Model-Predictive Control Enhanced Energy Management

and Analysis of Dual-Motor Electric Vehicles

by

Harsh Gangwar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2024 by the
Graduate Supervisory Committee:

Yan Chen, Chair
Junfeng Zhao
Dajiang Suo

ARIZONA STATE UNIVERSITY

May 2024

ABSTRACT

Electric vehicles (EVs) have emerged as a promising solution to reduce greenhouse gas emissions and dependency on fossil fuels in the transportation sector. However, limited battery capacity remains a significant challenge, impacting range and overall performance. This thesis explores the application of Nonlinear Model Predictive Control (NMPC) techniques to optimize energy management in EVs. The study begins with a comprehensive review of existing literature on EV energy optimization strategies and NMPC methodologies. Subsequently, a detailed model of the EV's dynamics, including the battery, motor, and vehicle dynamics, is developed to formulate the optimization problem. The NMPC controller is designed to dynamically adjust the power distribution among different vehicle components, such as the motor, battery, and regenerative braking system, while considering constraints such as battery state-of-charge, vehicle speed, and road conditions. Simulation studies are conducted to evaluate the performance of the proposed NMPC-based energy optimization strategy under various driving scenarios and compare it with conventional control strategies. The results demonstrate that NMPC offers superior performance in terms of energy efficiency, range extension, and overall vehicle dynamics. The findings of this research contribute to the advancement of energy optimization techniques for EVs, paving the way for more efficient and sustainable transportation systems in the future.

ACKNOWLEDGMENTS

I am deeply grateful to my advisor, Dr. Yan Chen, for his invaluable guidance throughout my academic journey and thesis completion. His mentorship has been instrumental in shaping my career trajectory, and I am indebted to him for the opportunities to refine my understanding of control theory and gain practical experience through various projects within the Dynamic Systems & Control Laboratory.

I extend my sincere appreciation to the esteemed committee members, Dr. Junfeng Zhao and Dr. Dajiang Suo, for their dedicated time and meticulous review of my thesis. Their insightful feedback has significantly enriched the quality of my work.

I am also profoundly thankful to my parents and brother for their unwavering support and encouragement throughout this endeavor. Their belief in me has been a constant source of motivation, and I am truly grateful for their love and guidance.

To everyone who has contributed to my academic and personal growth, I carry with me a deep sense of gratitude and appreciation. Your support has fueled my journey, and I am committed to carrying forward this positive energy in all my future endeavors.

TABLE OF CONTENTS

Page

LIST OF TABLES ... vii

LIST OF FIGURES ... viii

LIST OF SYMBOLS / NOMENCLATURE ... ix

CHAPTER

1 INTRODUCTION ... 1

Battery Electric Vehicles ... 1

Model Predictive Control ... 2

Nonlinear Model Predictive Control ... 4

2 LITERATURE REVIEW ... 8

Paper 1 ... 8

Paper 2 ... 9

Paper 3 ... 10

3 PROBLEM STATEMENT ... 12

MATLAB Simulink Challenge 246 ... 12

Requisites ... 12

4 MODEL PREDICTIVE CONTROL, STABILITY & FEASIBILITY ... 14

Stability ... 14

Feasibility ... 15

5 SOFTWARE IN LOOP ... 17

Importance in Vehicle Optimization and Related Fields ... 17

Advantages ... 18

CHAPTER Page

Disadvantages 19

6 VEHICLE COMPOSER AND ITS COMPONENTS 20

Environment 22

Scenarios....................23

Driver....................25

Vehicle Plant 25

Controller....................29

Visualization 30

Controller Design and Operation....................31

7 HAMILTONIAN OPTIMIZATION.................... 33

8 NMPC CONTROLLER 35

The System....................36

Objective function 36

Optimization Problem....................37

Implementation Methodology 37

Model Dynamics 39

Solver 40

9 EXPERIMENTAL SETUP 43

Vehicle Parameters....................43

Implementaion 44

MPC Parameters 48

CHAPTER Page

10 PERFORMANCE METRICS AND EVALUATION 49

Experimentation 49

Nominal MPC 49

Fixed-Timestep NMPC with Fixed Timestep Model 50

Optimized Parameter NMPC 51

Optimized Parameter NMPC with Delta U penalty 55

Optimized Parameter NMPC with Control feedback 58

Optimized Parameter NMPC with Non-overshooting constraints 63

Optimized Parameter NMPC with linear velocity estimation 63

Summary 67

11 IMPACT AND LIMITATIONS 66

Global impact of 1.07% decrease in EV SOC consumption 68

Comparison of Hamiltonian Optimization and NMPC 69

Hamiltonian Optimization 70

NMPC with fmincon 70

Future Directions 71

12 CONCLUSION AND RECOMMENDATIONS 72

Conclusion 72

Recommendation 72

REFERENCES 75

LIST OF TABLES

Table Page

1. Vehicle Parameters 33 41

2. NMPC Paramters 45

LIST OF FIGURES

Figure Page

1. MPC Overview 2
2. GPS infrastructure 9
3. Hybrid Vehicle Architecture 10
4. Challenges for MPC 14
5. MPC Feasibility Sets 16
6. Virtual Vehicle Composer App 20
7. Process flow diagram 22
8. VVC Overview 22
9. US EPA Urban Dynamometer Driving Schedule (FTP-75) 23
10. WLTC cycle for Class 3b vehicles 24
11. US06 Drive Cycle 24
12. Driver Block 25
13. Vehicle Physics 26
14. Drivetrain 27
15. Electric Machines 27
16. Maximum Torque Plot 28
17. Torque Speed Efficiency Map 28
18. Battery Management System 29
19. Battery Charge Model 29
20. Controller Block 30

Figure Page

21. Power vector calculation 31

22. Combination of front and rear torque vector 33

23. Hamiltonian function..34

24. EV-2M Architecture..35

25. Instructions 44

26. Instructions step -2 45

27. Instructions: Clock..46

28. Results for Nominal NMPC from VVC App 50

29. Results for Fixed Time step NMPC and Dynamics Model 51

30. Optimized NMPC Visualization 52

31. High Point Analysis- Optimized NMPC 53

32. Low Point Analysis- Optimized NMPC 53

33. Medim Point Analysis- Optimized NMPC 54

34. Control Policy- Optimized NMPC 54

35. Comparison of Baseline and Delta U penalty NMPC 55

36. High Point Analysis- Delta U NMPC..56

37. Low Point Analysis- Delta U NMPC 56

38. Medium Point Analysis- Delta U NMPC..57

39. Control Policy- Delta U NMPC 57

40. Full feedback NMPC..58

41. High Points- Full feedback NMPC 59

42. Low Points- Full feedback NMPC...60

Figure Page

43. Medium Points- Full feedback NMPC 60

44. Control Policy- Full feedback NMPC 61

45. High Power vector- Full feedback NMPC 61

46. Low Power vector- Full feedback NMPC 62

47. Medium Power vector- Full feedback NMPC..62

48. Non-overshooting NMPC 63

49. Blind NMPC..64

50. High Points- Blind NMPC 65

51. Low Points- Blind NMPC..65

52. Medium Points- Blind NMPC 66

53. Control Policy- Blind NMPC...66

CHAPTER 1

INTRODUCTION

Battery Electric Vehicles

The rapid growth of Electric Vehicle (EV) production is transforming the automotive industry, driven by increasing customer demand, stringent global policy changes aimed at reducing emissions, and significant advancements in battery technology. According to Reuters, EV sales are projected to capture up to 33% of the global market share by 2028 . EVs accounted for less than 8% of global sales last year, and just under 10% in the first quarter of 2022. To support that demand, automakers and suppliers now expect to invest at least $526 billion on EVs and batteries from 2022-2026 [6]. That is more than double the five-year EV investment forecast of $234 billion from 2020-2024.

The integration of multiple electric motors in EV powertrains enables features such as all-wheel drive, improved acceleration, and enhanced lateral vehicle control. However, to fully exploit the potential benefits of multi-motor configurations, efficient energy management is crucial. Maximizing the driving range and minimizing energy consumption are paramount objectives in the design and operation of EVs.

While current energy management strategies for EVs often rely on heuristic rules or simple control algorithms, there is a growing recognition of the potential for formal optimization methods to achieve superior performance. Model Predictive Control (MPC) algorithms have demonstrated promise in optimizing the energy management of electrified powertrains by considering various operating conditions and environmental factors in real-time.

This project addresses the need for advanced energy optimization strategies in EV powertrains by developing a Nonlinear Model Predictive Control (NMPC) algorithm tailored to divide torque between front and rear motors efficiently. By dynamically allocating torque based on vehicle dynamics, road conditions, and battery state-of-charge, the proposed NMPC algorithm aims to maximize energy efficiency, extend driving range, and enhance overall vehicle performance.

Through comprehensive simulation studies and validation tests, this research seeks to demonstrate the effectiveness and benefits of the NMPC-based energy optimization strategy in real-world driving scenarios. The outcomes of this project have the potential to significantly advance the state-of-the-art in energy management for electrified powertrains, contributing to the continued growth and sustainability of the EV market.

Model Predictive Control

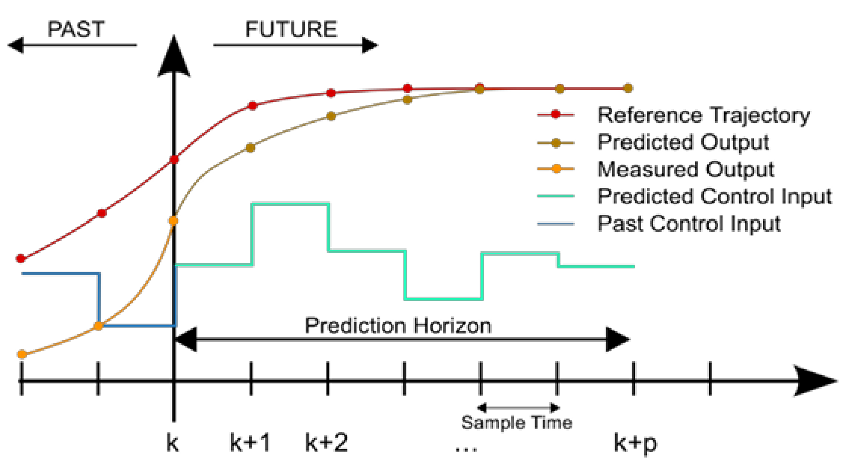

Figure 1: MPC Overview [10]

Model Predictive Control (MPC) is an advanced control strategy widely used in engineering and industrial processes for optimizing system performance while adhering to constraints. Unlike Traditional control methods, MPC utilizes a predictive model of the system to anticipate future behavior and determine control actions that minimize a cost function.

At its core, MPC consists of several key components:

1. Prediction Model: This component represents the dynamic behavior of the system being controlled. It could be a mathematical model derived from first principles, empirical data, or a combination of both. The model predicts how the system will evolve over a future time horizon in response to control inputs.

2. Objective Function: Also known as the cost function, this quantifies the performance goals and objectives of the control system. It typically includes terms related to desired setpoints, reference trajectories, and constraints. The objective function guides the optimization process to find control inputs that achieve the desired system behavior.

3. Constraints: MPC considers both control and state constraints that must be satisfied throughout the control horizon. These constraints may include limits on control inputs, operational limits of the system, and safety constraints. Ensuring constraint satisfaction is crucial for preventing undesirable behavior and maintaining system integrity.

4. Optimization Algorithm: MPC employs an optimization algorithm to solve the optimization problem defined by the objective function and constraints. The goal is to find the sequence of control inputs that minimizes the cost function while satisfying all constraints. Common optimization techniques include quadratic programming, linear

programming, and nonlinear programming.

5. Control Horizon: This defines the time horizon over which future predictions and control actions are considered. The length of the control horizon influences the trade-off between predictive accuracy and computational complexity. Shorter horizons provide faster responses but may sacrifice long-term performance, while longer horizons offer better prediction accuracy but require more computational resources.

6. Moving Horizon: MPC operates in a moving horizon fashion, where the control inputs are recalculated at each time step based on updated measurements and predictions. This enables MPC to adapt to changes in the system dynamics and disturbances in real-time, ensuring optimal performance under varying operating conditions.

By integrating these components, MPC enables precise control over complex systems, offering advantages in terms of stability, robustness, and efficiency compared to traditional control methods. Its widespread adoption across various industries underscores its effectiveness in addressing challenging control problems[8].

Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) builds upon the principles of Model Predictive Control (MPC) but accommodates nonlinear dynamics in the system model. This extension allows NMPC to handle a broader range of processes and systems that cannot be adequately represented by linear models. NMPC is particularly valuable in systems with nonlinear behavior, such as chemical processes, power systems, and robotic systems.

The components of NMPC are like those of MPC, but with some specific considerations for handling nonlinearities:

1. Nonlinear System Model: Unlike MPC, which often relies on linear or linearized models, NMPC employs nonlinear models that more accurately capture the system's dynamics. These models may involve nonlinear differential equations, algebraic equations, or even black-box models derived from empirical data. The complexity of these models can vary significantly depending on the system under control.

2. Nonlinear Optimization: NMPC requires specialized optimization algorithms capable of handling nonlinear objective functions and constraints. Nonlinear programming techniques, such as sequential quadratic programming (SQP), direct collocation methods, and particle swarm optimization, are commonly used to solve the optimization problem efficiently. These algorithms iteratively refine the control inputs to converge towards the optimal solution while satisfying the system constraints.

3. Prediction Horizon: The prediction horizon in NMPC determines the length of the time interval over which future predictions and control actions are computed. In nonlinear systems, longer prediction horizons may be necessary to capture complex nonlinear behavior accurately. However, longer horizons also increase computational complexity,

requiring efficient algorithms to balance prediction accuracy with computational efficiency.

4. Handling Nonlinear Constraints: NMPC must account for nonlinear constraints imposed by the system dynamics and operational limits. These constraints may include nonlinear inequalities or equalities representing physical limits, safety constraints, or constraints on control inputs. Efficient techniques, such as nonlinear complementarity constraints or constraint transformation methods, are employed to handle these nonlinear constraints effectively.

5. State Estimation: In nonlinear systems, accurate state estimation becomes more challenging due to the complexity of the system dynamics. Kalman filters, extended Kalman filters (EKF), unscented Kalman filters (UKF), and particle filters are common techniques used for state estimation in NMPC. These methods recursively estimate the system state based on available measurements and the nonlinear system model, enabling robust control performance even in the presence of uncertainties and disturbances.

6. Stability Analysis: Analyzing the stability properties of NMPC becomes more intricate due to the presence of nonlinearities. Lyapunov-based methods, stability margins, and robustness analysis techniques are employed to ensure the closed-loop stability of the nonlinear control system. These analyses are crucial for guaranteeing reliable and safe operation, especially in critical applications.

By incorporating these components, NMPC extends the capabilities of traditional MPC to handle nonlinear systems effectively. Its ability to accommodate complex dynamics and nonlinear constraints makes NMPC a powerful tool for controlling a wide range of dynamic systems in various fields, including process control, robotics, aerospace, and automotive applications[11].

CHAPTER 2

LITERATURE REVIEW

Electric vehicles (EVs) have emerged as promising solutions to address environmental concerns and reduce dependency on fossil fuels. Effective energy management strategies play a crucial role in optimizing the performance and efficiency of EVs, particularly those equipped with multiple motors. This literature review explores two pertinent studies in the field of energy management for electric ground vehicles, focusing on terrain profile preview and Model Predictive Control (MPC) implementation.

1. Paper 1- Energy Management and Driving Strategy for In-Wheel Motor Electric Ground Vehicles with Terrain Profile Preview**. [1]**

- Chen et al. (Year) introduced a novel energy management and driving strategy for in-wheel motor electric ground vehicles. The study emphasizes the significance of terrain profile preview in optimizing energy consumption and enhancing vehicle performance.

  According to Figure 2, energy consumption during a trip is influenced by both the terrain between the starting point A and destination B, which can be accessed through on-board vehicle geographical information systems (GIS) and global positioning systems (GPS), and by vehicle characteristics [4] and real-time traffic data [5]. Consequently, optimizing the driving velocity profile based on terrain and traffic preview information is crucial for minimizing energy consumption and increasing travel distance for Electric Ground Vehicles (EGVs).

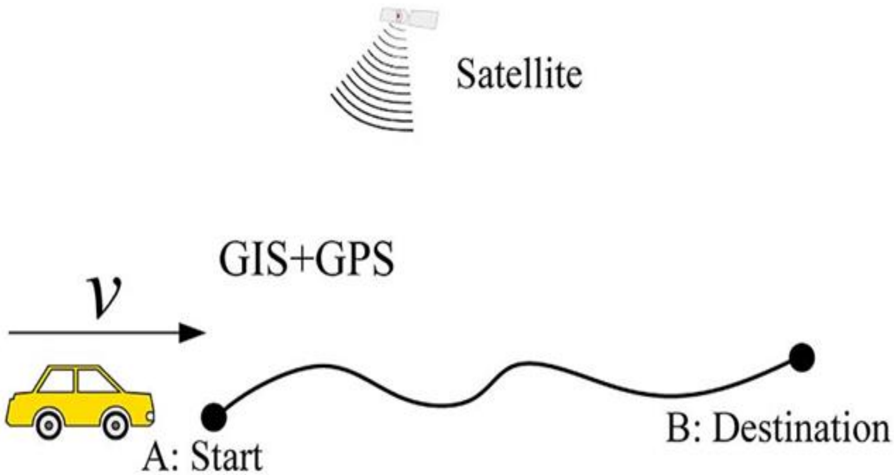

Figure 2: GPS infrastructure.

- By leveraging terrain profile information, the proposed strategy dynamically adjusts motor torque distribution to adapt to varying road conditions, thereby improving energy efficiency and extending driving range.
- Through experimental validation and simulation studies, Chen et al. demonstrated the effectiveness of their approach in real-world driving scenarios, highlighting its potential for practical implementation in electric ground vehicles.

2. Paper 2- Fast and Global Optimal Energy-Efficient Control Allocation with Applications to Over-Actuated Electric Ground Vehicles**.** [2]

- Chen and Wang (Year) proposed a novel energy-efficient control allocation scheme for over-actuated electric ground vehicles, focusing on optimizing operational energy efficiency.

- Their study introduces a fast and globally optimal optimization algorithm based on the Karush-Kuhn-Tucker (KKT) framework, which efficiently identifies all local optimal solutions and determines the global minimum despite non-convex optimization challenges.
- Through numerical validations, the algorithm demonstrates substantial computational efficiency improvements compared to classical methods, making significant strides towards more energy-efficient over-actuated vehicle systems.

3. Paper 3- Processor-In-the-Loop Management Demonstration of MPC for HEVs Energy System **[3]**

- Cavanini et al. (Year) presented a comprehensive demonstration of Model Predictive Control (MPC) for Hybrid Electric Vehicles (HEVs) energy management system using a Processor-In-the-Loop (PIL) setup.

