation, addressing the challenges and implementation strategies for integrating different modeling paradigms using co-simulation. To achieve this, the paper investigates the issues faced by simulation developers when implementing hybrid co-simulation. By studying examples from different papers, key problems that commonly arise in these scenarios are identified. Potential solutions to these challenges are then explored, along with an analysis of how these issues are interconnected. This paper aims to be a valuable resource for developers, offering a comprehensive list of recommendations to help implement hybrid co-simulation more effectively and enhance the accuracy and efficiency of energy system simulations.

2 Method

For this paper, a systematic review methodology was used to select and review papers. The starting point of this paper was Godfrey et al. (2010), provided by Delft University of Technology during the course Simulation Masterclass (SEN9110). Using the search term "Co-simulation" then quickly gave the paper on which this paper focuses: Gomes et al. (2018).

The search terms used for literature search is different for the different challenge clusters that were identified.

Challenge 1 The search terms used were: 'Semantic adaptation' AND 'Co-simulation'. Three papers were retrieved by backward snowballing from the initial source Gomes et al. (2018) and one paper was provided during the Delft University of Technology course Simulation Masterclass (SEN9110). One paper was retrieved through forward snowballing from the initial source. This was done because the original source was a systematic literature review recognising challenges and information about current developments on the recognised challenges were sought.

Challenge 2 The search terms used were: 'Discontinuity' AND 'Co-simulation'; 'Hybrid simulation' AND 'Energy sector'; 'Adaptive time steps' AND 'Co-simulation'; 'GSS' AND 'Simulation'; 'Embedding DEVS'. Two other papers used were snowballed from the papers found using the aforementioned search terms. There weren't strict requirements for the selection of papers, only that they were published recently. However, when it proved difficult to find suitable papers from the last eight years, the year constraint was loosed up a bit.

Challenge 3 Search terms used were: 'Co-simulation'; 'Time synchronisation' AND 'Co-simulation'; 'Co-simulation energy communication' AND ('use case' OR 'case study'); "power simulation" AND "communication simulation". There weren't strict requirements for a paper to be selected, as long as they related to the issue of time. Two papers were chosen as practical examples of how to synchronise time, and a more recent paper was chosen to explain novel ideas regarding time synchronisation in hybrid co-simulation. It appeared to be easier to find papers in this category when compared to the other categories; most co-simulation frameworks discuss how time is managed between different models. Therefore, no snowballing methods were needed.

Challenge 4 The search terms used were: "Co-simulation" AND "energy" AND "case study"; "Multi-paradigm" AND "energy" AND "case study." It was essential that the identified use cases were grounded in the energy field to ensure practical applicability. To delve deeper into the specific challenges defined by Gomes et al. (2018), particularly the issue of infinite loops, additional search terms like "illegitimacy" AND "co-simulation" and "algebraic loop" AND "co-simulation" were employed. For these terms, it was crucial that a solution to the challenges was presented. During this process, two relevant papers discussing the theory behind solutions for algebraic loops were identified, prompting the use of a snowballing approach to expand the review further. Notably, since the paper on algebraic loops was a review article, it is likely that a follow-up will be published in the future. To anticipate these developments, a search was conducted on recent publications by Gomes to explore additional contributions to the field. Additionally, 'zeno behaviour AND co-simulation' was also explored; however, this yielded limited results regarding a solution to zeno behaviour in co-simulation frameworks. The combination of these search terms provided a comprehensive understanding of the practical applications and theoretical issues related to co-simulation in energy systems.

3 Challenges to address

Hybrid co-simulation in energy systems reveals several technical challenges that must be addressed to ensure accurate and efficient simulations. Gomes et al. (2018) identified a range of these challenges, which arise mainly due to the integration of continuous-time and discrete-event simulation units (SUs), each of which operates under different principles and timescales. Achieving effective communication and synchronisation between these two paradigms is essential, but not straightforward.

To analyse these challenges, a literature review of relevant case studies and research papers that applied hybrid co-simulation techniques in energy systems was conducted. The focus was on examining how different authors addressed the challenges identified by Gomes et al. (2018) in their implementations or whether they did not address them. Their approaches were categorised into four main clusters:

- Semantic adaptation: involving issues regarding semantic adaptation and integration methods subsection 3.1
- Discontinuity: the identification and handling of transitions between continuous and discrete states, which is the focus of subsection 3.2
- The Issue of Time: focusing on synchronisation and handling of time across different simulation units, as discussed in subsection 3.3

– Infinite Loops: addressing scenarios such as algebraic loops, illegitimacy and zeno behaviour, which are explored in subsection 3.4

3.1 Challenge 1 : Semantic Adaptation

Semantic adaptation and model composition in hybrid co-ulation are important challenges, as they involve integrating submodels that follow different paradigms and thus have to meet different requirements. To enable effective hybrid modelling, it is essential to translate between these time-based signals (continuous) and event-based outputs (discrete) to ensure that the behaviour of one model accurately reflects changes occurring in the other. This requires mechanisms that can handle real-time data conversion and synchronisation, preventing miscommunication between models and ensuring overall system coherence (Liboni & Deantoni, 2021)

In order to make communication between the models possible two major challenges need to be addressed; signal data mismatch and hierarchy. The first issue reflects on the different time and data interpretations across simulation units. This is crucial when integrating continuous and discrete components, where semantic adaptation involves converting continuous data into discrete events or vice versa. This process requires precise event detection and synchronisation methods to ensure accurate and consistent communication between submodels. Another aspect is the hierarchical nature of some hybrid systems, where components might have nested or layered interactions that need coordination and specific interaction rules, like for example the reactions between consumers/producers in a smart grid where the electricity market supply can both affect as be affected by the consumer's behaviour. This setup requires semantic adaptation to ensure that all parts of the system can communicate effectively without conflicts or loss of information.

