Ecological Control and Coordination of Connected and Automated PHEVs at Roundabouts under Uncertainty

by

Sina Alighanbari

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2019

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

During the last decade, comprehensive research efforts were concentrated on autonomous driving. Annually, many car accidents happen as a result of human faults. Extreme traffic congestion prolongs commute time, increase air pollution and cause other transportation inefficiencies. Consequently, using advanced technologies to make vehicles less dependent on human drivers enable more efficient use of time for passengers and decrease car accidents. Connectivity between vehicles and automation provides a spectacular opportunity to improve traffic flow, safety, and efficiency. There are different main active research subjects under the broad domain of autonomous driving, one of them is intersection control for connected and automated vehicles (CAVs) which can be categorized into centralized and decentralized approaches.

The environmental and strict regulatory demands require automotive companies to reduce Carbon Dioxide emissions by investing more in Electric Vehicles (EVs) and Plug-in Hybrid Electric vehicles (PHEVs). A PHEV equipped with connectivity and automation looks more interesting to automobile consumers since they can have advantages of both fewer emissions and enhanced abilities. Since the powertrain of PHEVs consists of different sources of power, advanced control techniques such as Model Predictive Control (MPC) is needed.

Coordination of vehicles at roundabouts is a demanding problem especially by knowing that the chance of both lateral and longitudinal collision exists. To this end, first, we proposed a centralized nonlinear MPC-based controller to adhere to calculated priorities for connected and automated PHEVs (CA-PHEVs). We further continued this research by proposing an approach for solving nonlinear multi-objective optimal control problem of decentralized coordination of CA-PHEVs at roundabouts with consideration of fuel economy. It was found that the proposed controller can calculate priority based on a navigation function and provide a safe gap between vehicles. A novel priority calculation logic based on optimal control is proposed as well and its performance is compared with the navigation function approach. In addition to the decentralized control approach, we considered a more realistic robust tube-based nonlinear MPC decentralized approach to solve this problem in the presence of uncertainties. We used simulations to test the controller and a Toyota Prius PHEV high-fidelity model is used in this thesis for simulations. Simulation results show that the addition of robustness, and energy economy to performance index can improve the fuel consumption of the vehicle. One of the major concerns in designing a controller for automotive applications is real-time implementation. The results of hardware-in-the-loop experiments show the real-time implementation of the controllers.

Acknowledgements

I would like to thank my passionate and supportive supervisor, Prof. Nasser L. Azad for his valuable guidance and support in this research which this thesis would not have been possible without his encouragement and motivations.

I acknowledge Toyota and Natural Science and Engineering Research Council (NSERC) for their financial support in this research study with special thanks to Dr. Ken Butts from Toyota Technical Center for his valuable comments and suggestions.

I thank all of my friends in Smart Hybrid and Electric Vehicles (SHEVS) lab at the University of Waterloo, especially Dr. Mahyar Vajedi and Dr. Bijan Sakhdari who helped me in this research.

I thank my beloved parents Nosratollah Alighanbari and Forough MohammadAghaei whose infinite love is with me in any stage of my life. Also, I thank my sister Laya and my brothers Soroosh and Mohsen for their spiritual support and suggestions.

This thesis contains some material from the following multi-author publication:

S. Alighanbari and N. L. Azad, *”Ecological nmpc controller for connected and automated plug-in hybrid electric vehicles at roundabouts,”* in 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019, pp. 1069-1074

Dedication

This is dedicated to the one.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and challenges

Automated vehicles (AVs) have become a hot topic of research and provide a good opportunity to improve performance, traffic flow, vehicle safety, and traffic congestion [1]. Most of the major car manufacturers and other companies such as GOOGLE and NVIDIA are working on AVs. In some states of the USA, AVs are being tested within urban areas. There are 5 different levels of "automated" vehicles. The automated vehicle can only be called level 5 automated (autonomous) when the vehicle itself controls steering and acceleration/braking and the vehicle does not rely on a human driver to monitor the road. For instance, lane-keeping and other advanced driver assistant systems (ADAS) are considered as level one automated vehicles [2].

Moreover, connected vehicles are linked to one another to exchange information and operate better than vehicles without connectivity. Connected vehicles utilize different means of communication to exchange data and information such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication systems to operate with higher performance and efficiency. However, automotive manufacturers and governments realized that "net neutrality" is not applicable to ground vehicles due to the fact that some information such as safety-related information is much more valuable; hence, it needs higher bandwidth. Therefore, dedicated short-range communication (DSRC) introduced as a solution to this matter. Vehicle connectivity reduces the number of unknown states and disturbances, increases the amount of information that can be inferred and used in comparison to only investigating lead vehicle through the usage of radar, and provides a higher capability of co-operation and collaboration between vehicles in multi-vehicle scenarios [3].

Plug-in hybrid electric vehicles (PHEVs) and fully electric vehicles (EVs) have been a subject of extensive research in the past two decades. In comparison to internal combustion engines, PHEVs can improve environmental factors like energy efficiency and greenhouse gas emission problems. However, due to their complex powertrain architecture, advanced optimal control techniques like model predictive control are required to distribute power optimally between different energy sources [4, 5, 6]. PHEVs are not successful enough in the market due to their relatively high market price in comparison with ordinary gasoline engine vehicles. Adding connectivity and automation to PHEVs significantly makes their performance, functionality, and safety better; thus, make PHEVs more competitive in the market and more attractive to consumers. Connected and automated plugin hybrid electric vehicles (CA-PHEVs) have great potential to enhance safety and traffic flow by decreasing uncertainty from the surrounding traffic environment and uncertainty related to driver behaviour; thus, make designing the controller in the multi-vehicle situations more tractable.

Among different intersections, roundabouts have more promising features and have better performance. Roundabouts have the capability to improve traffic flow and their number is increasing within cities. Because there is no traffic light to beat, roundabouts provide better traffic flow and reduce delay due to the fact that drivers get through the traffic circle without waiting for the green light. Additionally, the number of fatal collisions is lower because of the slower speed of vehicles inside roundabouts [7]. However, due to roundabout's geometry and priority system, it can reach its capacity with even moderate traffic which causes a significant delay for minor-roads [8, 9].

Developing a controller for CA-PHEVs at roundabouts is a demanding problem since roundabouts consist of more than two approaching roads, the situation is a multi-vehicle scenario that is more convoluted. Moreover, passing through roundabouts requires merging with traffic inside it; thus, the chance of collision is higher than that in other scenarios. Therefore, designing a proper controller to make CA-PHEVs able of crossing roundabouts is a crucial and challenging problem.

1.2 Problem Statement

The main goal of this thesis is to solve the control and coordination problem of vehicles at roundabouts. We assume that the host vehicle which is a CA-PHEV is approaching a single lane roundabout where right-hand side driving is allowed. Other traffic vehicles exist in the scenario and they are approaching the same roundabout. It is assumed that all of the vehicles are equipped with V2V and V2I communication systems, therefore can get critical driving data of surrounding traffic participants. Another assumption is that there are no pedestrians crossing the roundabout. Our contributions to this problem are as follows:

- Proposing an NMPC controller that satisfies priorities in centralized coordination by consideration of fuel consumption.
- Development of decentralized control and priority calculation approach.
- Extension of decentralized priority calculation and control to robust tube-based in order to take the more realistic case into account.
- Comparison between different controllers using dspace traffic simulator, HIL, and MIL.

1.2.1 Coordination approaches

There exist two different approaches to solve this problem: centralized and decentralized. In centralized approaches, a central unit exits which gathers information of all of the vehicles and assigns a drive schedule or priority to each vehicle. Then the vehicles are responsible to adhere to the assigned task given to them by the central controller. In this thesis, we proposed one centralized coordination approach based on nonlinear model predictive control (NMPC) with consideration of energy cost. However, this approach does not seem practical at least in the near future since having information of all of the traffic participants is a requirement for the central controller which is hard to achieve. To propose a more realistic coordination approach, a method that works based on a subset of data or in other words local information is needed. Therefore, we proposed a decentralized coordination approach based on NMPC as well. In decentralized coordination approaches, the vehicle gets information of surrounding vehicles within a specific distance range to calculate priority and adjust its speed to obtain safe inter-vehicular distance before the merging zone.

1.2.2 Control-Oriented Model

Since the technique that we want to use in this thesis is MPC, an accurate control-oriented model is required. While the prediction model should be simple enough to guarantee fast and real-time implementation, it requires to be accurate in capturing the dynamics of the system. Therefore, the model that we use considers the longitudinal dynamics of the vehicle but ignors lateral dynamics. It is worth noting that ignoring lateral dynamics is not an unrealistic assumption since the speed at roundabouts is fairly low. Moreover, the other goal of this research is to propose a controller for the complicated powertrain of PHEVs; hence, the longitudinal dynamics of the vehicle will be adjusted for the case of PHEVs accordingly.

1.2.3 Control Design

To formulate the problem in the form of an optimal control problem, in addition to the control-oriented model, the performance index and constraints are needed to be specified. One of the main advantages of MPC is its capability to satisfy multiple objectives simultaneously. Each of the terms in the cost function has a weighting parameter whose proper value can be found by tuning. To have the best set of tuning parameters and robustness for different scenarios, some weightings are defined adaptively in this research. Also, MPC can handle equality and inequality type constraints on both states and inputs. To take into account and compensate for the effects of disturbances and uncertainties, robust tube-based NMPC was designed. Robust MPC requires the addition of linear stabilizing controller and calculation of robust positive invariant (RPI) set.

1.2.4 Control Evaluation

To investigate the performance of the proposed controllers, we tested controllers on the high-fidelity model of PHEV Toyota Prius on model in the loop (MIL) simulations. By doing MIL simulations, the overall performance of the controllers can be checked and the effect of different factors can be determined. Also, to make sure that the controllers are real-time and fast, we did Hardware-in-the-loop (HIL) test with high-fidelity of PHEV. We also used the traffic simulator to compare the efficiency of the controllers against a real driver.

1.3 Thesis Organization

This thesis consists of 5 chapters. Chapter 2 provides background information about model predictive control and powertrain of hybrid vehicles and a review of papers on centralized, decentralized and robust control will be presented.

Chapter 3 is about different aspects of the centralized coordination approach. First the problem of centralized coordination will be reviewed, then different parts of optimal control formulations will be covered in subsequent subsections and finally, results of the proposed approach will be presented.

Chapter 4 is dedicated to the decentralized coordination method. In this chapter, the decentralized approach and two different ways of defining the navigation function will be mentioned. Results of decentralized coordination method will be presented at the end of this chapter.

Chapter 5 is about robust tube-based decentralized coordination. First, we will talk about a robust tube-based approach for linear systems and we will extend it to nonlinear systems. The subsequent parts of this chapter are about the disturbance set and calculation of robust positive invariant (RPI) set. Finally, the robust nonlinear optimal control formula will be rewritten and its results will be shown.

The last chapter concludes this thesis by summarizing contributions and providing some suggested future works.

Chapter 2

Literature Review and Background

2.1 Model Predictive Control: Theory and Methods

Model Predictive Control (MPC) is one of the closed-loop modern control techniques. MPC is very powerful in controlling complex systems since it is a multi-input multi-output controller, can handle equality and in-equality state and control constraints, and is able to incorporate estimation of future information. The MPC, in its heart, has a simple but accurate model representing the system called the control-oriented model. The inputs of the MPC are previous states, controls, and reference trajectory and its goal is to minimize a performance index while satisfying constraints and minimize error between the actual trajectory and reference trajectory.

The MPC predicts the future output plant behavior using control-oriented model calculates the control input using an optimizer to minimizes the difference between the system output and a reference trajectory. The MPC at each sampling time determines a control sequence in the next N_c steps in order to minimize the error between predicted plant output and reference trajectory in the next N_p time steps. The N_c is the length of the control horizon and N_p is the length of prediction horizon. The first generated control input in the sequence is applied and the prediction horizon moves one step forward and once again the process repeats; thus, this control technique is also called receding horizon control Fig. 2.1.

There is a trade-off in choosing a proper control-oriented model. The more accurate model can better capture the dynamics of the system and improves the quality of prediction and performance of the MPC. However, it results in more complex, time-consuming and computationally expensive processes and makes the real-time implementation of the MPC for highly dynamic applications like automobiles more challenging.

An extended version of MPC is nonlinear MPC (NMPC). It enables defining a more realistic nonlinear control-oriented model and has more promising and useful features compared to linear MPC (LMPC). However, computational load is still a problem for the real-time application of NMPC [10]. There are many different approaches proposed in the literature to improve the computation load one of them is the explicit MPC (eMPC) [11]. In the eMPC, the optimization problem is solved offline and the solution is stored as some lookup tables of linear controller gains. Although the eMPC approach reduces computation during run time, it increases pre-processing computation offline to prepare lookup tables and it needs storing capacity to store huge data [12].

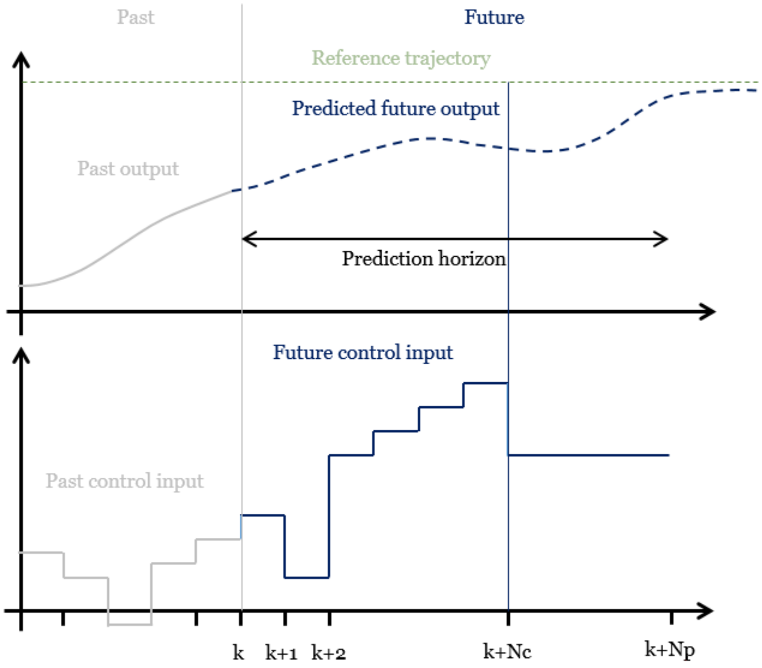


Figure 2.1: Principle of Model Predictive Control

2.2 Literature Review on Coordination Approaches and Robust Control

In this thesis, we want to solve the coordination and merging of connected and automated PHEVs at roundabouts. A large number of available research papers have focused on improving vehicle coordination at on-ramps and four-ways. The approaches can be categorized into centralized and decentralized approaches [13].

2.2.1 Literature Review on Roundabouts

Among different intersections, roundabouts have more promising features and have better performance [14, 15, 16] but they reach their capacity with moderate traffic and cause congestion. People living in metropolitan areas such as Washington DC, Los Angeles, and New York spend a significant amount of time and money on traffic congestions [17] which makes proposing a method to reduce chance of traffic congestion in roundabouts more demanding. An extensive amount of work has been done to improve the performance of roundabouts.

Former researches focused on finding proper metering or traffic signal control approaches. [8] proposed a simple metering model based on "Highway Capacity Manual" and investigated the performance using a simulation model. They showed that roundabouts with metering signals have the potential to reduce the delay in comparison to unmetered intersections. A method of evaluating performance and capacity of roundabouts with metering signals has been presented in [18, 19] and showed that metering can improve the performance of roundabouts significantly. [9] Presented a new method of traffic signal control for multi-lane roundabouts that eliminates conflicting points and weaving sections. In their method, they presented a second stop line for the left-turn besides the first stop line on the approach. On the other hand, coordination approaches try to solve the problem by controlling the flow of vehicles and it will be discussed in the subsequent sections.

2.2.2 Centralized Coordination Approaches

In centralized approaches, at least one of the tasks is done by a centralized unit. The centralized approaches can be categorized to optimization-based and heuristic.

One of the centralized heuristic methods to address coordination problems at intersections is the reservation scheme. In this approach, each of the vehicles approaching the intersection sends a request for a reservation of the space-time cells inside the intersection from a central controller. If the request has no conflict with others, the reservation will be granted. Otherwise, the vehicle is responsible to decelerate and send another request. Some examples of the papers in the literature that used this method are: [20, 21, 22, 23, 24, 25, 26].

The optimization-based centralized coordination methods can also be divided into different subgroups. One approach is to try to optimize time duration that vehicles spend inside the intersection by incorporating travel time in the objective function with some constraints to avoid collisions [27, 28, 29, 30, 31, 32]. Another approach, first presented by [33] and later extended by [34], is to minimize the overlap between the positions of vehicles by optimizing the number of vehicles inside an intersection based on the length of vehicles, intersection area, and a safe following distance. In the multi-objective centralized approach, it is common to assume that vehicles have been assigned a driving schedule and they try to adhere to it by minimizing speed and acceleration errors. In this approach, the cost function can have different objectives and MPC can be used to solve the problem [35, 36, 37].

In the approach proposed by [38], at the high-level, priorities are determined by a central controller and given to each of the vehicles, then in the low-level, every vehicle is responsible for adhering to the assigned priorities by solving an optimization problem with preceding vehicle's current and anticipated subsequent states.

2.2.3 Decentralized Coordination Approaches

In decentralized methods, each of the vehicles gathers local information from other vehicles within a certain range and based on those data, they calculate the proper control policy. One of the challenges associated with decentralized approaches is the situation of deadlock because of using only local information. The decentralized approaches can also be divided into heuristic and optimization-based methods.

Since the main goal in this research is to introduce decentralized optimization-based methods to solve coordination problem at roundabouts for PHEVs, after this point, we concentrate on surveying relevant papers in the literature with a focus on decentralized optimization-based approaches.

In [39], the authors assumed the arrival time of the vehicles to the intersection can be found since each of the vehicles is following the desired speed. To satisfy the time gap objective, each vehicle is responsible to find appropriate control based on the navigation logic while minimizing speed error. In their method, the heavier vehicles can be assigned to smaller acceleration values to provide smoother traveling. Other papers in the literature utilized MPC to solve this multi-task optimal control problem to benefit from advantages of this method.

Authors of [40] introduced an approach in which each of the vehicles gathers local information about other vehicles in a given scenario and calculates the time of arrival to the intersection and their priority for all of the participating vehicles by solving a linear-quadratic optimal control problem.

[41] presented a coordination approach by combining predictive and reactive control layers for autonomous vehicles at single lane roundabouts and a novel priority framework for decentralized coordination that enable the vehicles to calculate relative priority based on shared information. Moreover, [42] focused on cooperative conflict resolution problem for coordinating decentralized autonomous vehicles at an intersection without traffic control devices by solving a set of local optimization problems formulated for each vehicle.