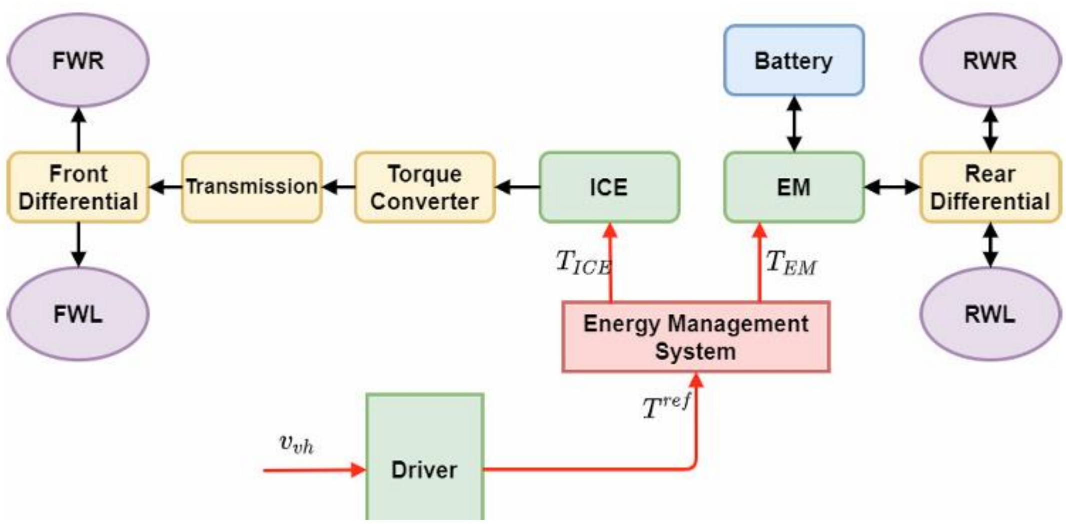

Figure 3: Hybrid Vehicle Architecture [3]

- The study focuses on the implementation of MPC algorithms to optimize the energy flow within HEVs, aiming to minimize fuel consumption and emissions while ensuring optimal vehicle performance.
- Through the PIL setup, Cavanini et al. validated the MPC algorithm's functionality and performance in real-time, considering various operating conditions and driving profiles.
- The results highlight the effectiveness of MPC in achieving superior energy management compared to traditional control strategies, emphasizing its potential for enhancing the overall efficiency and sustainability of HEVs.

By synthesizing insights from these studies, this thesis aims to build upon existing research and contribute to the development of advanced energy management strategies for electric vehicles. Specifically, the thesis seeks to integrate terrain profile preview and MPC techniques to optimize energy consumption and extend driving range in two-motor electric ground vehicles. Through simulation and experimental validation, the proposed approach aims to demonstrate its effectiveness in real-world driving scenarios, paving the way for enhanced performance and efficiency in electric vehicle applications.

CHAPTER 3

PROBLEM STATEMENT

MATLAB-Simulink-Challenge

Within the automotive industry, the widespread adoption of electric vehicles (EVs) underscores the urgency for sophisticated control algorithms to maximize their energy efficiency and driving range. This thesis endeavors to tackle the challenge posed by the "Design and Evaluation of Model-Predictive Control (MPC) Algorithm for Two-Motor Electric Vehicle (BEV) to Enhance Energy Efficiency and Driving Range." It is worth noting that this research project is part of the MATLAB Simulink Challenge Project 246, designated with a difficulty level suitable for Master's and Doctoral candidates[12].

The project proposes the implementation of a 2-motor Electric Vehicle architecture utilizing MATLAB and Simulink, leveraging the Powertrain Blockset and Model-Predictive Control Toolbox. The primary objective is to develop a robust MPC algorithm capable of optimally distributing torque to the electric motors, thereby reducing energy consumption while maximizing the vehicle's driving range.

Requisites

The key components of the project include:

1. Model Development: Utilizing the Powertrain Blockset, a 2-motor BEV model will be generated within the Simulink environment. This model will replicate the dynamics of the

electric vehicle during various drive cycles, providing a baseline for performance evaluation.

2. MPC Algorithm Design: Leveraging the capabilities of the Model-Predictive Control Toolbox, a sophisticated MPC algorithm will be designed. This algorithm will incorporate both linear and non-linear control strategies to efficiently manage torque distribution, with the overarching goal of enhancing energy efficiency and driving range.

3. Integration and Evaluation: The developed MPC algorithm will be integrated into the vehicle model as a model reference operating at a fixed-time step of 10ms. Through rigorous simulation and testing, the performance of the MPC algorithm will be evaluated across different drive cycles, including WLTP Class 3, HWFET, FTP75, and US06.

4. Performance Analysis: The effectiveness of the MPC algorithm will be assessed by comparing energy consumption and MPGe metrics against the baseline controller. Additionally, the adherence to implemented constraints within the MPC algorithm will be validated to ensure the stability and safety of the control system.

By addressing these objectives, this thesis endeavors to contribute to the advancement of electric vehicle technology by introducing a novel approach to optimize energy efficiency and driving range through Model-Predictive Control.

CHAPTER 4

MODEL PREDICTIVE CONTROL, STABILITY & FEASIBILITY

Model Predictive Control (MPC) offers a powerful framework for optimizing control actions over a finite time horizon based on a dynamic model of the system. However, two critical issues often arise in the implementation of MPC: stability and feasibility.

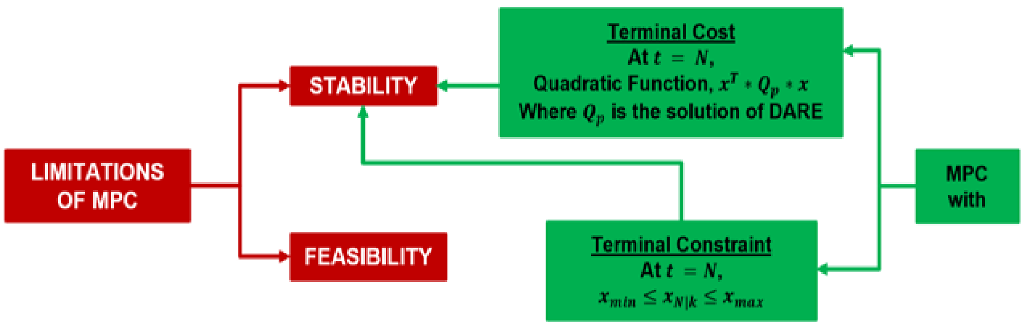

Figure 4: Challenges for MPC [10]

Stability:

Ensuring stability is crucial for the reliable operation of any control system. In MPC, stability concerns arise from the predictive nature of the controller. The closed-loop system's stability depends on the choice of prediction horizon, terminal conditions, and control law design.

Improper selection of these parameters can lead to instability, where the system's behavior diverges over time. To address stability concerns, MPC often employs terminal constraints and terminal costs, which guide the system towards a stable equilibrium over the prediction horizon[11].

Terminal Cost Function:

$$J(x) = x^T W x$$

Where J(x) represents the cost function, x denotes the state vector, and W is a positive-definite terminal cost matrix.

Lyapunov Matrix Equation:

$$A^T W + WA - WBR^{-1}B^T W + Q = 0$$

This equation ensures stability by guaranteeing that the closed-loop system remains stable over the prediction horizon.

From Infinite horizon to final horizon (terminal cost)

The terminal cost matrix P.

$$\begin{aligned} \mathrm{J}^*_{\mathrm{N}-1}(x_{\mathrm{N}-1}) = \min_{\mathrm{u}_{\mathrm{N}-1}} \{ & x^T_{N-1}(\mathrm{A}^T \mathrm{P}_\mathrm{N} \mathrm{A} + \mathrm{Q}) x_{N-1} + 2x^T_{N-1} \mathrm{A}' \mathrm{P}_\mathrm{N} \mathrm{B} u_{N-1} \\ & + u^T_{N-1}(\mathrm{B}^T \mathrm{P}_\mathrm{N} \mathrm{B} + \mathrm{R}) u_{N-1} \} \end{aligned}$$ [7]

Feasibility:

Feasibility refers to the ability of the control system to find a valid solution within the constraints imposed by the system and environment. In MPC, feasibility issues arise when the optimization problem becomes infeasible, meaning that no control input satisfies all constraints over the prediction horizon.

Feasibility problems can occur due to aggressive control objectives, inadequate modeling of the system dynamics, or constraints that are too restrictive. To mitigate feasibility issues, MPC algorithms incorporate mechanisms to relax constraints or adjust control objectives dynamically[11].

Addressing stability and feasibility concerns requires careful tuning of MPC parameters, robust modeling of system dynamics, and effective handling of constraints. By balancing these factors, MPC can deliver reliable and efficient control solutions across a wide range of applications.

Terminal Equality Constraint:

- x_N= 0, Ensures that the predicted state converges to the origin at the end of the prediction horizon, promoting feasibility.
- Control Law: u_k= K*x, Where K is the feedback gain matrix, determined to stabilize the system and satisfy constraints.

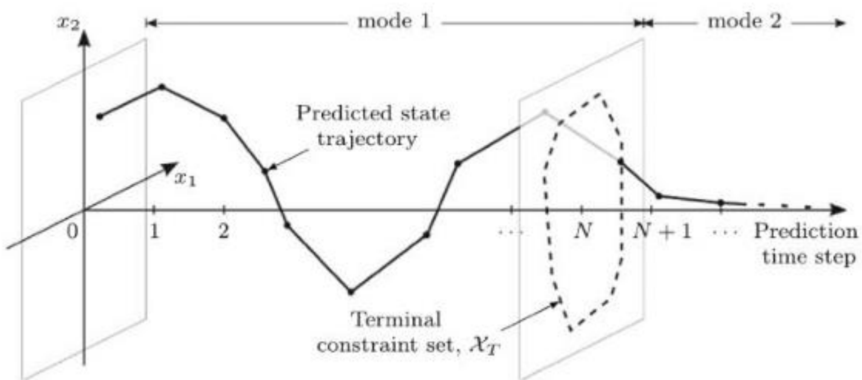

Figure 5: MPC Feasibility Sets [10]

By incorporating these equations into the MPC formulation and appropriately tuning parameters, stability and feasibility can be addressed effectively in control applications.

CHAPTER 5

SOFTWARE IN LOOP

Enhancing Vehicle Optimization and Beyond

Software-in-Loop (SIL) testing is a pivotal methodology in the realm of vehicle optimization and related fields. This technique involves the testing of software components in a simulated environment, allowing for comprehensive evaluation and validation of their functionality before integration with physical hardware. SIL testing holds immense importance in various industries, including automotive, aerospace, robotics, and more. Here, we delve into the significance of SIL testing, its advantages, and potential disadvantages.

Importance in Vehicle Optimization and Related Fields[13]:

1. Early Detection of Software Defects: SIL testing enables the early detection and rectification of software defects, thereby minimizing the risk of issues manifesting in the final product. By simulating real-world scenarios, developers can identify and address potential issues at an early stage of development.

2. Acceleration of Development Cycles: SIL testing accelerates development cycles by providing rapid feedback on software performance. Developers can iterate and refine software designs more efficiently, leading to faster time-to-market for vehicle optimization solutions.

3. Cost-Effectiveness: SIL testing is a cost-effective approach compared to testing with physical hardware. It reduces the need for expensive equipment and facilitates testing

across a wide range of scenarios without the logistical challenges associated with physical testing.

4. Comprehensive Testing: SIL testing allows for comprehensive testing of software under various conditions and scenarios, including extreme environments and edge cases. This ensures robustness and reliability in real-world applications.

5. Integration Testing: SIL testing facilitates integration testing of software components with minimal dependencies on physical hardware. It enables developers to validate the interaction between different software modules and ensure seamless integration into the overall system.

Advantages[13]

1. Early Validation: SIL testing enables early validation of software functionality, helping to identify and address issues at the initial stages of development.

2. Rapid Iteration: SIL testing facilitates rapid iteration and refinement of software designs, leading to faster development cycles and improved product quality.

3. Cost Savings: SIL testing reduces the need for expensive physical hardware and testing equipment, resulting in cost savings for organizations.

4. Comprehensive Testing: SIL testing allows for comprehensive testing of software under various conditions, including edge cases and extreme scenarios, ensuring robustness and reliability.

5. Integration Testing: SIL testing facilitates integration testing of software components, helping to validate their interaction and integration within the overall system architecture.

Disadvantages[13]

1. Limited Real-World Validation: SIL testing may lack the realism of physical testing, leading to potential discrepancies between simulation results and real-world performance.

2. Simulation Uncertainty: SIL testing relies on simulation models, which may introduce uncertainty and inaccuracies, particularly in complex systems with nonlinear behavior.

3. Hardware-Software Interaction: SIL testing focuses primarily on software validation and may overlook the interaction between software and hardware components, leading to potential integration issues.

4. Complexity of Simulation: Building and maintaining accurate simulation models for SIL testing can be complex and time-consuming, requiring specialized expertise and resources.

In conclusion, Software-in-Loop (SIL) testing is a valuable methodology for enhancing vehicle optimization and related fields. Despite its advantages, SIL testing may present certain challenges, including limited real-world validation and simulation uncertainties. However, with careful planning and execution, SIL testing can significantly accelerate development cycles, improve product quality, and drive innovation in vehicle optimization and beyond.

CHAPTER 6

VIRTUAL VEHICLE COMPOSER AND ITS COMPONENTS

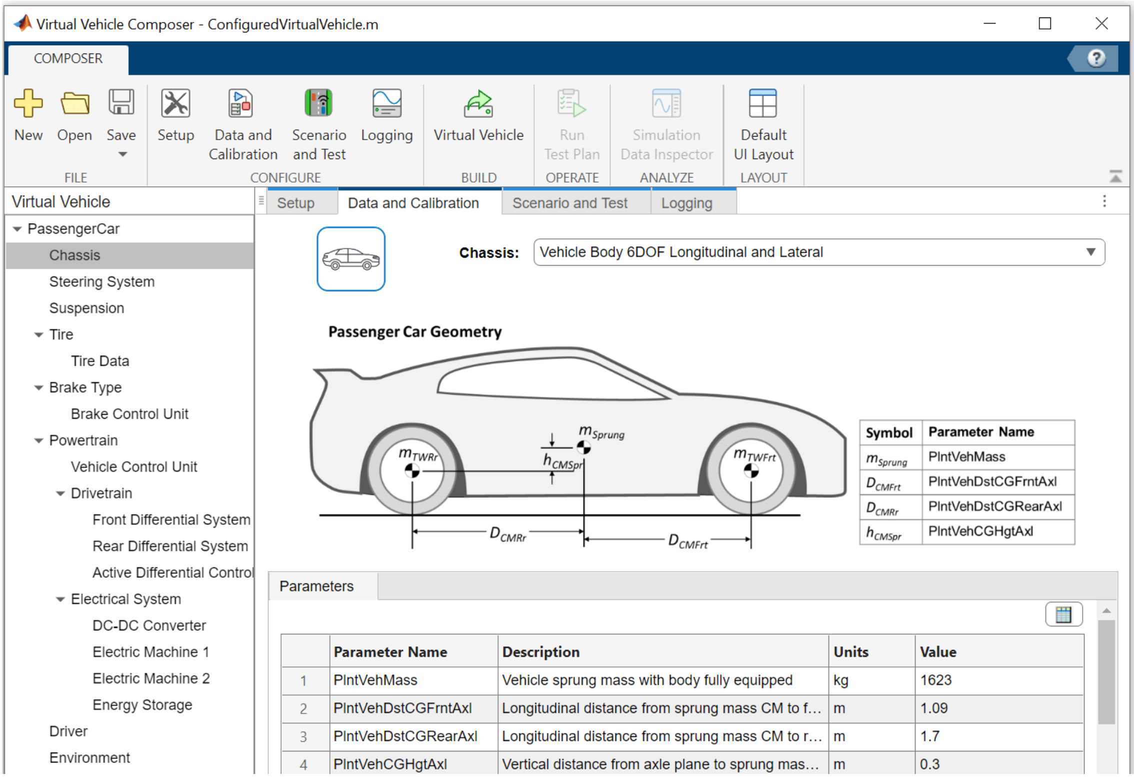

Figure 6: Virtual Vehicle Composer App [14]

To simulate the electric vehicle, EV-2M, we leverage the Virtual Vehicle Composer app provided by MATLAB to simulate the entire electric vehicle seamlessly. This application empowers us to swiftly configure and build a virtual vehicle for comprehensive system-level performance testing and analysis.

The Virtual Vehicle Composer app covers a wide array of functionalities, including component sizing, fuel economy assessment, battery state of charge analysis, drive cycle tracking, vehicle handling maneuvers, software integration testing, and hardware-in-the-loop (HIL) testing. It streamlines the configuration of architecture and inputting parameter data, making the process efficient and user-friendly.

The virtual vehicle model employed here utilizes sets of blocks and reference application subsystems from Powertrain Blockset™, Vehicle Dynamics Blockset™, and Simscape™ add-ons. With Powertrain Blockset, we can configure passenger cars with various powertrain architectures, whether conventional, electric, or hybrid-electric. It also allows us to operate the vehicle under specific test conditions, such as FTP drive cycles, facilitating in-depth design analysis and component sizing[14].

Vehicle Dynamics Blockset comes into play for configuring passenger cars and analyzing their ride-and-handling characteristics through standard test maneuvers. Additionally, it extends its capabilities to configuring and testing motorcycles in various scenarios, including drive cycles and ride-and-handling maneuvers.

Moreover, the Virtual Vehicle Composer app provides a visual representation of the virtual vehicle within the Unreal Engine® simulation environment. This visual aspect enhances our understanding and aids in refining the electric vehicle software efficiently.

The Vehicle Composer app in MATLAB provides a comprehensive platform for configuring and simulating various vehicle variants, including passenger cars and motorbikes, across different powertrain architectures. In our research focused on energy optimization through torque vectoring between front and rear motors, we will specifically explore the EV-2EM variant, which features two electric motors.

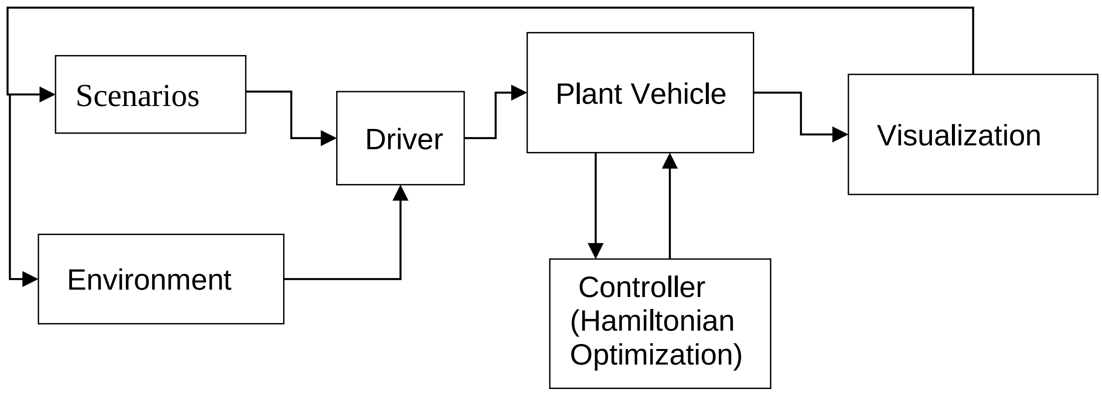

Figure 7: Process flow diagram.

