Vol. 20 (2026): Proceedings of 2026 International Conference on Materials Science, Energy Engineering and Automation Control (ICMSEAC 2026)
E-ISSN: 3105-0522 P-ISSN: 3105-0514
DOI: https://doi.org/10.70267/mseac.20260713

Modeling and Fault Analysis of Complex Mechatronic Systems Based on MBSE

Sichen Yao*

Department of Nanjing University of Aeronautics and Astronautics, Nanjing, China

**Corresponding author: Sichen Yao.*

Abstract

As high-end equipment evolves towards higher complexity and multi-disciplinary coupling, the interaction among mechanical, hydraulic, and control systems becomes increasingly close. The traditional document-based systems engineering (DBSE) method often encounters bottlenecks, such as fragmented information, poor traceability, and a disconnect between design and analysis, when dealing with the “logical-physical” coupling relationships of complex systems. Therefore, a new methodology capable of uniformly expressing the characteristics of multiple domains is urgently needed. This paper aims to systematically review the research progress in model-based systems engineering (MBSE) for modeling and fault analysis of complex mechanical-electro-hydraulic systems, summarize the current application status of MBSE in overcoming disciplinary barriers and achieving full life-cycle management, and provide references for engineering practice in related fields. Based on mainstream MBSE design processes such as MOFLP-R, and by integrating relevant domestic and foreign literature, this paper has conducted an induction and review from three dimensions: theoretical framework, modeling technology, and application cases. The focus is on analyzing the integration mechanism of multi-domain physical modeling languages such as SysML and Modelica, as well as the application models of MBSE in typical industries such as aviation and rail transit. Existing studies have shown that MBSE effectively achieves closed-loop traceability from requirements to physical realization by establishing a unified system architecture model. In fault analysis, model-based fault injection and joint simulation technologies have been proven to improve early fault prediction capabilities and significantly reduce R&D costs. However, current research still lacks interoperability standards for models, the development of interdisciplinary talent teams, and intelligent modeling empowered by AI. MBSE is an effective paradigm for solving the design and verification problems of complex mechanical-electro-hydraulic systems. Future research should focus on deep integration of MBSE with digital twins and artificial intelligence, and on the construction of standardized collaborative environments to promote the digital transformation of high-end equipment development models.

Keywords

model-based systems engineering (MBSE), review, fault analysis, mechanical-electro-hydraulic system, multi-domain modeling

1. Introduction

Modern high-end equipment, such as aerospace vehicles and complex industrial robots, has evolved into deeply integrated mechatronic systems where mechanical, hydraulic, and control disciplines are inseparable. However, traditional Document-Based Systems Engineering (DBSE) is struggling to cope with this complexity. As highlighted in existing research, the primary bottleneck lies in the "disconnection" between design and analysis [1]. In DBSE, design requirements, system architecture, and physical models often exist in silos, leading to high interdisciplinary communication costs and a lack of traceability. Furthermore, conventional fault analysis methods (such as FTA and FMEA) are predominantly reliant on post-hoc empirical data or physical testing. This reactive approach fails to capture the dynamic coupling faults—particularly the interaction between software logic and physical hardware—that are inherent in complex systems [2]. To transcend these limitations, Model-Based Systems Engineering (MBSE) has emerged as a paradigm shift. By utilizing formal modeling languages like SysML, MBSE transforms the workflow from a "document flow" to a "model flow." Crucially, by integrating multi-domain physical simulation technologies (e.g., Modelica), MBSE enables the establishment of a "logic-physical" closed loop, allowing for dynamic verification at the design stage [3]. This research focuses on the MBSE modeling and fault analysis of typical mechatronic objects to address the chronic "design-analysis disconnect" plaguing traditional methodologies. Methodologically, this study proposes a unified modeling framework that formally expresses mechanical structures, hydraulic power units, and control logic within a single architectural model. By correlating FMEA data with the system architecture, this approach achieves a structured expression of fault knowledge and facilitates multi-disciplinary collaboration, overcoming the limitations of isolated disciplinary models [4]. In engineering applications, integrating fault injection techniques with joint simulation enables precise identification of logical conflicts and physical failure modes during the early design phase. This proactive validation significantly reduces the reliance on costly physical prototypes, thereby shortening R&D cycles and cutting costs. By providing a quantitative analysis method for system reliability, this research offers direct technical support for the design of safer and more reliable complex electromechanical systems.