2.2.4 Robust Model Predictive Control

Non-robust MPC assumes that a simple certain prediction model is an accurate presentation of the actual vehicle and ignores the effects of disturbances and uncertainties (DUs) on vehicle control performance which is not true most of the time. The actual system is subject to different DUs that can affect the performance of the system and constraint handling, for instance, the radar and lidar are prone to weather conditions such as rain, snow, and fog which leads to inaccuracies in their measurements or V2V and V2I communication systems' performance can be negatively affected by delays. The DUs can be categorized into bounded additive disturbances, parametric uncertainties, and unstructured/unmodelled dynamics [43].

For the case of automotive applications, DUs can be assumed to be bounded to a certain limit. There are different methods to deal with bounded additive DUs such as the tube-based approach and the min-max approach which is explained in [43] and references therein. In [44], authors proposed a robust min-max approach for ACC with a linear model in presence of disturbances on states and showed that it can improve the tracking performance of the system at the cost of higher computational demand that makes the ACC not suitable for real-time applications.

In [45, 46] authors used tube-based approach to design semi-autonomous ground vehicles and their results showed robustness in the presence of disturbance and uncertainties. In this paper, we have used the tube-based approach since its tube can be calculated offline and it is still real-time, although it is computationally more expensive than non-robust MPC [43, 47, 48].

2.3 Hybrid Powertrain: Architecture and Control

Because of environmental issues and dangerous phenomena like global warming, there is an overwhelming tendency to reduce fossil fuel usage and rely on green sources of energy. The automotive companies contribute to these efforts by working on hybrid electric vehicles (HEVs) and PHEVs improvements. These types of vehicles have a more advanced architecture in comparison to gasoline vehicles and require more advenced methods to deal with them. In this section, we briefly talk about different PHEV architectures. The PHEVs can have series, parallel and power-split powertrain architecture which will be explained in the following paragraphs.

In series type powertrain as shown in Fig. 2.2, the electric motor is connected to the wheels and the engine is completely disconnected from the final drive. The engine is responsible for charging the battery of the electric motor by powering a generator while the motor is the only source of power that drives the wheels. Also like other powertrain architectures, some part of the kinetic energy can be restored by using Regenerative Braking Systems (RBS). This structure has very low fuel consumption and emission since the engine only works when it is needed. However, the biggest drawback of this structure is its relatively lower energy efficiency due to multiple energy conversions compared to other architectures[49].

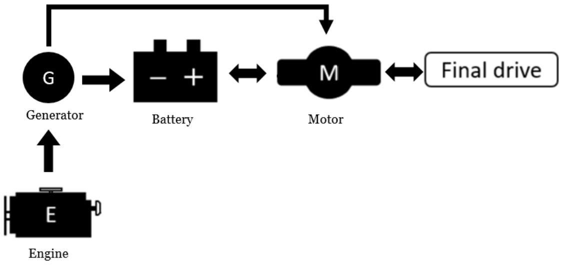


Figure 2.2: Series Powertrain Architecture of PHEVs

On the other hand, in parallel structure both engine and motors are connected to final drive through a mechanical coupling Fig. 2.3. In this method, the motor only helps engine and is not a driving source of power although this hybrid architecture is more efficient by using electric motor during propulsion and RBS during braking than non-hybrid vehicles [49].

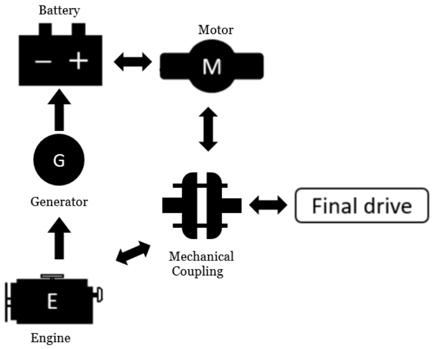


Figure 2.3: Parallel Powertrain Architecture of PHEVs

Finally, in power-split setup Fig. 2.4, the engine, motor and generator are connected to each other through a planetary gear. The gear splits power between wheels and generator. Additionally, decoupling the engine and electric motor is the main advantage of this structure which provides more flexibility [49].

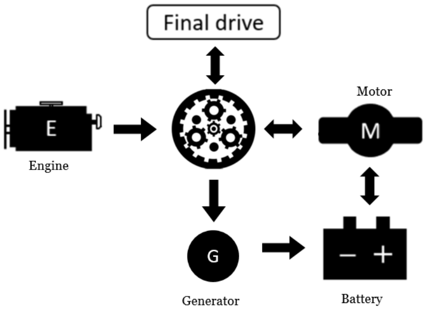


Figure 2.4: Power-split Powertrain Architecture of PHEVs

Comprehensive research efforts in the past twenty years in the fields of control and energy management have been concentrated on proposing supervisory controllers to improve the performance of PHEVS. For instance, several papers in the literature focused on different control techniques to design an optimal energy management system (EMS) [50, 51, 52, 53]. Other papers used dynamic programming (DP) to optimize the battery charge-depletion by having trip information [54].

2.4 Summary

Although PHEVs have more promising features than ordinary vehicles, they are not competitive enough in the market because they are more expensive. One solution is to append connectivity and automation to PHEVs. This enhancement makes designing a controller at roundabouts more manageable by enabling the vehicle to use data of other traffic participants.

Roundabouts enable smoother traffic flow, by eliminating traffic lights and the number of stops and also add to the beauty of cities. However, because of their special geometry, they reach their capacity with moderate traffic and cause traffic jams. Also, since the ego vehicle needs to keep a safe distance from vehicles on the same road plus vehicles approaching the same merging zone from other roads, they cause danger. Therefore, proposing a controller to coordinate vehicles at traffic circles improves their performance, which is a sophisticated problem [8, 9].

Most of the previously reported research in the literature has concentrated on solving the coordination problem at four-way intersections and on-ramps while few papers were addressing the coordination problem at roundabouts. Additionally, most of the previous papers used linear model predictive control (LMPC) to deal with the problem. However, using nonlinear model predictive control (NMPC) enables the consideration of nonlinear terms in the problem equations which results in more accurate and detailed models [55]. Different methods are available to solve NMPC problems and one of them is Generalized Minimal Residual (GMRES) that is able to quickly solve the problem for real-time applications [56, 57].

In this work, a CA-PHEV was used because more data and information is provided to the vehicle due to connectivity between vehicles and infrastructure. In this thesis, first, we propose a centralized coordination and control approach. We assume that priorities have been calculated and assigned by an intersection controller and we focus on proposing a controller for the host vehicle.

Second, We contribute to the decentralized coordination problem by i) formulating the problem in the form of an NMPC problem by considering more accurate vehicle longitudinal dynamics than other papers in the literature and a nonlinear cost function, ii) formulating the optimal control problem for the powertrain of PHEV with consideration of fuel economy, iii) solving the problem for the less investigated case of roundabouts (compared to intersections and on-ramps) and iv) proposing a new navigation function rule based on optimal control for priority calculation.

Last, a robust tube-based decentralized coordination approach is presented. Our contributions are i) formulating nonlinear cost function and incorporating more accurate longitudinal vehicle dynamics for NMPC problem, ii) considerating fuel economy for the complex powertrain of PHEVs, iii) designing robust NMPC to counteract the influence of DUs.

Chapter 3

Centralized NMPC Coordination Approach

3.1 Design Overview

The problem addressed in this chapter is as follows. An ego vehicle that is equipped with the controller is approaching a roundabout with one lane and four curved entrances. Each vehicle approaching the roundabout has a fixed path D and the traveled distance of each vehicle on its fixed trajectory is shown by d. It is assumed that there is an upper-level controller at the roundabout which specifies the preceding (p) and back vehicle (b) for the ego vehicle. Except for the host vehicle, other vehicles in the traffic are ordinary vehicles without the controller. The host vehicle has to yield the right of way to the preceding vehicle (p) and has priority over back vehicle.

We define a merging zone at the entry point of the entrances to the roundabout as shown in Fig. 3.1. In order to have a collision-free merging with traffic inside roundabout and prevent a lateral collision, only one vehicle is allowed to be in this region at the same time. Consequently, to adhere to the assigned priorities, the ego vehicle has to let the preceding vehicle leaves the merging zone and enters the zone before the back vehicle. Since the dynamics of the vehicles have been taken into account, the host vehicle is allowed to merge with traffic inside roundabout when a safe distance between vehicles exists.

The ego vehicle has information about the position, speed, and acceleration of surrounding vehicles through V2V and V2I communication systems. Since the speed in urban areas are not high, we can ignore the danger of lateral skidding and roll over and focus on longitudinal dynamics of the vehicle.

The controller generates the required torque (T) to the vehicle to follow a reference speed and keep a safe distance from surrounding vehicles while minimizing fuel consumption. The determined torque will be given to EMS of PHEV's powertrain.

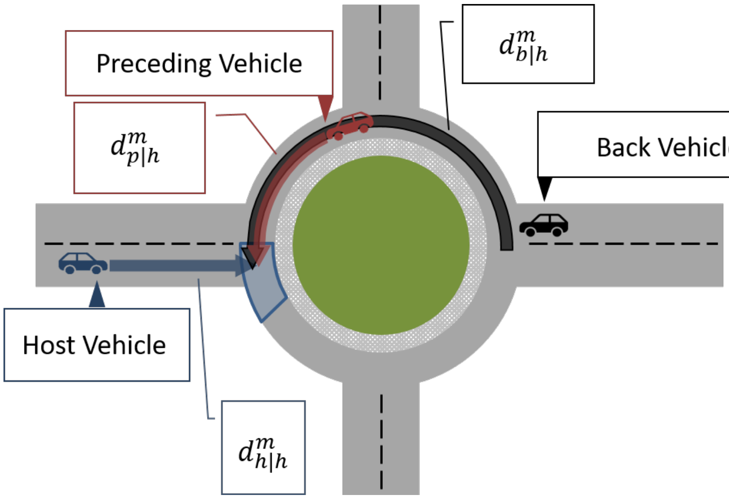


Figure 3.1: Single lane traffic circle with 4 entrances. Red vehicle is preceding vehicle and black is back vehicle. Blue is host vehicle. The light blue zone is called merging zone.

3.2 Control-oriented Model

There are two models available to use in the development of control systems. The high-fidelity model has all verified components of the plant and mainly it is used to verify the controller's performance. The high-fidelity model is computationally expensive; thus, to design the NMPC-based controller to solve this centralized coordination problem in real-time, a simple but accurate control-oriented model is required. The model which has been utilized in this paper is as follows:

$$\begin{bmatrix} \dot{x}_h(t) \\ \dot{v}_h(t) \\ \dot{T}_w \\ \dot{x}_p(t) \\ \dot{v}_p(t) \\ \dot{a}_p(t) \\ \dot{x}_b(t) \\ \dot{v}_b(t) \\ \dot{a}_b(t) \end{bmatrix} = \begin{bmatrix} v_h \\ \frac{T_w}{m r_w \eta_p} - \frac{F_{res}}{m} \\ -\frac{T_w}{\tau_a} + \frac{U}{\tau_a} \\ v_p \\ a_p \\ -\sigma.a_p \\ v_b \\ a_b \\ -\sigma.a_b \end{bmatrix} \tag{3.1}$$

$$F_{res} = \frac{1}{2}\rho c_d A v_h^2 + mg\sin\theta + \mu_r mg, \tag{3.2}$$

where F_{res} is a resistance force applied to the vehicle, T_w is wheel torque, r_w is wheel radius, η_p is powertrain efficiency, τ_a is constant coefficient and σ is a decaying factor and is considered to take the fact that vehicles mostly drive with constant speed into account. ρ is the air density, c_d is the drag coefficient, A is the frontal area of the vehicle, v_h is the vehicle speed, m is the vehicle mass, μ_r is the rolling resistance coefficient and θ is the road grade. The indices h, p, and b are for the host, preceding, and back vehicles respectively.

The host vehicle equipped with the proposed controller which is responsible for keeping a safe distance from the surrounding vehicles. The distance error between the preceding vehicle and host vehicle is defined as follows:

$$\beta = d^m_{p|h} - d^m_{h|h} - d^*, \tag{3.3}$$

$d^m_{p|h}$ means the distance of preceding vehicle p to merging zone m on its fixed path and $d^m_{h|h}$ means the distance of host vehicle to merging zone on its fixed path calculated on-board.

Similarly, the distance error between the back vehicle and host vehicle is defined as follows:

$$\beta^{'} = d^m_{h|h} - d^m_{b|h} - d^*, \tag{3.4}$$

One way to define safe distance between vehicles is based on the safe distance at parked position L_0 and vehicle speed:

$$d^* = L_0 + \kappa v_h, \tag{3.5}$$

In the above equation, κ is headway time. Since the safe distance between vehicles changes with speed, we consider headway time to take that fact into account. The proposed controller tries to minimize the difference between the actual distance between vehicles and the safe inter-vehicular distance equation (3.3) and equation (3.4).

To solve this NMPC problem, a prediction of the future acceleration of traffic vehicles should be considered in the receding prediction horizon. According to [58], vehicles mostly travel with near constant speeds and mostly avoid harsh accelerations or decelerations. Consequently, the below model has been used in this paper:

$$a_t(\tau) = e^{-\epsilon\tau} a_t(t), \tag{3.6}$$

where ϵ is a tuning parameter, $a_t(\tau)$ is predicted future acceleration and $a_t(t)$ is acceleration in current time.

3.3 Trip Cos

One of the objectives of the proposed controller is to decrease the host vehicles' fuel consumption by considering it in our cost function. Since the high fidelity model of Toyota Prius PHEV has been used in this paper, instead of using fuel rate or electricity current, the energy cost will be used. The schematic of Toyota Prius powertrain is shown in Fig. 3.2.

$$E_{cost} = -K_e \frac{S\dot{O}C}{v_h} - K_f \frac{\dot{m}_f}{v_h}, \tag{3.7}$$

In above equation, E_{cost} is energy cost of the trip and the effect of traveled distance can be excluded by dividing it by the host vehicle's velocity. SOC is the state of the charge of battery, $\dot{m}_f$ is fuel rate, K_e and K_f are unit cost of electricity and fuel. The PHEV Toyota Prius is equipped with an energy management system (EMS) which is responsible for distributing energy between the electric motors and engine; thus, we can assume that the engine is always working at its optimum working point. Therefore, the fuel rate can be found using the following equation:

$$\dot{m}_f = a_1 + a_2 P_e + a_3 P_e^2 + a_4 v_h, \tag{3.8}$$

Where P_e is the engine power and a_1 to a_4 are constant coefficients. Additionally, the electricity rate can be estimated as follows:

$$S\dot{O}C = \zeta_1 + \zeta_2 P_m + \zeta_3 P_m^2, \tag{3.9}$$

Where P_m is the motor's power and ζ_1 to ζ_3 are constant coefficients. Moreover, EMS of the vehicle specifies the power ratio; hence, affects the power demand for each source and the energy cost:

$$PR = \frac{P_e}{P_m}, \tag{3.10}$$

To recap, using above mentioned equations, the energy cost can be evaluated and taken into account in the performance index [59].

3.4 Optimal Control Problem Formulation

To formulate this coordination problem in a form of NMPC, defining the performance index, constraints, and prediction model is required. We have already specified the prediction model. This section is dedicated to formulating the performance index and in-equality type constraints.

One of main advantages of the receding horizon control is its capacity to incorporate different objectives and try to satisfy all of them simultaneously. In this paper, the following performance index is used:

$$PI = \sum_{i=0}^{N-1} U(i)^2 + R^T C_x(i) + w_1(v_h(i) - v_{ref}(i))^2 + w_2 E_{cost} + w_3 \beta e^{\beta}, \tag{3.11}$$

where R^T and w_1 to w_3 are weighting factors. The proper values for these parameters can be achieved by tuning. By adding the energy cost, E_{cost}, to the cost function, the controller's output will be calculated with the consideration of energy efficiency which results in lower fuel consumptions by the controller. Another feature of the proposed controller is a reference speed v_{ref} following which prevents the vehicle from powertrain shutdown. The last term considers safety concerning the inter-vehicular distance error. When the distance between vehicles is less than safe distance, the value of the last term goes up exponentially and becomes the most important since it is a safety issue. However, in the case that the distance is equal or more than the safe distance, the safety term becomes less important and the optimization can be solved by consideration of fuel economy and reference speed following terms.

In the PI equation (3.11), C_x is the vector of inequality type constraints. To define this vector, the constraints need to be specified first:

$$\begin{aligned} v_h &\leq v_{max}, \\ U &\leq U_{max}, \\ U_{min} &\leq U, \end{aligned} \tag{3.12}$$

where U is the powertrain torque which is the control input and is bounded by maximum and minimum values. Also, the speed of the vehicle has a maximum specified by the lateral controller to prevent lateral skidding and rollover in the curved paths of the roundabout. As a result, the C_x vector can be defined as follows:

$$C_x = \begin{bmatrix} v_h - v_{max} \\ U - U_{max} \\ U_{min} - U \end{bmatrix} \tag{3.13}$$

3.5 Fast Optimizer

To solve this NMPC problem in real-time, we need a fast solver. In this paper, we used the N-GMRES solver developed by our research group [57]. This solver is MATLAB-based and it can solve different nonlinear optimal control problems fast and accurate enough by solving differential equations at each time step while satisfying constraints. We used this solver because the performance of this solver has been verified, it is user-friendly and it is more compatible with our problem.

3.6 High-fidelity Model

The high-fidelity model presents plants in more details but it is not real-time. The baseline power-split powertrain of PHEV in this paper is developed in a new generation of Powertrain System Analysis Toolkit (PSAT), Autonomie software, by Argon National Lab. The schematic of the powertrain of PHEV is shown in Fig 3.2. The powertrain consists of one engine and two electric motors. The electric motors can be used as a simple motor as well as a generator. The components are coupled to the wheels through two planetary gears. In the first planetary gear, the ring is connected to the wheels while the sun and carriers are attached to the motor 1 and engine. In planetary gear two, the ring is again connected to the wheels and sun plus carriers are attached to motor 2 and chassis. At low speed, the engine is turned off and at normal speeds, some part of the engine power goes to the wheels and the rest applies to motor 1 which acts as a generator. The electric power in the generator turns into mechanical power by motor 2 Fig. 3.2.

The main components of the high-level Simulink are vehicle powertrain architecture (VPA), vehicle powertrain controller (VPC), driver, and environment which are connected via buses as shown in Fig. 3.3. The software has some libraries of vehicle models and components with features. The procedure for developing the high-fidelity model and verification of its performance has been investigated in our group's previous works [60]

.