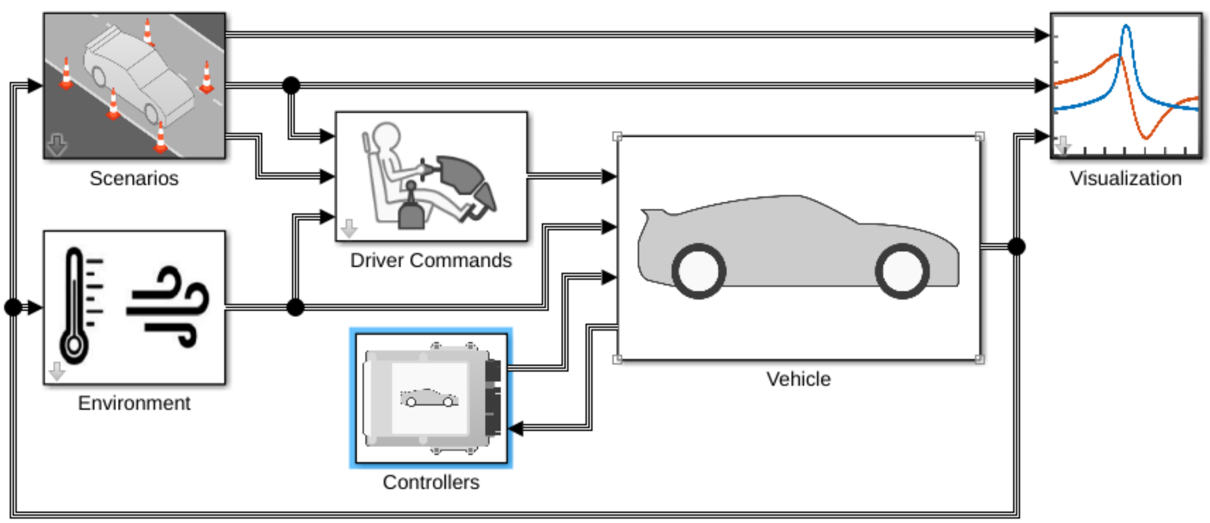

Figure 8: VVC Overview

Environment:

The Environment component within the Vehicle Composer app allows users to define and simulate the external conditions in which the vehicle operates. This includes factors such as road surface, weather conditions, and ambient temperature, which can significantly impact vehicle performance and energy consumption.

Scenarios:

The Vehicle Composer app offers a diverse set of predefined scenarios that simulate various driving conditions and challenges. These scenarios range from urban city driving to highway cruising and off-road terrain traversal. By incorporating these scenarios into our research, we can assess the performance and efficiency of the EV-2EM variant under different real-world driving conditions. This enables us to evaluate the effectiveness of our

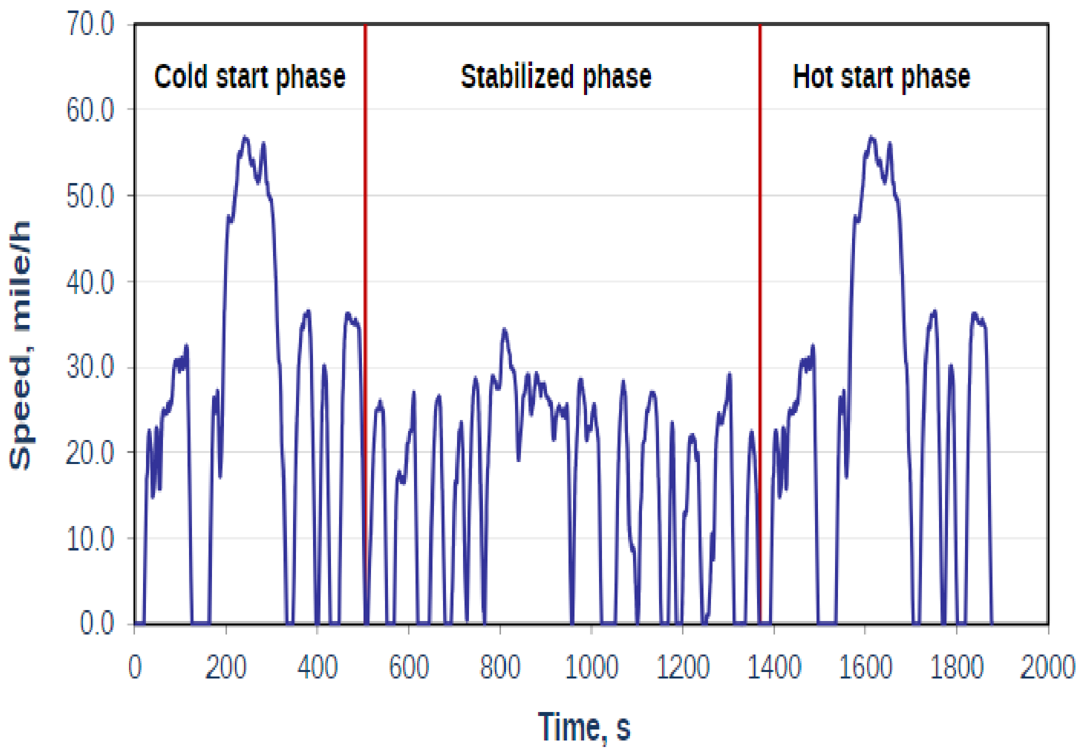

Figure 9: US EPA Urban Dynamometer Driving Schedule (FTP-75) [15]

energy optimization strategies across a wide range of scenarios, providing valuable insights into the practical applicability and robustness of our control algorithms.

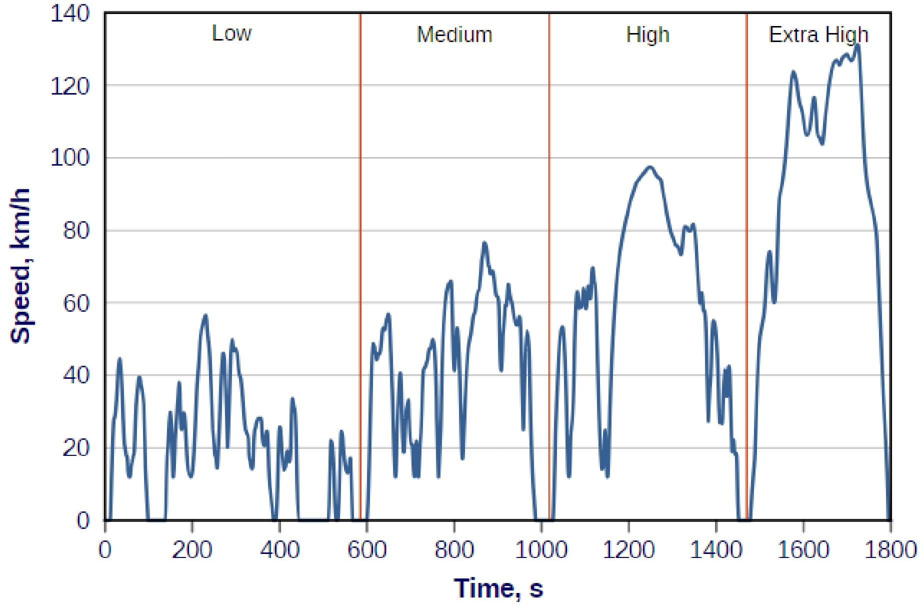

Figure 11: WLTC cycle for Class 3b vehicles.[15]

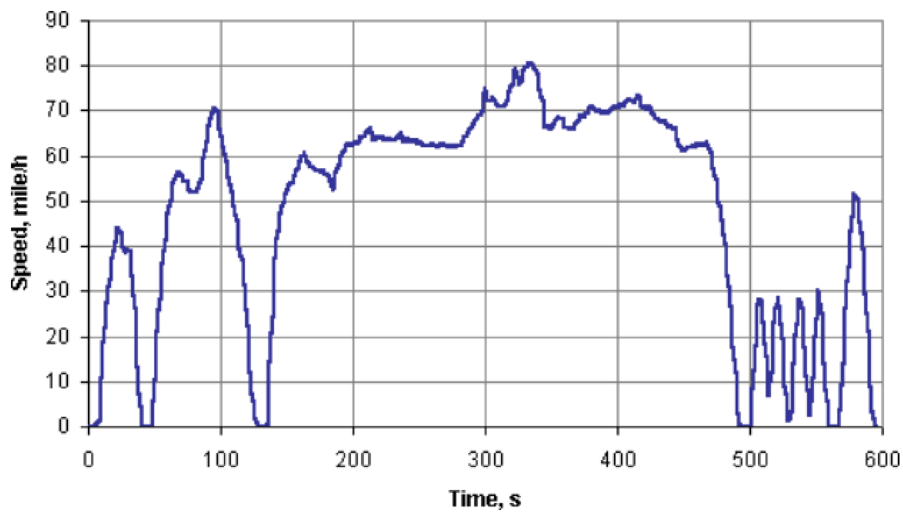

Figure 10: US06 Drive Cycle [15]

Driver:

The Driver module simulates the behavior and input of the vehicle's driver. Users can specify various driving profiles, such as aggressive, moderate, or eco-friendly driving styles, to evaluate the vehicle's response under different driving conditions. This component plays a crucial role in assessing energy consumption and optimizing vehicle performance based on driver behavior.

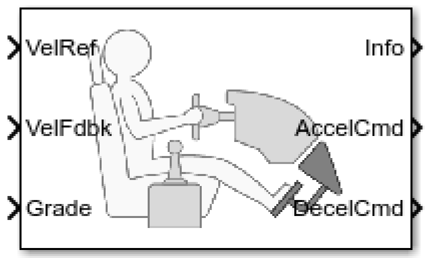

Figure 12: Driver Block

$$y = \frac{K_{ff}}{v_{\text{nom}}} v_{\text{ref}} + \frac{K_p e_{\text{ref}}}{v_{\text{nom}}} + \int \left(\frac{K_i e_{\text{ref}}}{v_{\text{nom}}} + K_{\text{aw}}\, e_{\text{out}} \right) dt + K_g \theta$$

$$e_{ref} = v_{ref} - v$$

$$e_{\text{out}} = y_{\text{sat}} - y$$

$$y_{\text{sat}} = \begin{cases} -1 & y < -1 \\ y & -1 \le y \le 1 \\ 1 & 1 < y \end{cases}$$

[16]

Vehicle Plant:

The Vehicle Plant represents the physical dynamics and characteristics of the vehicle itself. This includes the vehicle's chassis, suspension, tires, and powertrain components. Within the EV-2EM variant, the Vehicle Plant model encompasses the two electric motors, front and rear drivetrains, battery system, and other relevant components necessary for energy optimization through torque vectoring.

i. Vehicle Physics

Vehicle Physics is implemented using the Vehicle Dynamics Blockset, which are equipped with separate sub-blocks dedicated to various components involved, such as suspension, wheels, chassis, and more.

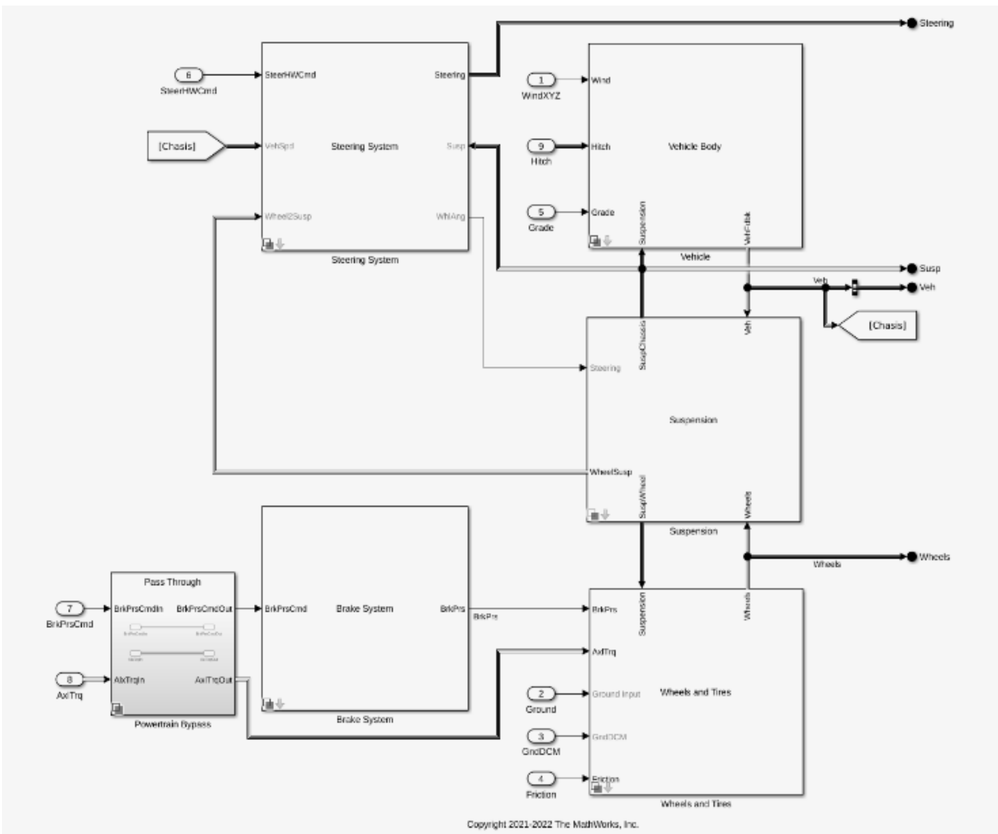

Figure 13: Vehicle Physics [9]

ii. Drivetrain

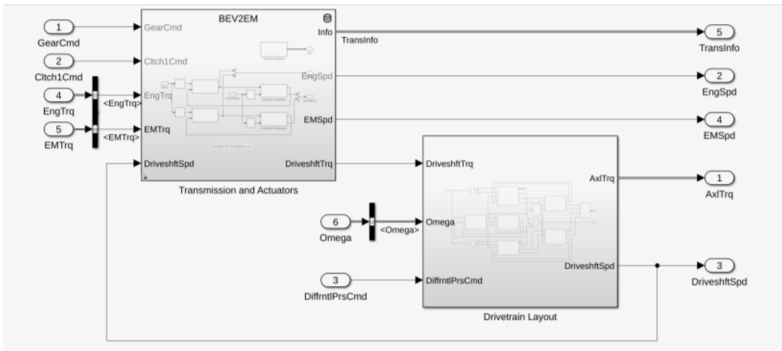

Figure 14: Drivetrain. [9]

iii. Electric Machines

The Mapped Motor block implements mapped motor and drive electronics operating in torque-control mode. Its output torque is designed to follow the torque reference demand, incorporating motor-response and drive-response time constants. It operates under the assumption that speed fluctuations resulting from mechanical loads do not impact the motor's ability to track torque.

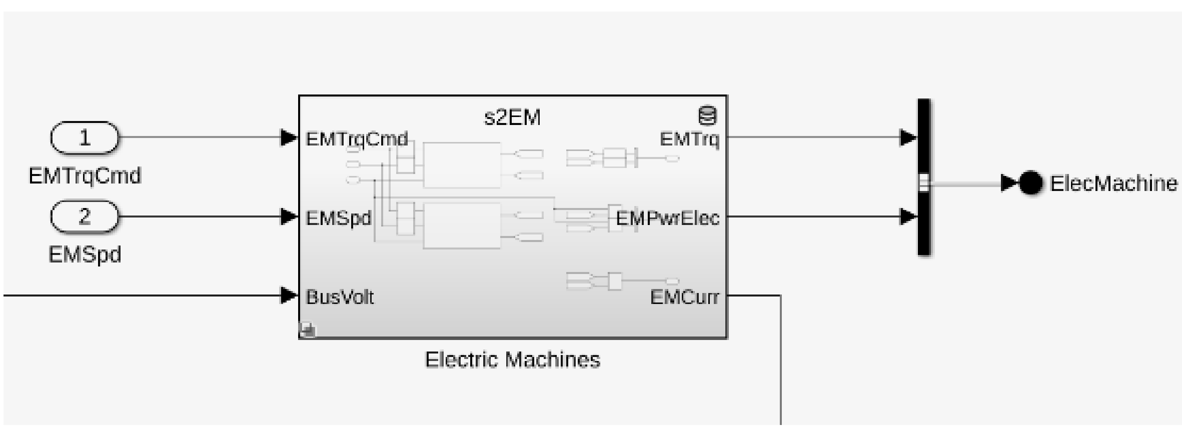

Figure 15: Electric Machines [9]

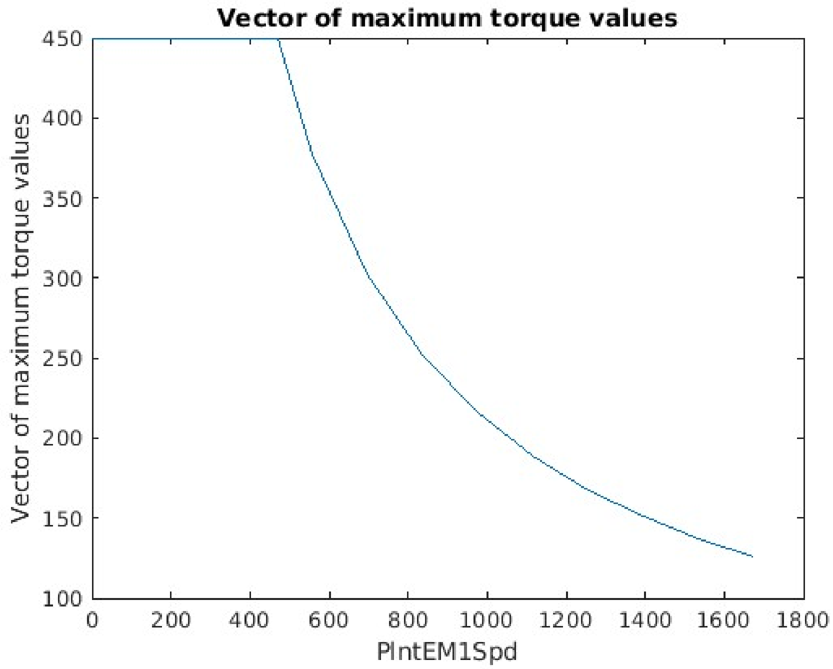

Figure 16: Maximum Torque Plot

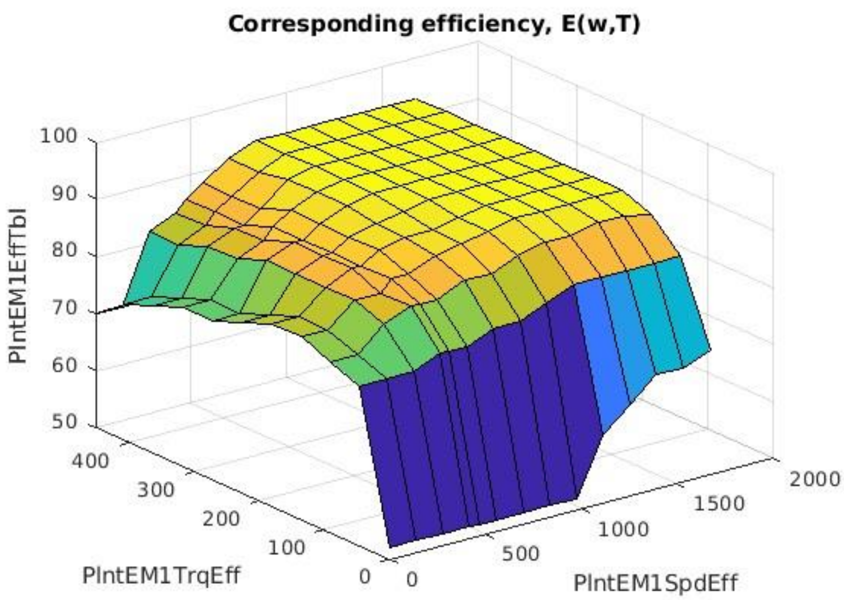

Figure 17: Torque Speed Efficiency Map

iv. Battery Management System

Simulink's Battery Management System (BMS) offers comprehensive control and monitoring for battery-powered systems. From state-of-charge estimation to thermal management, Simulink's BMS ensures optimal utilization and longevity of batteries. Seamlessly integrate BMS into your Simulink models for efficient design and analysis of battery systems.