It is important to note that resolving semantic adaptation involves three distinct levels: the theoretical base, denoting according to which formalism the hybrid model is defined, the integration strategy, reflecting upon the choice of the specific level in which the 'translation' of the sub-models happens and lastly the structural frameworks specifying the architecture of this 'translation'.

Theoretical base. The first level of semantic adaptation touches upon the theoretical basis that defines the hybrid model. Languages in which both paradigms can be described: super-formalism techniques such as DEVS/DESS and Hybrid DAE provide a conceptual foundation for hybrid modelling by offering frameworks that can describe both discrete-event and continuous-time dynamics within a unified model (Vangheluwe et al., 2002). By accommodating multiple paradigms, these super-formalisms allow for the expression of systems that need to interact across different domains and time scales.

Integration method. Hybrid models can be integrated in one of two ways: either a common formalism is used in which both submodels are translated in or the submodels are contained in their own specialised paradigm and an interface in-between these two models is added that transforms the output of one submodel in the input other submodels need (Vangheluwe et al., 2002).
The first method, aims to provide a unified framework that encapsulates various formalism structure transformations and could be re-used across studies. The second approach focuses on the independent operation of each simulator, allowing for distribution of expertise.

*Structural frameworks.*The structuring of the interaction between the simulation units of each submodel (SU's) calls for practical frameworks that facilitate the execution of hybrid models created using super-formalisms. These frameworks provide the necessary infrastructure to integrate diverse models, handling communication, synchronisation, and data exchange and aim to provide an architecture that can be followed in order to ensure successful integration.

Possible Strategies to address Semantic Adaptation. Both issues of signal mismatch and hierarchy in co-simulation can be address with the use if structural frameworks. The most commonly used standards for hybrid co-simulation as of 2019 are High Level Architecture (HLA) and the Functional Mock-up Interface (FMI) (Schweiger et al., 2019). Both define a structure in which each hybrid model is seen as a combination of its submodels (defined as federates or FMUnits repectivelly) and an orchestrator that manages this communication.
In the FMI framework a master algorithm orchestrates the co-simulation dictating the FMUnits in the order defined in the master algorithm.This way hierarchy is encapsulated ensuring an orderly interaction (Denil et al., 2015). In order to address the unit mismatch, a conversion of different units (FMU's) is realised by the creation of a new FMU with inputs/outputs resulting from an algebraic transformation of the FMU's that where involved in the conversion (Gomes et al., 2019).
The HLA framework the order in which federated should interact and exchange information is dictated by the Runtime Infrastructure (RTI), addressing issues of hierarchy. A way to handle the signal management is through a controller federate that has thresholds and established outputs it can communicate through the RTI when a threshold is reached (Dahmann et al., 1997).

Although these frameworks are widely used in hybrid co-simulation there are other multiple specialised tools modellers use to integrate their models. Research shows that multiple tools like SimuLink and Ptolemy are being used by experts to practically address integration issues however, there is a significant percentage that mentioned the use of 'self written' tools (Schweiger et al., 2019).This makes sense in a context where the physical systems and their depictions in formalisms can vary greatly from project to project, thus resulting a custom approach that will probably save the modellers time of extensively understanding and correctly adapting an already-existing tool. An example of such a tool is MOSAIK, developed with the purpose of co-simulation in complex energy systems such as smart grids (Steinbrink et al., 2018). In comparison to HLA, MOSAIK concentrates more functionality in the Sim-Manager, the managing unit of the tool (comparable to the RTI), and leaves much less room to the individual components. This allows for the user to control the data exchange through a script and not the units, making this approach more centralised. This approach can in general be implemented quicker compared to HLA but at the expense of extended functionality options for the individual units. Trade-offs like this, combined with intellectual property issues allow for such a great variety of tools to be available.

To conclude, it is important to define a practical integration strategy in which the alignment of data and time integration throughout the sub-models is managed in a clear way. As shown this can be accommodated in both a central and decentralised way in order to fit the specifications and interaction needs of the real systems.

3.2 Challenge 2 : Discontinuity

The Issue with Combining Continuous and Discrete Models. Discontinuity happens when continuous components, which are solved by a numerical integrator, are combined with discrete event components, which are scheduled by a DE simulator. As far as the integration goes, the discrete states behave exactly like parameters in the continuous components; they never change their value in between time steps, but only at event times (Cellier & Kofman, 2006). These events can cause immediate change in the continuous state, as this is part of both the internal and external transition function of the system, adapting the differential equation of the continuous part (Zeigler, 2006).