2. Fundamental Principles of MBSE Modeling and Fault Analysis

2.1 MBSE Structure

Building on the MOFLP-R (Mission-Operation-Function-Logic-Physical-Reliability) design flow, a full life-cycle management framework has been established, spanning from stakeholder needs to the implementation of physical components. Based on the research in [5], the researchers formalized the V model definition using OPM (Object-Process Methodology), thereby resolving the disconnect between MBSE and MBD. Also, the dynamic landing load constraint at the top level is decomposed step-by-step into functional requirements. In line with the practice outlined in [6], it is emphasized that an executable model should be introduced at the conceptual design stage. By establishing a mapping from the task scenario (Mission) to the operation mode (Operation), it ensures that parameter changes at the physical layer are promptly reflected in the requirement verification at the logical layer, ultimately achieving a true "digital twin" development.

To address the multidisciplinary coupling characteristics of the mechanical-electrical system, the Heterogeneous Model Integration Strategy has been utilized, with a specific focus on the Functional Mock-up Interface (FMI) standard. To clarify the interface relationships between components, the block definition diagram (BDD) and the internal block diagram (IBD) of SysML have been used to provide a standardized description of structure, behavior, and requirements in traditional MBSE [2]. For the actual simulation execution, the integration relies heavily on the exchange of Functional Mock-up Units (FMUs). Specifically, control logic and discrete events (typically modeled in Simulink) are integrated with multi-body dynamics and physical characteristics (modeled in Modelica) via the FMI standard [5, 6]. This approach addresses interoperability issues that SysML alone cannot.

Furthermore, this study delves into the challenges of co-simulation using FMI/FMU. While FMI provides a standardized interface, the efficiency of the co-simulation is heavily dependent on the data exchange step size and the model order reduction of complex physical components. In complex scenarios such as landing gear control, improper synchronization of FMUs can lead to numerical instability. Therefore, this research emphasizes optimizing the co-simulation configuration to ensure the digital twin accurately reflects real-time interactions between the control algorithm and the physical dynamics, thereby enhancing simulation fidelity.

2.2 Fault Analysis Mechanism

Unlike traditional fault analysis, independence lies in the design model; this research proposes to integrate FMEA into the system model architecture. Using the parametric method introduced in document [3], fault modes (“Hydraulic leakage” and “Sensor drift”) are defined in the model as a special “object” or “process”, instead of only text Annotation. Based on the system architecture diagram, a propagation path from the physical-layer failure source to the logical-layer functional failure is established. For example, map the failure of the damping mechanism in the physical model to the “insufficient landing impact absorption” functional anomaly in the logical model, thereby structuring and visualizing fault knowledge.

From the other side, traditional physical destructive tests can be replaced by building a “digital prototype” for active fault testing. The first way is to add typical fault models to the virtual model, combining the executable modeling described in document [6], which not only injects logic faults but also includes physical parameter drift (variation of the gas adiabatic index and changes in oil viscosity). The second way is response analysis. By co-simulation and analysis, the dynamic response characteristics of systems under abnormal conditions are examined. For example, simulating the free-fall impact during the emergency lowering of the landing gear, or the impact of minor hydraulic system leaks on the reliability of long-duration missions, and then identifying the weakness at an early stage of design [2].

2.3 Comparison with Conventional Methods

Traditional Design Methods have many limitations; the design of traditional planes and complex equipment always relies on DBSE. However, this approach results in the information being scattered across separate requirement documents, design specifications, and physical test reports. It depends on the transmission of manual experience, which lacks quantitative methods for correlation analysis. Fault diagnosis is usually used after manufacturing and testing of a physical prototype. Once a design flaw is discovered, the repair cost will be high, and tracing the source of the requirements will be difficult. Compared with the traditional approach, MBSE-based Analysis offers significant benefits. It breaks down disciplinary barriers and enables synchronous iteration between logical design and physical simulation. Designers can adjust the control logic and physical parameters within the same model simultaneously. Through executable models and parametric physical simulations [3], quantitative fault prediction and performance verification can be supported in the conceptual stage. This not only significantly reduces the risks of research and development and physical test costs, but also ensures the consistency and reliability of the product from “requirement” to “operation and maintenance” through two-way traceability throughout the entire life cycle [4].

3. Applications in Mechanical and Hydraulic Systems

3.1 Application of Aeronautic and Electrical Systems: Landing Gear and Hydraulic Control

This example focuses on highly dynamic, highly coupled avionics subsystems and demonstrates the advantages of MBSE for handling collaboration between complex physical fields and control logic. Based on research on “dynamic landing load constraints” [6], this research integrates a physical Modelica model into a SysML architecture model. It performs a multi-body dynamic simulation of the landing gear, precisely calculating stress concentrations at key hinge points and fatigue life. Breaking top-level task requirements (such as landing speed) into the structural parameters of the physical layer enables closed-loop verification from “requirements” to “structure”. This solved the problem of “toll interoperability deficiency” [1] and verified the superiority of the complex mechanical structure design.

