IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, August 2026
Impact Factor: 8.2

Model-Based Co-Simulation Approaches for Networked Control Systems

Sana A Bellary[1] and Dr. Nikhil Janardan Rathod[2]
[1]Research Scholar, [2]Associate Professor
Department of Electronics Engineering
Vikrant University, Gwalior M.P

Abstract: *Networked Control Systems integrate physical processes, controllers, sensors, actuators, and communication networks into a unified feedback architecture. Unlike conventional control systems, the performance of an NCS is strongly influenced by communication-related factors such as transmission delay, packet loss, jitter, bandwidth limitations, network congestion, scheduling, and topology changes. Consequently, evaluating an NCS only through conventional control-system simulation may not accurately represent its real operational behaviour.*
Model-based co-simulation has emerged as an effective approach for addressing this limitation by enabling independently developed models of physical plants, controllers, communication networks, and computing platforms to interact within a coordinated simulation environment. Early NCS co-simulation research demonstrated the importance of jointly simulating physical and communication dynamics, while later studies developed integrated platforms based on environments such as MATLAB/Simulink, System, True-Time, network simulators, and real-time simulation frameworks.

Keywords: Networked Control Systems, Network Simulation, Control Systems, FMI, Synchronization, MATLAB/Simulink, Communication Networks, Co-Design.

I. INTRODUCTION

Networked Control Systems have become an important research area because modern automation increasingly relies on communication networks to connect sensors, controllers, actuators, and physical plants. In a conventional feedback-control system, the controller and plant may be located within the same computational environment, making the communication path comparatively simple. In an NCS, however, feedback information is transmitted through a communication network, and the network itself becomes an important component of the control loop. Delays, packet losses, scheduling conflicts, variable transmission times, limited bandwidth, and network congestion can therefore affect stability, transient response, tracking accuracy, robustness, and overall system reliability. The interaction between control dynamics and communication behaviour makes NCS design considerably more complicated than conventional control-system design. Early work on NCS co-design recognized that communication and control should be studied jointly rather than independently, particularly when packet-level network behaviour influences plant dynamics.
Model-based co-simulation addresses this challenge by allowing different models and simulation environments to operate together while exchanging information during execution. Instead of forcing the complete NCS into a single simulator, the physical plant may be represented in MATLAB/Simulink or Modelica, while communication behaviour may be represented using SystemC, OMNeT++, ns-3, or another network simulator. The controller can then be represented either within the control simulator or as an independent executable component. Such an approach allows researchers to investigate how communication characteristics affect the closed-loop performance of the physical system. A major overview of NCS co-simulation platforms emphasized that the principal benefit of co-simulation is the ability to expose the coupled behaviour of control and communication systems, including the effects of topology, routing, network traffic, and communication protocols on control performance.

DOI: 10.48175/568

II. CONCEPT OF MODEL-BASED CO-SIMULATION

Model-based co-simulation can be defined as the coordinated execution of two or more independently developed simulation models that exchange data during simulation. In an NCS, the major models generally include the physical plant, controller, sensors and actuators, communication network, and computational resources. Each model may use a different mathematical representation and numerical solver. The co-simulation framework acts as an integration mechanism through which information is exchanged and simulation time is coordinated.

The fundamental advantage of this approach is modularity. A detailed network model does not need to be recreated inside a control-oriented simulator, and a sophisticated nonlinear plant does not need to be simplified merely to make it compatible with a network simulator. Instead, specialized simulators can be coupled according to the requirements of the research problem. For example, a plant model may calculate physical states using continuous-time differential equations, while the network simulator models packet transmission using discrete-event simulation. The controller receives measurements after the communication delay and generates control commands that are subsequently transmitted toward the actuator. This creates a more realistic representation of the closed-loop system.

The Functional Mock-up Interface provides an important standardized mechanism for model interoperability. FMI defines Functional Mock-up Units (FMUs) that package models and provide standardized interfaces for their execution. FMI supports Model Exchange, Co-Simulation, and Scheduled Execution; in Co-Simulation, an FMU can contain its own solver and participate in a larger simulation orchestrated by an importing environment.

III. ARCHITECTURE OF CO-SIMULATION FOR NETWORKED CONTROL SYSTEMS

A typical model-based NCS co-simulation architecture consists of five major components: the physical plant model, sensor and actuator models, controller model, communication-network model, and co-simulation coordinator. The physical plant represents the process being controlled, such as a robotic system, autonomous vehicle, industrial process, power converter, or manufacturing system. Sensors generate measurements from the plant, and these measurements are transmitted to the controller through a communication network. The controller processes the received information and generates control commands. These commands are then transmitted through the network to the actuators, which influence the physical plant.

