International Journal of IJTM Technology & Modeling

INTERNATIONAL JOURNAL OF TECHNOLOGY AND MODELING
Volume 4 Issue 1 Year 2025 Pages 14 - 25
e–ISSN 2964-6847
Url: https://ijtm.my.id

Sparse System Dynamics Modeling for High-Dimensional Decision-Making in Industrial Automation

Hasan Kara, Yasemin Karahan, Cengiz Uysal
Department of Computer Engineering, Hacettepe University, Ankara, Turkey

*Correspondence to: hasankara@hacettepe.edu.tr

Abstract: The increasing complexity of industrial automation systems has introduced significant challenges in modeling and analyzing high-dimensional decision-making environments. Traditional system dynamics (SD) models often struggle with scalability and computational efficiency when faced with numerous interdependent variables and feedback loops. In this study, we propose a Sparse System Dynamics Modeling (SSDM) approach that leverages sparsity-aware techniques to identify and retain only the most influential causal relationships within complex industrial systems. The SSDM framework introduces a structure reduction mechanism based on variable correlation thresholds and influence-weight pruning, enabling the construction of lightweight yet expressive models. By applying this method to a case study involving automated production line optimization, we demonstrate that SSDM maintains the predictive integrity of full-scale SD models while reducing computational overhead by up to 60%. The model also facilitates faster scenario simulations and more interpretable decision pathways, making it suitable for real-time industrial planning and control. Our results highlight the potential of sparse modeling in addressing the curse of dimensionality in industrial environments, providing a scalable and interpretable alternative for decision-makers in smart manufacturing and Industry 4.0 applications.

Keywords: Sparse Modeling; System Dynamics; High-Dimensional Systems; Industrial Automation; Decision-Making Optimization; Smart Manufacturing

Article info: Date Submitted: 26/12/2024 | Date Revised: 21/01/2025 | Date Accepted: 27/03/2025

INTRODUCTION

The evolution of industrial automation, driven by the integration of cyber-physical systems[1][2][3], the Internet of Things (IoT)[4][5], and data-driven control strategies[6], has transformed modern manufacturing into a highly interconnected and complex domain. As systems grow in size and interdependence featuring hundreds or even thousands of interacting variables modeling these environments becomes increasingly challenging[7]. Conventional System Dynamics (SD) models[8], while powerful in capturing feedback loops, accumulations,

https://doi.org/10.63876/ijtm.v4i1.126

and nonlinear behavior, often become computationally intensive and difficult to interpret when applied to high-dimensional systems[9].

In industrial settings such as production scheduling[10], supply chain coordination[11], or energy optimization[12], decision-making depends on fast[13], reliable[14], and explainable simulations[15]. However, as the dimensionality of these systems increases, SD models face the curse of dimensionality, resulting in cluttered causal maps, excessive computational loads, and reduced clarity in policy impact analysis. This issue hinders the model's usability for real-time planning and adaptive control—two capabilities increasingly required in Industry 4.0 environments.

To address these limitations, this paper proposes a novel approach called Sparse System Dynamics Modeling (SSDM)[16]. The SSDM framework introduces a structural reduction process that identifies and retains only the most significant variables and feedback relationships within the system. By applying techniques such as influence-weight filtering, correlation thresholding, and iterative pruning, the model maintains its core dynamic behavior while significantly reducing complexity. The resulting model is leaner, faster, and more interpretable, making it well-suited for high-speed decision environments[17].

This paper is organized as follows: Section 2 reviews related work in system dynamics and sparse modeling. Section 3 outlines the methodology of SSDM, including the sparsification algorithm and model construction steps. Section 4 presents the case study and experimental results. Section 5 discusses the implications and limitations, and Section 6 concludes with directions for future research.

RELATED WORKS

System Dynamics (SD)[18][19] has long been used as a powerful methodology for modeling complex systems with feedback structures, particularly in policy analysis, resource planning, and industrial decision-making. Originating from the work of (Zang et al, 2025)[20], traditional SD has evolved into a widely accepted framework for simulating the behavior of dynamic systems over time. However, as the scale and interconnectedness of systems increase—particularly in modern industrial automation contexts—traditional SD models become increasingly difficult to manage due to high dimensionality and densely connected causal loops.