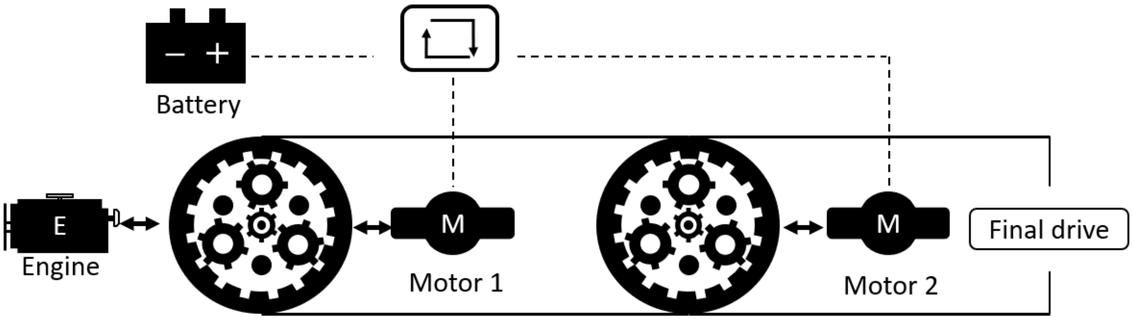


Figure 3.2: Toyota Prius powertrain components

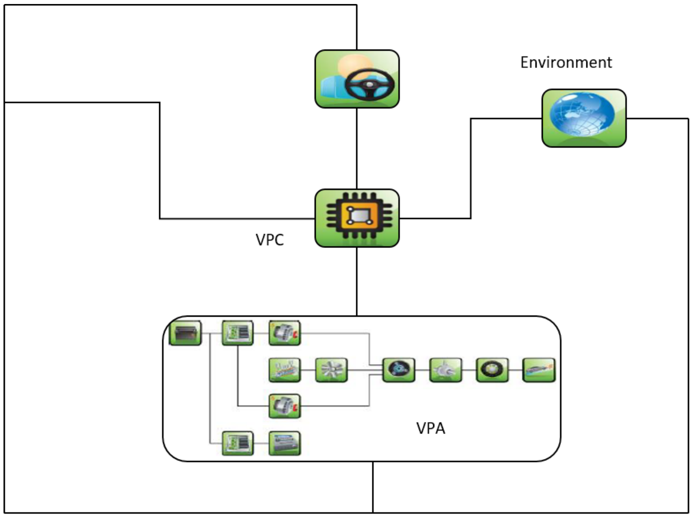


Figure 3.3: Main Components of High-fidelity model of Toyota Prius in Simulink

3.7 Control Evaluation Results

This section is dedicated to the investigation of the performance of the proposed controller in addition to comparing it to other controllers. In order to generate meaningful results, we have used special notation in this thesis. To better understand resulting graphs, the entrance of merging zone is put on the $y = 0$, therefore when the position of vehicles are -40, it means that the vehicle needs to travel 40 meters to enter merging zone.

The subsequent sections show different results of the centralized controller using the high-fidelity model. First, we compare the performance of the proposed controller with PID. The second part compares the fuel consumption of the centralized and PID controllers. The last part devoted to HIL test of real-time implementation of the centralized controller.

3.7.1 Centralized Controller vs PID

In this section, we compare the proposed centralized controller performance with a PID controller. For the sake of comparison, a limited part of HWFET drive cycle has been used and scaled to generate speed profile of the preceding vehicle in the roundabout scenario.

As shown in Fig. 3.4 and Fig. 3.5, the centralized controller has better performance in keeping a safe inter-vehicular distance between vehicles. Because the PID controller is a single task controller and in this comparison, it tries to minimize the speed error. As can be seen, both the preceding vehicle and PID are on the same distance from merging zone which is an indication of accident.

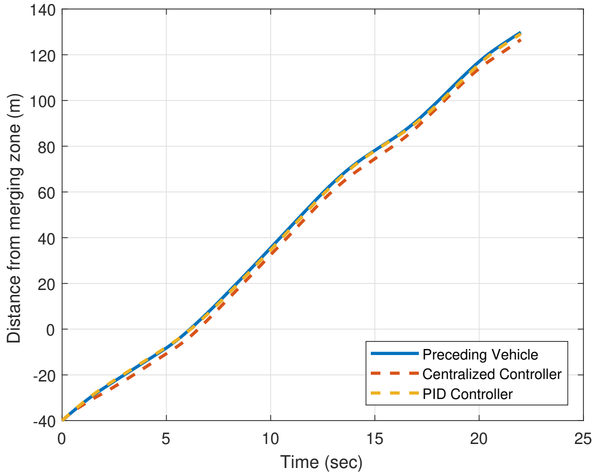


Figure 3.4: Distance from merging zone comparison between centralized controller and PID at roundabouts

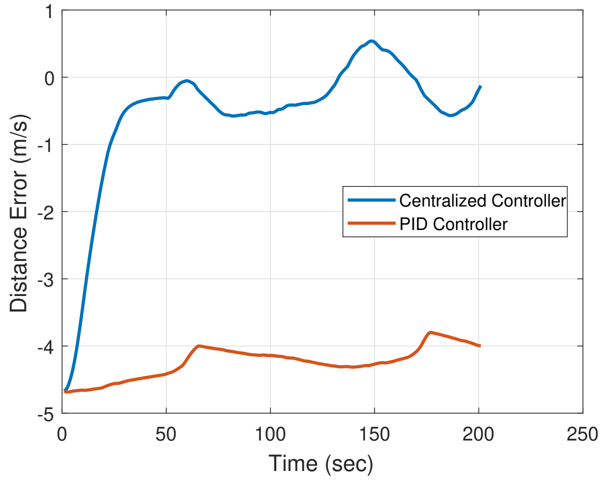


Figure 3.5: Inter-vehicular distance error comparison between centralized controller and PID at roundabouts

To further compare PID with centralized controller, we changed the simulation environment to preceding vehicle following. In the new environment, the host vehicle follows the preceding vehicle which has speed profile of 2*HWFET drive cycle. Fig. 3.6 and Fig.

3.7 compare reference speed following and distance error of both controllers. The centralized controller has less fluctuations in its reference speed following and its inter-vehicular distance error is closer to zero.

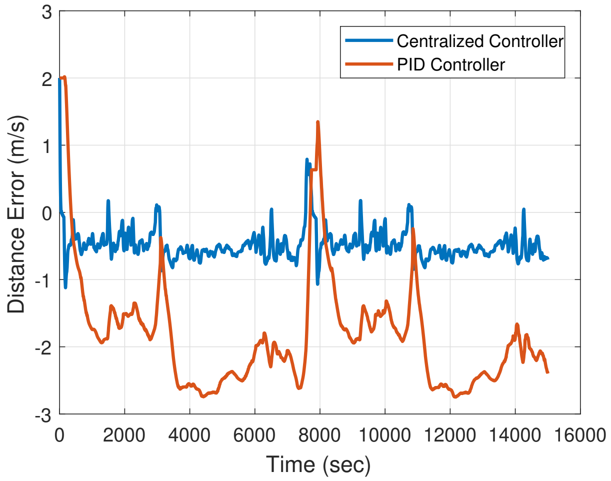


Figure 3.6: Distance error comparison between centralized and PID controllers for 2*HWFET drive cycles

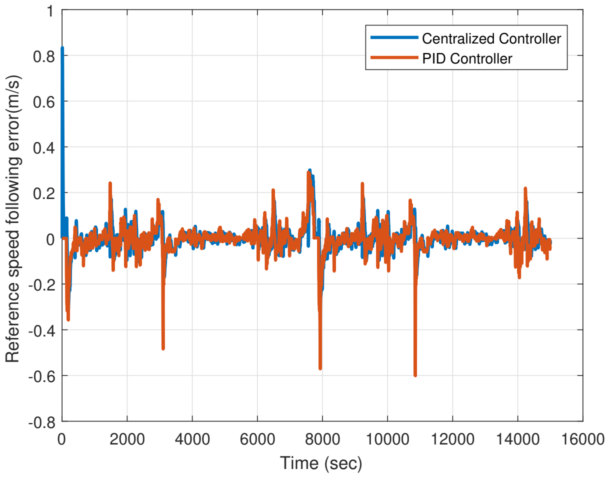


Figure 3.7: Reference speed error comparison between centralized and PID controllers for 2*HWFET drive cycles

3.7.2 Fuel Consumption

This section shows results of fuel consumption of proposed centralized controller and PID for 2*HWFET drive cycles. MPC is multi task controller which is capable of satisfying different objectives at the same time. The NMPC based central control can reduce energy cost of the trip while following reference speed and keeping inter-vehicular distance error around zero. On the other hand, the PID is a single task controller. Although the PID controller has better fuel-consumption than the proposed controller as shown in Fig. 3.8 it cannot provide safe inter-vehicular distance as shown in Fig. 3.5.

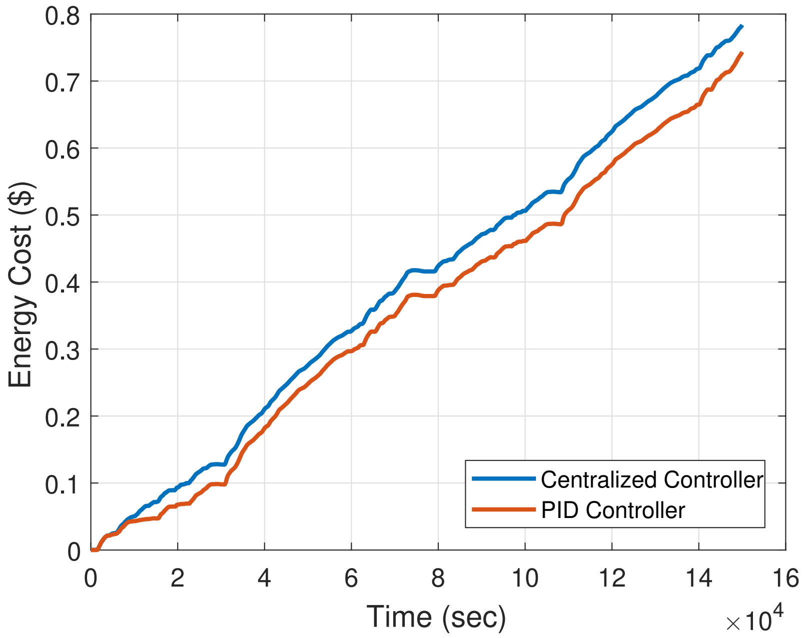


Figure 3.8: Energy cost comparison between centralized and PID controllers for 2*HWFET drive cycles

3.7.3 Hardware-in-the-loop (HIL)

HIL Introduction

Hardware-in-the-loop (HIL) tests are really common among automotive companies and they are really beneficial in the process of design and development of control systems. In some stages in the development of control systems, to better assess the fidelity of the performance evaluations physical prototyping is required. However, this physical prototyping can be very expensive, time-consuming and even dangerous. To reduce the associated risks, the HIL test which is one of the rapid prototyping techniques can be used. In HIL tests only special and complex components will be prototyped instead of all components which is more cost-effective and requires less effort but improves the fidelity of the evaluations. Another important factor in designing control systems is repeatability. To check repeatability physical prototyping is not a proper option but through HIL tests one specific simulation environment can be built and run multiple times. Additionally, investigation of the performance of the controller in harsh situations such as icy road in winter condition using physical prototyping is perilous. However, in HIL tests the simulation can be adjusted accordingly to check performance in a different situation without significant risk and danger. To recap, HIL tests enable us to better assess fidelity of the evaluations while avoiding expensive, complicated and sometimes destructive procedures which in turn cause faster design process and prepare a solid base for further developments.

Nowadays, most of the controllers are programmed on Electronic Control Units (ECUs) for implementation. The ECU is a component that receives sensor captured information and based on those data, it applies proper control action using actuators. Today, advanced vehicles have around 80 ECUs [61] to control different parts of the vehicle like engine, brake, suspension system, etc. Due to the increase in usage of ECUs, evaluating and validating new advanced ECUs in a short time is crucial which can happen using HIL testing. Therefore, we implemented our proposed controllers on an ECU to study its performance through HIL test.

The HIL for rapid control prototyping (RCP) is consist of three components:

- User interface to design simulation, program hardware and setup variables
- A real-time simulator which is a very fast and powerful microprocessor to run the high-fidelity model in real time
- A prototype ECU to check real time performance of the proposed controllers

The different components in the HIL platform are communicating with each other through Controller Area Network (CAN). The CAN enables microcontrollers and ECUs to exchange data without needing a computer. Especial I/O ports compatible with CAN interface enable efficient system and sending and receiving data. The HIL setup used in this thesis is provided by dSPACE which is well-known to most of the major car manufacturers and it is compatible with MATLAB/Simulink.

HIL Hardware and Programming

The HIL that we use in this research has a prototype ECU (Micro AutoBox II) and real time simulator (DS-1006 Processor) [62].

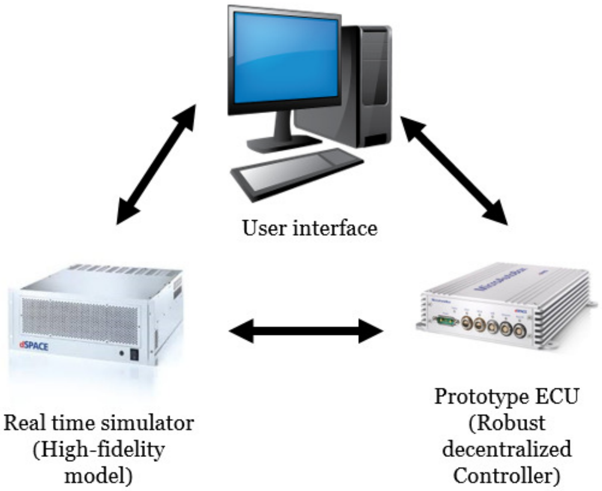

User interface
Real time simulator (High-fidelity model)
Prototype ECU (Robust decentralized Controller)

Figure 3.9: dSPACE hardware-in-the-loop setup

Table 3.1: Specifications of DSPACE HIL Setup

Specification	Prototype ECU	Real-time Simulator
Hardware	MicroAutoBox II	Processor board
Processor	DS 1401 PowerPC 750GL	DS 1006 Quad-Core AMD
I/O	DS 1511	DS 2202

The real-time simulator is responsible to run the high-fidelity model along with environment variables and prototype ECU calculates and sends control signals.

The dSPACE HIL platform works with MATLAB/Simulink and provides appropriate Real-Time Workshop code generators which comply with codes compatible with each hardware. The real-time processor only handles the high-fidelity model and the prototype ECU runs the proposed controller. Therefore, to generate compatible C codes for each hardware and do the testing we follow the following procedure. First, we divide the Simulink model into two separate models one contains only the proposed controller and the other possesses the rest of the original model. To generate proper C code to run on the DS1006 Processor Board, we use rti1006.tlc libraries presented by dSPACE. We add the required interface blocks to the *.mdl file. Then to do the same for MicroAutobox II, we use rti1401.tlc. The communicating signals should have the same CAN ID in both *.mdl files. By compiling each of the *.mdl files, System Description Files *.sdf files can be built. To measure and record different variables, another product of dSPACE called ControlDesk can be used [63]. Once the files uploaded to corresponding hardware, desired variables for both platforms such as ECU turnaround time can be watched and recorded. The turnaround time proves whether the controller is real-time implementable and fast enough for automotive applications. Turnaround time is the time that ECU requires to generate controller input based on the received information. In the next chapters, we do HIL simulation to make sure that the proposed controller is real-time on the prototype ECU and we report the turnaround time.

3.7.4 Hardware-in-the-loop (HIL) Results

One major concern for designing a controller for automotive applications is its real-time implementation. For designing this controller, we used a solver based on the GMRES method. There are two versions of GMRES solver, Newton/GMRES and Continuation/GMRES.

Based on previous investigations [10], C/GMRES solves problems in continuous time and takes into account the system's dynamics and is more accurate than N/GMRES. However, it makes the tuning process more difficult and adds an insignificant amount of improvement in comparison to N/GMRES. Therefore, N/GMRES is used in this paper.

As a matter of fact, the higher number of states, length of prediction horizon, and step time can increase the computational burden of N/GMRES solver and it might make it too slow. To make sure that the proposed controller is real-time, we did HIL test. Fig. 3.10 Shows the turnaround time of the controller. The turnaround time is between 5.9×10^{-5} and 5.2×10^{-5} with its average around $5.6 * 10^{-5}$. The results are measured on prototype ECU which is more powerful and faster than real vehicle ECU. Therefore, we can approximately calculate turnaround time of Toyota Prius by multiplying the above data by 7 [59]. Hence, the maximum approximated value of turnaround time for Toyota Prius is about 0.4 milliseconds which is less than 1 millisecond; thus, the controller is real-time.

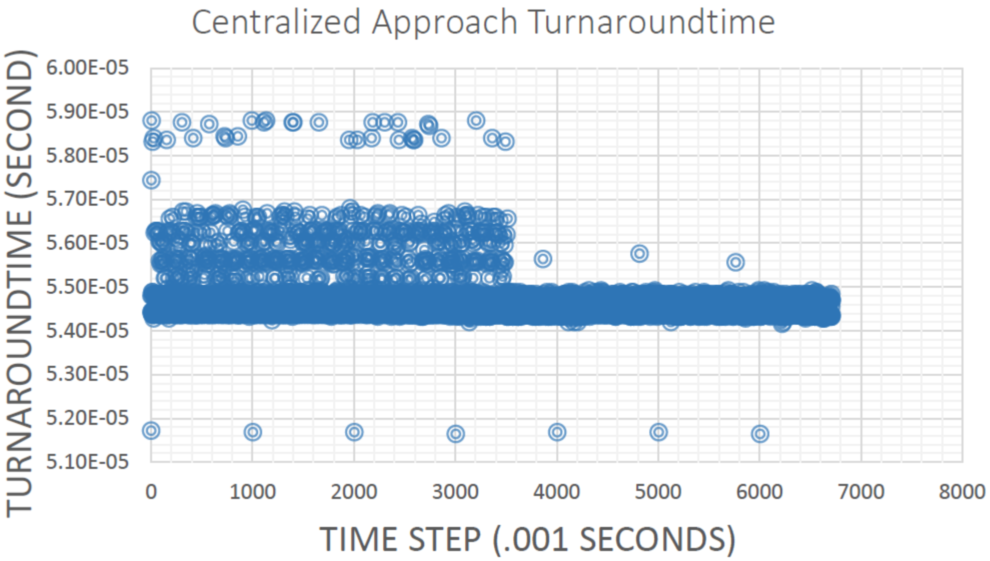


Figure 3.10: Hardware-in-the-loop (HIL) Results of Centralized Controller

3.8 dSPACE Traffic Simulator

Another product from dSPACE is the ASM traffic simulator [64]. In a traffic simulator, one can build a simulation environment, specify traffic scenarios with multiple traffic participants and implement the controller on the vehicle with full vehicle dynamics to test a controller, visualize results, and compare the controller with actual drivers. This product can be used for validation of the controller algorithm by Simulink simulations during the design phase. The dSPACE traffic simulator has full compatibility with MATLAB/Simulink which makes working with this system straightforward. Additionally, ModelDesk and MotionDesk are available to facilitate interaction with the complex Simulink model.