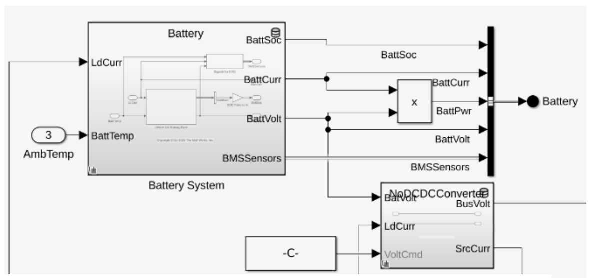

Figure 18: BMS. [9]

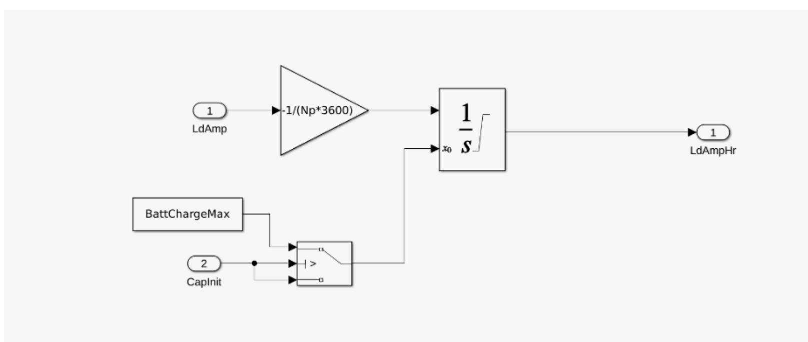

Figure 19: Battery Charge Model [9]

Controller

The Controller component implements control algorithms and strategies to optimize the vehicle's energy efficiency and performance. In our research, the Controller module will focus on torque vectoring algorithms that dynamically adjust the power distribution between the front and rear motors to maximize traction, stability, and energy utilization.

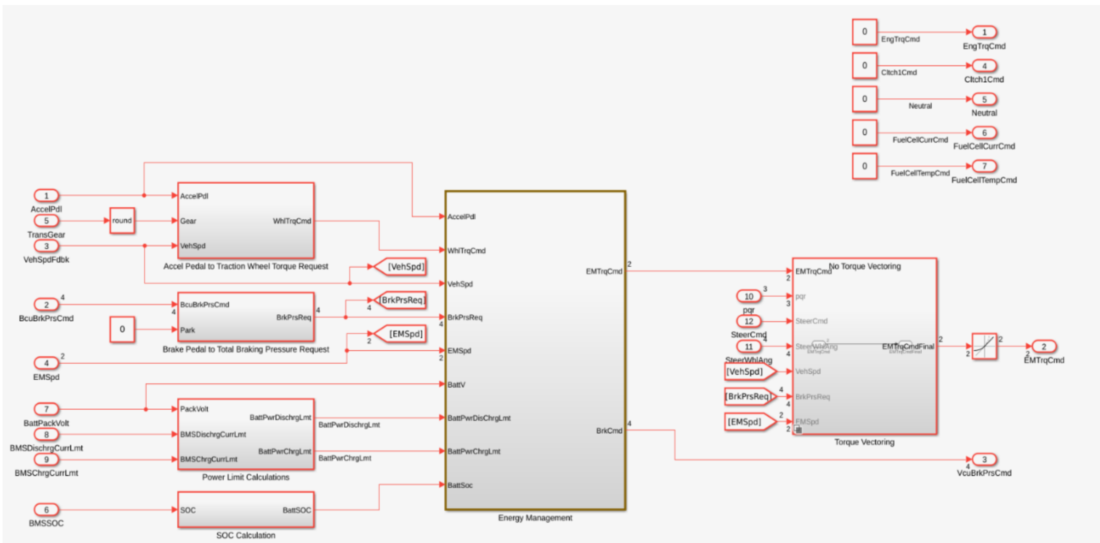

Figure 20: Controller Block [14]

Visualization

The Visualization module within the Vehicle Composer app is responsible for estimating and monitoring the vehicle's states and parameters in real-time. This includes variables such as vehicle speed, acceleration, battery state of charge, and motor torque. The Observer component plays a crucial role in providing feedback to the controller and ensuring accurate control and optimization of the vehicle's operation.

By leveraging the capabilities of the MATLAB Vehicle Composer app and focusing on the EV-2EM variant, we can conduct comprehensive research and analysis to optimize energy efficiency through torque vectoring.

Through simulation and experimentation within this framework, we aim to develop advanced control strategies that enhance the performance and sustainability of electric vehicles.

Controller Design and Operation

The controller block serves as a pivotal component within the vehicle's control system, receiving a primary input known as the Control command. This command variable ranges from -1 to 1, representing the accelerator and brake pedal commands. Upon receiving this input, the controller initiates a series of calculations to determine the torque required for vehicle propulsion or braking.

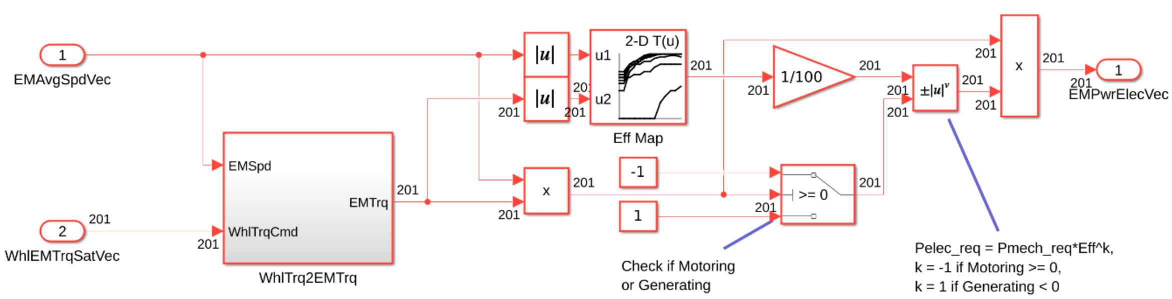

Figure 21: Power vector calculation. [14]

The first step involves interpolating the Control command to calculate the desired torque. Positive torque values denote acceleration, while negative values signify braking torque. Simultaneously, the controller computes the maximum torque that each motor can generate based on the current motor speed.

With the total torque requirement and the maximum torque capacity of each motor determined, the controller proceeds to create a 201x1 equally spaced vector. This vector serves as the decision space for a global search based on the Hamiltonian function.

The Hamiltonian function evaluates the energy of the system at each point in the decision space, considering factors such as motor torque distribution and vehicle dynamics. The goal is to minimize the Hamiltonian function, indicating an optimal torque distribution between the front and rear wheels for efficient vehicle operation.

Upon completion of the global search, the controller identifies the torque distribution configuration where the value of the Hamiltonian function is minimized. This optimized control command is then transmitted to the vehicle plant, instructing it on torque vectorization between the front and rear wheels. By dynamically adjusting torque distribution based on real-time inputs and system constraints, the controller plays a crucial role in enhancing vehicle performance, stability, and energy efficiency.

CHAPTER 7

HAMILTONIAN OPTIMIZATION

As part of the challenge provided by MATLAB, a baseline control algorithm called Hamiltonian Optimization has been introduced. This algorithm draws inspiration from Hamiltonian functions, a concept rooted in classical mechanics and mathematical optimization. In essence, the baseline control algorithm aims to optimize torque distribution between the front and rear motors of the vehicle by leveraging the principles of Hamiltonian functions.

Hamiltonian functions, named after the Irish mathematician Sir William Rowan Hamilton, are mathematical functions commonly used in the field of classical mechanics to describe the dynamics of physical systems. They encapsulate the total energy of a system in terms of its position and momentum variables, providing insights into the system's behavior and evolution over time.

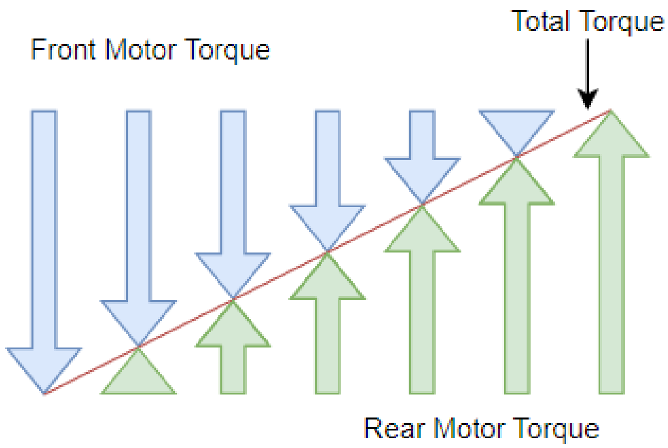

Figure 22: Combination of front and rear torque vector.

In the context of the baseline control algorithm, the Hamiltonian Optimization approach involves dividing the entire control decision space into equally spaced vectors. Subsequently, the algorithm calculates the value of the Hamiltonian function for each discrete point in the decision space to determine the optimal torque distribution between the front and rear motors. However, due to the discrete nature of the decision points, the resulting torque distribution may not always be optimal.

To gain a deeper understanding of how the Hamiltonian Optimization algorithm works and its implications for vehicle optimization, let us delve into its underlying principles and operational mechanisms. Through this exploration, we can uncover insights into the efficacy and limitations of this baseline control algorithm in optimizing vehicle performance and energy efficiency.

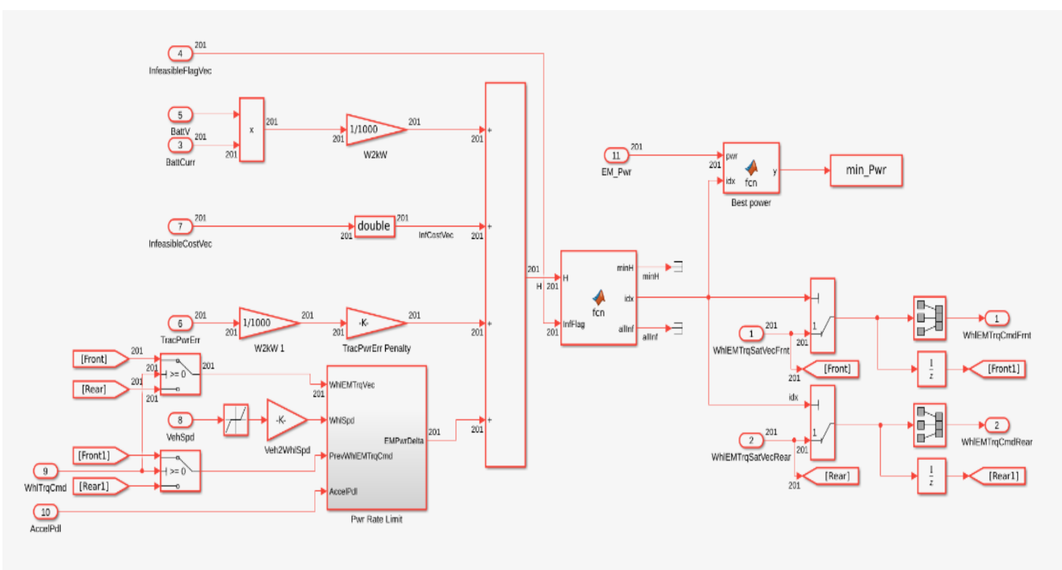

Figure 23: Hamiltonian function [14]

CHAPTER 8

NMPC CONTROLLER

The NMPC (Nonlinear Model Predictive Control) controller is integrated into the controller block, serving as an alternative to the Hamiltonian optimization block. Following a similar framework as the baseline controller, the NMPC controller computes the total torque required based on the acceleration/braking command provided to the system. Additionally, it supplies the essential system parameters to the NMPC block, facilitating the creation of an accurate predictive model of the vehicle.

The NMPC controller aims to minimize the cost function by iteratively adjusting the control inputs over a finite prediction horizon. It solves the optimization problem subject to various constraints, including:

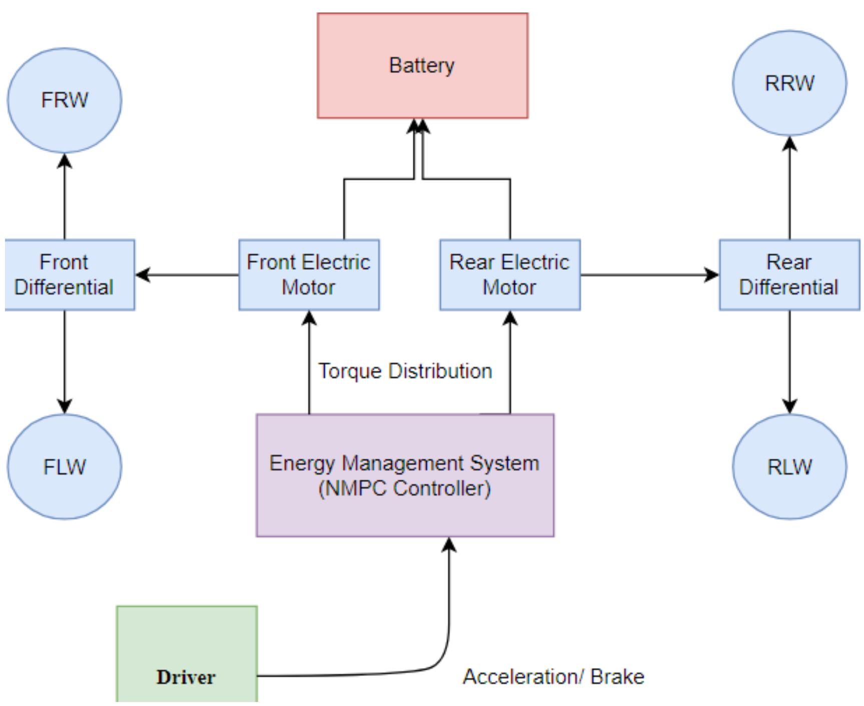

Figure 24: EV-2M Architecture

System

1. **System Dynamics**: Constraints imposed by the vehicle's dynamics and physical limitations, ensuring that the control inputs remain within feasible ranges.

   State Variables:

   i. x_1 - State of charge of the battery at time step k (SOC)

   ii. x_2 - Velocity of the center of gravity (C.G.) at time step k

   State Equations:

$$\dot{x}_1 = -\frac{\text{Battery Voltage} \times \text{Battery Current}}{E_{\text{batt\_Max}}}$$

$$\dot{x}_2 = -\frac{F_x}{M}$$

2. **State Constraints:** Restrictions on the vehicle's states, such as velocity, position, or battery state of charge, to ensure safe and stable operation.

3. **Control Input:**

   Torque vectorization parameter, u.

4. **Control Constraints:** Limits on the control inputs, such as torque distribution between front and rear motors, to prevent excessive energy consumption or vehicle instability.

   Thus u ∈ [0,1].

Objective function

$$\sum_{k=0}^{N-1} \left(\frac{P_{out\ 1}(k)}{\eta_{out\ 1}(k)} + \frac{P_{out\ 2}(k)}{\eta_{out\ 2}(k)} - P_{in1}(k)\eta_{in1}(k) - P_{in2}(k)\eta_{in2}(k)\right)$$

Optimization Problem

$$\min_{T_1,T_2} J = \sum_{k=0}^{N-1} \left(\frac{P_{out1}(k)}{\eta_{out1}(k)} + \frac{P_{out2}(k)}{\eta_{out2}(k)} - P_{in1}(k)\eta_{in1}(k) - P_{in2}(k)\eta_{in2}(k)\right)$$ [1]

$$s.t.\ \omega_{m\_min} \leq \omega_m(k) \leq \omega_{m\_max}$$

$$T_{min}(\omega_m) \leq T_i(k) \leq \min\{T_{max}(\omega_m), T_{i\_max}(k)\}$$

$$\mathrm{SOC}_{min} \leq \mathrm{SOC}(k) \leq \mathrm{SOC}_{max}$$

By solving the above optimization problem in real-time and generating optimal control inputs, the NMPC controller enables precise and adaptive control of the vehicle's behavior. This approach enhances vehicle performance, efficiency, and safety across various operating conditions, making it a valuable tool in advanced vehicle control systems.

IMPLEMENTATION METHODOLOGY

Model

1. State Variables:

   i. SOC (State of Charge): Represents the current charge level of the vehicle's battery.

   ii. v_curr (Current Velocity): Current velocity of the vehicle.

2. Input Variables:

   i. Torque_demand: Desired torque demanded by the vehicle.

ii. req_data: Data required for torque and speed interpolation.

iii. current_timestep: Current time step.

iv. EMspeed (Motor Speed): Speed of the electric motor.

3. Output Variables:

i. U_f, U_r: Front and rear motor torque commands.

ii. min_Pwr: Minimum power consumption.

iii. flag: Flag indicating the optimization result.