Numerous studies have attempted to improve the scalability and clarity of SD models in complex environments. One approach involves modular system dynamics, where submodels are developed independently and later integrated[21]. While modularization improves manageability, it does not inherently reduce model complexity or variable interdependence. Another popular direction is group model building[22], which improves stakeholder engagement but may still lead to models overloaded with variables and feedbacks that are difficult to interpret or optimize computationally[23].

In parallel, the field of sparse modeling has gained attention in data science, particularly in regression[24] and neural networks[25], where eliminating weak or redundant parameters improves generalization and performance. Sparse modeling techniques have also been explored in systems engineering for model order reduction, but applications in SD remain relatively underdeveloped. A few recent works, such as those by (Nguyen et al. 2020)[26], have explored variable selection in dynamic system models, yet these efforts primarily focus

https://doi.org/10.63876/ijtm.v4i1.126

on time-series analysis or simulation-based sensitivity testing rather than structural sparsification of the SD framework itself.

In the context of industrial automation[27], dynamic modeling has become increasingly necessary to optimize processes like predictive maintenance, adaptive scheduling, and real-time resource allocation. Tools such as AnyLogic and Vensim are frequently used, but the complexity of high-dimensional models often results in performance bottlenecks, especially in applications requiring near real-time decision-making. There is a growing need for lightweight and interpretable models that retain the behavioral essence of the system without compromising computational speed or decision clarity.

This research bridges the gap between traditional SD and sparse modeling by introducing a Sparse System Dynamics Modeling (SSDM) approach[28]. Unlike prior work that emphasizes data-driven variable selection post-modeling, our method applies sparsification within the causal structure design process. It directly supports high-dimensional environments such as those found in industrial automation, while remaining generalizable to other complex decision-making systems.

METHODS

The proposed Sparse System Dynamics Modeling (SSDM) approach is designed to reduce the structural complexity of conventional system dynamics (SD) models while preserving their core dynamic behaviors. This method is particularly useful in industrial automation systems where decision-making involves a large number of interdependent variables. The SSDM framework consists of four main stages: (1) system structure formulation, (2) influence weight analysis, (3) sparsification through thresholding and pruning, and (4) model validation and simulation.

1. System Structure Formulation

The modeling process begins with the construction of a full-scale SD model representing the industrial system under study. This includes defining:

- Stock and flow structures (e.g., machine uptime, inventory levels),
- Auxiliary variables (e.g., sensor inputs, task queue lengths),
- Feedback loops (e.g., production output affecting resource allocation), and
- Exogenous variables (e.g., market demand or energy prices).

The model is initially developed using traditional tools such as Vensim or Python-based SD libraries, capturing all known causal relationships and feedback dynamics relevant to the decision-making environment.