The co-simulation coordinator manages data exchange and time progression among the participating models. This role is particularly important because the participating simulators may operate with different time steps, event mechanisms, numerical solvers, and notions of simulation time. A continuous plant simulator may advance using fixed or adaptive integration steps, whereas a network simulator may advance from one communication event to another. Therefore, synchronization is one of the most important technical issues in NCS co-simulation. Previous research has specifically identified synchronization as a key factor determining the correctness and usefulness of integrated co-simulation.

IV. MAJOR MODEL-BASED CO-SIMULATION APPROACHES

Several approaches have been developed for integrating physical and communication models in NCS research. One important approach couples MATLAB/Simulink with a network simulator. MATLAB/Simulink provides extensive control-design and plant-modelling capabilities, while a separate network simulator provides detailed communication modelling. System/Matlab co-simulation is another example in which System represents networking aspects and MATLAB/Simulink represents control and physical-system aspects. The two environments operate as separate processes and exchange information through an inter-process communication mechanism.

Another approach is based on specialized NCS simulation platforms such as True Time. These platforms attempt to integrate control algorithms, task scheduling, communication networks, and physical dynamics into a common environment. Their advantage is that the researcher does not always need to construct a communication interface between two completely independent simulators. However, integrated platforms can sometimes provide less flexibility than loosely coupled approaches when specialized models or external simulation tools are required.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, August 2026
Impact Factor: 8.2

A third approach uses standardized model exchange mechanisms such as FMI. The use of FMUs permits models developed in different environments to be exported and incorporated into a common co-simulation framework. FMI 3.0 introduced additional capabilities intended to improve sophisticated co-simulation, including early return, intermediate updates, event modes, clocks, and clocked variables. These capabilities are particularly relevant to networked control applications because communication events do not necessarily occur at uniform intervals.

Table 1. Comparison of Major Model-Based Co-Simulation Approaches

Co-Simulation Approach	Main Models	Major Strength	Main Limitation	Typical Application
MATLAB/Simulink + Network Simulator	Plant, controller, network	Strong control modelling with detailed network representation	Interface and synchronization complexity	NCS and robotics
MATLAB/Simulink + SystemC	Control/plant + communication	Supports control and communication co-design	Inter-process communication overhead	Embedded and networked control
TrueTime-Based Approach	Control, tasks, network, plant	Integrated NCS modelling	Tool-specific modelling constraints	Wireless and embedded control
FMI-Based Co-Simulation	FMUs from different tools	Interoperability and modularity	Synchronization and FMU compatibility issues	Multi-domain CPS
Network Simulator + External Plant	Network + physical dynamics	Detailed packet-level analysis	Data exchange overhead	Industrial and communication-sensitive NCS
Real-Time Co-Simulation	Physical model + network/control hardware	High realism and hardware interaction	Higher cost and complexity	Validation and HIL testing

V. SYNCHRONIZATION IN NCS CO-SIMULATION

Synchronization is central to model-based co-simulation because the individual simulators may use different time advancement mechanisms. In a simple fixed-step method, each simulator advances by a predefined time interval and exchanges data at synchronization points. This approach is comparatively straightforward but can be computationally inefficient when important communication events occur between synchronization points. Smaller step sizes improve temporal resolution but increase computational cost.

Event-based synchronization is more suitable for systems dominated by discrete communication events. The network simulator can advance to the next packet-transmission event while the physical simulator is updated when necessary. However, integrating event-based communication with continuous-time physical dynamics remains challenging. Incorrect synchronization can lead to inaccurate representation of network delay, packet ordering, control inputs, or physical states.

Advanced synchronization mechanisms attempt to balance accuracy and computational efficiency. FMI 3.0 includes features such as early return and intermediate update mechanisms that can support more sophisticated co-simulation algorithms. Thus, synchronization research remains an important component of the development of accurate and efficient NCS co-simulation environments.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, August 2026
Impact Factor: 8.2

VI. COMMUNICATION NETWORK MODELLING

Communication modelling is particularly important because the network is an active component of an NCS rather than merely a data-transfer mechanism. A co-simulation model can incorporate transmission delay, packet loss, jitter, bandwidth restrictions, routing, congestion, scheduling, queueing, and communication protocols. These characteristics can then be related directly to control performance.

For example, a controller may achieve excellent tracking performance under an ideal communication channel but experience oscillations or instability when network delay increases. Similarly, packet loss can result in outdated sensor information reaching the controller, while network congestion can create variable delays. Co-simulation makes it possible to study such effects under controlled experimental conditions. The literature has therefore emphasized that integrated co-simulation is useful for examining how network topology, scheduling, routing, and traffic patterns affect the overall NCS.

