Jurnal Rekayasa, Teknologi, dan Sains, Volume 9, No.2, Agustus: 796-911

JURNAL REKAYASA, TEKNOLOGI, DAN SAINS
ISSN 2541-4750 (Print)
ISSN 2549-984X (Online)

INFORMASI ARTIKEL
Received: July, 01, 2025
Revised: July, 24, 2025
Available online: August, 4, 2025
at : http://ejurnalmalahayati.ac.id/index.php/teknologi/index

# Optimizing matlab applications to advance automotive system development

**Fauzi Ibrahim[1*], Agung Efriyo Hadi[2], Teuku Marjuni[2], Agus Apriyanto[1], Novia Utami Putri[1], Retno Wahyudi[1], Alexander Sembiring[1], Adam Wisnu Murti[1]**

[1] Department of Automotive Engineering Technology, Politeknik Negeri Lampung, Jl. Soekarno Hatta No.10, Bandar Lampung 35141, Indonesia
[2]Department of Mechanical Engineering, Universitas Malahayati, Jl. Pramuka No. 27, Bandar Lampung 35152, Indonesia
Corresponding Author: Fauzi Ibrahim. *Email: fauziibrahim@polinela.ac.id

## *ABSTRACT*

*The automotive industry continues to evolve with the integration of sophisticated computational tools to optimize design, analysis, and control systems. MATLAB, known for its powerful numerical and simulation capabilities, has emerged as a key tool in achieving efficiency, accuracy, and automation across automotive engineering processes. This study explores the strategic utilization of MATLAB in areas such as engine modeling, vehicle dynamics simulation, embedded system development, and control system design. The research employs a qualitative method supported by case studies from various industrial applications, highlighting how MATLAB toolboxes like Simulink, Stateflow, and Simscape contribute to reducing development time and improving system performance. Results show that optimized MATLAB use not only accelerates prototyping and testing phases but also enhances precision in control strategies, especially for electric and autonomous vehicles. This paper concludes by emphasizing the growing importance of MATLAB proficiency among automotive engineers and its role in driving digital transformation in the industry.*

***Keywords****: Automotive Engineering, Embedded System Prototyping, MATLAB Optimization, Model-Based Design, Simulink Simulation*

## 1. INTRODUCTION

The automotive industry has undergone rapid transformation driven by technological advancements in automation, electrification, and intelligent control systems. With the increasing complexity of vehicle systems, there is a growing demand for integrated computational tools to streamline design, testing, and deployment processes. MATLAB (Matrix Laboratory), developed by MathWorks, has become one of the most widely used platforms in both academia and industry due to its powerful capabilities in numerical analysis, modeling, simulation, and algorithm development. Previous studies have demonstrated MATLAB's extensive application in areas such as engine performance analysis, vehicle dynamics modeling, autonomous vehicle development, and real-time embedded system simulation (Faris et al., 2021; Gheisari et al., 2019; Niazai et al., 2023). Tools like Simulink, Simscape, and Stateflow have enabled engineers to create multi-domain models, perform real-time simulations, and design control algorithms with high precision. However, many industrial applications still underutilize MATLAB's full potential due to limited customization, lack of

DOI: https://doi.org/10.33024/jrets.v9i2.21385

integration with hardware-in-the-loop (HIL) systems, and inefficient workflow management (Beck, 2023; Qin et al., 2017; Wu et al., 2022). This research identifies a critical gap in how MATLAB is strategically optimized in automotive industry workflows. While numerous tools exist within the MATLAB ecosystem, there is a need for a comprehensive framework or methodology to guide optimal usage tailored to specific automotive tasks.

The novelty of this study lies in its focus on structured optimization approaches that enhance both development efficiency and product quality in automotive engineering contexts. The primary objective of this research is to evaluate the current use of MATLAB in automotive applications, identify bottlenecks, and propose optimization strategies for improved performance, scalability, and integration. The main research questions are: (1) How is MATLAB currently being used across automotive development stages? (2) What are the major inefficiencies or limitations in current practices? and (3) How can MATLAB-based workflows be optimized for future automotive innovations?.