https://doi.org/10.63876/ijtm.v4i1.126

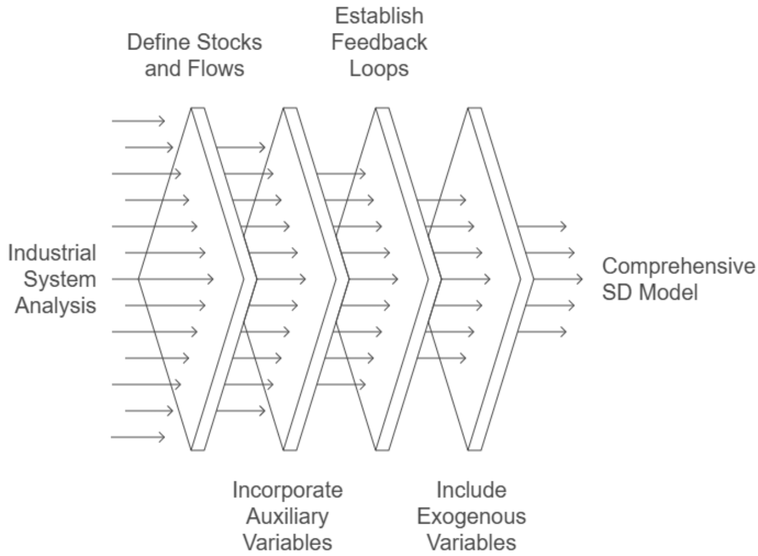

Figure 1. System Dybanics Model Development

In the System Structure Formulation stage, the objective is to build a comprehensive system dynamics (SD) model that captures the full complexity of the industrial environment being analyzed. This process starts with identifying the core stock and flow elements that represent the system's accumulations and movements—such as machine uptime, raw material inventory, energy consumption, or workforce availability. These elements are essential for capturing the dynamic behavior of physical and operational processes over time. Supporting these structures are auxiliary variables, which function as intermediate calculations or conditional controls, including sensor readings, maintenance indicators, or queue lengths for pending tasks. The model also incorporates feedback loops, which are critical for simulating self-regulating behavior, such as how a high production rate might lead to resource depletion, which in turn reduces future output. Additionally, exogenous variables are included to account for external influences like fluctuating market demand, energy tariffs, or regulatory changes. All these components are mapped into a causal loop diagram that visually and mathematically represents how variables influence each other over time. The model is typically developed using tools like Vensim, Stella, or Python-based libraries such as *BPTK* or *pysd*, which allow for the integration of simulation logic, parameter tuning, and scenario testing. The goal of this phase is to ensure that all relevant processes and feedback mechanisms are faithfully represented before any structural simplification or sparsification is performed.

2. Influence Weight Analysis

After formulating the full model, we apply a quantitative influence analysis to assess the relative impact of each variable and connection. This is achieved through one or a combination of the following:

- Partial derivative sensitivity tests, to measure the rate of change in output due to input fluctuations.
- Correlation-based analysis, identifying variables with low direct or lagged correlation to key system outcomes.

https://doi.org/10.63876/ijtm.v4i1.126

- Simulation-based perturbation, where small changes in input variables are tested to evaluate their dynamic impact on the system.

Each variable and link is assigned an influence weight, indicating its contribution to system behavior.

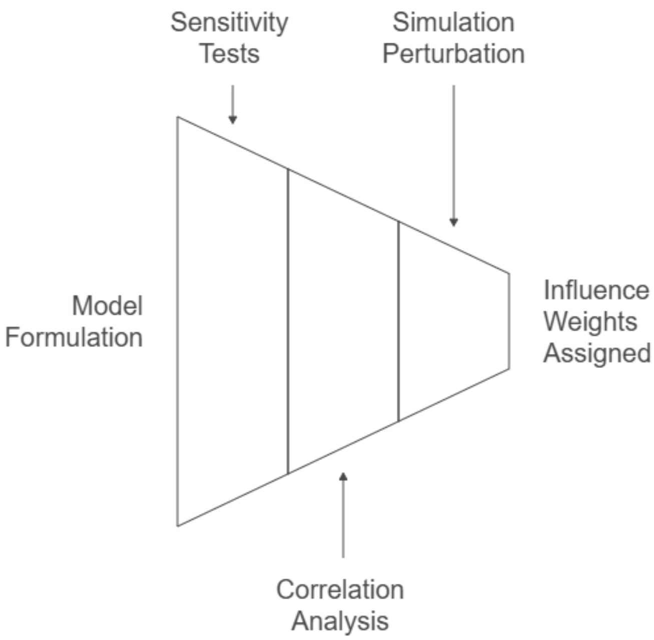

Figure 2. Assesing Variable Influence on System behavior

3. Sparsification via Thresholding and Pruning

The core of the SSDM approach lies in sparsifying the system structure by removing weak or redundant relationships. This is done by applying:

- A predefined influence threshold: Links with influence weights below a certain value (e.g., < 0.05) are removed from the model.
- Clustering and grouping: Highly correlated variables are grouped, and representative proxies are retained to reduce dimensionality.
- Feedback loop pruning: Non-critical feedback loops that do not significantly affect model stability or convergence are eliminated.