There are two types of events that can cause discontinuity in hybrid co-simulations; time-evens (also called *time delayed state* in discrete event simulation) and state events (called *conditional wait state* in discrete event simulation). Time events are triggered at a previously known time, and thus can be scheduled in the event list of the model (with their event time and type). While there might be some difficulties with race conditions and step size, this is is a well known and established field of study (Dejaco & Benedikt, 2017). Therefore, it won't oppose a big challenge when conducting co-simulations.

State events happen when a continuous state x(t) reaches a certain threshold. Opposed to time events, state events are more difficult to deal with in simulation, as the exact time instant of the discontinuity must be detected. In order to do so, they must be continuously monitored during the simulation run. Once they pass the threshold, an interpolation mechanism is started to determine the threshold-crossing time with a pre-specified precision (Cellier & Kofman, 2006). There are multiple algorithms to determine the next time at which the threshold needs to be evaluated; Regula Falsi, the Golden Section method, a Newton iteration, a cubic interpolation and many more. The last two of which are faster to converge, with only the cubic interpolation guaranteed to converge (Cellier & Kofman, 2006). The actual calculation of the threshold-function is done by the higher-order numerical integration scheme that is used throughout the simulation, limiting the approximation error (Cellier & Kofman, 2006).

Possible Strategies to Handle Discontinuity. As discrete and continuous times cannot be connected as such, it has to be specified when exactly data has to be exchanged between the different components of the model. One of the most basic/loose coupling strategies is the periodic data exchange, where the separate components run in parallel, extrapolate the necessary coupling data and synchronize at fixed points. This strategy does not require the simulation to jump back in time to make iterations, and therefore the communications does not require deep involvement in the simulation engine (Heinzl et al., 2018). The choice of the size of the macro-time step, as well as the type of subsystem scheduling, is an important trade-off between the overall simulation time and the accuracy of the simulation results (Stettinger et al., 2013).

While in principle this works, it becomes an issue when the continuous variable experiences a very steep incline or decline. The extrapolation will overshoot the value of the continuous variable during the next time step, after which it is over-corrected again and again leading to oscillation of the continuous variable. In other words, extrapolation techniques amplify noisy measurements (Stettinger et al., 2013). They lead to discontinuities in the values, and thus the solution, which arise after each time step, and disturb the numerical time integration in the subsystems (Busch, 2019). A way to work around this is by reducing the step-size or rejecting steps at the discontinuities, but this will cause the calculation effort to strongly increase (Busch, 2019). Another drawback of this loosely coupled integration is that events occurring between the macro-steps are delayed until the next data synchronization, increasing numerical errors in the calculations (Heinzl et al., 2018).

In cases where (discrete) variables can jump significantly between macro-steps and simple extrapolation is not accurate enough, a more tight coupling strategy is needed. One example of this is a rollback strategy, which provides a potential solution for capturing and handling state events (Heinzl et al., 2018). In a rollback-based model, the discrete simulator advances before the continuous simulator. When the continuous model generates a state event, it will send both the data as well as the time the state event occurred. The discrete simulator will rollback to the previous state and restore the data saved for that time step (Gheorghe, 2009). However, as this requires implementation deep into the inner workings of the simulator, it is not provided in many co-simulation frameworks as an option (as of 2018) (Heinzl et al., 2018).

Another way of dealing with discontinuity is by discretizing the state value, while keeping time continuous. Instead of asking the question; "given the state value at a certain time, what is the state value at the next time step", the question "given the value of the state at a certain time, what is the earliest time instant at which the state assumes the next value" is being asked. As these Quantized State Systems (QSS) don't require the system to replace the iteration on state events by interpolation, thereby saving a a lot of computational and memory power, QSS algorithms are very suitable to simulating systems with large discontinuities like switched power convert circuits, efficiently and accurately (Cellier et al., 2008).

3.3 Challenge 3 : The Issue of Time

In co-simulation, SUs are coupled to simulate a system (Gomes et al., 2018). These independent, often black-box SUs may not handle time the same way, requiring an orchestrator to manage time progression and data exchange between them (Gomes et al., 2018). The challenge of time synchronization arises in both hybrid DE and CT co-simulation, but for different reasons. In hybrid DE co-simulation, the orchestrator manages time around significant events, whereas CT SUs evolve continuously and cannot inherently know when a discrete event will occur. In hybrid CT co-simulation, DE SUs must react to continuous signals, raising the question of how they determine when to trigger an event based on those continuous signals.

In this section different solutions for these issues are provided. Two examples are provided of how one can instruct the orchestrator to synchronise time in hybrid DE co-simulation and posit solutions from Gomes et al. (2018) for hybrid CT co-simulation. A more theoretical paper from Cremona et al. (2019) is also examined which proposes more novel solutions to the issue. Finally, the results are summarised before providing a comprehensive list of possible solutions for the issue of time.

Potential Solutions Addressing the Issue of Time. Hopkinson et al. (2006) propose the EPOCHS platform for simulating power and communication systems by employing different simulators for each sector—for example, one Discrete Event (DE) simulator for network simulation and another Continuous Time (CT) simulator for power system simulation—connected through a Runtime Infrastructure (RTI) which serves as the orchestrator. To synchronise time across these simulators, EPOCHS uses a *fixed time-stepped method*. This entails that the RTI sets fixed synchronisation points at which the simulators can communicate with each other. This was also discussed in subsection 3.2, where the timing between synchronisation points were called macro-steps. As was mentioned, one issue that arises with this approach is that it can introduce errors in the model. In EPOCHS, messages from the DE SU that are meant for the CT SU have to wait in a queue for the next synchronisation point, but this situation is not ideal as it can introduce errors that accumulate over time. One solution to this is using smaller macro-steps to minimise errors, though this comes at the cost of efficiency and thereby does not fully resolve the issue.