In order to test the controller, we need to bring up the traffic simulator using the computer interface. This can be done by using dSPACE ASM packages and libraries provided by dSPACE for MATLAB/Simulink. There are different Automotive Simulation Models available in the ASM libraries. To get access to these models, one way is to call *ASM-VehicleDynamics-lib* on the MATLAB workspace or simply call asm. Each of these models has some unique specifications and features.

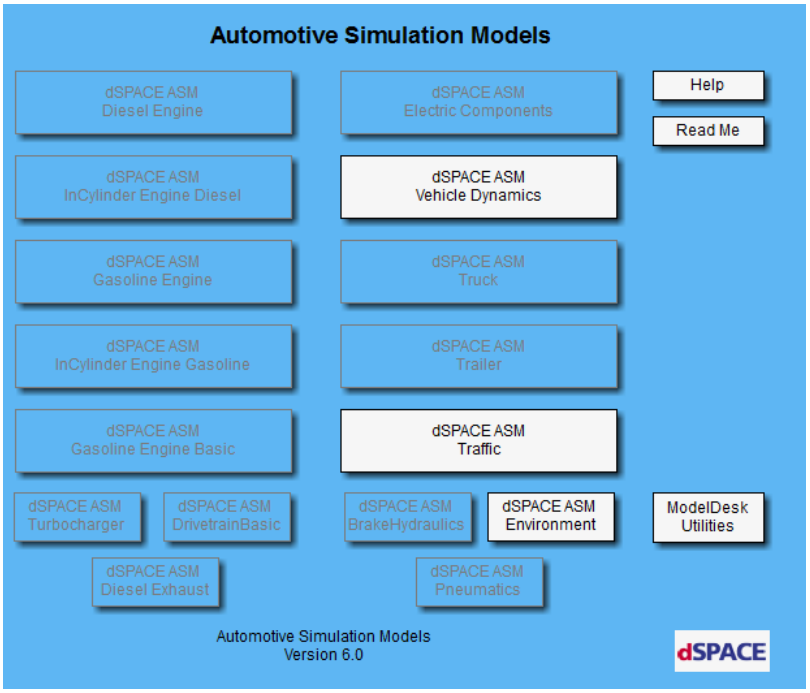

Automotive Simulation Models
dSPACE ASM Diesel Engine
dSPACE ASM Electric Components
Help
Read Me
dSPACE ASM InCylinder Engine Diesel
dSPACE ASM Vehicle Dynamics
dSPACE ASM Gasoline Engine
dSPACE ASM Truck
dSPACE ASM InCylinder Engine Gasoline
dSPACE ASM Trailer
dSPACE ASM Gasoline Engine Basic
dSPACE ASM Traffic
dSPACE ASM Turbocharger
dSPACE ASM DrivetrainBasic
dSPACE ASM BrakeHydraulics
dSPACE ASM Environment
ModelDesk Utilities
dSPACE ASM Diesel Exhaust
dSPACE ASM Pneumatics
Automotive Simulation Models
Version 6.0
dSPACE

Figure 3.11: Screenshot of Traffic Simulator Automotive Simulation Models pane

The dSPACE ASM Vehicle Dynamics only provides full vehicle dynamics for the host vehicle, however, dSPACE ASM Traffic enables the user to define traffic scenarios and traffic vehicles in addition to full host vehicle dynamics. Since in this research, the presence of traffic vehicles is essential, we choose dSPACE ASM Traffic.

The ASM VehicleDynamic library consists of different blocks for parts of vehicles such as vehicle dynamics, engine, drivetrain and soft ECUs. The ASM Traffic also offers a demo model with default parameters for a standard mid-size car called Automotive Simulation Models Traffic version3 which is used in this research. The complete demo project consists of the following [65]:

- Simulation: ASM vehicle dynamics model including initialization
- Parametrization: ModelDesk with default parameters of mid-size car
- Instrumentation: ControlDesk next generation experiment to control real-time simulation
- Animation: MotionDesk default scene and parametrization for road and maneuver
- Automation: Remote control of ModelDesk

By choosing the default model, the following *.mdl file will be generated.

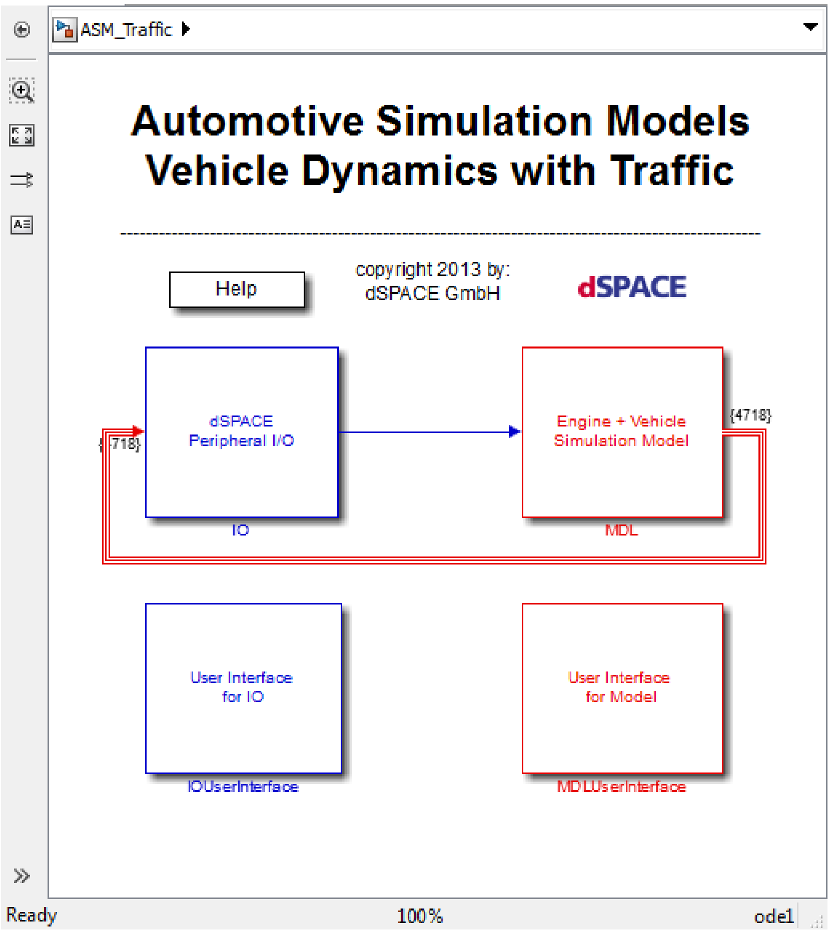


Figure 3.12: Screenshot of Traffic Simulator Automotive Simulation Models pane

There are two different ways to change this simulation environment and vehicle dynamics:

- Applying required changes directly through Simulink file
- Using user graphical interfaces ModelDesk and MotionDesk

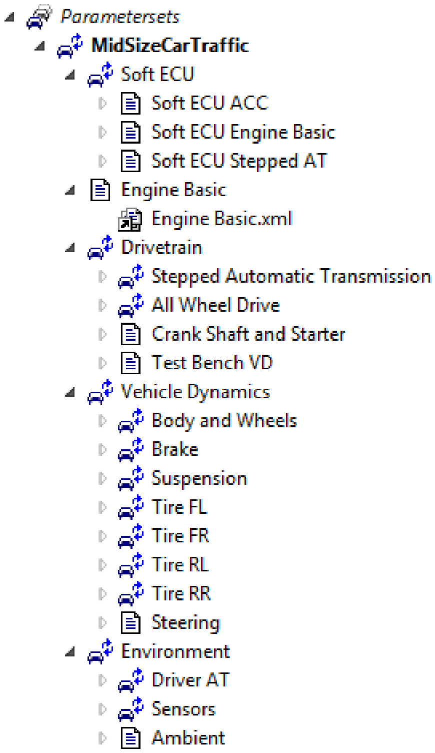
We can open the demo project & experiment on ModelDesk and MotionDesk by going to User Interface for Model and clicking on corresponding blocks in Projects and Experiment Handling.

3.8.1 ModelDesk

The ModelDesk let the user to specify and change parameters related to vehicle dynamics, build a maneuver for host vehicle, instruct and generate road profiles, add multiple traffic participant and build maneuver for each of them, and create and manage traffic objects. The following sections will explain each of these features briefly [66].

Parametrizing ASM Model

The parameters of vehicle dynamics can be edited in Parametersets section MidSizeCar-Traffic. It has following main components:



- *Parametersets*
  - **MidSizeCarTraffic**
    - Soft ECU
      - Soft ECU ACC
      - Soft ECU Engine Basic
      - Soft ECU Stepped AT
    - Engine Basic
      - Engine Basic.xml
    - Drivetrain
      - Stepped Automatic Transmission
      - All Wheel Drive
      - Crank Shaft and Starter
      - Test Bench VD
    - Vehicle Dynamics
      - Body and Wheels
      - Brake
      - Suspension
      - Tire FL
      - Tire FR
      - Tire RL
      - Tire RR
      - Steering
    - Environment
      - Driver AT
      - Sensors
      - Ambient

Figure 3.13: Screenshot of parameter set in ModelDesk

- *Soft ECU*: This section enables the user to edit and change soft electronic control units for ACC, Engine Basics and Stepped Automatic Transmission.

- *Engine Basics*: The user can change or edit the engine graph and specify some parameters of engine dynamics like time constant for torque increase, etc.

- *DriveTrain*: In this section, variables and graphs for Crankshaft and starter, transmission, final drive assembly, and test bench (speed controller for the engine under test) can be chosen and edited. In the final drive assembly, the user can choose between different drivetrains like all-wheel, front wheel, rear wheel, front wheel front engine drive and front-wheel front-engine 4WD drive. The transmission can be automatic stepped gearbox with torque converter or manual gearbox with the ability to change general parameters and graphs besides efficiency graphs.

- *Vehicle Dynamics*: In this section for each of the tires, user can select between TMeasy and Magic Formula tire models and specify relevant parameters (geometry, forces, self-alignment, scaling parameters and dynamics) for each of the models in dry, damp, wet, and icy condition. Also variables of components of brake system (brake booster, brake hydraulics, master brake cylinder and brake itself), steering (steering, power steering graph and steering compliance graph), body-wheels (aerodynamics, wheel mass and inertial, torsional frame, sensor position, body geometry and mass and additional loads) can be accessed and edited accordingly. Moreover, in this section, user can choose between different suspension systems (symmetric, asymmetric, McPherson, ASYM_ANC.spring, SYM_ANC.spring, ASYM_ANC.spring 3 DOF, SYM_ANC.spring 3DOF, Rigid Symmetric, symmetric 3DOF, asymmetric 3DOF) and change relevant parameters (suspension kinematics, suspension compliance and spring-damper-stabilizer each of which has different features) for both front and rear suspension system.

- *Environment*: Different selection of sensors, ambient temperature, and pressure, driver information (brake invariant, longitudinal controller, lateral controllers, and common driver parameters) in addition to the road, maneuver and traffic can be specified in this section.

Maneuver Editor

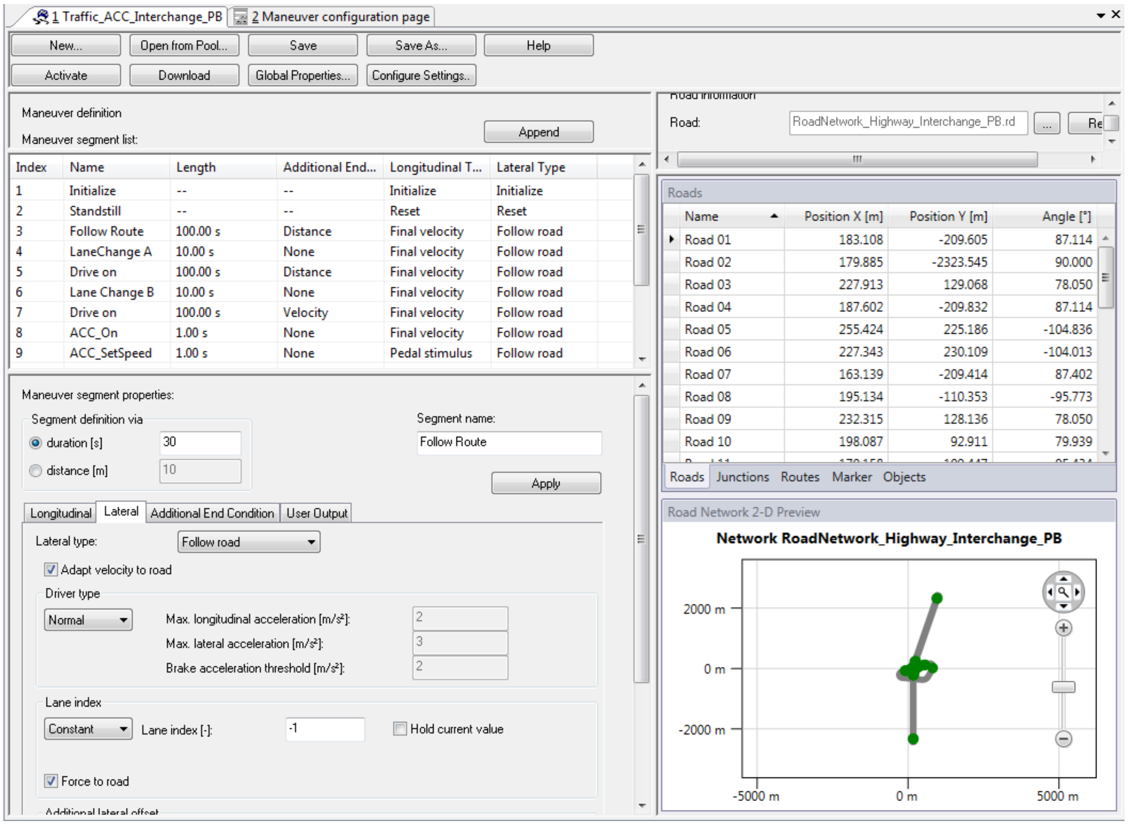
Maneuver defines how a vehicle moves. The user can choose between some already existing maneuvers or build a new maneuver. The vehicle can move on a flat surface or follow a route already specified with the road generator. In order to build a maneuver, we can append multiple maneuver segments in which each segment represents part of the overall plan and the length of each segment can be defined by duration or a distance. For each maneuver segment, different longitudinal and lateral driving types exits that can be picked with options of adding additional end condition for that segment and some user outputs.

Depending on the selected longitudinal and lateral type, further setup is required. There are two different maneuver types available on traffic simulator:

- *Stimulus maneuver*: This option allows controlling the vehicle model from outside by giving signals to the accelerator pedal, brake pedal, clutch pedal, gear, and steering wheel generated by an actual driver or additional controller developed by researchers.

- *Controlled maneuver*: In this case, the driver model specified in the vehicle dynamics section is responsible for controlling the vehicle. The steering wheel is controlled by lateral driver controllers and the rest of the pedals are controlled by a longitudinal driver controller.

In addition to maneuver segments, we have initialize, standstill, reset, and end segments. The initialize segment starts the vehicle dynamics model. The initial position, speed, and height can also be specified in the initialize segment. The Standstill segment mostly comes after initialize segment and before the reset segments to let the vehicle dynamics stop bouncing and go to the steady-state situation. The reset segment is responsible for re-initializing the vehicle dynamics. The end segment which is the last in the overall plan indicates the end of the maneuver.



Maneuver definition

Maneuver segment list:

Index	Name	Length	Additional End...	Longitudinal T...	Lateral Type
1	Initialize	--	--	Initialize	Initialize
2	Standstill	--	--	Reset	Reset
3	Follow Route	100.00 s	Distance	Final velocity	Follow road
4	LaneChange A	10.00 s	None	Final velocity	Follow road
5	Drive on	100.00 s	Distance	Final velocity	Follow road
6	Lane Change B	10.00 s	None	Final velocity	Follow road
7	Drive on	100.00 s	Velocity	Final velocity	Follow road
8	ACC_On	1.00 s	None	Final velocity	Follow road
9	ACC_SetSpeed	1.00 s	None	Pedal stimulus	Follow road

Maneuver segment properties:

Segment definition via

● duration [s] 30

○ distance [m] 10

Segment name: Follow Route

Lateral type: Follow road

☑ Adapt velocity to road

Driver type

Normal

Max. longitudinal acceleration [m/s²]: 2

Max. lateral acceleration [m/s²]: 3

Brake acceleration threshold [m/s²]: 2

Lane index

Constant Lane index [-]: -1 □ Hold current value

☑ Force to road

Additional lateral offset

Road information

Road: RoadNetwork_Highway_Interchange_PB.rd

Roads

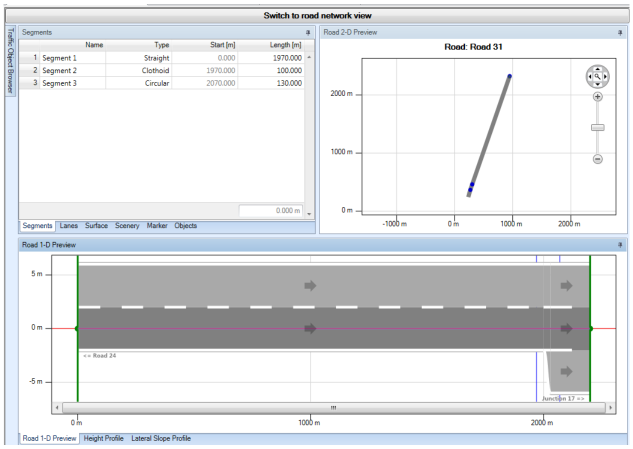
Name	Position X [m]	Position Y [m]	Angle [°]
Road 01	183.108	-209.605	87.114
Road 02	179.885	-2323.545	90.000
Road 03	227.913	129.068	78.050
Road 04	187.602	-209.832	87.114
Road 05	255.424	225.186	-104.836
Road 06	227.343	230.109	-104.013
Road 07	163.139	-209.414	87.402
Road 08	195.134	-110.353	-95.773
Road 09	232.315	128.136	78.050
Road 10	198.087	92.911	79.939

Figure 3.14: Screenshot of maneuver editor in ModelDesk

Road Creation

We use this component to generate road models. We define different segments of the road and specify its features. The generated road file integrates road with the animation scene in MotionDesk and in Simulink model. The road generator has the following capabilities:

- Creating road and specify features such as horizontal profile, height profile, lane selection, texture, scenery, and special surface condition
- Creating junctions and connecting them to roads
- Placing static objects around the road like a traffic sign
- Defining routes on the road for ASM maneuver segments
- Defining traffic type
- Placing some position markers and trigger points that can be used as additional end condition for maneuver segments or trigger a segment



	Name	Type	Start [m]	Length [m]
1	Segment 1	Straight	0.000	1970.000
2	Segment 2	Clothoid	1970.000	100.000
3	Segment 3	Circular	2070.000	130.000