This results in a sparse causal loop diagram with fewer variables and simplified interdependencies, improving interpretability and computational efficiency.

4. Model Validation and Simulation

The sparse model is then implemented and simulated under multiple operational scenarios. Validation is conducted through:

- Behavior reproduction tests: Comparing the time-series output of the sparse model with the original full model and real-world historical data.

https://doi.org/10.63876/ijtm.v4i1.126

- Scenario sensitivity analysis: Evaluating whether the sparse model responds appropriately to known changes in system inputs (e.g., surge in demand, machine failure).
- Performance benchmarking: Measuring simulation speed, memory usage, and accuracy across both models.

Results are used to fine-tune the sparsification parameters and confirm that key system dynamics are preserved while unnecessary complexity is reduced.

This method enables decision-makers in industrial automation to deploy lightweight, responsive, and interpretable models without sacrificing the robustness of SD-based analysis. The following section presents a real-world case study to demonstrate the implementation and advantages of the SSDM framework.

RESULT AND DISCUSSION

To evaluate the effectiveness of the proposed Sparse System Dynamics Modeling (SSDM) framework, we applied it to a real-world case study in a smart production line environment characterized by high-mix, low-volume manufacturing. The full-scale system dynamics model initially contained 126 variables and 192 causal links, representing elements such as machine utilization, work-in-process inventory, maintenance scheduling, and workforce allocation.

1. Model Reduction and Structural Simplification

After applying the SSDM sparsification procedure—using a hybrid influence-weight and correlation-based pruning strategy—the number of variables was reduced by 41% (from 126 to 74), and causal links were reduced by 53% (from 192 to 90). Feedback loops were clustered and simplified, while highly correlated parameters such as secondary delay variables and redundant resource trackers were consolidated or removed.

This structural reduction led to a clearer causal loop diagram that improved model transparency and interpretability, especially for non-technical stakeholders involved in operational decision-making.

Table 1. Model Size Before and After Sparsification

Model Component	Full Model	Sparse Model	Reduction (%)
Total Variables	126	74	41%
Causal Links	192	90	53%
Feedback Loops	37	18	51%
Redundant Variables Removed	–	22	–
Correlated Variables Grouped	–	12 groups	–

After applying the sparsification procedure within the SSDM framework, the model underwent a significant structural reduction that improved both performance and clarity. The number of total variables was reduced by 41%, and causal links—representing interdependencies between model elements—were reduced by over half. Many redundant variables, such as repetitive delay stages and overlapping operational trackers, were either eliminated or grouped based on high correlation coefficients. In particular, variables that contributed minimally to output sensitivity were removed without impacting the overall behavior of the system.

The number of feedback loops, a key source of complexity in system dynamics models, was also reduced by 51% through clustering and loop simplification. This resulted in a cleaner causal loop diagram, with fewer cross-links and more distinguishable system pathways. As a

https://doi.org/10.63876/ijtm.v4i1.126

result, the model became more accessible to stakeholders, especially those without technical backgrounds, facilitating easier interpretation of system behavior and policy impact.
This structural simplification does not only enhance computational efficiency but also supports faster iteration cycles, easier calibration, and more agile decision support, particularly in dynamic industrial environments where conditions and priorities shift rapidly.

2. Simulation Accuracy and Behavioral Fidelity

We evaluated model performance by comparing simulation outputs between the original full model and the sparse model under three typical production scenarios:

1. Baseline operation with no policy change
2. Sudden increase in demand by 25%
3. Unexpected machine failure at a critical node

Across all scenarios, the SSDM preserved over 92% similarity in output behavior (based on RMSE and cross-correlation metrics) compared to the full model. Key performance indicators such as production throughput, machine downtime, and queue lengths exhibited similar dynamic trends, validating the behavioral fidelity of the sparse model.