4. Parameters:

i. Vehicle constants like mass, wheel radius, aerodynamic coefficients, gear ratio, inertia, dimensions, etc.

ii. Control parameters such as number of time steps (N), sampling time (Ts), and initial control inputs (u0).

Dynamics Overview:

i. State Update: The code iterates over a prediction horizon (N time steps) to calculate the future velocity (v_ref) based on a velocity reference vector. It then updates the vehicle's velocity over each time step using the applied torque, aerodynamic resistance, rolling resistance, and mass dynamics.

ii. Torque Calculation: The torque demand is calculated based on the desired acceleration, aerodynamic resistance, and rolling resistance.

iii. Optimization: The code utilizes an optimization algorithm (fmincon) to find optimal front and rear motor torque commands (U_f and U_r) that minimize power consumption while meeting torque demand and motor efficiency constraints.

iv. Constraints: Constraints are applied to ensure the state of charge (SOC) of the battery remains within predefined limits, and motor torque commands adhere to motor torque limits.

v. Objective Function: The objective function aims to minimize the deviation between the actual and reference power while penalizing excess power generation.

Model Dynamics

The model equations include:

i. Newton's second law to calculate vehicle acceleration.

$$M_{veh}\frac{dv_{veh}}{dt} = F_{\text{trac}} - F_{\text{roll}} - F_{\text{aero}}$$

$$F_{\text{trac}} = F_{pwt} - F_{\text{brake}}$$

$$F_{\text{roll}} = (c_{r0} + c_{r1}v_{veh})M_{\text{veh}}g$$

$$F_{\text{aero}} = \frac{1}{2}\rho_{\text{air}}A_f C_d v_{veh}^2$$

ii. Power balance equations considering motor efficiency, torque demand, and speed.

Power(k) = $(\frac{P_{\text{out1}}(k)}{\eta_{\text{out1}}(k)} + \frac{P_{\text{out2}}(k)}{\eta_{\text{out2}}(k)} - P_{\text{in1}}(k)\eta_{\text{in1}}(k) - P_{\text{in2}}(k)\eta_{\text{in2}}(k))$

iii. Constraints on battery state of charge and motor torque limits.

- u_1 < Torque limit for motor 1

- u_2 <Torque limit for motor 2
- Charge > x_1 > Discharge limit.

Solver: fmincon

The optimization framework implemented in this research leverages the fmincon solver from the MATLAB Optimization Toolbox. fmincon stands as a robust tool for solving constrained optimization problems, particularly suited for scenarios where both the objective function and constraints are nonlinear in nature.

Algorithm:

- fmincon adopts SQP methods to tackle constrained optimization problems. These methods navigate iteratively from an interior point of the feasible region towards the optimal solution while adhering to the specified constraints.
- This solver is adept at handling optimization challenges characterized by a substantial number of constraints and variables.

Usage:

- Users of fmincon are required to provide the objective function, constraints (both equality and inequality), initial guess, and any additional parameters essential for optimization.
- The function syntax typically resembles:

[x, fval, exitflag, output, lambda] = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, options)

Features:

- fmincon offers flexibility in optimization goals, accommodating minimization or maximization of the objective function.
- It supports the inclusion of both linear and nonlinear constraints.
- Decision variable bounds can be specified to refine the optimization process.
- Diverse options are available to fine-tune solver behavior, encompassing tolerance levels, maximum iterations, and display preferences.

Performance:

- While ‘fmincon’ demonstrates efficiency for medium-sized optimization problems, scalability concerns may arise with very large-scale problems due to computational and memory demands.
- Performance is contingent upon the problem's characteristics, including the smoothness of the objective function, presence of constraints, and the chosen starting point.

Output:

Upon successful completion, fmincon furnishes the optimized solution (x), objective function value at the solution (fval), exit flags denoting optimization status (exitflag), optimization output (output), and, if requested, Lagrange multipliers (lambda).

Considerations:

- Prudent selection of optimization options and algorithm parameters is crucial to strike a balance between solution accuracy and computational efficiency.
- Sensitivity to initial guesses and constraint violations should be duly

acknowledged.

- Proper scaling of variables and constraints can enhance solver performance.
- Compatibility:
- fmincon is compatible with MATLAB versions equipped with the Optimization Toolbox.

In this thesis, fmincon serves as the cornerstone of the optimization framework, adeptly navigating the solution space to yield efficient and reliable outcomes while adhering to the formulated constraints.

CHAPTER 9

EXPERIMENTAL SETUP

Vehicle Parameters [14]

Define the constants related to the vehicle, including mass, wheel radius, coefficients of friction and air drag, air density, surface area, final gear ratio, and motor inertia. These constants are crucial for accurately modeling the vehicle dynamics.

Parameter	Value	Description
Vehicle Mass	1623 kg	Mass of the vehicle
Wheel Radius	0.327 m	Effective radius of the wheel
Coefficient of Friction	0.012	Friction coefficient between tires and road
Air Drag Coefficient	0.389	Coefficient of drag for air resistance
Air Density	1.202 kg/m^3	Density of air, also used for battery calculations
Surface Area	2.27 m^2	Surface area of the car
Final Gear Ratio	3.32	Gear ratio for final drive

Motor Inertia	0.01 kg/ m^2	Inertia of the motor

Implementation

To run the NMPC energy optimization for an electric vehicle (EV) in Matlab, follow these steps:

1. Open the Virtual Vehicle from Matlab apps.

- Launch Matlab and navigate to the apps tab.
- Open the Virtual Vehicle app.

2. Click on the "new" on the left-hand top-side icon. Select the EV-2M model from Powertrain architecture and click on the configure icon.

- Create a new project, select the EV-2M model from the Powertrain architecture, and click on the configure icon.

3. Build the model using the virtual vehicle build icon in the virtual vehicle composer.

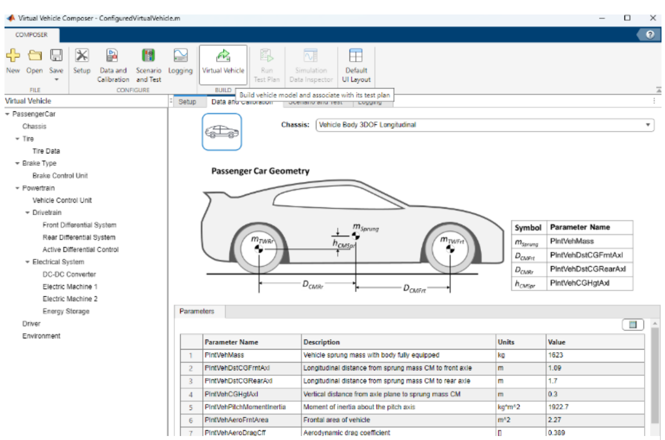

Figure 25: Instructions [14]

4. Once the ConfiguredVirtualVehicleModel is open, go inside the controller\VehicleControlUnit\EnergyManagement\.

- Navigate to the EnergyManagement section within the VehicleControlUnit controller in the ConfiguredVirtualVehicleModel.

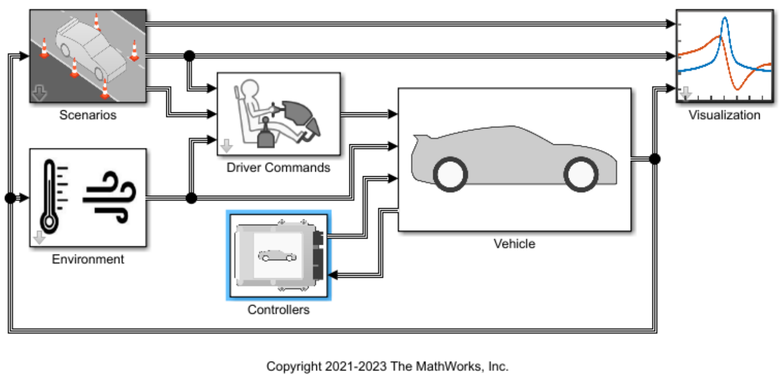

Figure 26: Instructions step -2. [14]

5. Create a MATLAB Simulink user-defined function. Copy the code from EVFmincon_test_v4.m inside the block and name it NMPC controller.

- Inside Energy Management, create a MATLAB Simulink user-defined function block.
- Copy the code from EVFmincon_test_v4.m into the block and name it NMPC controller.

6. Create a digital clock block, a subtract block, and a memory block just outside the NMPC controller block.

- Place a digital clock block, a subtract block, and a memory block adjacent to the

NMPC controller block.

7. Connect the output of the digital clock to the plus input of the subtract block. Connect the output of the subtract block to the memory block. Connect the output of the memory block to calculate the variable step size of the model during each iteration.

- Establish connections as described: digital clock to subtract block, subtract block to memory block, and memory block to control the variable step size.

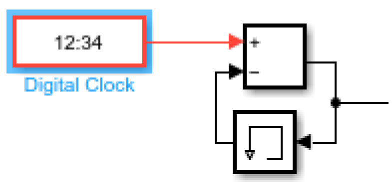

Figure 27: Instructions: Clock [14]

8. Run the global_init.m file in matlab to load the necessary data in Simulink.

- Execute the global_init.m file to load the required data into Simulink.

9. Create a constant block. Select the struct variable “custom_struct” in the main option of the block. Under attributes, select the “Bus:Data_bus” as the output data type.

- Create a constant block, choose "custom_struct" in the main options, and set the output datatype to "Bus:Data_bus" under attributes.

10. Delete the connection between WhlTrqCmdFrnt coming out of the “Hamiltonian computation and minimization block” to the “Convert Whl to EMtrq Vector block” and the “WhlEMTrqCmdFrnt" Tag. Attach it to a terminator block. Do the same for

WhlTrqCmdRear.

- Disconnect WhlTrqCmdFrnt and WhlTrqCmdRear from the "Hamiltonian computation and minimization block" and connect them to terminator blocks.

11. Start making the connections.

- Connect a branch from the output block of the subtract block to the input parameter “dt” of the NMPC controller block.
- Connect a branch from the main digital clock to the current_time input port of the NMPC function.
- Connect the constant block to the req_data block of NMPC.
- Connect the SOC to BattSoc (indicated by signal 9).
- Connect v_curr to the “VehSpd” signal.
- Connect Torque_demand to “WhlTrqCmdOut”.

12. Attach the outputs of the NMPC block U_f and U_r to WhlEMTrqCmdFrnt and WhlEMTrqCmdRear in the “Convert Whl to EMtrq Vector" block and tags.

- Establish connections between the NMPC block outputs (U_f and U_r) and the

corresponding inputs in the “Convert Whl to EMtrq Vector " block, ensuring proper tagging.

13. Once done, the subsystem should look like this.

- Verify that the subsystem matches the provided configuration.

14. Move back to the ConfiguredVirtualVehicleModel and press run.

- Return to the ConfiguredVirtualVehicleModel and initiate the simulation by clicking on the run button.

By following these steps, you should be able to successfully run the NMPC energy optimization for the electric vehicle model in MATLAB.

MPC parameters

Ts = varied for experimentation

Symbol	Parameter	Value
Np	Prediction Horizon	varied for experimentation
Nu	Control Horizon	10
Q	Control action weight matrix	1e03
R	Control action rate weight matrix	1e02

Algorithm: SQP (Semi Quadratic Programming)

CHAPTER 10

PERFORMANCE METRICS AND EVALUATION

In evaluating the performance of the controllers, the FTP75 driving cycle served as the benchmark. The primary criterion for performance assessment centered on State-of-the-art SOC consumption and speed tracking error. Leveraging Model Predictive Control (MPC), optimization occurred over a horizon, enabling the system to capitalize on the battery's function as an energy buffer while constraining control input variations to a relatively narrow range. Notably, the speed tracking error for all controllers remained within ±2 mm/ss, indicating precise speed control across the board.

Experimentation

Throughout my experimentation in MATLAB Simulink, I conducted a comprehensive analysis of various versions of Electric Vehicle Energy Optimization via torque distribution between the front and rear motor. Each version was meticulously designed and tested to explore different parameters and configurations, aiming to achieve optimal energy efficiency and performance. Here is a detailed description of the experiments conducted:

Nominal MPC:

The Nominal MPC version represents an initial exploration into the realm of Electric Vehicle Energy Optimization. By employing Nominal Model Predictive Control (MPC) with a horizon of N=4 and a timestep of Ts=10ms, this version serves as a foundational step in comparing and evaluating the efficacy of MPC against Hamiltonian Optimization methods. MPC, a powerful control technique, optimizes control actions over a finite horizon while considering system dynamics and constraints. By utilizing MPC, the objective is to achieve optimal torque distribution between the front and rear motors of an electric vehicle, thereby enhancing energy efficiency and performance. Through this experimentation, insights into the benefits and limitations of MPC in the context of energy optimization are sought, laying the groundwork for subsequent iterations and refinements. The comparison with Hamiltonian Optimization offers valuable insights into the relative strengths of these approaches, guiding future research and development efforts in the

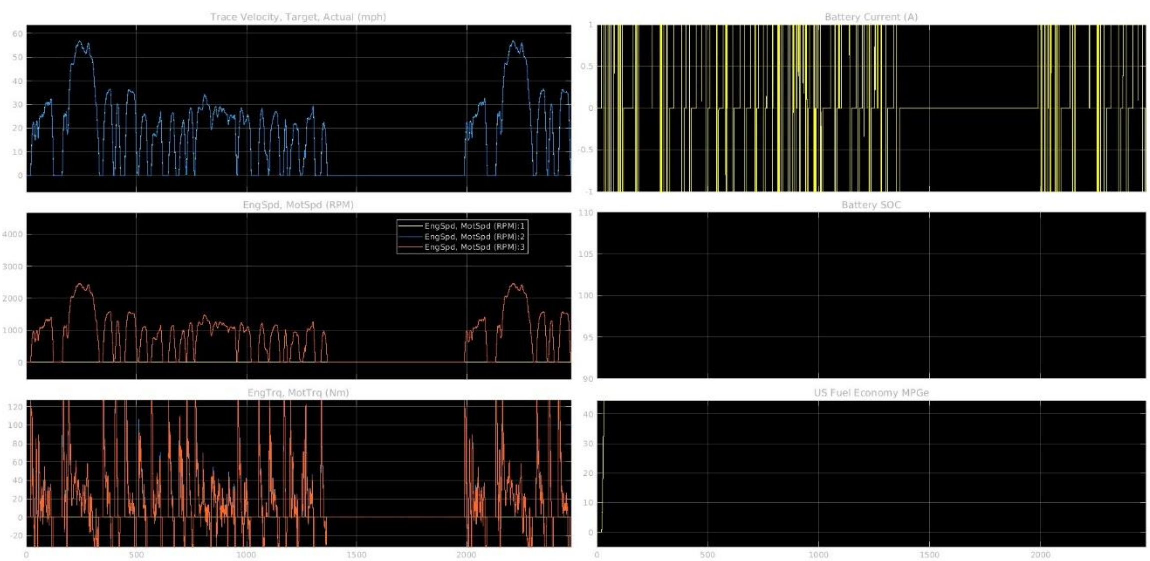

Figure 28: Results for Nominal NMPC from VVC App. [14]

Fixed-Timestep NMPC with Fixed Timestep Model:

The Fixed-Timestep NMPC with Fixed Timestep Model represents a deliberate refinement in the exploration of Electric Vehicle Energy Optimization. Employing a fixed timestep for both the vehicle model and the Model Predictive Control (MPC) algorithm, with N=4 and Ts=10ms, this version facilitates a controlled comparison with previous iterations while offering nuanced insights into the impact of fixed timestep on optimization outcomes. By maintaining a consistent timestep throughout the simulation, this configuration ensures stability and reproducibility, enabling a comprehensive evaluation of the optimization process. Furthermore, the fixed timestep approach provides valuable insights into the effects of discretization on MPC performance, shedding light on potential limitations and trade-offs associated with this method. Through meticulous analysis of optimization outcomes, including torque distribution and energy consumption patterns, this experiment aims to elucidate the implications of fixed timestep modeling in the context of electric vehicle systems. The findings gleaned from this iteration will contribute to refining optimization strategies and advancing the understanding of MPC's applicability in real-world scenarios.

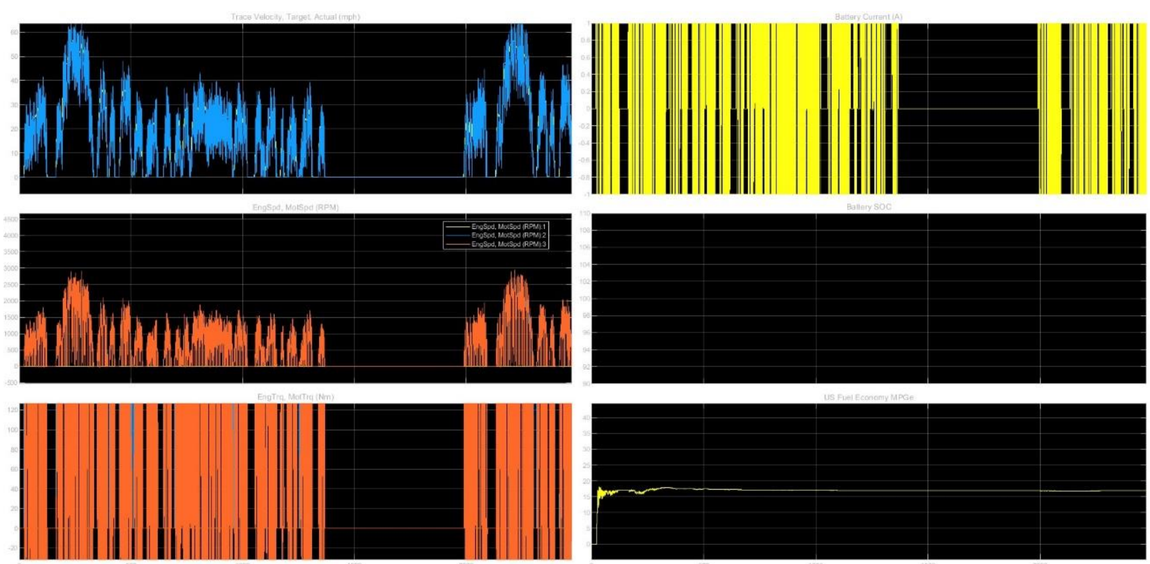

Figure 29: Results for Fixed Time step NMPC and Dynamics Model [14]

Optimized Parameter NMPC (N=10, Ts=100ms) for Variable Timestep Plant:

In the Optimized Parameter NMPC (N=10, Ts=100ms) for Variable Timestep Plant iteration, a significant evolution in the experimental setup is evident. Following iterative refinement, this version embraces a variable timestep vehicle model alongside a fixed timestep Model Predictive Control (MPC) configuration, with N=10 and Ts=100ms. The decision to increase both the horizon and timestep underscores a strategic effort to elevate optimization accuracy and efficiency. By extending the horizon to N=10, the MPC algorithm gains a broader perspective, allowing for more informed decision-making over a longer prediction horizon. Additionally, the adoption of a larger timestep of Ts=100ms facilitates smoother computation and reduces computational overhead, potentially enhancing overall efficiency. This adjustment aims to strike a balance between computational complexity and optimization performance, aiming to achieve a favorable trade-off between accuracy and computational cost.