Otherwise, for the high-reliability hydraulic system of civil aircraft, the SysML State Machine Diagram is used to describe the start-stop and emergency logic of the hydraulic pump. Based on the ideas about “system model exploration technology” [7], logical deadlocks and conflicts were identified through simulation. Establishing a fault injection model (such as pipeline rupture and sensor drift) to simulate the dynamic response of the system under abnormal conditions. Combined with the method of “validation of satellite energy management software requirements” [8], the system's emergency response capability under fault modes was verified, ensuring the robustness of the control logic.

3.2 Application of Urban Rail Transit System: Brake System and Energy Management for Railway Vehicles

The railway vehicle braking system is a typical mechatronic system; its reliability is directly related to traffic safety. Based on the research on “Railway System Structural Modeling Platform Based on MBSE” [9], the research develops a joint simulation model that includes BCU, pneumatic components, and mechanical brake discs. Define the system architecture by using SysML and integrate the physical model with the control model through interfaces. By adding typical fault models (such as “brake cylinder leakage” and “sensor signal loss”), the fault propagation paths in the system were analyzed, and the effectiveness of MBSE in diagnosing faults in railway systems was verified.

Based on research on “Complex Equipment Full Life Cycle Management” [10], the research applies MBSE to an energy management system for railway vehicles. By building the traceability chain from the top-level operational requirements to the specific component design, we ensure the traceability of design changes. Using the method of”Fault Mode Analysis of Relay Satellite Tracking System Based on MBSE” [4], systematically analyzes the fault modes of the energy management system, identifies potential design flaws, and reduces the risks of research and development.

4. Breakthrough Applications in Control Systems

4.1 Limitations of Traditional Control Design

In traditional design processes, control algorithm engineers often rely on simplified mathematical models for offline simulation; however, the mechanical structure is designed independently at the physical level. This makes it difficult to identify the coupling failure between “software logic” and “physical characteristics”. For instance, the timing logic of the control instructions may operate perfectly at the software level. Still, when driving actual physical loads (such as inertial loads or hydraulic dampers), there could be response lag or oscillation. Because of the lack of bidirectional traceability between logic design and physical realization, extensive design rework is required [1].

The traditional “trial-and-error” debugging method relies on the manufacture of physical prototypes and actual machine testing; matching the control logic and physical actuators (such as motors and hydraulic cylinders) often requires multiple rounds of expensive physical debugging. Once it is discovered during later testing that the control strategy does not match the mechanical structure (e.g., a mismatch in bandwidth or nonlinear friction interference), the cost of modifying the physical structure is extremely high, and it is difficult to trace back to the initial requirement definition. In the absence of early behavioral simulation support, the evaluation of factory concepts often lags, making it difficult to identify potential logical conflicts during the requirements phase (RFLP) [11].

4.2 Technical Advantages of MBSE in Control

Based on “A digital twin modeling method for robots integrating MBSE and Modelica”[3], this study constrains a digital twin model for the control system, which jointly simulates the control algorithm (usually described in Simulink or Stateflow) with high-fidelity multi-disciplinary physical models (such as mechanical dynamics and hydraulic characteristics described in Modelica). This technology enables engineers to debug control and mechanical structure parameters simultaneously, enabling early alignment of the control law with the physical characteristics. To demonstrate this capability, we applied this framework to a six-degree-of-freedom industrial robot. We utilized SysML to define the system architecture and interface relationships (e.g., between servo drivers and the robotic arm). By integrating the multi-body dynamics model (including gear transmission nonlinearities and joint flexibility) established in Modelica with the control logic in Stateflow, we conducted joint simulations. The results not only verified the correctness of the trajectory planning algorithm but also quantitatively analyzed the impact of flexible deformation during high-speed movement on control accuracy. This effectively solved the “decoupling” problem described in Section 4.1, achieving collaborative optimization of the control algorithm and the mechanical structure.