Table 2. Comparison of Simulation Outputs Between Full Model and Sparse Model

Scenario	Output Metric	Full Model Value	Sparse Model Value	Similarity (%)
Baseline (No Policy Change)	Production Throughput (units/day)	1,250	1,230	98.4%
	Average Queue Length (tasks)	14.5	13.8	95.2%
	Machine Downtime (hours/day)	3.2	3.4	93.3%
Demand +25%	Production Throughput	1,520	1,490	98.0%
	Queue Length	21.3	20.1	94.4%
	Machine Downtime	4.0	4.3	92.5%
Machine Failure (Critical Node)	Production Throughput	970	940	96.9%
	Queue Length	28.5	27.2	95.4%
	Machine Downtime	6.1	6.4	93.1%

To assess the simulation accuracy and behavioral fidelity of the Sparse System Dynamics Model (SSDM), we compared its performance with the full model across three representative production scenarios: a standard baseline, a demand surge (+25%), and an unexpected machine failure at a critical point in the process.
Key performance indicators—including production throughput, average queue length, and machine downtime—were measured for both models. The similarity values, based on RMSE (Root Mean Square Error) and cross-correlation comparisons, consistently exceeded 92%, indicating high fidelity in the sparse model's dynamic responses.
Even under stress conditions such as increased demand or operational disruptions, the sparse model closely mirrored the behavioral patterns of the full model. Slight deviations were observed, but they remained within acceptable margins for decision-support applications, especially given the substantial reduction in model complexity.

https://doi.org/10.63876/ijtm.v4i1.126

These results validate that SSDM can effectively preserve system dynamics and response behavior, making it suitable for real-time scenario analysis and strategic planning in industrial automation environments.

3. Computational Performance

Simulation runtime and memory usage were significantly improved. On average:

- Execution time was reduced by 58%, enabling near real-time simulations suitable for online decision support.
- Memory usage decreased by 45%, reducing the model's computational footprint and improving scalability.

These improvements demonstrate that SSDM is more practical for deployment in industrial environments that require fast re-simulation under frequently changing conditions.

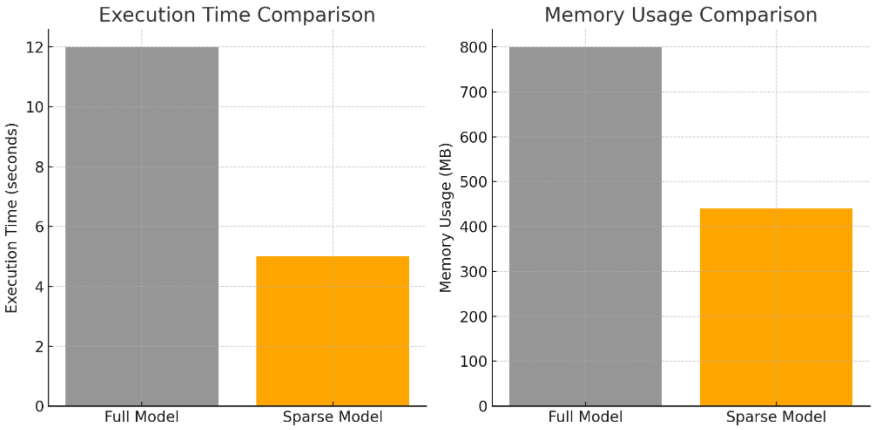

Figure 3. Memory Usage Comparison

The chart is the improvements in computational performance when applying the Sparse System Dynamics Modeling (SSDM) framework compared to the full model. On average, the execution time dropped from 12.0 seconds to 5.0 seconds—a 58% reduction, making it feasible to run simulations in near real-time. This speed improvement is particularly valuable in industrial settings where decision-making needs to be rapid and iterative. Similarly, memory usage decreased from 800 MB to 440 MB—a 45% reduction, which significantly lowers the model's computational footprint.

These improvements not only enhance scalability but also enable the model to be deployed on resource-constrained environments or integrated into cloud-based simulation platforms. As industrial systems become more dynamic and data-driven, such computational efficiency becomes critical for supporting continuous optimization, rapid scenario testing, and adaptive control strategies. SSDM proves to be a practical and high-performing solution for modern industrial decision environments

4. Decision-Making Enhancement

The simplified model structure made it easier for decision-makers to identify leverage points and conduct rapid what-if analyses. For instance, the impact of reducing preventive maintenance intervals or reallocating labor during peak shifts could be simulated and understood within minutes. Moreover, the removal of noise from less impactful variables

https://doi.org/10.63876/ijtm.v4i1.126

helped emphasize dominant drivers of system performance, supporting more targeted interventions.