Lin et al. (2011) propose a different solution which they apply in two use cases. Time synchronisation in these cases happens through a *global scheduler*, which processes events in real-time without needing explicit synchronisation points between the simulators. This means that the simulators use the same timeline instead of running separately. The global timeline includes all timesteps of the CT SU as 'events' and thereby merges the two SUs into one big event list. This approach eliminates errors and delays that would otherwise occur if events had to wait for fixed synchronisation intervals. However, a problem that arises here is that the number of events increase drastically, especially since CT SUs often use very small timesteps (Lin et al., 2011). This would lead to very slow simulations. To combat this issue, the global scheduler only handles events when a relevant event occurs. However, this would expose the inner workings of the SUs, since it is impossible to predict a relevant event if you do not know the context of that event. This would make this method less accessible to parties who wish to leave their SUs as black boxes. Additionally, using a global event list still opens up the possibility of errors, though the possibility is smaller when compared with the fixed time-step approach. If a discrete event happens in between the time interval of the CT SU, does it wait until the end of the interval to take that discrete event into account? This can still induce errors into the model. Lin et al. (2011) also underline that these issues are a trade-off based on the size of the time-step that is chosen.

No cases were found to illustrate solutions that can be used to synchronise time in hybrid CT co-simulation. Therefore the paper of Gomes et al. (2018) is used to illustrate this theoretically. In hybrid CT co-simulation, the DE SUs need to react to continuous signals from the CT systems. One solution to have this happen smoothly is to *make the DE SUs predict when an event will occur* based on the continuous signals it receives. This prediction helps align the DE SU's discrete events with the continuous simulation. One issue with this is that the orchestrator has to be able to adaptively adjust the step size based on whether the DE SU predicts a discrete event is going to occur. If the orchestrator calculates that a threshold will be crossed in the CT SU right in between a time interval of that CT SU, the step size should be decreased so that the CT SU can communicate right at the moment the threshold is crossed. An important note here is that 'in CT-based co-simulation, error is an accepted, well studied, and controllable, feature.' (Gomes et al., 2018). This accepted error is an issue here, since it can influence the prediction of when an event has to fire.

Finally, Cremona et al. (2019) posit the idea that the representation of time is the root cause of the issues relating to time in hybrid co-simulation, and propose to *represent time as integers* to prevent quantization errors. This can help with detecting when significant events happen in CT simulation. For example, the orchestrator might detect an event at t = 0.4000000001 seconds instead of exactly t = 0.4 seconds. These small errors can accumulate over time. Representing time as integers can reduce the room for errors in this case. Additionally, they propose a method where the *orchestrator negotiates the time steps between different SUs*. The orchestrator determines a global time resolution for the simulation and negotiates the time steps with the SUs. If an SU cannot handle the proposed step size, the orchestrator adjusts it to meet the component's requirements. This is similar to the concept of QSS discussed in subsection 3.2.

Conclusion. There are a number of proposed solutions for the issue of time in hybrid co-simulation that were listed in this section.

- Fixed time-stepped method (hybrid DE co-simulation): the orchestrator sets fixed synchronisation points where SUs can communiate. This can introduce accumulating errors in the simulation over time. Reducing error by decreasing step size also increases inefficiency of the simulation.
- Global scheduler (hybrid DE co-simulation): integrating both SUs to one global event list reduces errors greatly. This necessitates prediction of relevant events by the CT SU, as the simulation would be very slow otherwise. However, this would expose the inner workings of the SU. Finally, the possibility of errors still exists, although it is much smaller.
- Event prediction based on continuous signals (hybrid CT co-simulation): allows DE SUs predict when they should fire events based on continuous inputs. This necessitates the orchestrator to be able to adaptively adjust the step size of the CT SU (this is proposed in Cremona et al. (2019)). Additionally, one should pay attention to the acceptability of errors that is common in continuous simulation. This is not common in DE simulation and could introduce significant errors.
- Representing time as integers: representing time as an integer (with a predefined time resolution) can help in reducing quantization errors as discussed in the section about the fixed time-stepped method. This is not a definite solution of synchronising time but is mentioned seperately here due to its novelty, though not studies were found that have implemented this solution for hybrid co-simulation.

3.4 Challenge 4 : Infinite Loops

The final challenge is the occurrence of infinite loops, where the simulation process gets trapped in an endless cycle of data exchanges or event handling without advancing in time (Meerbaum-Salant et al., 2011). Understanding these loops in real-world scenarios illustrates their impact on simulation accuracy and stability.