The core advantage of MBSE is the support for the preposition of “destructive testing”. By injecting typical faults (such as motor stall, sensor zero drift, and communication packet loss) into the model, the effectiveness of the fault-tolerant control strategy can be verified without a physical prototype. This model-based fault injection (MBFI) method enables designers to assess the system's safety boundaries under extreme conditions. Drawing on the idea of “system model utilization technology” [7], by formally defining fault modes, test cases can be automatically generated to verify the robustness of the control logic in fault states, ensuring the system can maintain basic functions or perform safe shutdown even when a single point of failure occurs. Taking the industrial robot case as an example, we developed a specialized fault-injection model to simulate scenarios such as “failure of position feedback sensors” and “loss of emergency stop button signals”. Simulation analysis indicated that the original control logic was prone to malfunction under abnormal sensor signals. Based on these simulation results, we optimized the “emergency stop logic” and “position feedback abnormal handling mechanism” by introducing redundant verification and safety torque shutdown (STO) strategies. The verification showed that the improved control system could quickly enter a safe state when encountering similar faults. This highlights how MBSE, through early virtual testing, significantly enhances the reliability and safety of complex control systems, addressing the high-cost “trial-and-error” issue mentioned in Section 4.1.

5. Challenges and Future Perspectives

Although MBSE has shown significant advantages in modeling and fault analysis of complex mechanical and electrical systems, it still faces severe challenges, such as model heterogeneity, the lack of standardized specifications, and the discontinuity of the talent pool. At the same time, with the development of artificial intelligence and cloud-native technologies, MBSE is evolving towards an intelligent, collaborative form.

5.1 Technical Challenges

During the process of achieving the “logical-physical” closed loop, maintaining consistency between heterogeneous models (SysML and Modelica/Simulink) is still a technical challenge. Although the missing links in MBSE have been demonstrated [1], due to differences in the internal cores of various tools, there is often semantic loss or precision degradation during data transmission in actual engineering applications. Especially when dealing with systems with high real-time requirements, such as the “flight sequence of launch vehicles” [10], the synchronization mechanism between the discrete-event model of SysML and the continuous-time model of the physical simulator imposes extremely high demands on computing resources and algorithm efficiency. Currently, there is a lack of unified MBSE modeling standards and fault analysis standardization in the industry, making it difficult for models from different teams and projects to be reused and shared. In the discussion of the ground segment engineering framework for ESA (European Space Agency) missions [12], it is noted that although promoting open-source MBSE (model-based systems engineering) frameworks helps standardization, in practical applications, compatibility issues across toolchains still hinder large-scale collaborative development. This “information island” phenomenon often forces enterprises to invest heavily in customized development when implementing MBSE, thereby raising the implementation threshold. MBSE requires engineers not only to understand system engineering but also to be proficient in multiple fields of physics (such as mechanics, hydraulics, and control), modeling languages (such as SysML), and simulation tools. In the literature [4], when analyzing the application of MBSE in the aviation field, it is emphasized that the current engineering education system lacks a mechanism to develop such versatile talents. Engineers need to go through a steep learning curve to master the full process, from requirements analysis to behavior simulation. This “talent gap” has become a key factor restricting the popularization of MBSE in small enterprises.

5.2 Future Directions

In the future, AI technology will deeply integrate with MBSE processes, solving the problem of traditional modeling, which relies on manual experience and is low-efficiency. Specifically, Large Language Models (LLMs) will automate the generation of SysML models from natural language requirements, while Knowledge Graphs (KGs) will ensure rigorous traceability from stakeholder needs to physical components. In combination with the research approach of “Satellite Fault Modeling Analysis” [2], the MBSE platform will use machine learning algorithms to mine massive simulation data in the future, enabling automatic optimization of model parameters and intelligent identification of fault modes. For instance, using deep learning algorithms to automatically generate SysML state machine diagrams, or optimizing control strategies through reinforcement learning, thus enabling the transition from "model-based systems engineering" to "intelligent systems engineering". To address distributed team collaboration and computing capacity bottlenecks, a cloud-native MBSE collaborative design environment will become mainstream. Drawing on experience with moving ESA tasks to the cloud [12], future platforms will support multiple teams developing models in different geographic locations simultaneously. They will utilize high-performance computing (HPC) resources in the cloud for large-scale fault-injection simulations. This "cloud + edge" model not only reduces enterprise hardware investment costs but also ensures the consistency and traceability of model data through version control and real-time collaboration. The ultimate goal of MBSE is to build a digital main thread that runs through the entire life cycle of the equipment. Future MBSE models will not be limited to the design stage; instead, they will seamlessly connect with the digital twin during operation and maintenance. As the study on factory concept evaluation [11] indicates, models during the design stage will directly drive predictive maintenance in the operation and maintenance stage. By comparing the design model with actual operational data in real time, the system can dynamically update the fault prediction model, achieving closed-loop feedback from "design verification" to "operation optimization" and truly realizing intelligent full-life-cycle management of the equipment.