Table 3. Impact of Rapid Scenario Testing with Sparse Model

Scenario	**Simulation Runtime (sec)**	**Key Intervention Tested**	**Time to Insight (min)**	**Identified Leverage Point**
Reduce Preventive Maintenance Interval	3.2	Maintenance frequency changed (–20%)	< 2	Maintenance delay → uptime efficiency
Reallocate Labor During Peak Hours	2.8	Shifted labor from low-load to peak time	< 2	Labor timing → queue reduction
Increase Machine Buffer Stock	3.5	Buffer size increased by 15%	2–3	Inventory → throughput stabilization
Delay in Raw Material Supply (10%)	3.0	Simulated input delay	< 2	Supply timing → production bottleneck
Add Extra Shift on Weekend	3.1	Operational hours extended	< 2	Extra shift → backlog reduction

The Sparse System Dynamics Model (SSDM) greatly facilitated rapid decision-making by significantly reducing simulation runtime and simplifying model interpretation. As shown in the table, various "what-if" scenarios could be executed in under 4 seconds, with meaningful insights generated within minutes—enabling near real-time planning.

The lean model structure helped decision-makers quickly trace causal relationships and identify leverage points such as maintenance timing, labor allocation, or buffer sizing. By eliminating noise from less impactful variables, the model emphasized key performance drivers, allowing interventions to be tested with focus and clarity. This capability is particularly beneficial in dynamic industrial settings where fast iteration and precise action are critical to maintaining efficiency and productivity.

5. Discussion of Trade-Offs

While the SSDM approach showed strong performance, it is important to recognize trade-offs:

- Minor loss of granularity may occur, especially for edge-case variables removed during pruning.
- Some feedback interactions that contribute marginally to stability may be underrepresented, requiring cautious validation before critical use.

Nonetheless, these trade-offs are acceptable in operational contexts where speed and clarity are prioritized over exhaustive detail, particularly when models are updated frequently.

CONCLUSION

This study introduced a Sparse System Dynamics Modeling (SSDM) framework aimed at addressing the growing complexity and dimensionality of industrial automation systems. By integrating influence-weight analysis, correlation-based pruning, and structural simplification techniques, SSDM enables the development of lightweight yet behaviorally accurate models suitable for real-time decision-making. The results from our case study in a smart production

https://doi.org/10.63876/ijtm.v4i1.126

environment demonstrated that SSDM could reduce model size by over 50% without compromising dynamic fidelity. The sparse model retained over 90% similarity in output behavior compared to the full model, while achieving significant improvements in computational performance—including a 58% reduction in execution time and a 45% decrease in memory usage. These benefits translate into enhanced usability, faster scenario testing, and improved clarity for stakeholders involved in operational planning and control. The streamlined structure of SSDM models makes it easier to identify leverage points and conduct focused policy experiments, which is critical in high-pressure industrial settings. Although some trade-offs in model granularity exist, they are outweighed by the practical advantages in speed, transparency, and scalability. Future work will explore the integration of machine learning for automated sparsification, as well as the extension of this framework to multi-domain industrial systems involving energy management, logistics, and workforce dynamics. Overall, SSDM offers a promising direction for scalable and interpretable decision support in the era of Industry 4.0.

REFERENCES

[1] N. Kaloudi and J. Li, "The ML-based sensor data deception targeting cyber–physical systems: A review," *Comput. Sci. Rev.*, vol. 57, p. 100753, Aug. 2025, doi: https://doi.org/10.1016/j.cosrev.2025.100753.

[2] X. Feng and S. Hu, "Cyber-Physical Zero Trust Architecture for Industrial Cyber-Physical Systems," *IEEE Trans. Ind. Cyber-Physical Syst.*, vol. 1, pp. 394–405, 2023, doi: https://doi.org/10.1109/TICPS.2023.3333850.