VII. APPLICATIONS OF MODEL-BASED CO-SIMULATION

Model-based co-simulation has applications across robotics, automotive systems, smart grids, industrial automation, aerospace systems, intelligent transportation, and the Industrial Internet of Things. In robotic systems, co-simulation can be used to examine the effects of wireless communication delay and packet loss on robot control. In automotive applications, it can support analysis of distributed electronic control systems and communication networks. In smart grids, communication and power-system dynamics can be simulated jointly to investigate protection and control performance.

The same principle is applicable to cyber-physical systems more broadly. Research on collaborative model-based systems engineering has shown that co-modelling and co-simulation can couple diverse discrete-event and continuous-time models and support early analysis before expensive physical prototypes are constructed. This makes co-simulation particularly valuable during the design stage because alternative architectures can be evaluated before hardware implementation.

VIII. ADVANTAGES AND LIMITATIONS

The most important advantage of model-based co-simulation is its ability to represent interactions that are difficult to capture through isolated simulation. It permits detailed physical modelling and detailed communication modelling to be combined. It also promotes modularity, reuse of existing models, early design validation, and comparison of alternative network and control strategies. Furthermore, standardized interfaces such as FMI can improve interoperability among simulation environments.

However, co-simulation also has significant limitations. Synchronization errors can reduce simulation accuracy. Communication between simulators can introduce computational overhead. Different numerical solvers can create compatibility problems, and complex models may require substantial computational resources. Reproducibility is another concern. A broader 2023 taxonomic review of co-simulation research reported that reproducibility remains a significant weakness, with nearly 70% of the reviewed publications lacking reproducibility according to the study's assessment.

Table 2. Advantages and Limitations

Aspect	Advantages	Limitations
Modelling	Supports detailed multi-domain models	Model compatibility may be difficult
Control analysis	Captures network effects on control	Synchronization errors can affect results
Communication analysis	Enables packet-level network modelling	Detailed simulation may be computationally expensive
Interoperability	FMI supports standardized model exchange	Different tools may implement features differently

Validation	Allows early testing before physical implementation	Validation against real hardware may still be required
Scalability	Models can be developed independently	Large numbers of coupled models increase complexity
Reusability	Existing models can be reused	Model version and configuration management are necessary

IX. CONCLUSION

Model-based co-simulation has become an important methodology for the analysis and design of Networked Control Systems because it addresses the limitations of isolated control and communication simulations. NCS performance depends on the interaction between physical dynamics, controller behaviour, computation, and communication networks; therefore, these components need to be studied within an integrated simulation framework. Existing approaches based on MATLAB/Simulink, SystemC, network simulators, specialized NCS platforms, and FMI demonstrate different strategies for achieving this integration. The literature shows that synchronization, data exchange, computational efficiency, interoperability, scalability, and validation remain central challenges. Early co-simulation research established the importance of jointly representing network and physical dynamics, while standardized approaches such as FMI have increasingly supported modular and interoperable model integration. Overall, model-based co-simulation provides a valuable foundation for the development of reliable NCSs and broader cyber-physical systems. Future research should emphasize standardized benchmarks, reproducible experiments, intelligent synchronization, cybersecurity-aware simulation, real-time execution, digital twins, and scalable multi-domain modelling.

REFERENCES

1. Bouchhima, A., Briere, D., Nicolescu, G., Abid, M., & Aboulhamid, E. M. (2008). Co-simulation tools for networked control systems. *Lecture Notes in Computer Science, 4981*, 16–29.
2. Branicky, M. S., Liberatore, V., & Phillips, S. M. (2003). Networked control system co-simulation for co-design. *Proceedings of the American Control Conference*.
3. Fitzgerald, J., Gamble, C., Payne, R., Larsen, P. G., & Basagiannis, S. (2016). Collaborative model-based systems engineering for cyber-physical systems. *INCOSE International Symposium*.
4. Frontiers in Energy Research. (2021). Review of modeling and simulation methods for cyber physical power systems.
5. Modelica Association. (2022). *Functional Mock-up Interface Specification, Version 3.0*.
6. Özbek, N. S. (2020). Co-design approach and co-simulation tools for networked cyber-physical control systems. *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies*.
7. Quaglia, D., Muradore, R., Bragantini, A., & Fiorini, P. (2012). A SystemC/Matlab co-simulation tool for networked control systems.
8. Shah, S., et al. (2014). Co-simulation platforms for co-design of networked control systems: An overview. *Control Engineering Practice, 23*, 44–56.
9. *Simulation Modelling Practice and Theory*. (2023). Co-simulation for buildings and smart energy systems—A taxonomic review. *126*, 102770.