Real-world Scenarios. The first example is the co-simulation framework developed by Miller et al. (2018), which integrates EnergyPlus and CitySim to model energy flows at both urban and building levels, aiming to enhance simulation accuracy by dynamically adjusting each tool's inputs based on the other's outputs. The data exchange at each simulation timestep between these two engines is facilitated using Functional Mock-up Units (FMUs), enabling them to share variables and occupancy levels. FMUs are standard tools used in simulation that allow different models to work together. They package up a model's behaviour and properties, making it easy to connect and share information between different simulation programs (Nouidui et al., 2014). CitySim supplies EnergyPlus with inputs to simulate building energy consumption, while EnergyPlus provides CitySim with data to refine urban-scale calculations. This setup implies a tightly coupled simulation, where each engine's outputs immediately influence the other engine's inputs in the next timestep. This setup aims to enhance overall simulation accuracy, however it also increases the risk of infinite loops.

Given the bidirectional exchange of information, an algebraic loop may emerge if data flow between CitySim and EnergyPlus leads to unresolved circular dependencies. For example, if CitySim calculates urban temperature for EnergyPlus, which then recalculates the building's heating demand and sends it back to CitySim, the simulation might endlessly cycle without advancing in time.

Illegitimacy could arise in this co-simulation setup due to the way data is exchanged between CitySim and EnergyPlus using FMUs. If FMUs improperly manage data exchanges, it will lead the orchestrator to repeatedly process events with the same timestamp. This could trap the simulation in an infinite loop of event handling.

Zeno behaviour poses yet another risk in this co-simulation framework. The frequent interactions between EnergyPlus and CitySim could lead to a rapid sequence of events being processed in increasingly short intervals, resulting in the simulation handling more and more events without significant advancement in the overall simulation time.

The second example is the multi-paradigm energy model developed by Hodge et al. (2009), which analyzes the United States' natural gas system by combining agent-based modeling for supply dynamics, system dynamics for demand analysis and a market module for facilitating interactions. The interconnected nature of these components, while innovative, can give rise to similar issues of infinite loops.

In this model, algebraic loops could occur during the continuous feedback exchange between supply and demand modules. For example, the system dynamics model calculates the energy demand, which directly influences the agent-based supply response. If the feedback loop between these modules results in continuous recalculation without stabilising, the simulation could get trapped in a cycle, repeatedly adjusting supply based on demand changes and vice versa, without progressing in simulation time.

The market module operates through a double-blind auction mechanism at each discrete time step. This setup involves processing bids from suppliers and consumers and determining market prices. Illegitimacy could occur if the system repeatedly processes these bids without advancing to the next state, possibly due to unresolved conditions or failure to reach an equilibrium. If the market dynamics depend too heavily on repeated iterations within the same time frame, the orchestrator could become stuck in an infinite loop of event handling, where the simulation remains at the same point in time.

Additionally, the hybrid nature of this model makes it susceptible to zeno behaviour. If interactions between the agent-based supply module and the system dynamics demand model cause state changes to accelerate, the simulation could process an infinite number of events within a finite time period.

Definitions and potential solutions. After examining the case studies, it's important to define the issues of algebraic loops, illegitimacy and zeno behaviour in more detail and discuss potential solutions to these problems.

Algebraic loops occur in continuous-time co-simulation when outputs from one unit immediately affect another's inputs (Gomes et al., 2018). Without fixed-point iteration to resolve these dependencies, the orchestrator cycles through the same data without progressing in time. The paper of S. T. Hansen et al. (2021) propose managing these loops with graph-based techniques and fixed-point iteration. This involves constructing a step operation graph to capture simulation dependencies and using the algorithm of Tarjan (1972) to identify algebraic loops. They distinguish between reactivity loops, which depend on time progression, and feed-through loops, which create immediate circular dependencies (Kübler & Schiehlen, 2000). To resolve these loops, a fixed-point iteration method is employed, iteratively refining initial guesses for input values until convergence is achieved. This iterative process is supported by reduction techniques that temporarily simplify the graph. The approach relies on SUs supporting state restoration, enabling the algorithm to revert to prior states if convergence fails. Implementing fixed-point iteration requires carefully chosen initial input values and iterative adjustments to ensure convergence, as poor starting points can hinder the process (S. Hansen et al., 2022). This method suits scenarios with continuous feedback, where effective convergence is crucial for simulation progress, as noted by Gomes et al. (2018) and S. T. Hansen et al. (2021). However, the need for state restoration and risk of non-convergence suggest further exploration of adaptive algorithms or alternative techniques to improve convergence in complex systems.

Illegitimacy is a common issue in discrete-event co-simulation, occurring when the orchestrator repeatedly processes events with the same timestamp, unable to move to the next state (Gomes et al., 2018). Schenk et al. (2015) address this in the CoSMOS framework with a random order evaluation strategy. Circular dependencies, like when event A depends on B and vice versa, can create a loop that stalls progress. By occasionally changing the processing order, such as handling Event B before Event A, the simulation breaks these cycles, reducing the risk of endless loops. Evaluating components in a random order decreases the likelihood of the orchestrator falling into an endless loop of the same events.

Zeno behaviour involves events occurring at shorter intervals, leading to an infinite number of events within a finite period (Gomes et al., 2018). This type of behaviour is harder to detect since it does not rely on clear cyclic dependencies or fixed timestamps but instead arises from the model's design. Although it can be intentional in some scenarios, a specific solution to address it remains underdeveloped. The concept of superdense time, highlighted in the paper by Cremona et al. (2019) from subsection 3.3, offers a potential solution. Superdense time enables events to occur at the same model time while maintaining clear ordering through microsteps, represented as pairs (t,n). However, limitations exist: it is not supported in FMI for co-simulation version 2.0, restricting its applicability in some hybrid simulations, and its implementation can introduce complexity in managing and representing microsteps. Thus, while superdense time shows promise for mitigating Zeno behavior, practical implementation may face significant challenges, necessitating further research.