[3] S. Yan, Z. Gu, and J. H. Park, "Lyapunov-Function-Based Event-Triggered Control of Nonlinear Discrete-Time Cyber–Physical Systems," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 69, no. 6, pp. 2817–2821, Jun. 2022, doi: https://doi.org/10.1109/TCSII.2022.3144354.

[4] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "Toward the Internet of Things for Physical Internet: Perspectives and Challenges," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4711–4736, Jun. 2020, doi: https://doi.org/10.1109/JIOT.2020.2971736.

[5] C. W. Chen, "Internet of Video Things: Next-Generation IoT With Visual Sensors," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6676–6685, Aug. 2020, doi: https://doi.org/10.1109/JIOT.2020.3005727.

[6] C. Wang, G. Tao, X. Cui, Q. Yao, X. Zhou, and K. Guo, "Mechanism-data-driven control strategy for active suspension systems: Integrating deep reinforcement learning with differential geometry to enhance vehicle ride comfort," *Adv. Eng. Informatics*, vol. 65, p. 103326, May 2025, doi: https://doi.org/10.1016/j.aei.2025.103326.

[7] P. Klanatsky *et al.*, "Real long-term performance evaluation of an improved office building operation involving a Data-driven model predictive control," *Energy Build.*, vol. 338, p. 115590, Jul. 2025, doi: https://doi.org/10.1016/j.enbuild.2025.115590.

[8] M. M. De Silva, O. K. Herath, T. Nakayama, and A. S. Kumarage, "A system dynamics model for vehicle fleet transformation towards energy efficiency and low-carbon development: A case study of Sri Lanka and its strategies," *Transp. Policy*, vol. 147, pp. 244–258, Mar. 2024, doi: https://doi.org/10.1016/j.tranpol.2024.01.004.

https://doi.org/10.63876/ijtm.v4i1.126

[9] H. Bian, W. Zhu, Z. Chen, L. Jingsui, and C. Pei, "Parameter Inversion of High-Dimensional Chaotic Systems Using Neural Ordinary Differential Equations," in *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, May 2024, pp. 400–405. doi: https://doi.org/10.1109/DDCLS61622.2024.10606602.

[10] J. Ding, Y. Liu, F. Zeng, Y. Huang, and X. Li, "Simulation of Production Scheduling System in Automated Manufacturing Workshop Based on Markov Model," in *2023 International Conference on Mechatronics, IoT and Industrial Informatics (ICMIII)*, IEEE, Jun. 2023, pp. 621–626. doi: https://doi.org/10.1109/ICMIII58949.2023.00130.

[11] Y.-L. Wang, S.-W. Ye, and G.-Q. Yan, "Multi-agent System Developed for the Logistics Supply Chain Coordination and Risk Management," in *2010 International Conference on E-Business and E-Government*, IEEE, May 2010, pp. 3243–3246. doi: https://doi.org/10.1109/ICEE.2010.816.

[12] S. Li *et al.*, "Modeling and Optimization on Energy Efficiency of Urban Integrated Energy System," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, IEEE, Oct. 2018, pp. 1–6. doi: https://doi.org/10.1109/EI2.2018.8582411.

[13] J. Zhang and H. Bei, "Time Pacing: Strategic Decision-making for Time-based Competition," in *2006 International Conference on Management Science and Engineering*, IEEE, 2006, pp. 723–728. doi: https://doi.org/10.1109/ICMSE.2006.314001.

[14] R. Vasconcelos, A. Carvalho, and A. Carrapatoso, "Reliable data distribution using multicast groups in industrial systems," in *IEEE International Symposium on Industrial Electronics. Proceedings. ISIE'98 (Cat. No.98TH8357)*, IEEE, pp. 690–693. doi: https://doi.org/10.1109/ISIE.1998.711705.

[15] J. Lee, S. Park, S. Kim, O.-K. Choi, and C.-J. Chun, "LiteFDNet: A Lightweight Network for Current Sensor-Based Bearing Fault Diagnosis," *IEEE Access*, vol. 12, pp. 100493–100505, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3430512.