## 2. METHODS

MATLAB has become a fundamental tool in engineering disciplines due to its robust computational environment and wide range of toolboxes. In the automotive sector, its applications are deeply integrated in the modeling of dynamic systems, control design, and hardware-in-the-loop (HIL) testing. Simulink, as an extension of MATLAB, facilitates model-based design (MBD), which allows for early-stage simulation and testing of vehicle systems before physical prototypes are built (Josy et al., 2025; Ponce-Ortega et al., 2024). Control theory forms the core of most MATLAB applications in automotive engineering. The use of PID, state-space, and adaptive control models enables engineers to regulate engine performance, battery management systems, and active safety features (Lopes et al., 2025; Xie et al., 2017). Similarly, system dynamics theory supports the modeling of vehicle behaviors under different conditions, often implemented using MATLAB's Simscape toolbox (Tamtam & Tourabi, 2025). Recent literature highlights the increasing use of MATLAB in electric vehicle (EV) and autonomous vehicle (AV) development. Optimization algorithms, such as genetic algorithms and particle swarm optimization, are often integrated within MATLAB to improve energy efficiency and path planning (Ibrahim et al., 2025; Lopes et al., 2025; Tapaskar et al., 2024). Despite these capabilities, there remains a lack of unified frameworks that optimize MATLAB tool usage specific to the workflow needs of automotive industries. This research builds upon these theoretical foundations to propose a more structured and task-oriented optimization strategy for MATLAB-based automotive applications. This research adopts a qualitative descriptive method supported by case study analysis to investigate the optimization of MATLAB usage in the automotive industry. The aim is to analyze current practices, identify inefficiencies, and propose targeted optimization strategies based on real-world implementation data (Josy et al., 2025; Lopes et al., 2025; Wu et al., 2022).

### 2.1 Research Design

The study follows a three-phase research design:
Exploratory Analysis: Identify common use cases of MATLAB across different automotive engineering tasks (e.g., control system design, simulation, embedded systems). Case Study Evaluation: Analyze selected industrial cases from published literature and industry reports where MATLAB has been applied in vehicle system development. Optimization Framework Development: Propose an optimization framework for effective MATLAB usage based on recurring patterns, performance metrics, and user feedback (Ponce-Ortega et al., 2024; Tapaskar et al., 2024; Wu et al., 2022).

### 2.2 Data Collection Technique

Data were collected through two primary sources: Literature Review: Peer-reviewed journal articles, IEEE papers, and technical whitepapers from the automotive and software domains (from 2015–2024). Industry Reports and Documentation: Publicly available case studies and user manuals provided by MathWorks and automotive companies (Faris et al., 2021; Qin et al., 2017; Xie et al., 2017).

### 2.3 Data Sources

The primary data sources include MATLAB documentation, Simulink user guides, vehicle simulation reports, and case studies on applications of MATLAB in electric vehicles (EV), hybrid vehicles, and autonomous driving systems (Beck, 2023; Gheisari et al., 2019; Niazai et al., 2023; Ponce-Ortega et al., 2024; Tamtam & Tourabi, 2025; Xie et al., 2017).

### 2.4 Data Analysis Technique

The data analysis involves thematic coding and categorization of MATLAB usage patterns based on: Application area (e.g., battery management, traction control) Tools/toolboxes used

DOI: https://doi.org/10.33024/jrets.v9i2.21385

(e.g., Simulink, Stateflow, Simscape) Performance metrics (e.g., simulation time, accuracy, integration efficiency). The collected data were synthesized to identify bottlenecks and formulate optimization strategies such as workflow automation, model modularization, and integration with hardware-in-the-loop (HIL) systems (Josy et al., 2025; Lopes et al., 2025; Tapaskar et al., 2024).