Conclusion. Infinite loops, including algebraic loops, illegitimacy, and zeno behaviour, present significant challenges to co-simulation frameworks. Techniques like fixed-point iteration and random order evaluation offer potential solutions but are not always applied in practice, as seen in the case studies of Miller et al. (2018) and Hodge et al. (2009). While the concept of superdense time presents a promising avenue for mitigating Zeno behavior, its lack of support in FMI for co-simulation version 2.0 and the complexity of its implementation highlight the need for more robust mechanisms. Addressing these challenges is crucial to ensure convergence and prevent infinite loops in future co-simulation frameworks.

3.5 Discussion

This section provides a table summarizing the findings of this paper. Different possible solutions to the issues presented in section 3 are outlined, along with an examination of the implications of these results. Subsequently, the limitations of the research are listed, followed by an outline of areas for further investigation.

Results

Table 1: Overview of co-simulation challenges and potential solutions

Challenges	**Issues**	**Possible solutions**
1: Semantic Adaptation	Signal data mismatch.	Apply a super-formalism that is able to transform the output signals in the desired inputs for each connection of sub-models
	Hierarchical nature of some hybrid systems.	Specify an architecture of handling internal communication and managing the order of units interaction. This can be done centrally by applying a structural framework like HLA or FMI, or decentralised by constructing a self-written tool that dictates this ordering.
2: Discontinuity	Time events are triggered at known times. Race conditions might occur.	Adjust macro-steps and subsystem scheduling to improve accuracy and manage timing.
	State events are triggered when a continuous state hits a threshold, making detection harder and requiring constant monitoring. This might happen between macro-steps, delaying communication and increasing numerical errors.	Rollback strategies to revert to older states, or Quantized State Systems (QSS) to discretize state values while keeping time continuous.
	Rapid changes in continuous variables cause inaccurate predictions, causing oscillations.	Reduce step size or reject steps at discontinuities to minimise errors, though this may increase computational effort.
3: The Issue of Time	Hybrid DE co-simulation: CT SUs cannot inherently know when a discrete event will occur.	Fixed time-stepped method, though it may cause accumulating errors over time. Reducing errors by decreasing step-size increases computational load.
		Global event scheduler reduces accumulating errors but requires event prediction of the CT SU, exposing its inner workings.
	Hybrid CT co-simulation: DE SUs cannot determine when to trigger an event based on continuous signals.	Event prediction based on CT signals. Requires an orchestrator that can adaptively adjust step size, as proposed by (Cremona et al., 2019). Using integer time-representation (Cremona et al., 2019) can help prevent quantization errors.
4: Infinite Loops	Algebraic loops occur when outputs from one unit immediately affect another's inputs, leading to unresolved circular dependencies.	Fixed-point iteration and graph-based techniques to manage dependencies and resolve loops. A step operation graph helps identify loops, and iterative adjustments refine input values until convergence.
	Illegitimacy occurs when the orchestrator repeatedly processes events within the same timestamp, unable to move to the next state.	Random order evaluation strategy to change the processing order of events.
	Zeno behaviour involves an infinite number of events occurring in a finite time due to design choices.	Use superdense time model to handle simultaneous event, though not supported in FMI-CS 2.0.

Several connections can be observed between the proposed solutions across different challenges in co-simulation, highlighting the interconnectedness of these issues.

- **Reducing step size** to manage rapid changes in continuous variables can effectively minimize oscillations and noise. However, this approach may also lead to zeno behavior, as the simulation might encounter an infinite number of changes within a very short interval. Consequently, while reducing the step size can be beneficial, it often necessitates the use of superdense time, which provides a structured framework for managing events and prevents simultaneous occurrences.
- **QQS.** Both the discontinuity and time synchronization challenges can be addressed by predicting when a SU will reach a specific value. In the context of challenge 2, this is accomplished using Quantized State Systems (QSS), which discretize state changes while maintaining continuous time. In challenge 3, the methodology proposed by Cremona et al. (2019) incorporates a similar predictive strategy.

Limitations This research has several limitations:

1. Due to time constraints, it was not possible to examine all the relevant papers published on this topic. Consequently, the recommendations and conclusions are based solely on the findings presented, which means that some potential solutions or issues may have been overlooked.
2. Terms related to this topic (i.e. co-simulation, hybrid co-simulation, multi-paradigm simulation, etc.) are often used interchangeably and not all authors provide a clear definition of what they mean when using these terms. For instance, the paper by Cremona et al. (2019) addresses co-simulation but does not define the term, leading to ambiguity in the literature.
3. This paper specifically examines the challenges of co-simulation identified by Gomes et al. (2018) and does not explore other potential challenges that could arise.

Further research For further research it is recommended to focus on examining practical applications of different theoretical solutions. Case studies often do not mention the challenges and issues they face on a more fundamental level. This omission raises questions about whether the author of the paper did not encounter these issues or failed to mention them, which creates a skewed image of the issues. Additionally, expanding the research to include more literature can contribute to a more comprehensive understanding of the co-simulation landscape and the associated challenges.