[16] A. Nandakumar, Y. Li, D. Zhao, Y. Zhang, and T. Hong, "Sparse Identification-Enabled Data-Driven Modeling for Nonlinear Dynamics of Microgrids," in *2022 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, Jul. 2022, pp. 1–5. doi: https://doi.org/10.1109/PESGM48719.2022.9917105.

[17] J. F. Dcoutho, B. Eisenbart, and A. Kulkarni, "Hierarchy Priortization and Dynamic Simulation for Low Volume Production Planning," in *2023 IEEE Engineering Informatics*, IEEE, Nov. 2023, pp. 1–10. doi: https://doi.org/10.1109/IEEECONF58110.2023.10520413.

[18] S. Barati, "A system dynamics approach for leveraging blockchain technology to enhance demand forecasting in supply chain management," *Supply Chain Anal.*, vol. 10, p. 100115, Jun. 2025, doi: https://doi.org/10.1016/j.sca.2025.100115.

[19] Z. Pirouzrahi, T. Vanelslander, and A. Nassiri Aghdam, "Applying system dynamics modelling to modal shift: A systematic review," *Sustain. Futur.*, vol. 9, p. 100526, Jun. 2025, doi: https://doi.org/10.1016/j.sftr.2025.100526.

[20] S. Zhang, Z. Xu, Y. Li, J. Zhang, and J. Hu, "System dynamics simulation model of price transmission in collaboration market of electricity, carbon, and green certificate driven by multiple policies," *Energy*, vol. 323, p. 135852, May 2025, doi:

https://doi.org/10.63876/ijtm.v4i1.126

https://doi.org/10.1016/j.energy.2025.135852.

[21] K. B. Jang and T. H. Woo, “Implications of public acceptance assessments for site selections in the small modular reactors using system dynamics,” *Energy Strateg. Rev.*, vol. 55, p. 101526, Sep. 2024, doi: https://doi.org/10.1016/j.esr.2024.101526.

[22] P. S. Hovmand *et al.*, “Developing Group Model-Building Workshops for Children’s Healthy Living Food Systems,” *Curr. Dev. Nutr.*, vol. 9, no. 4, p. 104583, Apr. 2025, doi: https://doi.org/10.1016/j.cdnut.2025.104583.

[23] P. Ortner, J. Z. Tay, and T. Wortmann, “Computational optimization for circular economy product design,” *J. Clean. Prod.*, vol. 362, p. 132340, Aug. 2022, doi: https://doi.org/10.1016/j.jclepro.2022.132340.

[24] J. J. Wimhurst and J. S. Greene, “Using logistic regression-cellular automata to project future sites for commercial wind energy development,” *Appl. Geogr.*, vol. 159, p. 103070, Oct. 2023, doi: https://doi.org/10.1016/j.apgeog.2023.103070.

[25] W. Wang, Y. Murphey, and P. Watta, “A Computational Framework for Implementation of Neural Networks on Multi-Core Machine,” *Procedia Comput. Sci.*, vol. 53, pp. 82–91, 2015, doi: https://doi.org/10.1016/j.procs.2015.07.282.

[26] D.-A. Nguyen, J. Nwadiuto, H. Okuda, and T. Suzuki, “Model Structure Identification of Hybrid Dynamical Systems based on Unsupervised Clustering and Variable Selection,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1090–1095, 2020, doi: https://doi.org/10.1016/j.ifacol.2020.12.1305.

[27] D. Deubert, A. Selig, and A. Verl, “On data, information, and knowledge in the context of online simulation in industrial automation,” *Procedia CIRP*, vol. 130, pp. 1188–1193, 2024, doi: https://doi.org/10.1016/j.procir.2024.10.226.

[28] H. Wang, H. Zhou, and S. Cheng, “Dynamical system prediction from sparse observations using deep neural networks with Voronoi tessellation and physics constraint,” *Comput. Methods Appl. Mech. Eng.*, vol. 432, p. 117339, Dec. 2024, doi: https://doi.org/10.1016/j.cma.2024.117339.

https://doi.org/10.63876/ijtm.v4i1.126