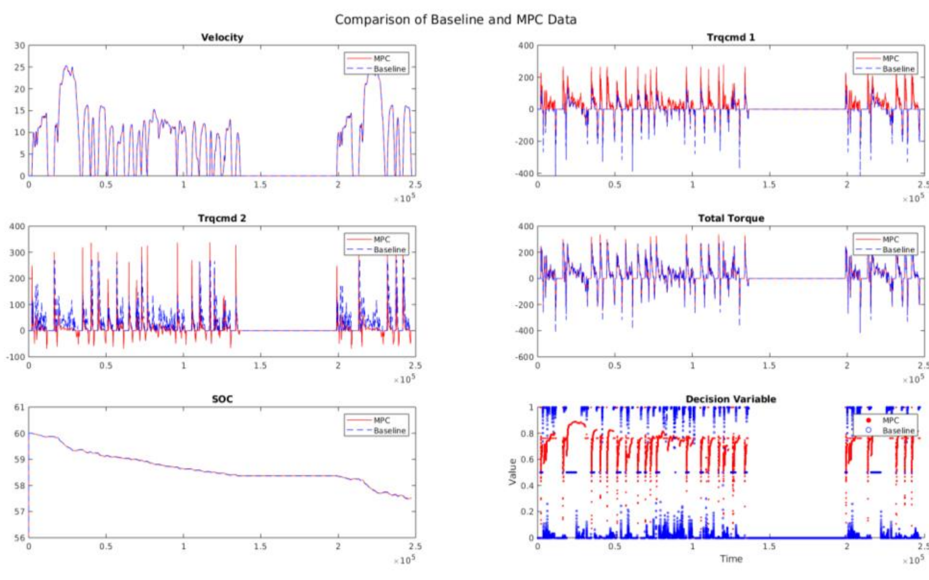

Figure 30: Optimized NMPC Visualization

Point Analysis:

Case 1: NMPC Power consumption < Baseline Power Consumption

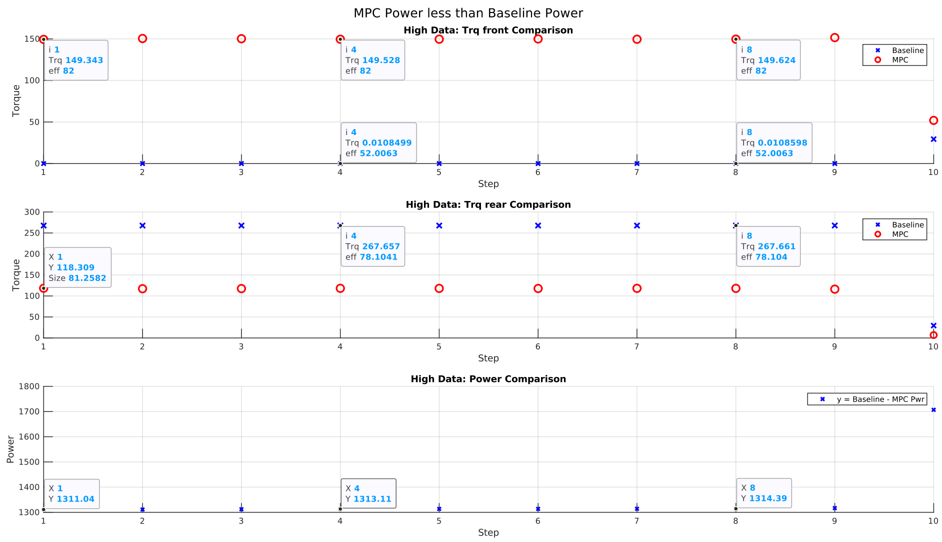

Figure 31: High Point Analysis- Optimized NMPC

Case 2: NMPC Power consumption > Baseline Power consumption

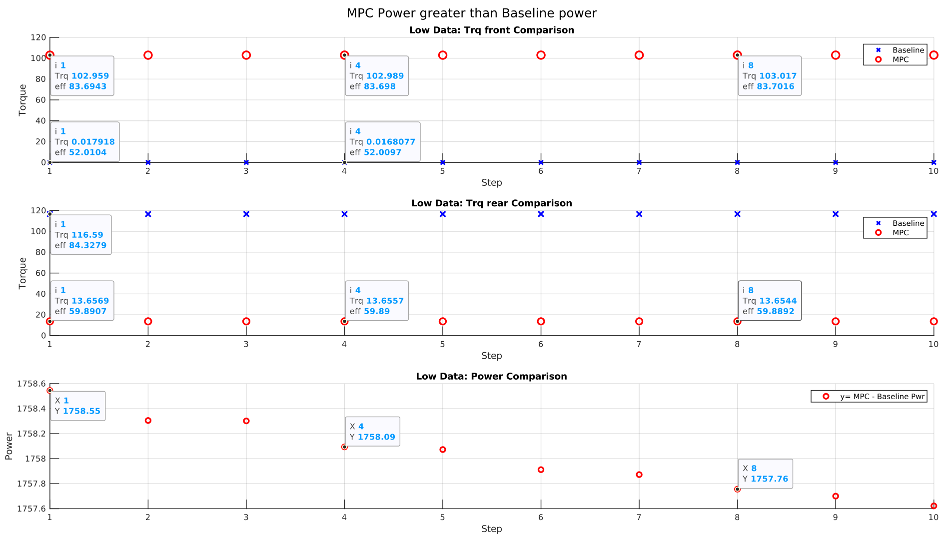

Figure 32: Low Point Analysis- Optimized NMPC

Case 3: NMPC Power consumption < Baseline Power consumption (Middle Range)

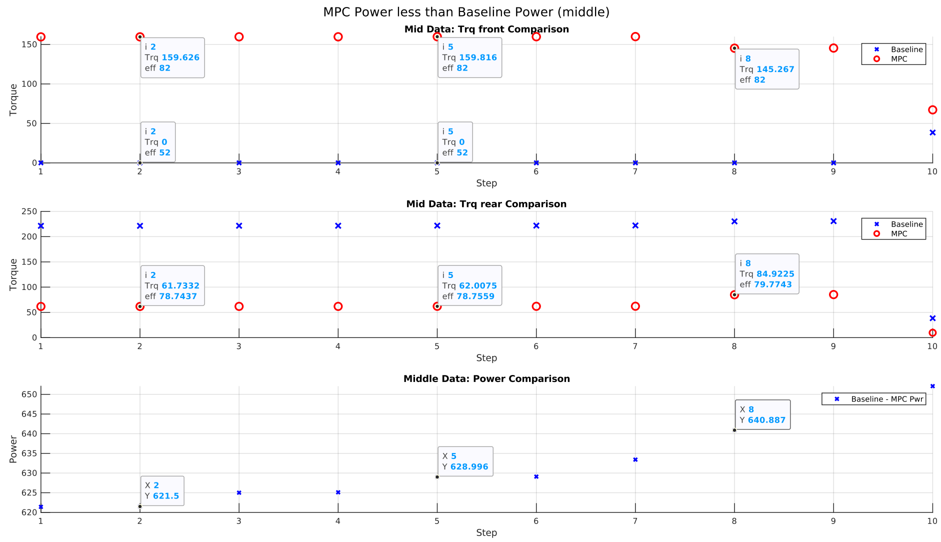

Figure 33: Medium Point Analysis- Optimized NMPC

Control Policy Comparison:

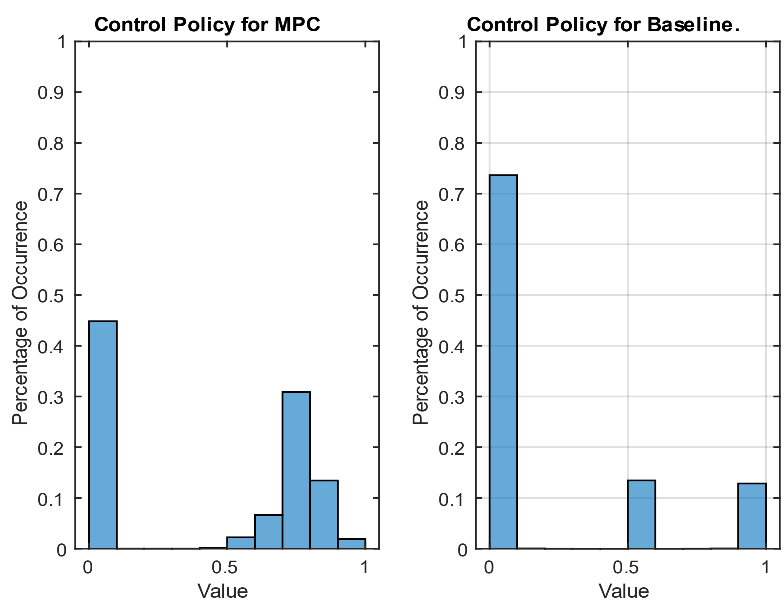

Figure 34: Control Policy- Optimized NMPC

Optimized Parameter NMPC with Delta U penalty:

In this experiment, a progressive refinement in optimization objectives is pursued. Expanding upon the previous version, this iteration introduces a penalty for the rate of change of control in addition to trace power, with N=4 and Ts=10ms. Like its predecessor, only the applied optimized control parameter extracted from the N-length control vector is fed back to the Nonlinear Model Predictive Control (NMPC) algorithm. This selective feedback mechanism remains instrumental in expediting convergence towards the optimal solution. By incorporating punishment for the rate of change of control, the optimization process seeks to further refine control actions and enhance energy consumption patterns.

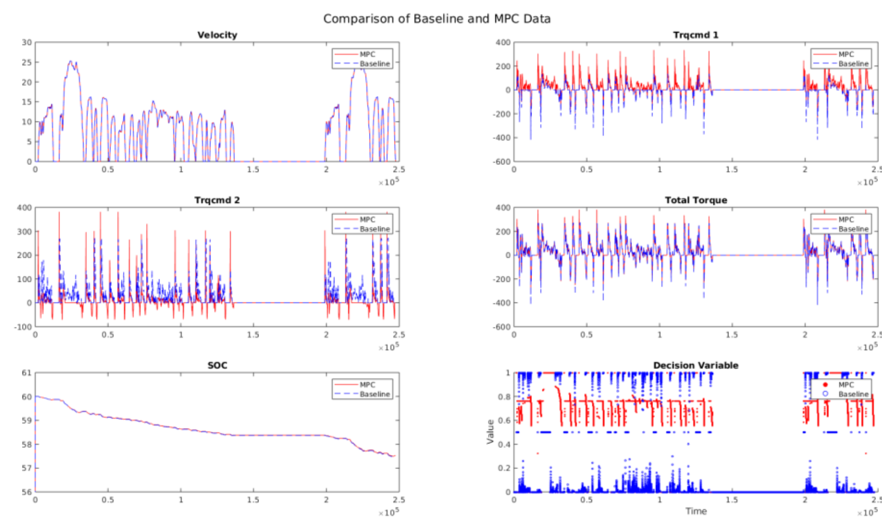

Figure 35: Comparison of Baseline and Delta U penalty NMPC.

Point Analysis:

Case 1: NMPC Power consumption < Baseline Power Consumption

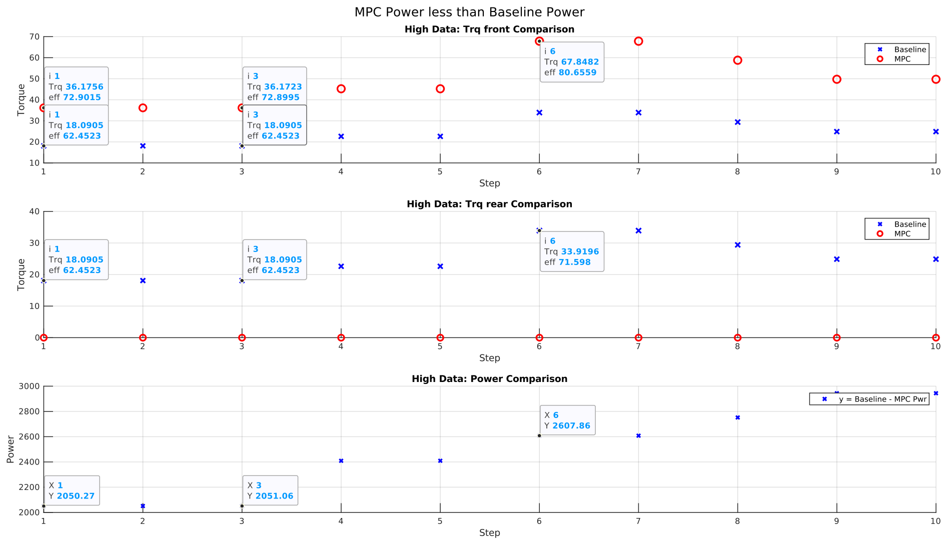

Figure 36: High Point Analysis- Delta U NMPC

Case 2: NMPC Power consumption > Baseline Power consumption

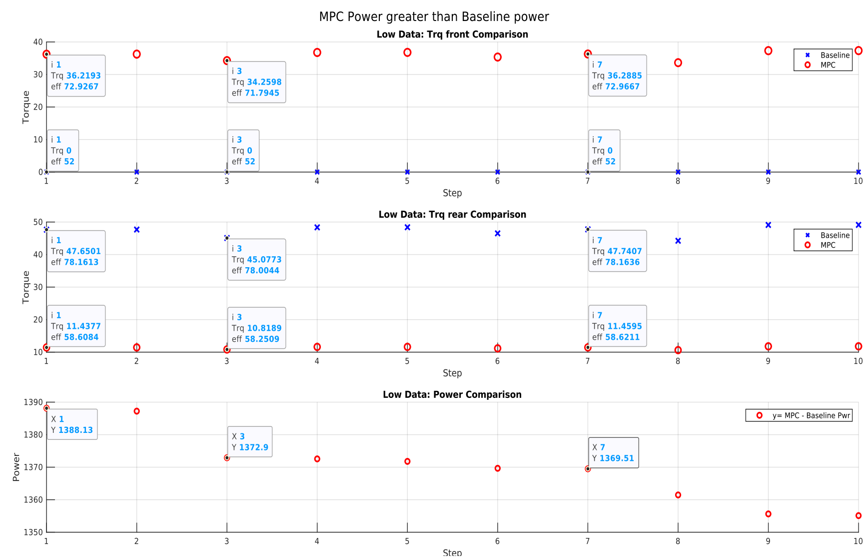

Figure 37: Low Point Analysis- Delta U NMPC

Case 3: NMPC Power consumption < Baseline Power consumption (Middle Range)

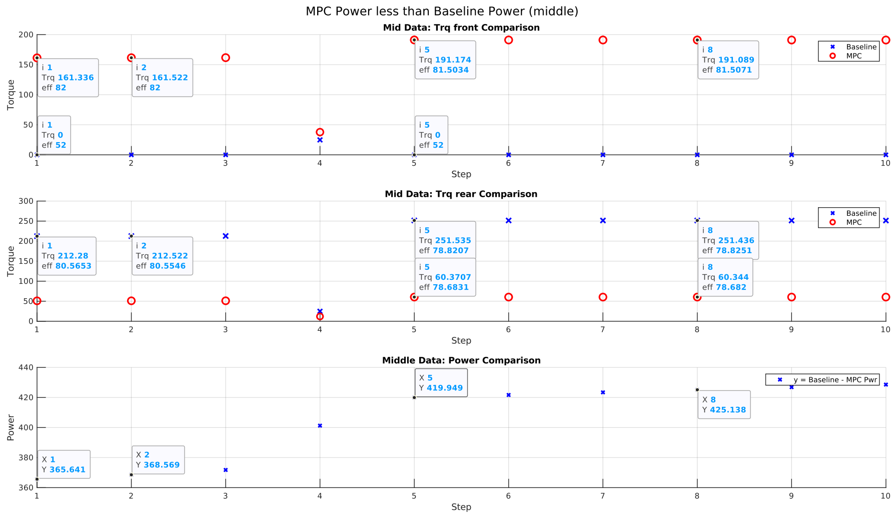

Figure 38: Medium Point Analysis- Delta U NMPC

Control Policy Comparision

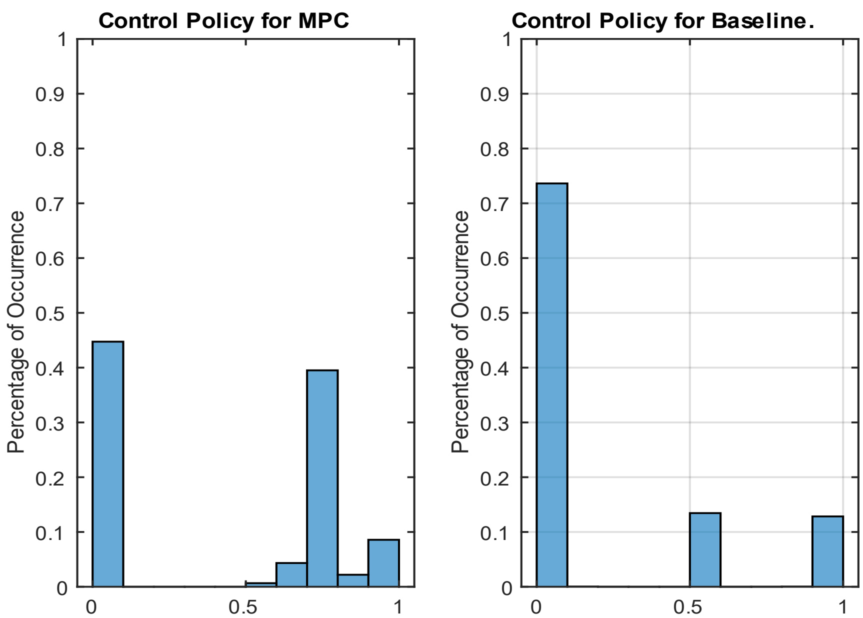

Figure 39: Control Policy- Delta U NMPC

Optimized Parameter NMPC with Control feedback (N=10, Ts=100ms/200ms/300ms...):

In the series of experiments conducted with NMPC with Control feedback, varying decision feedback policies were investigated to discern their influence on optimization outcomes. With N=10 and different timestep parameters ranging from Ts=100ms to Ts=400ms. The utilization of feedback from past decision policies in Nonlinear Model Predictive Control (NMPC) is beneficial in that it initializes the decision variable for the subsequent iteration of optimization. This initialization occurs in the proximity of the previously computed optimal control policy. However, a notable drawback arises due to the inherent nature of Model Predictive Control (MPC), as it does not guarantee convergence to global minima. Consequently, incorporating feedback of the control vector may compromise MPC's ability to reach global minima, potentially leading to convergence at local minima instead.