3.6 Conclusion

The aim of this paper was to explore the solution space for hybrid co-simulation, addressing the challenges and implementation strategies for integrating different modeling paradigms using co-simulation. Therefore this study has examined key issues in hybrid co-simulation and outlined potential solutions, while also evaluating how these solutions might apply across different challenges.

The findings reveal that multiple solutions exist for various challenges, with significant areas of commonality among them. However, differences in terminology and specific implementation details often obscure these shared aspects, making it difficult to identify and leverage the full range of solutions. While these results offer a valuable framework for hybrid co-simulation developers to select suitable strategies for their projects, the effectiveness of these solutions ultimately depends on their application in real-world scenarios.

Future research should expand this framework by exploring a broader range of solutions and validating their effectiveness in practical applications. Even more importantly, a shift in the research approach is needed—acknowledging that challenges in hybrid co-simulation do exist, instead of presenting simulations as flawless or issue-free. A more open discussion about these difficulties could encourage the development of targeted solutions, filling the current gap in research on how to address these complex problems effectively. Standardising terminology and creating a unified language for hybrid co-simulation might also make it easier to compare different methodologies and integrate advances in the field. Encouraging transparency about these challenges can pave the way for innovative solutions, ultimately leading to the development of more robust and reliable hybrid simulation frameworks.

References

Busch, M. (2019). Performance improvement of explicit co-simulation methods through continuous extrapolation. In B. Schweizer (Ed.), *Iutam symposium on solver-coupling and co-simulation* (pp. 57–80). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-14883-6_4

Cellier, F. E., & Kofman, E. (2006). Simulation of discontinuous systems. In *Continuous system simulation* (pp. 397–478). Springer. https://doi.org/10.1007/0-387-30260-3_9

Cellier, F. E., Kofman, E., Migoni, G., & Bortolotto, M. (2008). Quantized state system simulation. *Proceedings. GCMS'08, Grand Challenges in Modeling and Simulation*, 504–510. https://people.inf.ethz.ch/~fcellier/Pubs/DEVS/scsc_08.pdf

Cremona, F., Lohstroh, M., Broman, D., Lee, E. A., Masin, M., & Tripakis, S. (2019). Hybrid co-simulation: It's about time. In A. Wasowski (Ed.), *Acm/ieee 21th international conference on model driven engineering languages and systems (models '18)* (pp. 1655–1679). Springer. https://doi.org/https://doi.org/10.1145/3239372.3242896

Dahmann, J. S., Fujimoto, R. M., & Weatherly, R. M. (1997). The department of defense high level architecture. *Proceedings of the 29th Conference on Winter Simulation*, 142–149. https://doi.org/10.1145/268437.268465

Dejaco, D., & Benedikt, M. (2017). A novel approach for handling discontinuities in non-iterative co-simulation. In F. D. Rango & T. Ören (Eds.), *Proceedings of the 7th international conference on simulation and modeling methodologies, technologies and applications (simultech 2017)* (pp. 288–295). Springer. https://doi.org/10.5220/0006440402880295

Denil, J., Meyers, B., De Meulenaere, P., & Vangheluwe, H. (2015, January). Explicit semantic adaptation of hybrid formalisms for fmi co-simulation. In F. Barros & M. H. Wang (Eds.), *Proceedings of the symposium on theory of modeling simulation: Devs integrative ms symposium* (pp. 99–106, Vol. 47).

Gheorghe, L. (2009, August). *Continuous/discrete co-simulation interfaces from formalization to implementation* [Doctoral dissertation, École Polytechnique de Montréal]. https://publications.polymtl.ca/137/

Godfrey, T., Mullen, S., Griffith, D. W., Golmie, N., Dugan, R. C., & Rodine, C. (2010). Modeling smart grid applications with co-simulation. *2010 First IEEE International Conference on Smart Grid Communications*, 291–296. https://doi.org/10.1109/SMARTGRID.2010.5622057

Gomes, C., Meyers, B., Denil, J., Thule, C., Lausdahl, K., Vangheluwe, H., & De Meulenaere, P. (2019). Semantic adaptation for fmi co-simulation with hierarchical simulators. *Simulation*, *95*(3), 241–269.

Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2018). Co-simulation: A survey. *ACM Computing Surveys*, *51*(3), 1–33. https://doi.org/10.1145/3179993

Hansen, S., Thule, C., Gomes, C., Pol, J., Palmieri, M., Inci, E., Madsen, F., Alfonso, J., Castellanos, J., & Rodriguez-Fortun, J.-M. (2022). Verification and synthesis of co-simulation algorithms subject to algebraic loops and adaptive steps. *International Journal on Software Tools for Technology Transfer*, *24*. https://doi.org/10.1007/s10009-022-00686-8

Hansen, S. T., Gomes, C., Larsen, P. G., & Van de Pol, J. (2021). Synthesizing co-simulation algorithms with step negotiation and algebraic loop handling. *2021 Annual Modeling and Simulation Conference (ANNSIM)*, 1–12. https://doi.org/10.23919/ANNSIM52504.2021.9552073