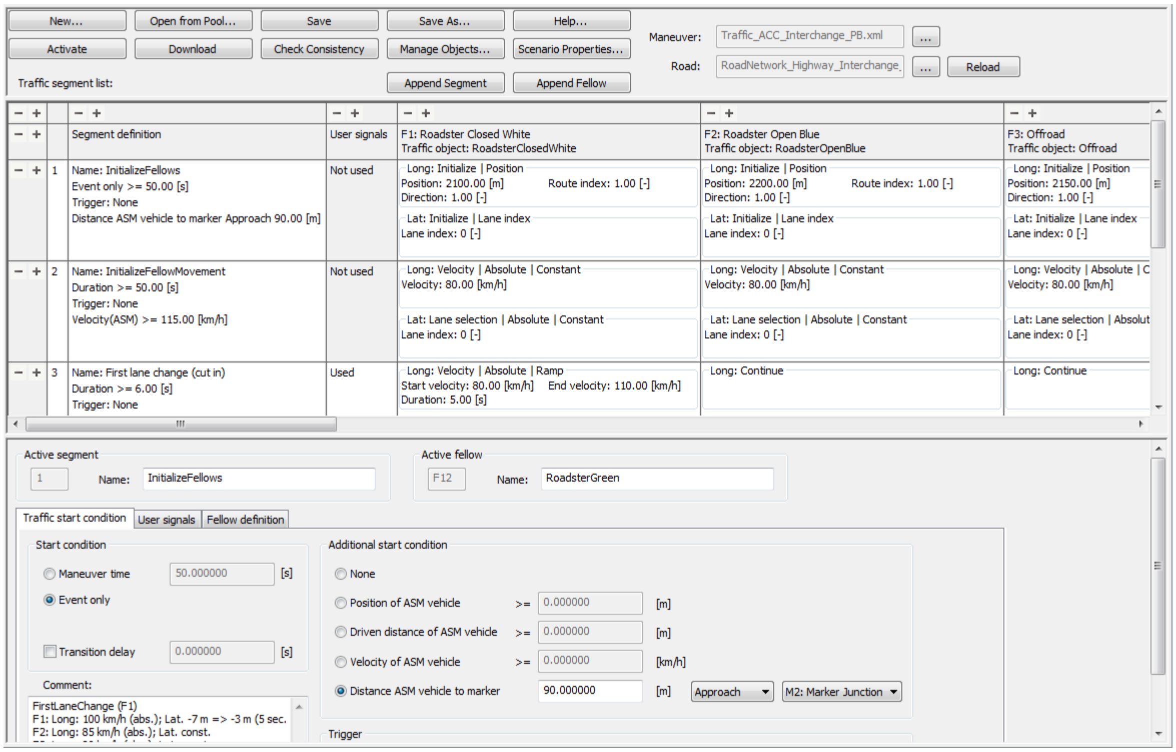
Figure 3.15: Screenshot of road creation pane in ModelDesk

Traffic

The traffic editor section let the user to create a traffic scenario with some traffic participants (traffic fellows) that follow a certain maneuver such as vehicle overtaking, changing lane, and reacting to other traffic fellows or ASM vehicle. In this section, we can append fellow vehicles and assign a chassis or chassis with wheels of different vehicles to it. Also some dynamic and geometric parameters can be set for each fellow vehicle.

The user defines maneuver for each fellow vehicle to follow. The maneuver consists of sequence of segments to follow. The initial segment is always the first segment which characterize start condition of fellow vehicle with respect to ASM vehicle or maneuver. Moreover, multiple initial conditions can be set and the fellow starts when one of those start conditions is met. The normal segments come after initial segment and defines longitudinal and lateral movements of a fellow in traffic scenario. There are couple of longitudinal and lateral type of maneuvers exists and each of them has further setup. The conditions for transmission from one normal segment to next segment can be specified. The movements can be defined with absolute values or relative to the ASM vehicle or other fellow vehicles. Each fellow vehicle needs to be assigned to a route created in the road creation section.

After making proper changes and saving vehicle dynamics, ASM vehicle maneuver, road, and traffic, we download each of the sections. By downloading, the blocks in the Simulink, *.mdl file changes accordingly and scene comes up in the MotionDesk.



Maneuver: Traffic_ACC_Interchange_PB.xml

Road: RoadNetwork_Highway_Interchange_

Traffic segment list:

	Segment definition	User signals	F1: Roadster Closed White Traffic object: RoadsterClosedWhite	F2: Roadster Open Blue Traffic object: RoadsterOpenBlue	F3: Offroad Traffic object: Offroad
1	Name: InitializeFellows Event only >= 50.00 [s] Trigger: None Distance ASM vehicle to marker Approach 90.00 [m]	Not used	Long: Initialize | Position Position: 2100.00 [m] Route index: 1.00 [-] Direction: 1.00 [-] Lat: Initialize | Lane index Lane index: 0 [-]	Long: Initialize | Position Position: 2200.00 [m] Route index: 1.00 [-] Direction: 1.00 [-] Lat: Initialize | Lane index Lane index: 0 [-]	Long: Initialize | Position Position: 2150.00 [m] Direction: 1.00 [-] Lat: Initialize | Lane index Lane index: 0 [-]
2	Name: InitializeFellowMovement Duration >= 50.00 [s] Trigger: None Velocity(ASM) >= 115.00 [km/h]	Not used	Long: Velocity | Absolute | Constant Velocity: 80.00 [km/h] Lat: Lane selection | Absolute | Constant Lane index: 0 [-]	Long: Velocity | Absolute | Constant Velocity: 80.00 [km/h] Lat: Lane selection | Absolute | Constant Lane index: 0 [-]	Long: Velocity | Absolute | C Velocity: 80.00 [km/h] Lat: Lane selection | Absolut Lane index: 0 [-]
3	Name: First lane change (cut in) Duration >= 6.00 [s] Trigger: None	Used	Long: Velocity | Absolute | Ramp Start velocity: 80.00 [km/h] End velocity: 110.00 [km/h] Duration: 5.00 [s]	Long: Continue	Long: Continue

Active segment

1 Name: InitializeFellows

Active fellow

F12 Name: RoadsterGreen

Start condition

○ Maneuver time 50.000000 [s]

● Event only

□ Transition delay 0.000000 [s]

Comment:

FirstLaneChange (F1)
F1: Long: 100 km/h (abs.); Lat. -7 m => -3 m (5 sec.
F2: Long: 85 km/h (abs.); Lat. const.

Additional start condition

○ None

○ Position of ASM vehicle >= 0.000000 [m]

○ Driven distance of ASM vehicle >= 0.000000 [m]

○ Velocity of ASM vehicle >= 0.000000 [km/h]

● Distance ASM vehicle to marker 90.000000 [m] Approach M2: Marker Junction

Trigger

Figure 3.16: Screenshot of traffic pane in ModelDesk

3.8.2 MotionDesk

MotionDesk is a product of dSPACE for visualizing traffic scenario and lets the user to directly make some changes. One capability of MotionDesk is to record a simulation of traffic scenario as a *.mdf file and reply existing *.mdf files.

The scene synchronization feature in ModelDesk synchronize MotionDesk, ModelDesk, and Simulink model and updates scenery of MotionDesk [67].

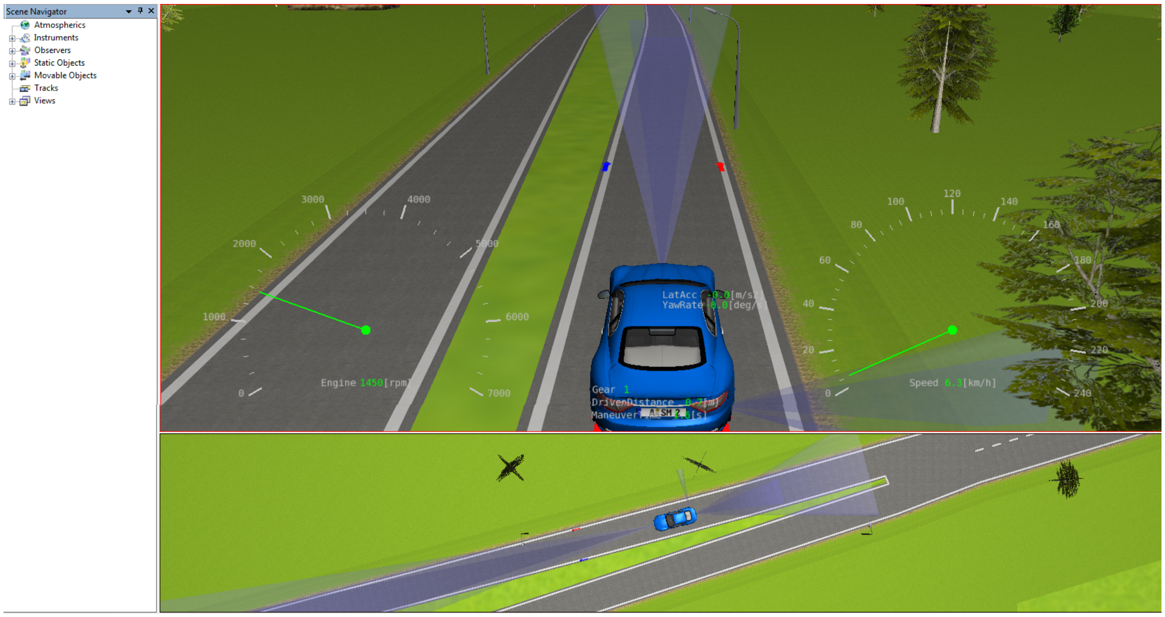


Figure 3.17: Screenshot of MotionDesk

3.8.3 Traffic Simulator Results

In this thesis, videos of the performance of the proposed controllers have been generated using dSPACE traffic simulator. The videos are available upon request.

Chapter 4

Decentralized NMPC Coordination Approach

4.1 Design Overview

In this method, we assume that the host vehicle h equipped with the proposed controller approaches the roundabout and wants to merge with the traffic inside it and pass through without collision with other vehicles. The traffic vehicle t is also approaching the same roundabout. We define a merging zone which is the light blue zone shown by m in Fig. 4.1. The distance that each vehicle should travel to arrive at merging zone is d. In order to have collision free merging with traffic inside roundabout, the host vehicle should enter merging zone before or after vehicle t.

Also, we define a control zone shown by light orange in Fig. 4.1. The length from the entrance of control zone to entrance of merging zone is L. When the host vehicle enters this control zone, it gets information of surrounding traffic vehicles within a specific range from it. Based on the local information, the controller determines the following two outputs:

- In decentralized coordination, each of the vehicles calculates their own priority with respect to other traffic vehicles. This is possible since the vehicles are connected via V2V and V2I communication systems. Additionally, because of the fact that each of the vehicles adjust their position with respect to other vehicles, all of the N number of vehicles can pass through roundabout without collision and situation of deadlock can be prevented.

- The controller calculates the needed torque to satisfy all of the objectives which have been considered in the development phase.

The ego vehicle gets information about traffic through communication systems, then it calculates priority and the required torque to adjust its position and speed with respect to other vehicles to have a safe and smooth merging with the traffic inside the roundabout while minimizing fuel consumption of the vehicle when safe inter-vehicular distance is available. As a result, priority determination and solving the optimization problem fast and in real-time is required for successful implementation of the controller.

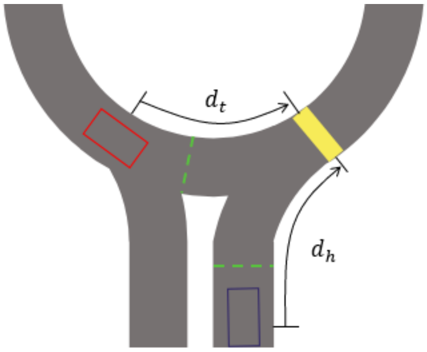


Figure 4.1: Single lane traffic circle. The blue vehicle is the host vehicle. The red vehicle is traffic vehicle. The yellow zone is the merging zone. Dashed green lines are the entrance to control zone.

4.2 Priority Calculation

4.2.1 Navigation Function Approach

If we want to have collision free merging in a roundabout, defining a navigation function is needed. When vehicle h is approaching the roundabout Fig. 4.1, first it enters the control zone. Since the vehicles are connected and automated, it gets information of other vehicles which are close to the same merging zone. Therefore, the vehicle h is informed of presence of another vehicle t and receives critical driving data of vehicle t such as distance of vehicle t to the merging zone, velocity and acceleration. Then by having critical driving data and distance of control zone, each of the vehicles can calculate their arrival time to the merging zone. In order to achieve safe merging objective, vehicle h should be in front of or behind vehicle t with a safe inter-vehicular distance between them when it enters the merging zone. Consequently, the following navigation function can be defined:

$$\beta = \begin{cases} d_t - d_h - (d^*) & T_h > T_t, \\ d_h - d_t - (d^*) & T_t \geq T_h, \end{cases} \tag{4.1}$$

Where d_t is distance of traffic vehicle t to the merging zone and d_h is distance of host vehicle to the same merging zone. T_h is arrival time of host vehicle at the entrance of merging zone calculated on-board by host vehicle and T_t is the time takes for the traffic vehicle to be at the merging zone and given to the host vehicle via V2V communication system.

According to the navigation function equation (4.1), when the arrival time of vehicle t is smaller than arrival time of vehicle h, the vehicle t which is the preceding vehicle should accelerate and vehicle h should decelerate since it is the back vehicle until a safe gap exists between them before entering merging zone and vice-versa. This can be done by considering β term in our cost function. This is possible because all of the vehicles are equipped with the controller and connected to each other. Moreover, this is also valid for the case that both of the vehicles are in a same path because the vehicle which is closer to the merging zone is preceding and the other vehicle is back vehicle and they adjust their position with respect to each other to provide safe gap [68].

4.2.2 Optimal Control Approach

In this novel approach, instead of using a navigation function to calculate priorities for each of the vehicles, we formulate the problem in a form of nonlinear optimal control problem with different objectives and solve that problem using a fast optimizer. One advantage of this approach is that we can consider different objectives while calculating priorities or put more weight on one of the objectives such as energy-cost. In this approach, the controller on the host vehicle once assumes to be the preceding then the back vehicle and calculates cost function for each case and makes decision based on the determined cost functions.

The pseudo code for the optimal control approach for calculation of priority is shown in Algorithm 1:

The code accepts the position, speed, and distance of both vehicles t and h to the merging zone. At each time step, once it assumes the vehicle h, for instance, is the preceding vehicle and solves the NMPC problem to find J_p which is the cost when the host

Algorithm 1 Priority Calculation Logic- Decentralized Coordination

```
Require: $x_h, v_h, x_t, v_t, \beta, \beta', d_t, d_h, d_c$
Ensure: $T_w$
$J_p \leftarrow NMPC(x_h, v_h, x_t, v_t, d_t, d_h, \beta)$
$J_b \leftarrow NMPC(x_h, v_h, x_t, v_t, d_t, d_h, \beta')$
while $d_h/d_c > 1$ do
    if $J_p \leq J_b$ then
        $T_w \leftarrow T_{w,p}$
    else
        $T_w \leftarrow T_{w,b}$
    end if
end while
while $d_h/d_c \leq 1$ do
    if $d_h < d_t$ then
        $T_w \leftarrow T_{w,b}$
    else
        $T_w \leftarrow T_{w,p}$
    end if
end while
```

vehicle is the preceding vehicle. Then, it supposes the vehicle h to be the back vehicle and determines J_b that is the cost of being the back vehicle. By a comparison between J_p and J_b, the more cost-effective option will be picked and the powertrain torque calculated based on that priority will be returned as the output.

By implementing this priority calculation logic, the vehicles can be the preceding or back vehicle depending on the situation and there is flexibility in providing a better fuel economy. To illustrate more, consider a situation when the vehicle t is on a different path than the vehicle h and it is closer to the merging zone with lower speed. By implementing this logic, the vehicle h decides to be preceding and keeps its current speed to go in front of the vehicle t which results in better fuel consumption. However, if the vehicle is in the same trajectory, it causes a collision.

To avoid the collision when the vehicles are on the same trajectory, we incorporate a collision-avoidance layer in our logic. The collision avoidance layer is shown below:

$$\frac{d_h}{d_c} \leq 1 \tag{4.2}$$

Where d_h is the distance of host vehicle to the merging zone calculated on-board and d_c is the distance to collision with the other vehicle. When both of the vehicles are on the same path, d_c given by the car-mounted sensors is lower than or equal to d_h. In these situations, there is no flexibility for the vehicles to calculate priority; hence, the vehicle which is closer to the merging zone has priority over the other vehicle.

4.3 Prediction Model

As mentioned in the centralized approach, the prediction (control-oriented) model needs to be specified for the controller to predict the dynamical behavior of the system. Since in this chapter we are introducing a different and more promising coordination approach, decentralized approach, the prediction model changes accordingly as follows:

$$\begin{bmatrix} \dot{x}_h(t) \\ \dot{v}_h(t) \\ \dot{T}_w \\ \dot{x}_t(t) \\ \dot{v}_t(t) \\ \dot{a}_t \end{bmatrix} = \begin{bmatrix} v_h \\ \frac{T_w}{m r_w \eta_p} - \frac{F_{res}}{m} \\ -\frac{T_w}{\tau_a} + \frac{U}{\tau_a} \\ v_t \\ a_t \\ -\sigma . a_t \end{bmatrix} \tag{4.3}$$

$$F_{res} = \frac{1}{2}\rho c_d A v_h^2 + mg\sin\theta + \mu_r mg, \tag{4.4}$$

The parameters are the same as before. Moreover, the safe inter-vehicular distance formula to determine distance between vehicles, the prediction model to anticipate acceleration of traffic vehicle, and trip cost formula are the same as centralized approach with some notation differences.

4.4 Optimal Control Problem Formulation

The purpose of this section is to define cost function with the following constraints:

$$\begin{aligned} v_h &\leq v_{max}, \\ U &\leq U_{max}, \\ U_{min} &\leq U, \end{aligned} \tag{4.5}$$

Where U and v_{max} are the same parameters as centralized controller.

The performance index for this problem is as follows:

$$\begin{aligned} PI = \sum_{i=0}^{N-1} U(i)^2 + R^T C_x(i) + w_1(v_h(i) - v_{ref}(i))^2 \\ + w_2 E_{cost} + w_3 \beta e^{\beta}, \end{aligned} \tag{4.6}$$

The constant parameters in the above formula can be found by trial-and-error experiments or based on more structured logic.