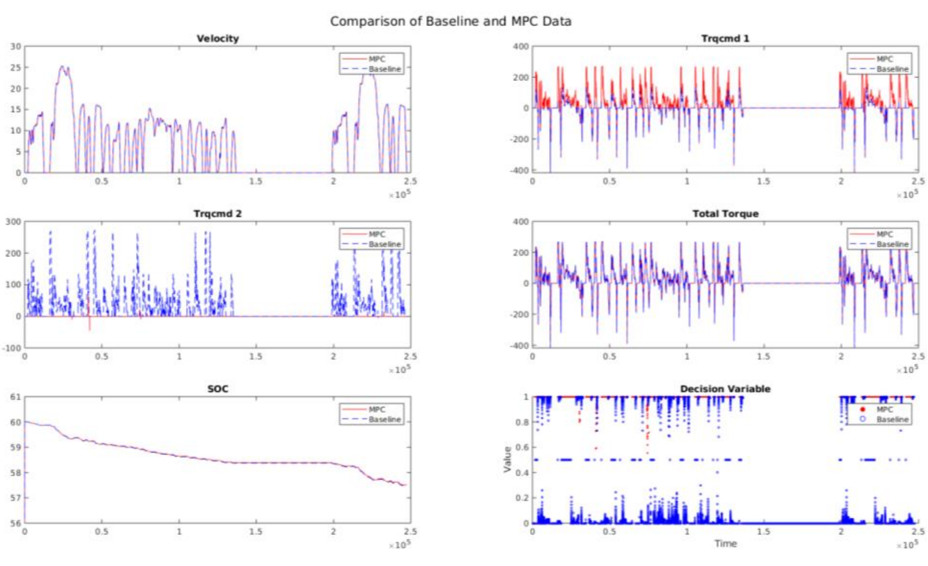

Figure 40: Full feedback NMPC.

Point Analysis:

Out of the entire NMPC run, we've isolated 20 points based on the disparity in power consumption between the Baseline and MPC strategies. Specifically, we've identified 10 points where the difference (Baseline power – MPC power) was most pronounced, as well as 10 points where it was least significant.

Additionally, the plot illustrates the distribution of torque and the corresponding motor efficiency at which it would have been supplied.

Case 1: NMPC Power consumption < Baseline Power Consumption

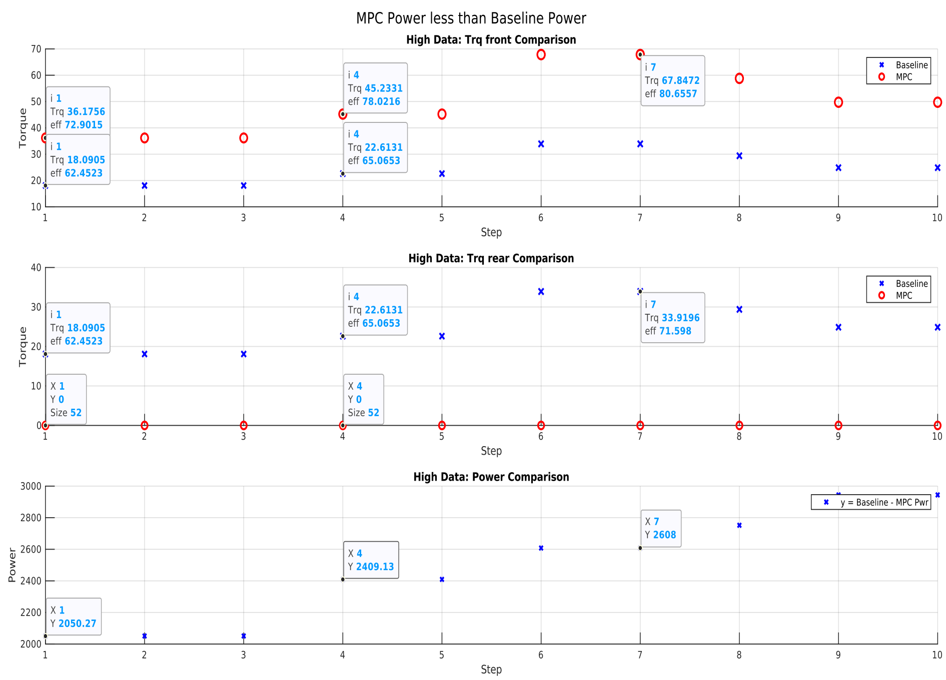

Figure 41: High Points- Full feedback NMPC.

Case 2: NMPC Power consumption > Baseline Power consumption

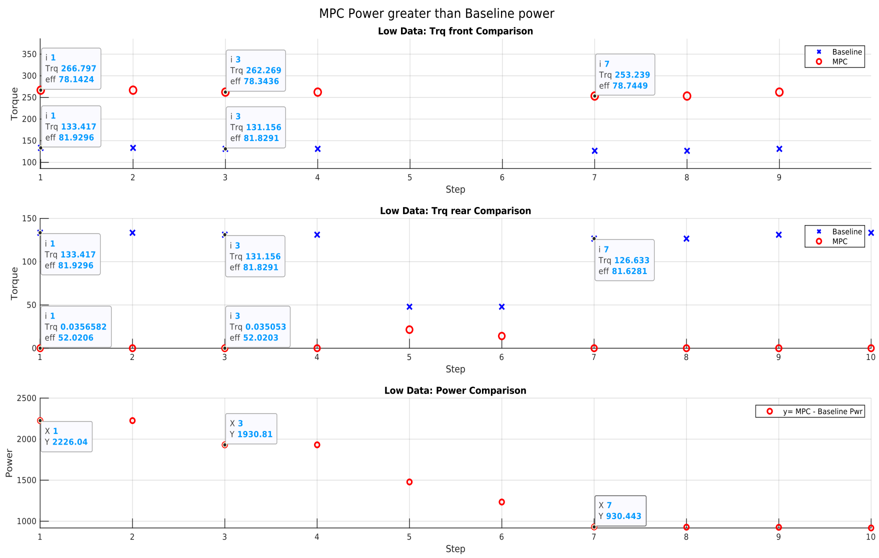

Figure 42: Low Points- Full feedback NMPC.

Case 3: NMPC Power consumption < Baseline Power consumption (Middle Range)

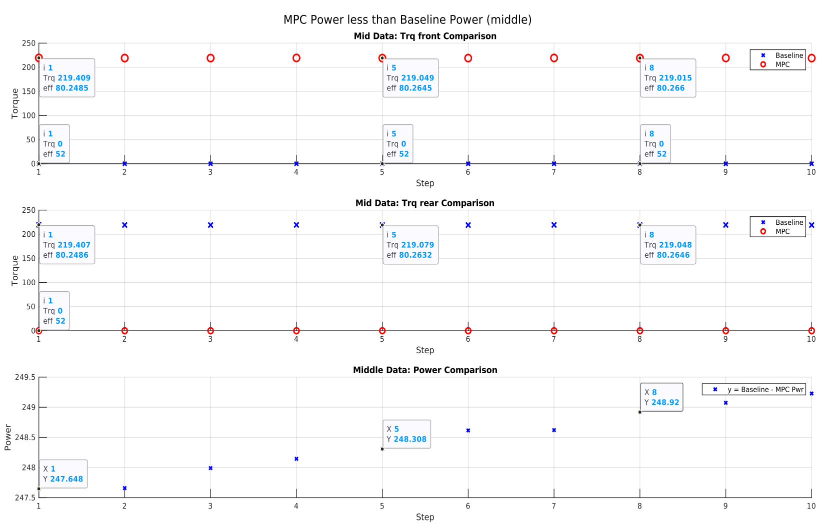

Figure 43: Medium Points- Full feedback NMPC.

Control Policy Comparision

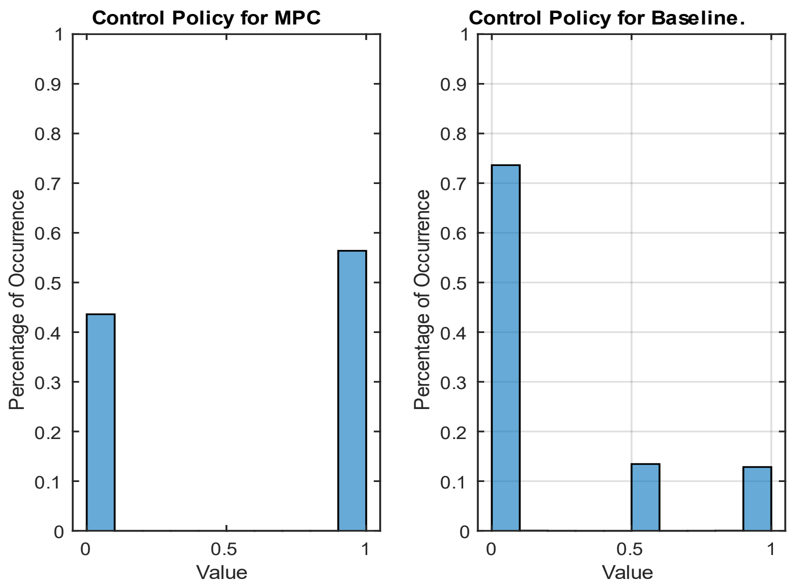

Figure 44: Control Policy- Full feedback NMPC.

Power Vector Comparison:

Case 1: NMPC Power consumption < Baseline Power Consumption

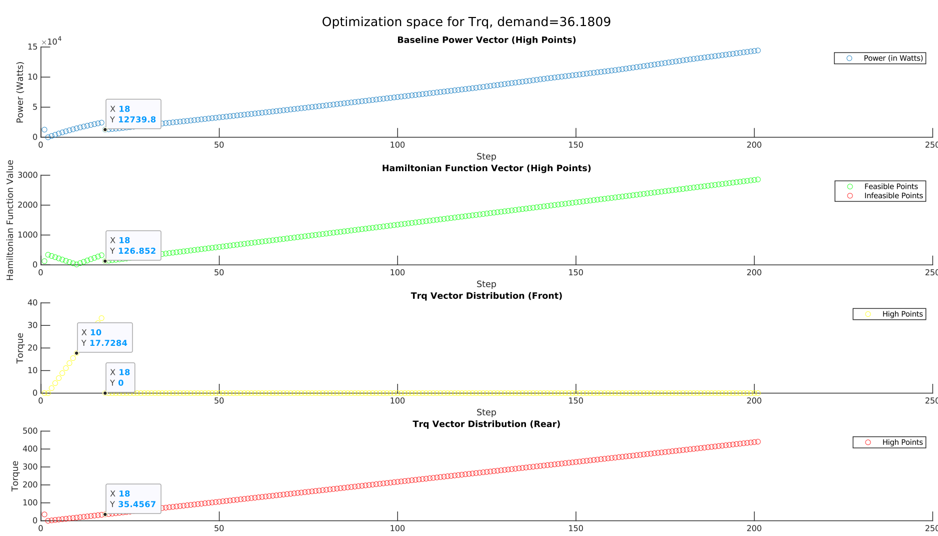

Figure 45: High Power vector- Full feedback NMPC.

Case 2: NMPC Power consumption > Baseline Power consumption

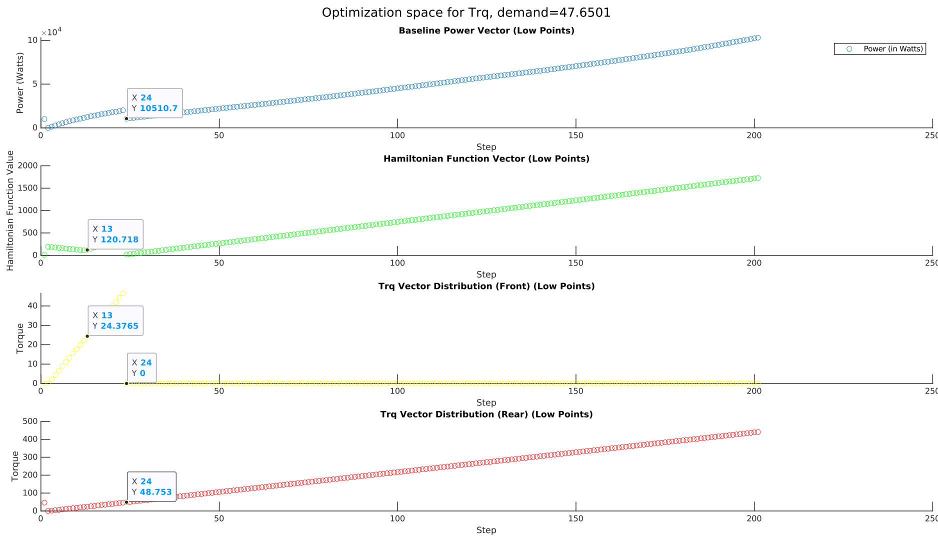

Figure 46: Low Power vector- Full feedback NMPC.

Case 3: NMPC Power consumption < Baseline Power consumption (Middle Range)

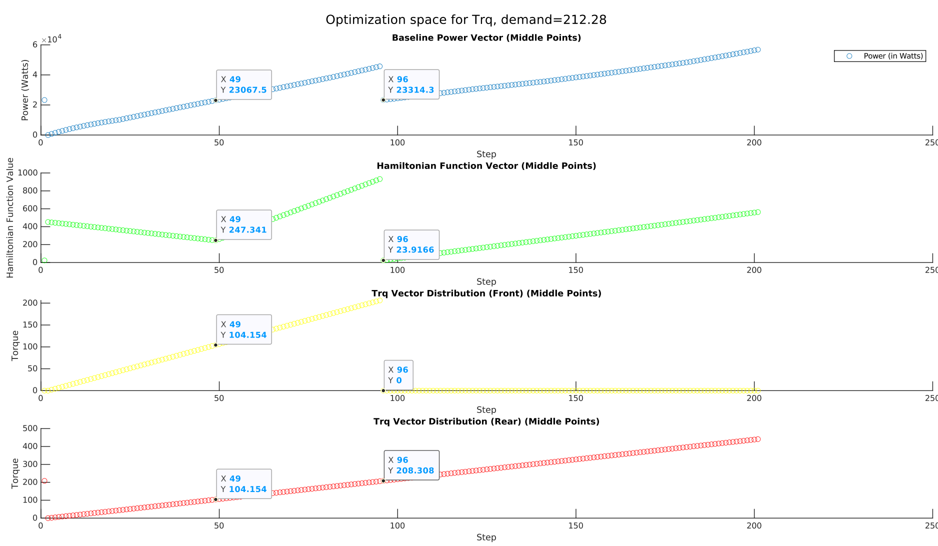

Figure 47: Medium Power vector- Full feedback NMPC.

Optimized Parameter NMPC with Non-overshooting constraints:

The NMPC with non-overshooting constraints implements a fixed timestep Model Predictive Control (MPC) algorithm for electric vehicles (EVs) with a variable timestep vehicle model. Unlike other versions of Nonlinear Model Predictive Control (NMPC), this version has an additional contstraint on each predicted state which prevents it from overshooting. The control horizon is set to (N = 10) steps with a timestep of (Ts = 200ms). This approach enables the MPC controller to make decisions that always stays under direct velocity reference signal, thereby facilitating non-overhshooting control of the EV under varying conditions.

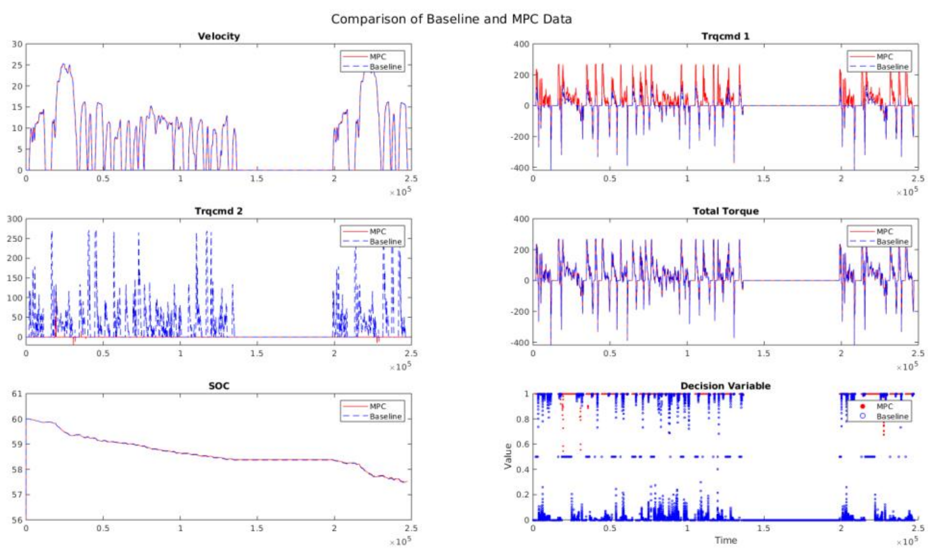

Figure 48: Non-overshooting NMPC.

Optimized Parameter NMPC with linear velocity estimation:

The NMPC with linear velocity estimation implements a fixed timestep Model Predictive Control (MPC) algorithm for electric vehicles (EVs) with a variable timestep vehicle model. Unlike other versions of Nonlinear Model Predictive Control (NMPC), this version does not rely on a true velocity reference signal. Instead, it employs linear projection using constant acceleration or deceleration to predict future velocity. The control horizon is set to (N = 10) steps with a timestep of (Ts = 200ms). This approach enables the MPC controller to make decisions about vehicle control without requiring a direct velocity reference signal, thereby facilitating effective control of the EV under varying conditions.

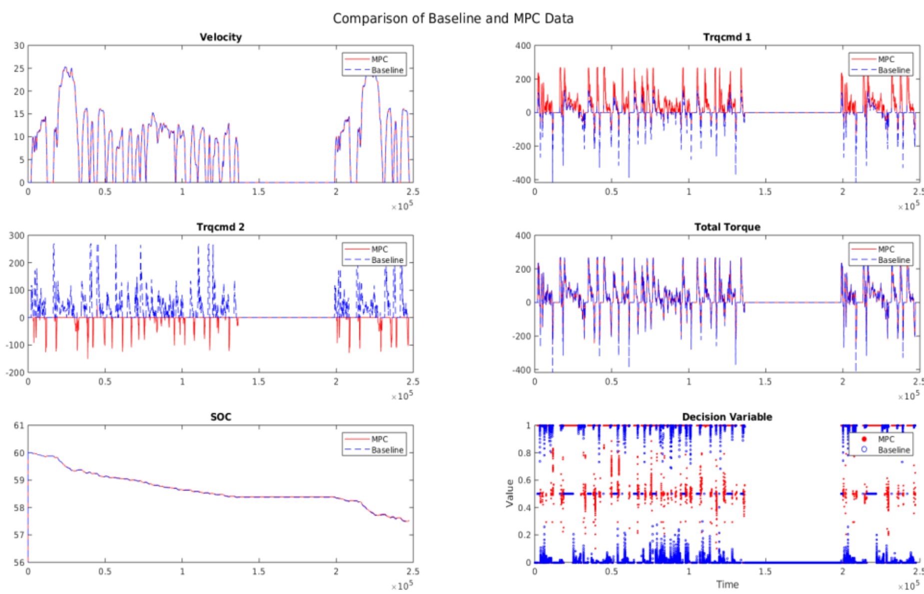

Figure 49: Blind NMPC.