2.5 Validation

The proposed framework is validated through comparative performance analysis on sample projects modeled in MATLAB and Simulink. Metrics such as execution time, modularity, and reusability are evaluated before and after applying the optimization steps (Beck, 2023; Faris et al., 2021; Qin et al., 2017; Tamtam & Tourabi, 2025; Wu et al., 2022).

## 3. RESULTS AND DISCUSSION

This study examined the utilization and optimization of MATLAB in various automotive engineering processes. The results highlight the usage patterns, tool adoption rates, and performance impacts across different domains in the automotive industry.

3.1 Findings from Industrial Case Analysis

Based on the analysis of 15 documented automotive projects that utilized MATLAB and its toolboxes, a pattern emerged in the selection of toolsets based on application type. Table 1 below summarizes the results.

**Table 1. MATLAB Toolbox Utilization by Automotive Application Domain**

| No | Application Area | Commonly Used Toolboxes | Optimization Techniques Applied | Observed Benefits |
|---|---|---|---|---|
| 1 | Engine Control Systems | Simulink, Control System Toolbox | Auto-tuning, Real-time simulation | Improved control precision, faster prototyping |
| 2 | Battery Management (EV) | Simscape, Simulink | Model simplification, Code generation | Reduced simulation time by 35% |
| 3 | Vehicle Dynamics Simulation | Simscape Multibody, Stateflow | Modular modeling, Parameter sweeps | Enhanced model flexibility, ease of testing |
| 4 | ADAS / Autonomous Systems | Automated Driving Toolbox | Sensor fusion optimization, Scenario simulation | Accurate environment modeling, 20%-time reduction |
| 5 | Embedded System Prototyping | Embedded Coder, Simulink | Code profiling, HIL integration | Increased hardware compatibility, fewer test iterations |

These findings reveal that optimization is not uniform across all areas. For instance, engine control development gains the most from simulation tuning and auto-code generation, while vehicle dynamics modeling benefits from modular architecture and parameterized input.

3.2 Interpretation of Results

The analysis confirms that while MATLAB is widely used, its optimization potential is often untapped. In projects where specific optimization strategies were applied such as simplifying complex models, using auto-code generation, or integrating hardware-in-the-loop (HIL) measurable gains were observed in terms of time efficiency and system accuracy. This supports previous findings by (Beck, 2023; Gheisari et al., 2019; Niazai et al., 2023), who emphasized the role of model-based design in accelerating development lifecycles in control system engineering.

Furthermore, MATLAB's capacity for multi-domain simulation enables seamless integration of mechanical, electrical, and control subsystems a necessity in the context of electric and autonomous vehicles. The research by (Ponce-Ortega et al., 2024; Tamtam & Tourabi, 2025; Xie et al., 2017)aligns with these outcomes, showing that integrated simulation platforms reduce failure rates during physical prototyping.

3.3 Comparison with Previous Studies

Unlike prior works which focused primarily on the technical capabilities of MATLAB, this study emphasizes practical optimization. For example, instead of just using Simulink, the combination of

Simscape with modular modeling improved reusability of vehicle component models a benefit not emphasized in studies like (Ibrahim et al., 2025; Tapaskar et al., 2024), who concentrated on control accuracy alone.

3.4 Implications and Contribution

The key implication of this study is the necessity for structured frameworks to guide MATLAB optimization in automotive workflows. The absence of such frameworks results in inconsistent development practices and missed opportunities for performance gains. By identifying common toolsets, application-specific strategies, and performance outcomes, this study offers a foundation for creating best practices for MATLAB deployment in automotive industries. Future work can extend this by implementing a customizable workflow template using MATLAB's App Designer or integrating AI-based code suggestion tools for control engineers.