In the PI equation (4.6), C_x is a vector of constraints defined as follows:

$$C_x = \begin{bmatrix} v_h - v_{max} \\ U - U_{max} \\ U_{min} - U \end{bmatrix} \tag{4.7}$$

One way to define the weighting parameter w_1 is: [58]:

$$w_1 = \begin{cases} e^{\gamma_1(\beta - 0.6\beta_{max})} & if 0.6\beta_{max} < \beta, \\ 50 & if -0.6\beta_{max} \leq \beta \leq 0.6\beta_{max}, \\ e^{\gamma_2(-\beta - 0.6\beta_{max})} & if \beta < -0.6\beta_{max}, \end{cases} \tag{4.8}$$

To solve this problem, the trajectory cost equation 4.6 is replaced by:

$$PI' = PI + M, \tag{4.9}$$

$$M_j = \begin{cases} 0 & C_x(x_k, u_k) \leq 0, \\ m_j C_x(x_k, u_k) & C_x(x_k, u_k) > 0, \end{cases} \tag{4.10}$$

This means that the constraint is considered as a penalty function when it is violated. From the optimal control theory, by defining Hamiltonian H, co-state Λ and Lagrangian multiplier vector ψ, the optimal control problem can be rewritten as a two-point boundary value problem.

$$\begin{cases} x_{k+1} = x_k + f(x_k, u_k)\delta\tau, \\ C_x(x_k, u_k) \leq 0, \\ H_u^T(x_k, u_k, \Lambda_k, \psi_k), \\ \Lambda_k = \Lambda_{k+1} + H_x^T(x_k, u_k, \Lambda_k, \psi_k), \end{cases} \tag{4.11}$$

Solving state equation forward in time, and the co-state equation backward, the problem can be formulated as follows:

$$U(t) = [u_0, \psi_0^{'}, u_1, \psi_1^{'}, u_2, \psi_2^{'}, ..., u_{N-1}, \psi_{N-1}^{'}]^T, \tag{4.12}$$

$$F(U, x, t) = \begin{bmatrix} H_u(x_0, u_0, \psi_0, \Lambda_1) \\ g_m(x_0, u_0,) \\ H_u(x_1, u_1, \psi_1, \Lambda_2) \\ g_m(x_1, u_1,) \\ . \\ . \\ . \\ H_u(x_{N-1}, u_{N-1}, \psi_{N-1}, \Lambda_N) \\ g_m(x_{N-1}, u_{N-1}) \end{bmatrix} = 0 \tag{4.13}$$

Where g_m is the former inequality constraint modified by dummy inputs as described in [69]. To solve this problem, we need a specific method capable of solving the problem sufficiently fast like GMRES method [10, 70].

4.5 Control Evaluation Results

L_0 and κ in safe distance equation (5.1) were chosen as 3 and 0.2 respectively. The prediction horizon length is 10 and the step time is 0.25 seconds. The high-fidelity model of Toyota Prius PHEV was used to check the performance of the proposed vehicle controller in different scenarios.

In simulations, the origin was put in the merging zone. Therefore, when x_h is -30 meters, it means that the vehicle h needs to travel 30 meters on its fixed path to be in the merging zone. During the simulations, the following weighting factors were used: $w_1 = adaptively$, $w_2 = 0$ for NMPC without energy cost and $w_2 = 10$ for NMPC with energy cost, $w_3 = 15$ and R=[150 0.1 0.1] for v_h, U_{max} and U_{min} respectively.

4.5.1 NMPC vs PID Performance

In the first step, we compared the proposed NMPC-based controller's performance with a PID controller. For the sake of comparison between controllers, the scenario is the same and a limited portion of HWFET drive cycle has been used and scaled to generate the speed profile of vehicle t.

For the special case of this thesis, because we get information of surrounding vehicles through V2V and V2I communications rather than radar, it is possible that the vehicle h be closer to the vehicle t than the safe gap while they are on a different path.

As shown in Fig. 4.2, we assume that the vehicle t has lower arrival time to the merging zone and has priority over the vehicle h. So, the vehicle h needs to decelerate to provide a safe gap for merging before the merging zone. The PID controller tries to provide a safe gap between the vehicles. Both controllers reduced the speed of vehicle h and at the merging zone which is located at $y = 0$, a safe gap exists between vehicles. However, the NMPC approach generates slightly better results in comparison to PID.

The NMPC controller can adhere to the priorities when the arrival time of vehicle h is lower and has priority over the vehicle t as shown in Fig. 4.3, but the PID controller lets the vehicle t enters the merging zone first and violates the priority.

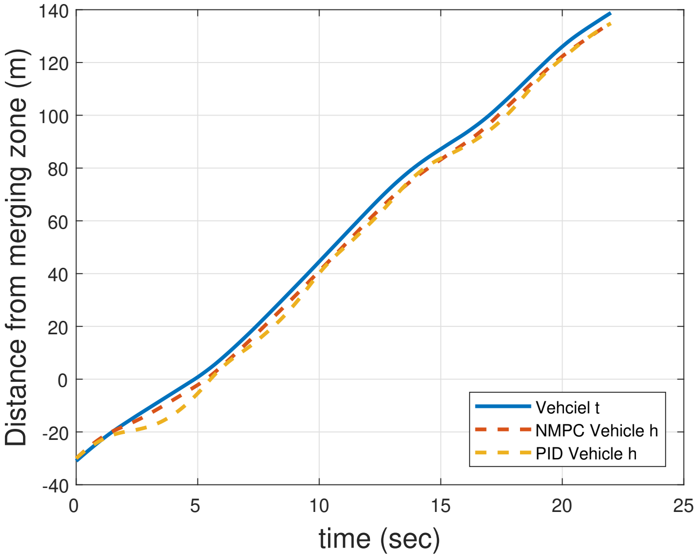


Figure 4.2: Comparison between NMPC and PID when vehicle t has priority © 2019 IEEE

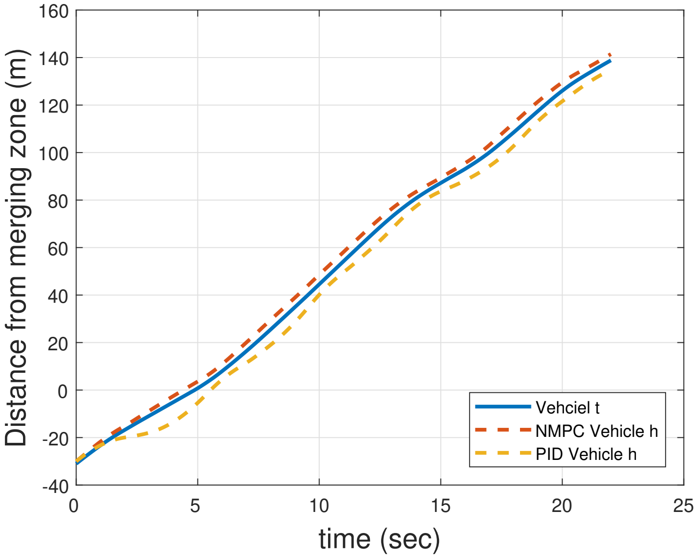


Figure 4.3: Comparison between NMPC and PID when vehicle h has priority © 2019 IEEE

Another feature in this multi-objective optimal control problem is following the desired speed. We assumed that the desired speed in this thesis is the speed profile shown in Fig. 4.4. The NMPC controller has much better performance at following the desired speed than PID because PID does not consider other objectives than minimizing the error between actual and the desired inter-vehicular distance.

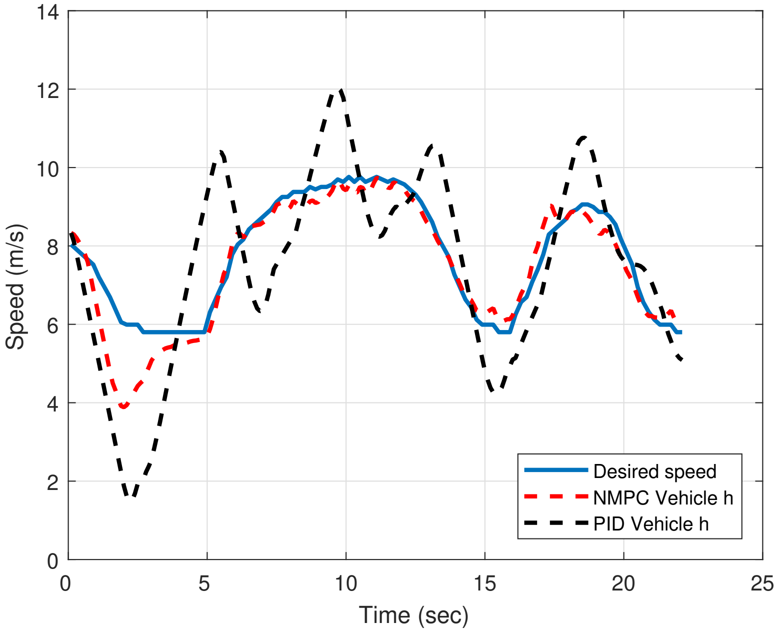


Figure 4.4: Speed following comparison between NMPC and PID © 2019 IEEE

4.5.2 Energy-cost Comparison

In the NMPC formulation, we considered the energy cost of the complex powertrain of PHEV. The NMPC with and without energy cost in its cost function is compared as shown in Fig. 4.5. The NMPC controller is better at reducing energy consumption when the energy cost of the vehicle is considered in its performance index.

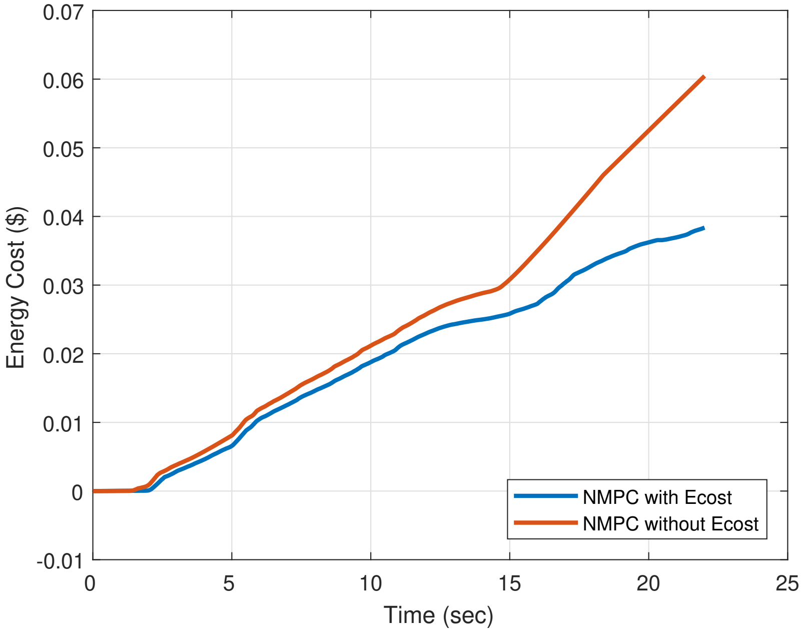


Figure 4.5: Energy cost comparison between different NMPC © 2019 IEEE

Table 4.1: Final energy cost for different NMPC in different drive cycles © 2019 IEEE

Cycle	with E_{cost}	without E_{cost}
portion of HWFET	0.0384	0.0605
3*HWFET	5.5192	8.5942

4.5.3 Comparison between Priority Calculation Approaches

As mentioned before, this thesis presents a novel priority calculation logic for decentralized coordination. In this section, we want to compare the performance of the two priority calculation logics. Fig. 4.6 compares the performance of these two approaches. The initial condition is the same and the results are for 2×HWFET driving cycles. According to Fig. 4.6(a) both methods have acceptable performance in following the reference speed but optimal control priority calculation logic's results are closer to zero. Fig. 4.6(b) compares the position error and again both methods results are within safe limits. Therefore, the optimal control priority rule has acceptable performance compared to the other logic.

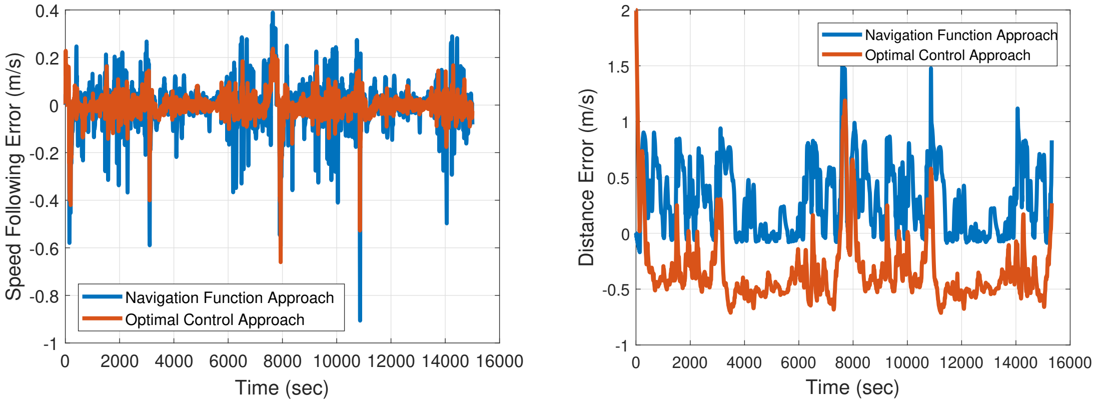


Figure 4.6: (a) Velocity, and (b) position error comparison between different priority calculation logic in 2×HWFET driving cycles

Chapter 5

Robust tube-based Decentralized NMPC Coordination Approach

5.1 Preliminaries

The following notations have been used in this chapter. For sets X and Y: $X = \{x|x \in X\}$ and $Y = \{y|y \in Y\}$ while A and B are constants, then $AX = \{Ax|x \in X\}$, $AX + BY = \{Ax + By|x \in X, y \in Y\}$. $\oplus$ presents Minkowski sum $X \oplus Y = \{y + x|x \in X, y \in Y\}$ and $\oplus_{k=0}^{\infty} X_k = X_1 \oplus X_2 \oplus ... \oplus X_{\infty}$.

5.2 Modeling

The models in the control systems development process can be divided into control-oriented models and high-fidelity models. For the sake of simulation, the high-fidelity model of a baseline vehicle, Toyota Prius Plug-in hybrid, was used in this research. The high-fidelity model has necessary verified components related to longitudinal dynamics and fuel consumption measurements. Although this model's fidelity enables investigation of the proposed controller, it is computationally expensive which makes defining a new model to be used inside controller compulsory. The control-oriented model describes the behavior of the system and is fast enough to ensure real-time application. The high-fidelity model has been explained in the previous chapters, therefore we concentrate on particular control-oriented model of robust decentralized control approach in this chapter. For safe merging of a vehicle with traffic inside the roundabout, a safe distance between vehicles need to be specified. Since the distance of vehicles to the merging zone is known, among different definitions for a safe distance, constant time headway policy suits our need. In this rule, the safe distance between vehicles is a function of the safe distance at parked position L_0 and headway time κ to have a fixed time gap between vehicles in different speeds:

$$d^* = L_0 + \kappa v_h, \tag{5.1}$$

Based on the above policy, the safe distance between vehicles can be found by the following formula:

$$\begin{aligned} \beta &= d^m_{t|h} - d^m_{h|h} - d^*, \\ \beta' &= d^m_{h|h} - d^m_{t|h} - d^*, \end{aligned} \tag{5.2}$$

$d^m_{t|h}$ is the distance traffic vehicle needs to travel to be at the merging zone and $d^m_{h|h}$ means the distance between merging area and host vehicle calculated on-board.

The proposed controller tries to minimize the difference between the actual distance between vehicles and the safe inter-vehicular distance equation (5.2). β is when the traffic vehicle is preceding vehicle and β' is when the host vehicle is preceding.

The following model in state-space form can be used to design the controller:

$$\begin{bmatrix} \dot{x}_h(t) \\ \dot{v}_h(t) \\ \dot{T}_w \\ \dot{x}_t(t) \\ \dot{v}_t(t) \\ \dot{a}_t(t) \end{bmatrix} = \begin{bmatrix} v_h \\ \frac{T_w}{m r_w \eta_p} - \frac{F_{res}}{m} \\ -\frac{T_w}{\tau_a} + \frac{U}{\tau_a} \\ v_t \\ a_t \\ -\sigma.a_t \end{bmatrix} \tag{5.3}$$

$$F_{res} = \frac{1}{2}\rho c_d A v_h^2 + mg\sin\theta + \mu_r mg, \tag{5.4}$$

Where F_{res} is the applied resistive force to the vehicle, T_w is the torque applied to the wheels, r_w is the radius of the wheel, η_p is the efficiency of the powertrain, τ_a is a constant and σ is a decaying parameter. ρ is density of the air, c_d is drag constant, vehicle frontal area is shown by A, v_h is speed of the vehicle, m is mass, μ_r is rolling resistance constant and θ is grade of the road. The host vehicle is presented by index h and traffic vehicle by t.

For non-robust MPC the prediction model is assumed as an accurate description of the real plant. Therefore, we need to solve the following problem:

$$\begin{gathered}\bar{x}_{k+1} = A_k\bar{x}_k + B_k\bar{u}_k + \bar{g}(\bar{x}_k),\\ \bar{x} \in X, \bar{u} \in U,\end{gathered} \tag{5.5}$$

$$\begin{gathered}J_N(x_k, u) = \bar{x}_{k+n|k}^T P\bar{x}_{k+n|k} +\\ \sum_{i=0}^{N-1} \bar{x}_{k+i|k}^T Q\bar{x}_{k+i|k} + \bar{u}_{k+i|k}^T R\bar{u}_{k+i|k}\end{gathered} \tag{5.6}$$

In equation (5.5), nominal state is presented by $\bar{x}$, nominal controls by $\bar{u}$, and $\bar{g}$ is a representative of nonlinear terms to take into account the effect of nonlinearities.

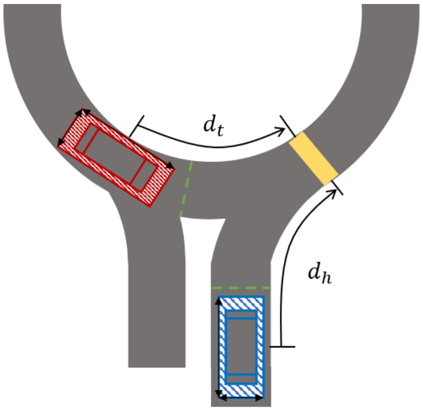


Figure 5.1: Roundabout with single-lane roads. The blue vehicle is the host vehicle. The traffic vehicle is red. The yellow zone is the merging zone. Dashed green lines are the entrance to the control zone before the merging zone. A safety envelope exists around vehicles for safety and consideration of DUs.

However, we can almost never meet this assumption in practice, since the real plant is subject to some disturbances and uncertainties (DU) which cause deviations of the real plant from the nominal states.

There are different methods to consider the effect of DUs, one of which is robust tube-based NMPC that considers a safety envelope around the vehicles shown in Fig. 5.1. In the tube-based NMPC, we assume the system is subject to bounded additive disturbances.