Heinzl, B., Raich, P., Preyser, F., & Kastner, W. (2018). Simulation-based assessment of energy efficiency in industry: Comparison of hybrid simulation approaches. In F. Breitenecker, W. Kemmetmüller, A. Körner, A. Kugi, & I. Troch (Eds.), *9th vienna international conference on mathematical modelling* (pp. 689–694, Vol. 51). Springer. https://doi.org/https://doi.org/10.1016/j.ifacol.2018.03.117

Hodge, B.-M. S., Pekny, J. F., & Reklaitis, G. V. (2009). A multi-paradigm energy model for liquid natural gas analysis. In H. E. Alfadala, G. Rex Reklaitis, & M. M. El-Halwagi (Eds.), *Proceedings of the 1st annual gas* *processing symposium* (pp. 1–9, Vol. 1). Elsevier. https://doi.org/https://doi.org/10.1016/B978-0-444-53292-3.50004-3

Hopkinson, K., Wang, X., Giovanini, R., Thorp, J., Birman, K., & Coury, D. (2006). Epochs: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. *IEEE Transactions on Power Systems*, *21*(2), 548–558. https://doi.org/10.1109/TPWRS.2006.873129

Kübler, R., & Schiehlen, W. (2000). Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*, *6*(2), 93–113. https://doi.org/10.1076/1387-3954(200006)6:2;1-M;FT093

Li, W., & Zhang, X. (2014). Simulation of the smart grid communications: Challenges, techniques, and future trends. *Computers Electrical Engineering*, *40*(1), 270–288. https://doi.org/https://doi.org/10.1016/j.compeleceng.2013.11.022

Liboni, G., & Deantoni, J. (2021). A semantic-aware, accurate and efficient api for (co-)simulation of cps. In L. Cleophas & M. Massink (Eds.), *Software engineering and formal methods. sefm 2020 collocated workshops* (pp. 280–294). Springer International Publishing.

Lin, H., Sambamoorthy, S., Shukla, S., Thorp, J., & Mili, L. (2011). Power system and communication network co-simulation for smart grid applications. *Innovative Smart Grid Technologies (ISGT) 2011, IEEE PES*, 1–6. https://doi.org/10.1109/ISGT.2011.5759166

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). Habits of programming in scratch. *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 168–172. https://doi.org/10.1145/1999747.1999796

Mets, K., Ojea, J. A., & Develder, C. (2014). Combining power and communication network simulation for cost-effective smart grid analysis. *IEEE Communications Surveys Tutorials*, *16*(3), 1771–1796. https://doi.org/10.1109/SURV.2014.021414.00116

Miller, C., Thomas, D., Kämpf, J., & Schlueter, A. (2018). Urban and building multiscale co-simulation: Case study implementations on two university campuses. *Journal of Building Performance Simulation*, *11*(3), 309–321. https://doi.org/10.1080/19401493.2017.1354070

Nadeem, T. B., Siddiqui, M., Khalid, M., & Asif, M. (2023). Distributed energy systems: A review of classification, technologies, applications, and policies. *Energy Strategy Reviews*, *48*, 101096. https://doi.org/https://doi.org/10.1016/j.esr.2023.101096

Nouidui, T., Wetter, M., & Zuo, W. (2014). Functional mock-up unit for co-simulation import in energyplus. *Journal of Building Performance Simulation*, *7*. https://doi.org/10.1080/19401493.2013.808265

Oberst, C. A., Schmitz, H., & Madlener, R. (2019). Are prosumer households that much different? evidence from stated residential energy consumption in germany. *Ecological Economics*, *158*, 101–115. https://doi.org/https://doi.org/10.1016/j.ecolecon.2018.12.014

Schenk, T., Gilg, A. B., Mühlbauer, M., Rosen, R., & Wehrstedt, J. C. (2015). Architecture for modeling and simulation of technical systems along their lifecycle. *Computational Visual Science*, *17*, 167–183. https://doi.org/10.1007/s00791-015-0256-9

Schweiger, G., Gomes, C., Engel, G. P., Schoeggl, J., Posch, A., Hafner, I., & Nouidu, T. (2019). An empirical survey on co-simulation: Promising standards, challenges and research needs. *CoRR*, *abs/1901.06262*. http://arxiv.org/abs/1901.06262

Steinbrink, C., van der Meer, A. A., Cvetkovic, M., Babazadeh, D., Rohjans, S., Palensky, P., & Lehnhoff, S. (2018). Smart grid co-simulation with mosaik and hla: A comparison study. *Computer Science-Research and Development*, *33*, 135–143.

Stettinger, G., Benedikt, M., Thek, N., & Zehetner, J. (2013). On the difficulties of real-time co-simulation. In S. Idensohn, M. Papadrakakis, & B. Schrefler (Eds.), *5th international conference on computational methods for coupled problems in science and engineering* (pp. 1–11). http://hdl.handle.net/2117/192671

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, *1*(2), 146–160. https://doi.org/10.1137/0201010

Vangheluwe, H., Lara, J., & Mosterman, P. (2002). An introduction to multi-paradigm modelling and simulation. *Proceedings of the AIS'2002 Conference*.

Zeigler, B. P. (2006). Embedding dev & dess in devs : Characteristic behaviors of hybrid models. *Simulation Series*, 125–132. https://api.semanticscholar.org/CorpusID:15201287