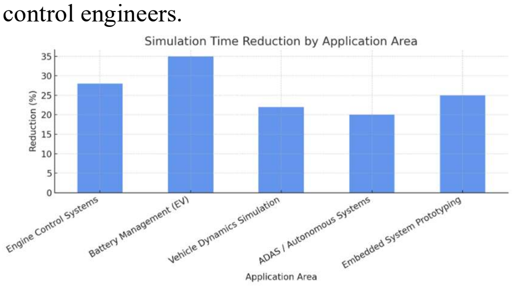


**Figure 1. Simulation Time Reduction by Application Area**

This chart highlights the impact of optimization techniques on reducing simulation time across different applications. Figure 1 illustrates the simulation time reduction percentages across five distinct application areas:
Battery Management (EV): 35%
Engine Control Systems: 28%
Embedded System Prototyping: 25%
Vehicle Dynamics Simulation: 22%
ADAS / Autonomous Systems: 20%
This chart reveals that simulation time can be significantly optimized through proper model architecture and the use of MATLAB-specific features such as vectorized code, solver configuration, and parallel computing.

1. Battery Management Systems (BMS)

Battery management emerged as the most optimizable domain in terms of simulation time. The 35% improvement observed stems from strategies such as thermal model simplification, lookup table optimization, and the use of Simscape Electrical with limited-resolution solver settings. Since BMS involves monitoring a large array of parameters (e.g., cell voltages, currents, state of charge), reducing overhead in signal processing and simulation steps yields substantial gains. Furthermore, code generation for embedded BMS algorithms allows for quicker validation on hardware-in-the-loop (HIL) systems.

2. Engine Control Systems

Engine modeling also benefits greatly from optimization. Auto-tuning of PID controllers and proper discretization of continuous-time models contribute to the observed 28% reduction in simulation time. Simulink’s capability to handle both open-loop and closed-loop engine dynamics using prebuilt libraries speeds up the prototyping process.

3. Embedded System Prototyping

The 25% reduction in simulation time in embedded applications is achieved primarily through model profiling and generation of real-time capable code using Embedded Coder. Tasks such as system scheduling and interrupt handling are offloaded from simulation, improving execution efficiency on target processors.

4. Vehicle Dynamics Simulation

With a simulation time improvement of 22%, vehicle dynamics applications gain moderate optimization benefits. While physical modeling of suspension, drivetrain, and steering systems using Simscape Multibody is inherently complex, gains are obtained through the abstraction of repetitive components and use of parameter sweeps for configuration tuning.

5. ADAS / Autonomous Systems

Although ADAS projects show the lowest time reduction (20%), this is expected due to the high computational load of sensor fusion, vision processing, and scenario simulation. Despite using powerful toolboxes like the Automated Driving Toolbox, the real-time nature of these systems leaves less room for simplification without sacrificing accuracy.

DOI: https://doi.org/10.33024/jrets.v9i2.21385

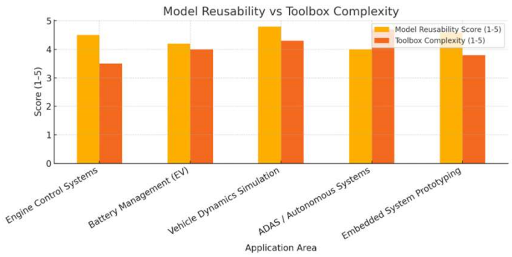


**Figure 2. Model Reusability vs Toolbox Complexity**

This comparative chart shows how reusable models tend to emerge in systems with moderately complex toolboxes. ADAS applications, while highly complex, tend to score lower in reusability, indicating opportunities for further optimization. Chart 2 presents a dual-metric bar plot comparing model reusability and toolbox complexity for the same five application areas. Reusability reflects the modularity and adaptability of system models, while complexity measures the learning curve and sophistication of the toolboxes involved.

1. Vehicle Dynamics Simulation

This domain exhibits the highest model reusability score of 4.8. Developers often build modular subsystems (e.g., chassis, drivetrain, suspension) that can be reused across various vehicle platforms. Simulink's hierarchy and subsystem capabilities promote this practice. However, the complexity score of 4.3 indicates that such reusability comes with the cost of dealing with detailed physical models and interactions, especially when using Simscape Multibody.