Point Analysis

Case 1: NMPC Power consumption < Baseline Power Consumption

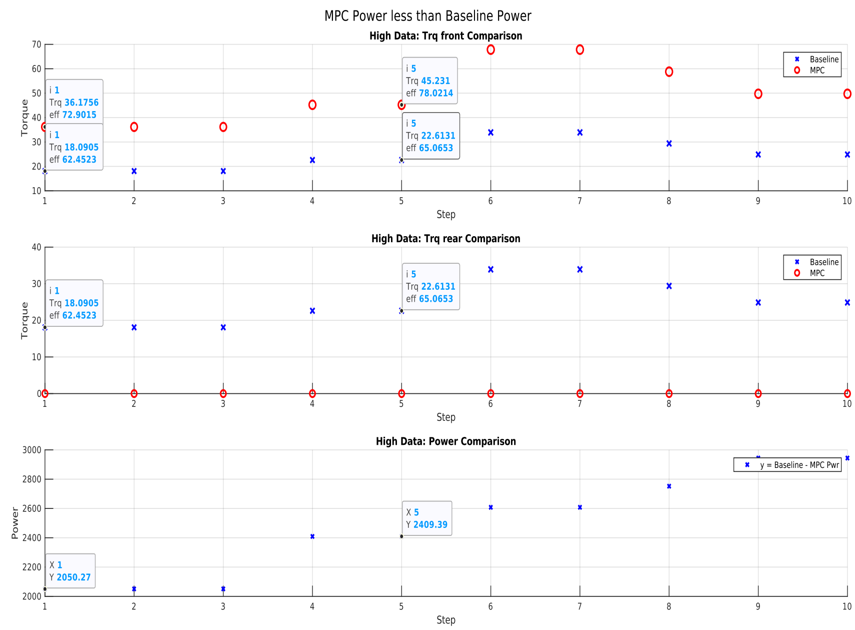

Figure 50: High Points- Blind NMPC

Case 2: NMPC Power consumption > Baseline Power consumption

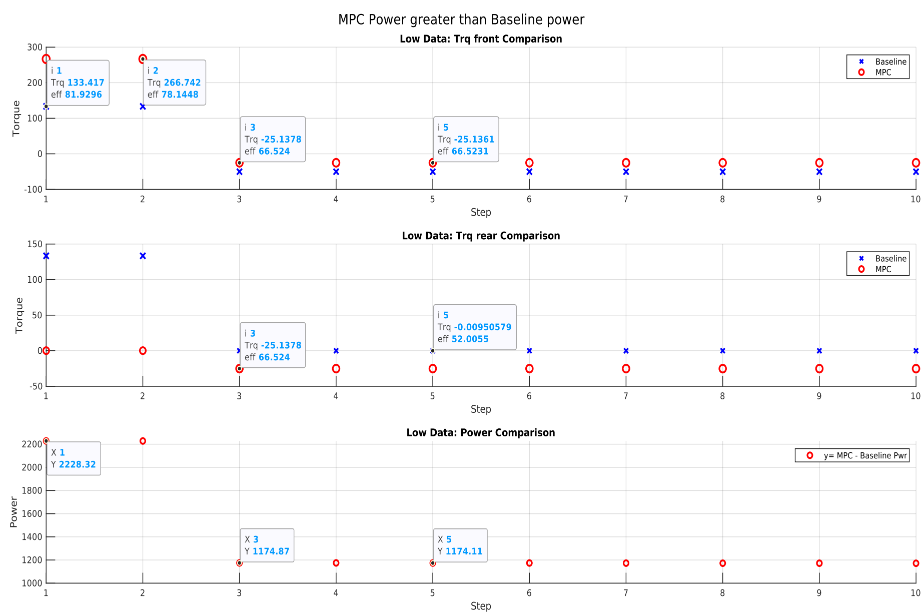

Figure 51: Low Points- Blind NMPC

Case 3: NMPC Power consumption < Baseline Power consumption (Middle Range)

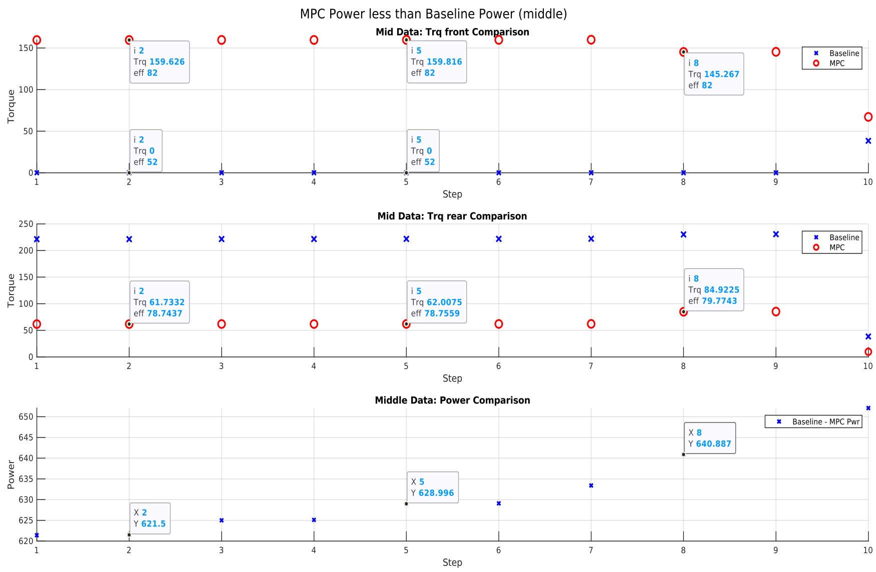

Figure 52: Medium Points- Blind NMPC

Control Policy Comparision:

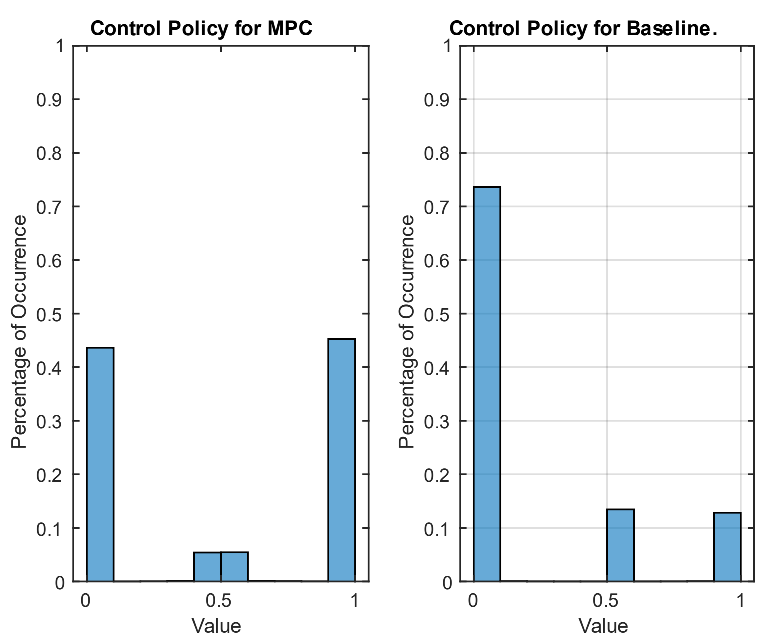

Figure 53: Control Policy- Blind NMPC

Summary

The experimentation conducted encompassed a comprehensive exploration of Electric Vehicle Energy Optimization strategies using Model Predictive Control (MPC) techniques. Beginning with foundational approaches like Nominal MPC , the study progressed to more sophisticated iterations involving variable timestep modeling (Optimized Parameter NMPC) and the incorporation of additional optimization objectives (NMPC with Delta U Penalty).

Subsequent experiments refined the optimization process by introducing penalties for the rate of change of control (NMPC with Delta U Penalty) and evaluating different decision feedback policies (NMPC with full feedback) across various timestep parameters (Ts=100ms to Ts=400ms).

Additionally, experiments such as Fixed-Timestep NMPC with Fixed Timestep Model 5ms and Fixed-Timestep NMPC with Variable Timestep Model 5ms provided insights into the effects of fixed timestep modeling on optimization outcomes. Overall, these experiments aimed to enhance energy efficiency and performance in electric vehicles by optimizing torque distribution between front and rear motors. Through meticulous analysis of optimization outcomes, including torque distribution and energy consumption patterns, the study provided valuable insights into the efficacy of different MPC configurations and control strategies. These findings contribute to advancing the understanding and implementation of MPC techniques for Electric Vehicle Energy Optimization, guiding future research and development efforts in this crucial field.

CHAPTER 11

IMPACT AND LIMITATIONS

Global Impact of 1.07% Decrease in EV SOC Consumption

- According to the Global EV Outlook 2023 [6] report by the International Energy Agency: Over 26 million electric cars were on the road in 2022, up 60% relative to 2021 and more than 5 times the stock in 2018.
- According to the U.S. Energy Information Administration: As of January 2024, the average price of electricity in the U.S. is 12.68 cents per kilowatt-hour (kWh) for the transportation sector.
- Calculation:
    - Original SOC consumption per 100 miles: 25 kWh
    - Decrease: 1.07% of 25 kWh = 0.2675 kWh
    - New SOC consumption per 100 miles: 50 kWh - 0.2675 kWh = 24.7325 kWh
- Total vehicles: 26 million
    - Energy savings per vehicle per 100 miles: 0.2675 kWh
    - Total energy savings per 100 miles driven: 26 million vehicles × 0.2675 kWh = 6,965,000 kWh

Impact: A 1.07% decrease in SOC consumption for EVs globally results in approximately 6,965,000 kWh of energy savings per 100 miles driven.

To calculate the money saved, we'll use the average price of electricity in the U.S. for the transportation sector, which is 12.68 cents per kilowatt-hour (kWh).

Total energy savings per 100 miles driven: 6,965,000 kWh.

Average price of electricity: 12.68 cents/kWh

Now, let's calculate the total money saved:

Total energy savings = 13,910,000 kWh

Price per kWh = 12.68 cents = $0.1268

Total money saved = Total energy savings × Price per kWh

Total money saved = 6,965,000 kWh × $0.1268/kWh

Total money saved = $ 883,162

Therefore, a 1.07% decrease in SOC consumption for electric vehicles globally could save approximately $ 883,162 for every 100 miles driven.

Comparison of Hamiltonian Optimization and NMPC

Hamiltonian optimization and Nonlinear Model Predictive Control (NMPC) are two widely used approaches in optimizing control strategies for dynamic systems. While Hamiltonian optimization offers a discrete yet global lookout optimization strategy, NMPC provides local optima solutions. In this section, we delve into the characteristics of both methods and highlight the challenges posed by the reliance on initializations in NMPC.

Hamiltonian Optimization

Hamiltonian optimization involves calculating the value of the Hamiltonian function at regular discrete intervals and selecting the control effort that yields the best outcome. This approach provides a global perspective on optimization, enabling a comprehensive evaluation of control options. By considering the entire system dynamics, Hamiltonian optimization aims to identify the most efficient control strategy across various operational scenarios.

NMPC with fmincon

On the other hand, NMPC optimization using fmincon typically yields local optima solutions. The optimization process heavily depends on the initial conditions of the control inputs, which can significantly impact the resulting optimal solution. Consequently, different initialization points may lead to distinct action policies, affecting the system's performance and efficiency.

Addressing Initialization Dependency:

The dependency on initialization in NMPC poses a significant obstacle in achieving consistent and reliable control strategies. To overcome this challenge, there is a pressing need for techniques that mitigate the influence of initial conditions on the optimization process. By developing methods that ensure robustness and stability regardless of initialization variations, researchers can enhance the applicability and effectiveness of NMPC in real-world applications.

Future Directions

Future research endeavors should focus on exploring innovative approaches to address the initialization dependency issue in NMPC. By incorporating robust optimization techniques or adaptive algorithms, it may be possible to achieve more consistent and reliable control strategies that offer improved performance and efficiency. Additionally, advancements in optimization algorithms tailored specifically for NMPC applications could lead to significant breakthroughs in overcoming this obstacle, paving the way for enhanced control of dynamic systems in various domains.

In summary, while Hamiltonian optimization provides a global outlook on control strategy optimization, NMPC offers local optima solutions but faces challenges related to initialization dependency. Overcoming these obstacles is crucial for advancing the efficacy and practical applicability of NMPC in dynamic system control.

CHAPTER 12

CONCLUSION AND RECOMMENDATIONS

Conclusion

In conclusion, the utilization of an MPC (Model Predictive Control) controller to anticipate the future dynamics of a vehicle and subsequently determine the optimal torque distribution holds immense promise within the realm of energy-efficient vehicle control. Undoubtedly, this approach presents a viable solution to minimize energy consumption, thereby contributing significantly to the sustainability and efficiency of automotive systems.

However, the implementation of MPC for such purposes is not without its challenges. High computation load, the risk of getting trapped in local minima, and the necessity for accurate prediction of future velocity profiles and environmental conditions emerge as significant roadblocks. Addressing these hurdles is imperative to ensure the practical applicability and effectiveness of MPC-based control strategies in real-world scenarios.

Recommendations:

1. Optimization of Computational Efficiency: Efforts should be directed towards optimizing the computational efficiency of MPC algorithms without compromising on their predictive capabilities. Exploring parallel computing techniques, algorithmic enhancements, and hardware advancements could aid in mitigating the computational burden associated with MPC-based control systems.

2. Enhancement of Prediction Accuracy: Improving the accuracy of future velocity profile and environmental predictions is essential for the robust performance of MPC controllers. Integration of advanced sensor technologies, data fusion algorithms, and machine learning approaches can enhance the predictive capabilities of MPC models, thereby enabling more precise decision-making in dynamic driving scenarios.

3. Incorporation of Necessary Dynamics: To further solidify the applicability of MPC in vehicle control, it is imperative to include the necessary dynamics in the current simplified MPC model. This entails incorporating comprehensive vehicle dynamics, environmental constraints, and driver behavior models into the MPC framework, striking a balance between accuracy and computational load.

4. Validation through Simulation and Real-World Testing: Rigorous validation of MPC-based control strategies through extensive simulation studies and real-world testing is essential to assess their performance, robustness, and reliability under diverse driving conditions. Collaborations with automotive industry partners and research institutions can facilitate the validation process and accelerate the adoption of MPC-based control solutions in practical automotive applications.

5. Continuous Research and Development: Continuous research and development efforts are crucial to keep pace with advancements in control theory, computational algorithms, and automotive technologies. Collaborative research initiatives, knowledge sharing platforms, and interdisciplinary collaborations can foster innovation and drive the evolution of MPC-based control strategies towards achieving sustainable and energy-efficient mobility solutions.

In summary, while challenges persist, the potential benefits of employing MPC controllers for energy-efficient vehicle control are undeniable. By addressing the identified roadblocks and implementing the recommended strategies, we can pave the way for the widespread adoption of MPC-based control solutions, thereby contributing to a greener and more sustainable future in the automotive industry.

REFERENCES

1. Y. Chen, X. Li, C. Wiet and J. Wang, "Energy Management and Driving Strategy for In Wheel Motor Electric Ground Vehicles with Terrain Profile Preview," in IEEE Transactions on Industrial Informatics, vol. 10, no. 3, pp. 1938-1947, Aug. 2014, Doi: 10.1109/TII.2013.2290067.

2. Y. Chen and J. Wang, "Fast and Global Optimal Energy-Efficient Control Allocation with Applications to Over-Actuated Electric Ground Vehicles," in IEEE Transactions on Control Systems Technology, vol. 20, no. 5, pp. 1202-1211, Sept. 2012, doi: 10.1109/TCST.2011.2161989.

3. Luca Cavanini, Pawel Majecki, Mike J. Grimble, Lakshmy Vazhayil Sasikumar, Richard Li, Curt Hillier, Processor-In-the-Loop Demonstration of MPC for HEVs Energy Management System, IFAC-Papers Online, Volume 55, Issue 24, 2022, Pages 173-178, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.10.280.

4. W. Huang, D. M. Bevly, X. Li, and S. Schnick, "Using 3D road geometry to optimize heavy truck fuel efficiency," in Proc.11thInt. IEEE Conf. Intell. Transportation Syst., 2008, pp. 334–339

5. B. Asadi and A. Vahidi, "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," IEEE Trans. Control Syst. Technol., vol. 19, no. 3, pp. 707–714, Mar. 2011.

6. Global EV Outlook 2023 by International Energy Agency (https://iea.blob.core.windows.net/assets/dacf14d2-eabc-498a-8263-9f97fd5dc327/GEVO2023.pdf)

7. F. Borrelli, A. Bemporad, M. Morari, Predictive Control for linear and hybrid systems, Date Oct 26, 2016.

8. Basil Kouvaritakis, Mark Cannon, Model Predictive Control: Classical, Robust and Stochastic.

9. Powertrain Blockset User's Guide, PDF Documentation, Powertrain Blockset automotive powertrain system," https://www.mathworks.com/help/autoblks/".

10. Non-Overshooting Model Predictive Control (MPC) Design for Vehicle Lateral Stability, Sudhakhar Monish Dev. Arizona State University ProQuest Dissertations Publishing, 2023. 30249547.

11. L. Grüne, J. Pannek, Nonlinear Model Predictive Control, Communications and

Control Engineering, DOI 10.1007/978-0-85729-501-9_1, ©Springer-Verlag London Limited 2011.

12. MATLAB-Simulink-Challenge-Project-Hub, Energy Management for a 2-Motor BEV using Model-Predictive Control 246, https://github.com/mathworks/MATLAB-Simulink-Challenge-Project-Hub/tree/main/projects/Energy%20Management%20for%20a%202-Motor%20BEV%20using%20Model-Predictive%20Control#energy-management-for-a-2-motor-bev-using-model-predictive-control

13. Software-in-the-Loop Simulation, Documentation, https://www.mathworks.com/help/ecoder/software-in-the-loop-sil-simulation.html

14. Virtual Vehicle Composer, Documentation, https://www.mathworks.com/help/vdynblks/ref/virtualvehiclecomposer-app.html

15. Emission Test Cycles, Standards, DiselNet, https://dieselnet.com/standards/cycles/index.php.

16. Longitudinal Driver, Longitudinal speed-tracking controller, Documentation, https://www.mathworks.com/help/vdynblks/ref/longitudinaldriver.html.

17. U.S. Energy Information Administration https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a.