Consequently, the problem can be rewritten as follows:

$$\begin{gathered}\hat{x}_{k+1} = A_k\hat{x}_k{+}B_k\hat{u}_k + \hat{g}(\hat{x}_k) + w_k,\\ w \in W,\end{gathered} \tag{5.7}$$

$$\begin{aligned}&J_N(x_k, u) = \hat{x}^T_{k+n|k}P\hat{x}_{k+n|k}+\\ &\sum_{i=0}^{N-1}\hat{x}^T_{k+i|k}Q\hat{x}_{k+i|k} + \bar{u}^T_{k+i|k}R\bar{u}_{k+i|k}\end{aligned} \tag{5.8}$$

Where $\hat{x}$ and $\hat{u}$ present actual state and control. w takes value from W and is the disturbance term.
For the case of a linear system:

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{5.9}$$

The robust tube-based MPC deploys a stable Luenberger observer to estimate states of the real system. The observer will be controlled such that the following equation satisfies the control and state constraints:

$$x_k = \hat{x}_k + \tilde{x}_k, \tag{5.10}$$

Where x_k is the actual state of the system, $\hat{x}_k$ is the measured state and $\tilde{x}_k$ is the estimation error guaranteed to be bounded by an invariant set.

To achieve this goal, the MPC will be responsible to solve the nominal system. The nominal system can be found by neglecting the effects of DUs. To consider the presence of DUs, the control input is a combination of control input generated by MPC for the nominal system and the input generated by linear stabilizing controller [71, 72].

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k, \tag{5.11}$$

$$u_k = \bar{u}_k + ke_k, \tag{5.12}$$

$$e_k \cong \hat{x}_k - \bar{x}_k, \tag{5.13}$$

Where e_k is the tracking error and is bounded by an invariant set.
Similarly, for a nonlinear system, the same procedure can be applied [73, 45]:

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k\bar{u}_k + g(\hat{x}_k) + w_k, \tag{5.14}$$

$$e_{k+1} = A_ke_k + w_k + (g(\hat{x}_k) - \bar{g}(\bar{x}_k)), \tag{5.15}$$

$$\begin{aligned} e_{k+1} &= A_k e_k + w_k^t, \\ w^t &\in W, \end{aligned} \tag{5.16}$$

Where e_{k+1} is also bounded by a robust positively invariant set.

5.3 Trip Cost

For the purpose of enhancing fuel consumption of the vehicle, the appropriate term should be included in the control-oriented model. The trip cost formula is the same as centralized and decentralized approaches.

5.4 Robust Positive Invariant Set Calculation

To design a robust MPC controller, the DUs must be bounded and this bound can be found by robust positive invariant set.

Definition1 (RPI set): The set φ is robust positive invariant (RPI) set, if for a system with the equation $x_{k+1} = Ax_k + bu_k + w$, $x_{k+1} \in \varphi$ for all $x_k \in \varphi$ and all $w \in W$ [74].

In other words, if there exists a set φ in X which the initial state x_k is in φ, the solution will be in X for all disturbance sequence w and for all time steps.

The RPI set is equal to the maximum error that can be caused by DUs. In order to use this method, all of the sources of the uncertainties needed to be considered. As shown by the authors of [45], if the nonlinear term $(g(x_k))$ is Lipschitz continuous and $L(x)$ is Lipschitz constant over X, then:

$$W_g = \{\Xi \in \mathbf{R}^n | \|\Xi\|_\infty \leq L(X) max \|e\|_2\}, \tag{5.17}$$

Therefore, a bound on the error of the nonlinear term Ξ can be found.

Another main source of the uncertainty for connected and automated vehicles is delay time. The authors of [75] proved that the bound on disturbance caused by the delay time can be calculated by finding maximum state changes due to maximum delay time T_d as follows:

$$W_T = T_d B_d K_c (AX \oplus (BU \oplus (\vartheta_t A_t \oplus W))) \tag{5.18}$$

The final source of uncertainty which is taken into account in this paper is the acceleration of traffic vehicles. A bound can also be defined on the acceleration by knowing the maximum acceleration of traffic vehicle A_t:

$$W_a = \vartheta_t A_t \tag{5.19}$$

The Mikowski sum of all of the sources of uncertainty can be used to define the bounded additive disturbance set W_t:

$$W_t = W_T \oplus W_g \oplus W_a \tag{5.20}$$

Definition2: The minimal RPI (mRPI) set φ_∞ is the RPI set contained in every closed RPI set.

Therefore, by having the bounded additive disturbance and Minkowski sum, the mRPI (φ_∞) set for the tracking error can be determined as follows [76, 77]:

$$e_{k+1} = A_k e_k + w_k^t, \tag{5.21}$$

$$e_{k+1} \in \varphi_\infty = \oplus_{i=0}^{\infty} A_k^i W, \tag{5.22}$$

As a result, the problem can be rewritten as the following nonlinear tube-based MPC problem:

$$\begin{split} J_N(x_k, u) = \bar{x}^T_{k+n|k} P \bar{x}_{k+n|k} + \\ \sum_{i=0}^{N-1} \bar{x}^T_{k+i|k} Q \bar{x}_{k+i|k} + \bar{u}^T_{k+i|k} R \bar{u}_{k+i|k}, \end{split} \tag{5.23}$$

subject to:

$$\begin{aligned} \bar{x}_{k+1} &= A_k \bar{x}_k + B_k u_k + \bar{g}(\bar{x}_k), \\ u_k &= \bar{u}_k - k\bar{x}_k, \\ \bar{x}_{k+1} &\in X \ominus \varphi, \\ u_{k+1} &\in U \ominus \varphi, \end{aligned} \tag{5.24}$$

5.5 Optimal Control Problem Formulation

The cost function for robust tube-based decentralized controller is the same as decentralized approach as shown below:

$$
\begin{aligned}
PI = \sum_{i=0}^{N-1} U(i)^2 + R^T C_x(i) + w_1(v_h(i) - v_{ref}(i))^2 \\
+w_2 E_{cost} + w_3 \beta e^{\beta},
\end{aligned}
\tag{5.25}
$$

In the cost function equation (5.25), C_x presents the inequality constraints as a vector. For the case of robust NMPC approach, we have more constraints than before:

$$
\begin{aligned}
\beta &\leq \beta_{max}, \\
\beta_{min} &\leq \beta, \\
v_h &\leq v_{max}, \\
U &\leq U_{max}, \\
U_{min} &\leq U,
\end{aligned}
\tag{5.26}
$$

As a result, the new C_x vector can be defined as follows:

$$
C_x = \begin{bmatrix} \beta - \beta_{max} \\ \beta_{min} - \beta \\ v_h - v_{max} \\ U - U_{max} \\ U_{min} - U \end{bmatrix}
\tag{5.27}
$$

Remark1: The control-oriented model and optimal control formulas are based on the nominal state of the system. The nominal states are states excluding the effect of disturbances and uncertainties. The NMPC is responsible for keeping nominal states of the system within constraints.

Remark2: The constraints on the system are tighter than non-robust case to make sure that states of the system will remain inside set regardless of the amount of the applied bounded additive disturbance. Moreover, tighter constraints reserve a space for control input generated by the linear stabilizing controller.

Remark3: This research's main aim is not to find the best set of weighting parameters. Therefore, the parameters have been found by trial-and-error up to a point that the proposed controller satisfies performance requirements. Enhanced performance can be achieved by finding a better and more efficient way of tuning.

5.6 Fast Optimizer

To solve this NMPC problem in real-time for automotive applications, we need a fast solver. In this research, we used Newton-GMRES based solver called MPsee [57] which is a MATLAB-based mathematical program. To make this solver faster, when the constraint is violated it should be incorporated in the cost function with a penalty weight and the violated constraint around current time step has priority over violation in prediction horizon which will be dealt with in future time steps. This will help the solver to work faster. To define the problem in Newton-GMRES solver, constraints and field vector need to be specified:

$$F\{x,u\} = \frac{d}{dt}\{\bar{x}_h, \bar{v}_h, \bar{T}_w, \bar{x}_t, \bar{v}_t, \bar{a}_t\} = \begin{bmatrix} \bar{v}_h \\ \frac{\bar{T}_w}{m r_w \eta_p} - \frac{F_{res}}{m} \\ -\frac{\bar{T}_w}{\tau_a} + \frac{U-Ku}{\tau_a} \\ \bar{v}_t \\ \bar{a}_t \\ -\sigma.\bar{a}_t \end{bmatrix}, \tag{5.28}$$

$$C\{x,U\} = AC_x + W \tag{5.29}$$

5.7 Control Evaluation Results

In subsequent sections, we investigate different performance features of the proposed controller. First, the performance and robustness of the robust tube-based decentralized controller are tested and compared with decentralized non-robust controller and centralized controller. The next section is dedicated to economic improvements of the proposed controller. Last but not least, our hardware-in-the-loop (HIL) setup is explained and results are displayed to prove the controller is real-time for automotive applications. The high-fidelity model has been used to generate all of the simulation results and HIL results are taken from our dSPACE HIL setup experiment.

In this research, the origin is put on the merging zone. The notation of the results is the same as centralized and decentralized case. Therefore, the error between actual and safe inter-vehicular distance must be zero or positive before merging zone to merge with traffic safely.

5.7.1 Robust Constraint Handling

To verify the constraints handling of robust NMPC controller in the presence of bounded additive disturbance, we conducted the following test using the high-fidelity model of the baseline vehicle. In this test, the host vehicle enters the control zone 40 meters before the entrance of the merging zone while the traffic vehicle is 35 meters away from the merging zone. To take into account the presence of disturbances, we added different variable delay and noise to the received position and velocity signals of the traffic vehicle which fuses into the proposed controller. We utilized the method presented in the robust tube-based decentralized approach section to find disturbance set and new constraints for robust NMPC controller. As it is shown in Fig. 5.2(a), both the robust and non-robust decentralized controller and centralized controller can calculate priority w.r.t. the traffic vehicle and can provide a safe distance between vehicles required for merging before merging zone. In order to better assess and compare performance, we changed the simulation environment. In the new environment, the host vehicle is following the traffic vehicle while noises and disturbances affecting the system. According to Fig. 5.2(b) the non-robust controllers have more fluctuations and harsher spikes in their reference speed following because of the uncertainties in the system, however, the robust NMPC controller can better handle constraints. Fig. 5.2(c) compares position error of the controllers. The non-robust controllers cannot handle defined constraints and have a higher position error. On the other hand, since the effects of DU has been considered in the design of robust NMPC, it can handle constraints under uncertainty.

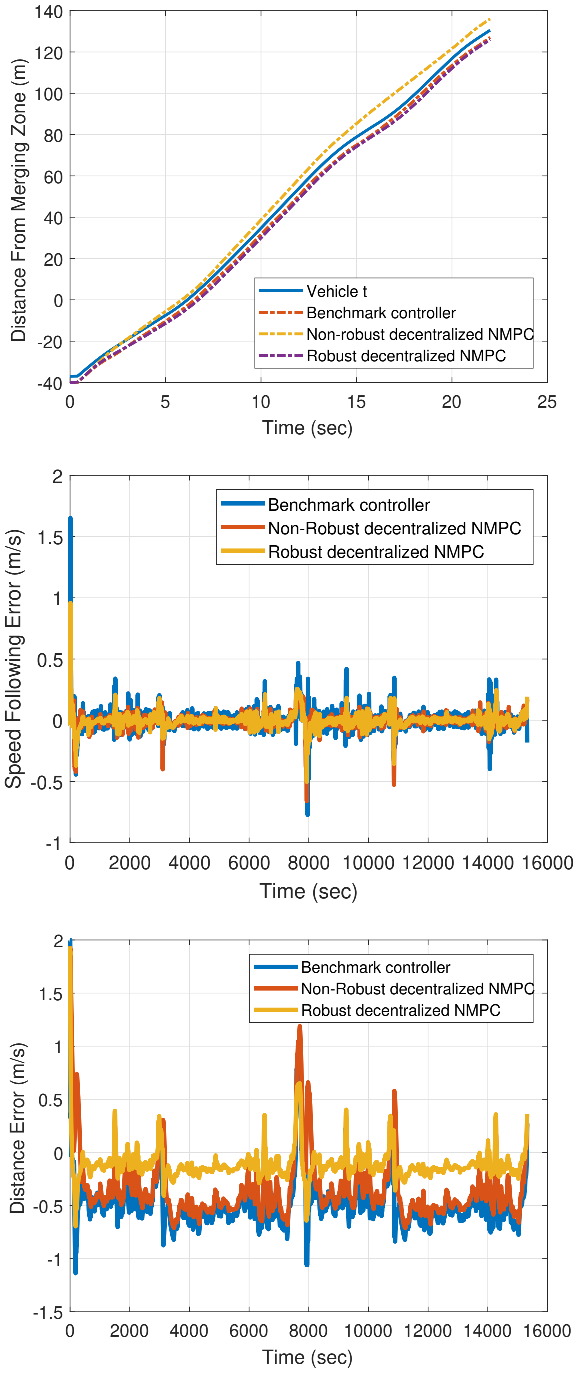


Figure 5.2: (a) Distance to merging zone comparison, (b) velocity, and (c) position error comparison between robust and non-robust controllers in 2×HWFET driving cycle

5.7.2 Fuel consumption comparison

One of the contributions is the consideration of fuel consumption. Fig. 5.3 compares energy cost for the non-robust NMPC and robust NMPC in 2×HWFET driving cycles. The non-robust NMPC at the end of the test has \$0.778 costs. In the case of non-robust controller, the disturbance affects constraints handling of the controller and the vehicle goes out of limit and tries to return inside the limit by harsh acceleration or braking which negatively influence energy economy of the vehicle. On the other hand, for the same scenario, the energy cost of the robust NMPC is \$0.743 which is about 4.7 % lower than the non-robust controller. This is due to the consideration of disturbances in the design of constraints in robust NMPC controller. Also, for the sake of comparison, Fig. 5.4 compares reference speed following, inter-vehicular distance error and fuel consumption of robust and non-robust decentralized controllers in 3×FTP driving cycles.

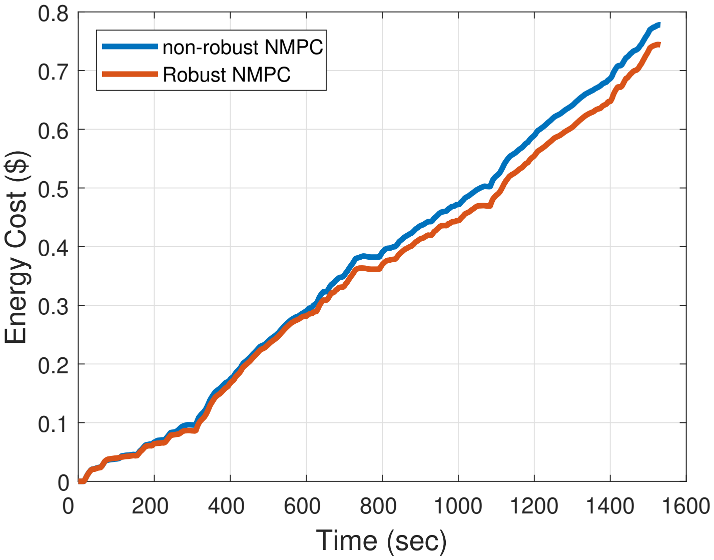


Figure 5.3: Energy cost comparison between robust and non-robust NMPC

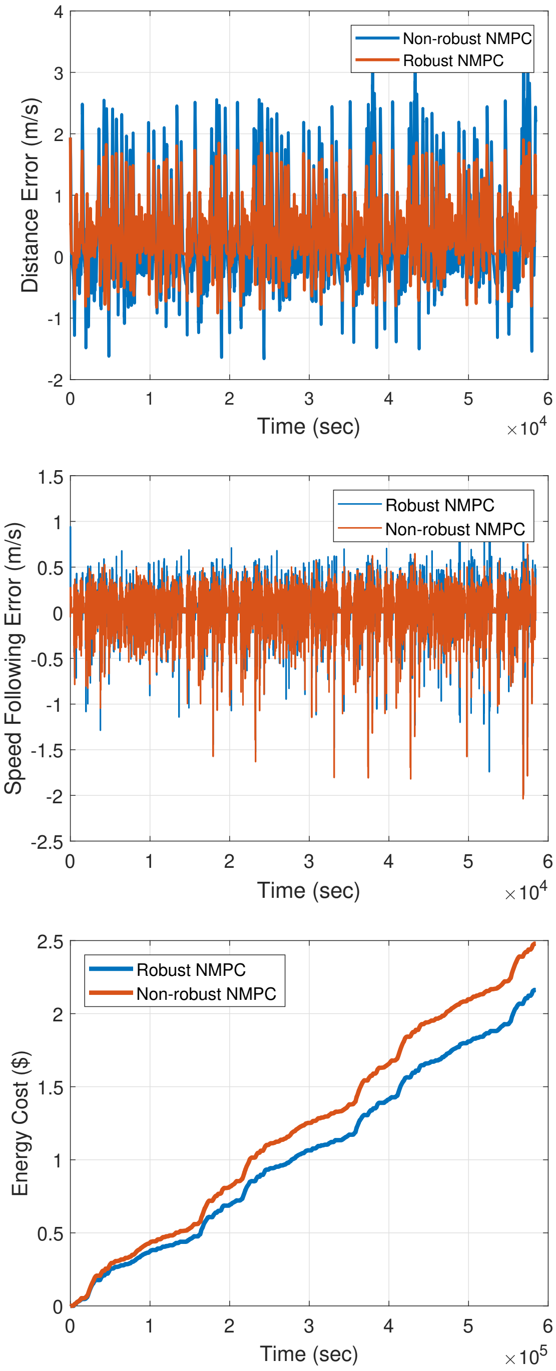


Figure 5.4: (a) position error, (b) velocity error, and (c) fuel consumption comparison between robust and non-robust decentralized controllers in 3×FTP driving cycle

5.7.3 Hardware-in-the-Loop (HIL) Experiment Results

There will be no concern about the computational demand of the controller if the turnaround time of the proposed controller is less than sampling time for the controller. The HIL result of the controller in priority following is shown in Fig. 5.6. The turn-around time is always between 500 μ seconds and 700 μ seconds for the prediction horizon length of N_p =10. The maximum inner and outer iterations are set to be less than 5. By comparison between Fig. 5.5 and Fig. 5.6, robust controller is computationally more expensive. However, as mentioned before the nonlinear solver of Newton-GMRES has been used in this research which results in the sampling time of less than 1 ms for the robust controller; thus, the controller is implementable on ordinary automotive ECUs without any concern about computational expenses.