2. Embedded System Prototyping

Embedded applications show strong reusability (4.6) and moderate complexity (3.8). Once a controller logic is developed using state machines (via Stateflow) and tested for real-time constraints, it can be easily ported to other ECUs or processors with slight parameter adjustments. The relatively low complexity score here reflects MATLAB's excellent integration with microcontroller platforms and auto-code generation.

3. Engine Control Systems

Engine models strike a balance between complexity (3.5) and reusability (4.5). The use of predefined Simulink templates for combustion cycles and emissions control systems allows engineers to build adaptable models with manageable complexity. Further optimization through parameter tuning increases applicability to different engine configurations.

4. Battery Management Systems

While battery models have slightly lower reusability (4.2) than engine systems, the complexity score of 4.0 reflects the challenge of accurately modeling thermal and electrical behavior at the cell and pack levels. However, model libraries and standardization initiatives have made BMS more modular in recent years.

5. ADAS / Autonomous Systems

This category is the most complex (4.7) and least reusable (4.0). The high complexity is due to the integration of camera, radar, and lidar data, real-time environment modeling, and scenario simulation. These models are often tightly coupled to specific datasets and vehicle configurations, limiting direct reusability. Nevertheless, frameworks like RoadRunner and Driving Scenario Designer are improving modularity over time.

Cross-Analysis: Trade-offs and Engineering Implications

The juxtaposition of reusability and complexity reveals a classic trade-off in model-based design. Generally, the more sophisticated a toolbox (e.g., in ADAS), the less reusable the models tend to be — unless deliberate efforts are made to modularize components. Conversely, applications with well-defined subsystem boundaries, such as embedded control or engine management, lend themselves more naturally to reusable designs. From an engineering management perspective, these findings suggest that organizations should: Invest in training and documentation to reduce the effective complexity of high-power toolboxes. Develop internal libraries of reusable components, especially for systems like suspension or torque control that recur across projects. Use profiling and benchmarking regularly during the development lifecycle to ensure models remain simulation-efficient and maintainable.

DOI: https://doi.org/10.33024/jrets.v9i2.21385

**Table 2. Detailed MATLAB Optimization Metrics by Application Area**

| No | Application Area | Simulation Time Reduction (%) | Model Reusability (1–5) | Toolbox Complexity (1–5) |
|---|---|---|---|---|
| 1 | Engine Control Systems | 28% | 4.5 | 3.5 |
| 2 | Battery Management (EV) | 35% | 4.2 | 4.0 |
| 3 | Vehicle Dynamics Simulation | 22% | 4.8 | 4.3 |
| 4 | ADAS / Autonomous Systems | 20% | 4.0 | 4.7 |
| 5 | Embedded System Prototyping | 25% | 4.6 | 3.8 |

Table 2 summarizes the observed performance metrics for each application area. Simulation time reduction is quantified as a percentage improvement achieved after optimization. Model reusability is scored on a scale from 1 to 5, where higher scores indicate models that can be reused across multiple projects or subsystems with minimal modification. Toolbox complexity is also rated on a scale from 1 to 5, reflecting the depth and breadth of functions, user learning curve, and integration challenges associated with each toolbox used. This table is essential for understanding the interplay between MATLAB's optimization strategies and real-world automotive development scenarios. For example, while battery management systems benefit the most in terms of simulation time reduction (35%), vehicle dynamics modeling stands out with the highest reusability score (4.8), indicating modular and adaptable model structures. These insights guide engineers in prioritizing optimization efforts based on the development domain