6. Conclusion

By deeply integrating the SysML system architecture model with the Modelica/Simulink multi-domain physical model, this article has constructed a unified system model covering mechanics, hydraulics, and control. This approach effectively addresses the problem of model separation in traditional design, enabling the transformation from "document flow" to "model flow". In applications such as aircraft landing gear and railway vehicle braking systems, this framework successfully enabled collaborative simulation of structure, control, and physical properties, verifying its universality and effectiveness in handling complex coupled systems. Otherwise, the fault injection and joint simulation method proposed in this paper breaks through the limitation of traditional FMEA, which relies on later physical testing. By injecting typical faults such as "pipeline rupture" and "sensor failure" into the virtual model, it can accurately identify logical conflicts and physical failures at an early design stage, significantly reducing development risks and costs. The case of the industrial robot control system further demonstrates that this method can effectively optimize the fault-tolerant control strategy and enhance the system's safety and reliability. Applications in both the aviation and railway fields have demonstrated that MBSE is not only a modeling language but also a systematic engineering methodology. By establishing digital threads throughout the entire life cycle, it achieves closed-loop management of requirements, design, simulation, and verification. Although there are still challenges in model interoperability and standardization, with the development of AI-assisted modeling and cloud collaboration technologies, MBSE will undoubtedly become the core engine driving the development of intelligent manufacturing and digital twin technologies, providing strong theoretical support and technical guarantees for the reliability design of high-end equipment.

References

[1] Khandoker, A.; Sint, S.; Gessl, G.; Zeman, K. Toward Demystifying the Missing Links in Model-Based Systems Engineering (MBSE). Systems 2026, 14, 158.

[2] Yan Senhao, Chen Yujun, Yang Qinglong, et al. Fault modeling analysis of relay satellite acquisition and tracking system based on MBSE [J]. Spacecraft Engineering, 2025,34(04):71-77.

[3] Mei Zaiwu, Zhou Jiangman, Zhang Cong, et al. A robot digital twin modeling method integrating MBSE and Modelica [J/OL]. Journal of System Simulation, 1-11 [2026-03-12]. https://link.cnki.netUrlid/11.3092.V.20260213.0950.002.

[4] Yin Wang. Application research of MBSE in the aviation field [J]. China Information Industry, 2026, (01):23-25.

[5] L. Li, N. L. Soskin, A. Jbara, M. Karpel, and D. Dori, "Model-Based Systems Engineering for Aircraft Design With Dynamic Landing Constraints Using Object-Process Methodology," in IEEE Access, vol.

7, pp. 61494-61511, 2019, doi: 10.1109/ACCESS.2019.2915917.keywords: {Aircraft;Atmospheric modeling;Aircraft propulsion;Aircraft manufacture;Load modeling;Model-based systems engineering (MBSE);model-based design (MBD);civil transport aircraft design;dynamic landing constraints;object-process methodology (OPM) ISO 19450}

[6] Epp J. Executable Model-Based Systems Engineering for Aircraft Systems: Landing Gear Extension and Retraction Use Cases[D]. Toronto Metropolitan University, 2022.

[7] Estable, S., Estanguet, R., Cortier, A. *et al.* System model exploitation techniques to foster MBSE adoption at Airbus. *CEAS Space J* (2026). https://doi.org/10.1007/s12567-026-00703-5

[8] Jiang Siyue, Wang Ye, Zheng Ze, et al. Requirements modeling and validation of satellite energy management software using MBSE methodology [J]. Spacecraft Engineering, 2026,35(01):107-114.

[9] Xie Ze, Zhang Zheming, Zhang Kexin, et al. Development and Application of Railway System Architecture Modeling Platform Based on MBSE [J]. Railway Computer Applications, 2026,35(02):28-36.

[10] Liu Yuanpeng, Jin Yili, Zhang Wenfeng. Modeling and Collaborative Research on Launch Vehicle Flight Sequencing Based on MBSE [J]. Aerospace Control, 2026,44(01):46-51. DOI: 10.16804/j.cnki.issn1006-3242.2026.01.008.

[11] Satwan, Philip, Ghanjaoui, Yassine, Biedermann, Jörn, and Nagel, Björn. "Kopplung von MBSE und Verhaltenssimulation für die Auswertung von Fabrikkonzepten: Mit RFLP und ereignisorientierter Simulation zu einer frühen Fabrikkonzeptbewertung" *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 120, no. 12, 2025, pp. 873-878.

[12] Wallum, M., Foley, S., Mody, R. *et al.* Onboarding of ESA missions to the Ground Segment Engineering Framework: an open source MBSE framework for ESA mission and science ground segments. *CEAS Space J* (2026).

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access