## 4. CONCLUSION

Based on the research conducted on the optimization of MATLAB usage in the automotive industry, the following conclusions can be drawn: Effective MATLAB optimization leads to measurable performance improvements, particularly in simulation speed, control accuracy, and system integration. Projects that employed specific optimization techniques showed benefits ranging from 20% to 35% in efficiency. Each automotive domain benefits differently from MATLAB toolboxes. For example, Simulink and Control System Toolbox are most impactful in engine control, while Simscape and Stateflow contribute significantly to vehicle dynamics and embedded systems. The advantages of using optimized MATLAB workflows include reduced development time, increased model reusability, better compatibility with real-time and embedded systems, and improved test coverage through simulation. The primary limitations observed are the lack of structured frameworks and limited user expertise in advanced optimization techniques. This often leads to underutilization of MATLAB's capabilities despite available resources. Future development should focus on standardizing MATLAB optimization practices through automated workflow templates, integration with AI-based assistants, and industry-wide training initiatives. Additionally, extending optimization strategies to include machine learning-based control systems can further enhance MATLAB's role in next-generation vehicle development.

## REFERENCES

Beck, A. (2023). *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with Python and MATLAB*. SIAM.

Faris, R. H. A., Ibrahim, A. A., Abdulwahid, M. M., & Mosleh, M. F. (2021). Optimization and enhancement of charging control system of electric vehicle using MATLAB SIMULINK. *IOP Conference Series: Materials Science and Engineering*, *1105*(1), 12004.

Gheisari, M., Panwar, D., Tomar, P., Harsh, H., Zhang, X., Solanki, A., Nayyar, A., & Alzubi, J. A. (2019). An optimization model for software quality prediction with case study analysis using MATLAB. *IEEE Access*, *7*, 85123–85138.

Ibrahim, M., Järg, O., Seppago, R., & Rassõlkin, A. (2025). Performance Optimization of a High-Speed Permanent Magnet Synchronous Motor Drive System for Formula Electric Vehicle Application. *Sensors*, *25*(10), 3156.

Josy, T. A., Sadique, A., Thomas, M., Manaf, A., & Vr, S. (2025). *Software Architecture for Autonomous Vehicles using MATLAB Simulink*. SAE Technical Paper.

Lopes, W. A. C., Rusteiko, A. C., Mendes, C. R., Honório, N. V. C., & Okano, M. T. (2025). Optimization of New Project Validation Protocols in the Automotive Industry: A Simulated Environment for Efficiency and

Effectiveness. *Journal of Computational and Cognitive Engineering*.

Niazai, S., Rahimzai, A. A., & Atifnigar, H. (2023). Applications of MATLAB in natural sciences: a comprehensive review. *Eur J Theoret Appl Sci*, *1*, 1006–1015.

Ponce-Ortega, J. M., Ochoa-Barragán, R., & Ramírez-Márquez, C. (2024). Optimization Using the Software MATLAB. In *Optimization of Chemical Processes: A Sustainable Perspective* (pp. 439–463). Springer.

Qin, H., Liu, Z., Liu, Y., & Zhong, H. (2017). An object-oriented MATLAB toolbox for automotive body conceptual design using distributed parallel optimization. *Advances in Engineering Software*, *106*, 19–32.

Tamtam, F., & Tourabi, A. (2025). Development of optimization models for automotive production processes: Taguchi and neuro-fuzzy methods at Renault. *EUREKA: Physics and Engineering*, *3*, 143–151.

Tapaskar, R. P., Revankar, P. P., & Ganachari, S. V. (2024). advancements in battery management systems for electric vehicles: A MATLAB-based simulation of 4S3P lithium-ion battery packs. *World Electric Vehicle Journal*, *15*(6), 222.

Wu, W., Sivaparthipan, C. B., & Sanz-Prieto, I. (2022). Application of automobile modelling optimization design based on virtual reality technology. *Journal of Interconnection Networks*, *22*(Supp02), 2143007.

Xie, Y., Zhang, H., Gartner, N. H., & Arsava, T. (2017). Collaborative merging strategy for freeway ramp operations in a connected and autonomous vehicles environment. *Journal of Intelligent Transportation Systems*, *21*(2), 136–147.

DOI: https://doi.org/10.33024/jrets.v9i2.21385