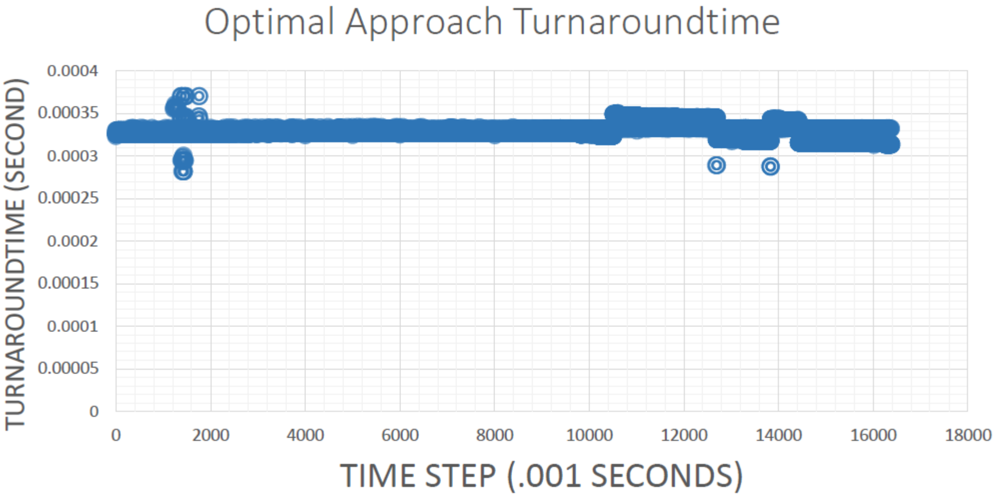


Figure 5.5: HIL experiment result of non-robust decentralized optimal control approach.

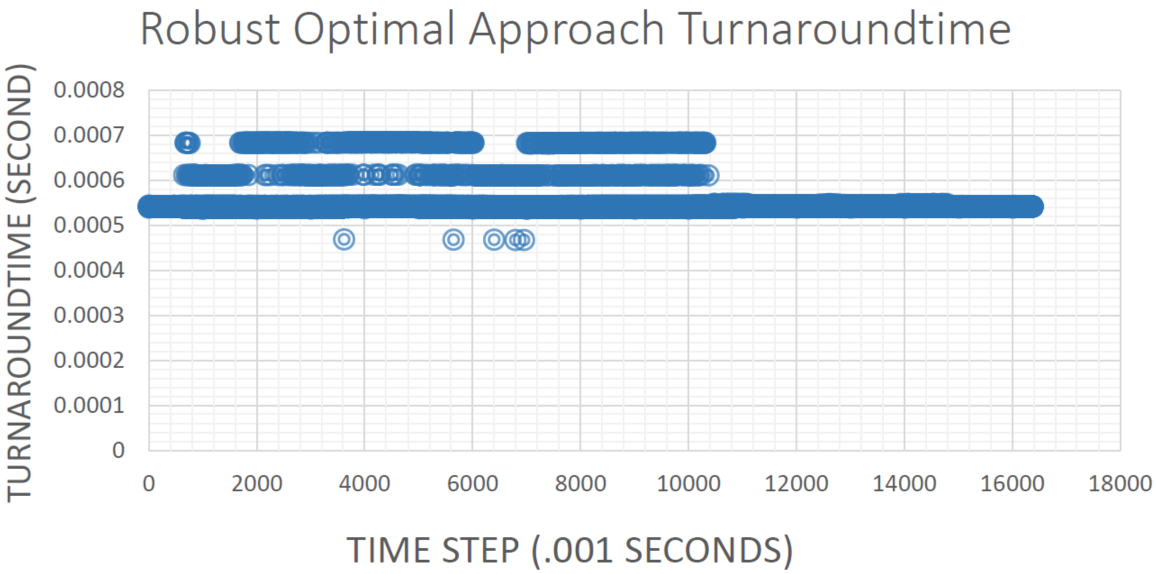


Figure 5.6: HIL experiment result of robust decentralized optimal control approach.

Table 5.1: Average turnaround time of robust decentralized optimal-control approach and decentralized optimal-control approach based on protoytpe ECU measurements and estimated Prius ECU turnaround time

Approach	Prototype ECU turnaround time	Estimated Prius turnaround time
Robust decentralized	0.0006	0.0048
non-robust decentralized	.00035	.0028

Chapter 6

Conclusions

In this thesis, we propose controller for connected and automated PHEVs (CA-PHEVs) since they can have better performance and wider access to the critical data. CA-PHEVs get information of other traffic participants through communication systems that enables coordination of vehicles at multi-vehicle situations like roundabouts. However, merging with traffic inside a roundabout is more complex because it is a multi-vehicle scenario and the chance of both lateral and longitudinal collision exits. Therefore, designing a controller for CA-PHEVs at roundabouts is a crucial problem.

In this thesis, first, we developed an NMPC controller for following assigned priorities in a centralized fashion. We considered the energy cost of the trip in our formulation for the complex powertrain of PHEVs. The results showed that the proposed controller outperforms the single task PID controller. Also, results of the Hardware-in-the-loop experiments was shown as a proof of real-time implementation of the NMPC controller for automotive applications.

Due to the assumption of having access to data of whole traffic participants in centralized methods, we proposed a decentralized controller which makes CA-PHEVs able to drive at the traffic circle in multi-vehicle situations. The performance was compared with PID in some scenarios using a high-fidelity PHEV model. It was shown that the NMPC controller can adjust the vehicle speed to increase the distance between vehicles while adhering to calculated priorities and the overall energy cost of the proposed controller becomes better by considering energy cost in performance index for the same CA-PHEV. We also introduced a novel priority calculation logic and its performance was compared with the navigation function approach. It was shown that the proposed logic can successfully calculate priorities while solving two multi-objective nonlinear optimal control problems.

Finally, in this thesis to compensate for the effect of DUs, we designed a robust tube-based NMPC controller for the decentralized coordination of vehicles at roundabouts. The results proved the robust NMPC has better performance in keeping the inter-vehicular distance error around zero and following the reference speed in the presence of DU compared to the non-robust NMPC. The results showed the addition of energy cost to the performance index plus robustness to disturbances improves the fuel consumption of the vehicle. The real-time execution of the proposed decentralized controllers was checked in the HIL test. It was demonstrated that the proposed controllers' turnaround time is below the safe threshold and are real-time for automotive applications.

6.1 Summary of Contributions

The following summarizes the main contributions of this thesis:

1. Development of an NMPC N/GMRES controller that adheres to assigned priorities in centralized coordination of PHEVs, minimizes fuel-consumption and follows reference speed with some deviations at roundabouts.

2. Proposing a decentralized coordination and control approach based on NMPC for CA-PHEVs:

   - Formulation of the problem in the form of an NMPC problem by considering more accurate vehicle longitudinal dynamics and nonlinear cost function.
   - The optimal problem formulated for the powertrain of PHEV with consideration of fuel economy.
   - Presentation of a navigation function approach and a novel optimal control approach for the calculation of priorities for the case of merging with traffic inside roundabouts.

3. Development of robust tube-based nonlinear optimal control problem to extend the decentralized optimal control priority calculation logic to more realistic case of control and coordination under uncertainties:

   - Consideration of different sources of uncertainties such as V2V and V2I communication delay, unknown traffic vehicle acceleration, and disturbance bound on non-linearities.

    - Calculation of robust positive invariant set for bounded additive disturbances and uncertainties acting on the system.
    - Compensate for uncertainties by tighter constraints and addition of linear stabilizing controller.

4. Investigation and comparison of the performance of all of the controllers through MIL simulations using the High-fidelity model of Toyota Prius.

5. Integration of controllers with the high-fidelity model of ASM Traffic Simulator to further examine the performance of the controllers at roundabouts and visualize their results.

6. Study real-time implementation of the controllers by hardware-in-the-loop experiments for automotive application.

6.2 Some Recommended Future Works

Some of the recommended ways to further expand this research are as follows:

- Find a better and more efficient way of tuning parameters to achieve a better performance

- Investigate performance of model-free control approaches like reinforcement learning to solve this complex problem and comparison between model-based and model-free approaches

- Extend decentralized coordination approach to consider multi vehicles at the same time

- Consideration of both lateral and longitudinal vehicle dynamics at the same time

- Consideration of pedestrians crossing roundabouts in our formulation

References

[1] Rasheed Hussain and Sherali Zeadally. Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials*, 21(2):1275–1313, 2018.

[2] Jeffery B Greenblatt and Susan Shaheen. Automated vehicles, on-demand mobility, and environmental impacts. *Current sustainable/renewable energy reports*, 2(3):74–81, 2015.

[3] Huei Peng. Connected and automated vehicles: The roles of dynamics and control. *Mechanical Engineering*, 138(12):S4, 2016.

[4] Kaijiang Yu, Junqi Yang, and Daisuke Yamaguchi. Model predictive control for hybrid vehicle ecological driving using traffic signal and road slope information. *Control theory and technology*, 13(1):17–28, 2015.

[5] Lars Grüne and Jürgen Pannek. Stability and suboptimality without stabilizing constraints. In *Nonlinear Model Predictive Control*, pages 113–163. Springer, 2011.

[6] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[7] Richard A Retting, Bhagwant N Persaud, Per E Garder, and Dominique Lord. Crash and injury reduction following installation of roundabouts in the united states. *American journal of public health*, 91(4):628, 2001.

[8] Joseph E Hummer, Joseph S Milazzo, Bastian Schroeder, and Katy Salamati. Potential for metering to help roundabouts manage peak period demands in the united states. *Transportation research record*, 2402(1):56–66, 2014.

[9] Xiaoguang Yang, Xiugang Li, and Kun Xue. A new traffic-signal control for modern roundabouts: method and application. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):282–287, 2004.

[10] Sadegh Tajeddin. Automatic code generation of real-time nonlinear model predictive control for plug-in hybrid electric vehicle intelligent cruise controllers. Master's thesis, University of Waterloo, 2016.

[11] Konstantinos I Kouramas, Christos Panos, Nuno P Faísca, and Efstratios N Pistikopoulos. An algorithm for robust explicit/multi-parametric model predictive control. *Automatica*, 49(2):381–389, 2013.

[12] Alessandro Alessio and Alberto Bemporad. A survey on explicit model predictive control. In *Nonlinear model predictive control*, pages 345–369. Springer, 2009.

[13] Jackeline Rios-Torres and Andreas A Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017.

[14] Hashim MN Al-Madani. Dynamic vehicular delay comparison between a police-controlled roundabout and a traffic signal. *Transportation Research Part A: Policy and Practice*, 37(8):681–688, 2003.

[15] Aimee Flannery, Lily Elefteriadou, Paul Koza, and John McFadden. Safety, delay, and capacity of single-lane roundabouts in the united states. *Transportation Research Record*, 1646(1):63–70, 1998.

[16] Aimee Flannery and Tapan Datta. Operational performance measures of american roundabouts. *Transportation Research Record*, 1572(1):68–75, 1997.

[17] David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015 urban mobility scorecard. 2015.

[18] Rahmi Akçelik. Roundabout metering signals: capacity, performance and timing. *Procedia-Social and Behavioral Sciences*, 16:686–696, 2011.

[19] Mariló Martín-Gasulla, Alfredo García, Ana Tsui Moreno, and Carlos Llorca. Capacity and operational improvements of metering roundabouts in spain. *Transportation research procedia*, 15:295–307, 2016.

[20] Kurt Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 530–537. IEEE Computer Society, 2004.

[21] Shan Huang, Adel W Sadek, and Yunjie Zhao. Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1201–1214, 2012.

[22] Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656, 2008.

[23] Tsz-Chiu Au and Peter Stone. Motion planning algorithms for autonomous intersection management. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[24] Arnaud de La Fortelle. Analysis of reservation algorithms for cooperative planning at intersections. In *13th International IEEE conference on intelligent transportation systems*, pages 445–449. IEEE, 2010.

[25] Kailong Zhang, Arnaud De La Fortelle, Dafang Zhang, and Xiao Wu. Analysis and modeled design of one state-driven autonomous passing-through algorithm for driverless vehicles at intersections. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 751–757. IEEE, 2013.

[26] Kailong Zhang, Dafang Zhang, Arnaud de La Fortelle, Xiao Wu, and Jean Gregoire. State-driven priority scheduling mechanisms for driverless vehicles approaching intersections. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2487–2500, 2015.

[27] Li Li and Fei-Yue Wang. Cooperative driving at blind crossings using intervehicle communication. *IEEE Transactions on Vehicular technology*, 55(6):1712–1724, 2006.

[28] Fei Yan, Mahjoub Dridi, and Abdellah El Moudni. Autonomous vehicle sequencing algorithm at isolated intersections. In *2009 12th International IEEE conference on intelligent transportation systems*, pages 1–6. IEEE, 2009.

[29] Ismail H Zohdy, Raj Kishore Kamalanathsharma, and Hesham Rakha. Intersection management for autonomous vehicles using icacc. In *2012 15th International IEEE conference on intelligent transportation systems*, pages 1109–1114. IEEE, 2012.

[30] Qiu Jin, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew Barth. Multi-agent intersection management for connected vehicles using an optimal scheduling approach. In *2012 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 185–190. IEEE, 2012.

[31] Jia Wu, Florent Perronnet, and Abdeljalil Abbas-Turki. Cooperative vehicle-actuator system: A sequence-based framework of cooperative intersections management. *IET Intelligent Transport Systems*, 8(4):352–360, 2013.

[32] Feng Zhu and Satish V Ukkusuri. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:363–378, 2015.

[33] Joyoung Lee and Byungkyu Park. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):81–90, 2012.

[34] Joyoung Lee, Byungkyu Brian Park, Kristin Malakorn, and Jaehyun Jason So. Sustainability assessments of cooperative vehicle intersection control at an urban corridor. *Transportation Research Part C: Emerging Technologies*, 32:193–206, 2013.

[35] Gabriel Rodrigues de Campos, Paolo Falcone, and Jonas Sjöberg. Autonomous cooperative driving: a velocity-based negotiation approach for intersection crossing. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1456–1461. IEEE, 2013.

[36] Md Abdus Samad Kamal, Jun-ichi Imura, Akira Ohata, Tomohisa Hayakawa, and Kazuyuki Aihara. Coordination of automated vehicles at a traffic-lightless intersection. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 922–927. IEEE, 2013.

[37] Md Abdus Samad Kamal, Jun-ichi Imura, Tomohisa Hayakawa, Akira Ohata, and Kazuyuki Aihara. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1136–1147, 2015.

[38] Xiangjun Qian, Jean Gregoire, Arnaud De La Fortelle, and Fabien Moutarde. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In *Control Conference (ECC), 2015 European*, pages 3452–3458. IEEE, 2015.

[39] Laleh Makarem and Denis Gillet. Fluent coordination of autonomous vehicles at intersections. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 2557–2562. IEEE, 2012.

[40] Laleh Makarem and Denis Gillet. Model predictive coordination of autonomous vehicles crossing intersections. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1799–1804. IEEE, 2013.

[41] Ezequiel Debada, Laleh Makarem, and Denis Gillet. Autonomous coordination of heterogeneous vehicles at roundabouts. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1489–1495. Ieee, 2016.

[42] Gabriel R Campos, Paolo Falcone, Henk Wymeersch, Robert Hult, and Jonas Sjöberg. Cooperative receding horizon conflict resolution at traffic intersections. In *53rd IEEE Conference on Decision and Control*, pages 2932–2937. IEEE, 2014.

[43] Graham C Goodwin, He Kong, Galina Mirzaeva, and María M Seron. Robust model predictive control: reflections and opportunities. *Journal of Control and Decision*, 1(2):115–148, 2014.

[44] Daniele Corona, Ion Necoara, Bart De Schutter, and Ton van den Boom. Robust hybrid mpc applied to the design of an adaptive cruise controller for a road vehicle. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1721–1726. IEEE, 2006.

[45] Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.

[46] Andrew Gray, Yiqi Gao, J Karl Hedrick, and Francesco Borrelli. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 208–213. IEEE, 2013.

[47] D Limon, I Alvarado, T Alamo, and EF Camacho. Robust tube-based mpc for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3):248–260, 2010.

[48] Bijan Sakhdari, Ebrahim Moradi Shahrivar, and Nasser L Azad. Robust tube-based mpc for automotive adaptive cruise control design. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[49] Maryyeh Chehresaz. Modeling and design optimization of plug-in hybrid electric vehicle powertrains. Master's thesis, University of Waterloo, 2013.

[50] Sanjaka G Wirasingha and Ali Emadi. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Transactions on vehicular technology*, 60(1):111–122, 2010.

[51] Pierluigi Pisu and Giorgio Rizzoni. A comparative study of supervisory control strategies for hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 15(3):506–518, 2007.

[52] Stephanie Stockar, Vincenzo Marano, Marcello Canova, Giorgio Rizzoni, and Lino Guzzella. Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles. *IEEE Transactions on Vehicular Technology*, 60(7):2949–2962, 2011.

[53] Nasser L Azad, Ahmad Mozaffari, Mahyar Vajedi, and Yasaman Masoudi. Chaos oscillator differential search combined with pontryagin's minimum principle for simultaneous power management and component sizing of phevs. *Optimization and Engineering*, 17(4):727–760, 2016.

[54] Qiuming Gong, Yaoyu Li, and Zhong-Ren Peng. Trip-based optimal power management of plug-in hybrid electric vehicles. *IEEE Transactions on vehicular technology*, 57(6):3393–3401, 2008.

[55] Md Abdus Samad Kamal, Masakazu Mukai, Junichi Murata, and Taketoshi Kawabe. Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Transactions on control systems technology*, 21(3):831–841, 2013.

[56] Mike Huang, Hayato Nakada, Ken Butts, and Ilya Kolmanovsky. Nonlinear model predictive control of a diesel engine air path: A comparison of constraint handling and computational strategies. *IFAC-PapersOnLine*, 48(23):372–379, 2015.

[57] Sadegh Tajeddin and Nasser L Azad. Ecological cruise control of a plug-in hybrid electric vehicle: A comparison of different gmres-based nonlinear model predictive controls. In *American Control Conference (ACC), 2017*, pages 3607–3612. IEEE, 2017.

[58] Mahyar Vajedi and Nasser L Azad. Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):113–122, 2016.

[59] Mahyar Vajedi. Real-time optimal control of a plug-in hybrid electric vehicle using trip information. 2016.

[60] Amir Taghavipour, Ramin Masoudi, Nasser L Azad, and John McPhee. High-fidelity modeling of a power-split plug-in hybrid electric powertrain for control performance evaluation. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V001T01A008–V001T01A008. American Society of Mechanical Engineers, 2013.

[61] Eugene L Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 2012.

[62] dSPACE. ASM MicroAutoBox II. https://www.dspace.com/en/inc/home/products/hw/micautob/microautobox2.cfm.

[63] dSPACE. ASM ControlDesk. https://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/controldesk.cfm.

[64] dSPACE. ASM Products. https://www.dspace.com/en/inc/home/products/products.cfm#.

[65] dSPACE. ASM Vehicle Dynamics. https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/vehicle_dynamics_models.cfm.

[66] dSPACE. ASM ModelDesk. https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/modeldesk.cfm.

[67] dSPACE. ASM MotionDesk. https://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/modesk.cfm.

[68] Sina Alighanbari and Nasser L Azad. Ecological nmpc controller for connected and automated plug-in hybrid electric vehicles at roundabouts. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1069–1074. IEEE, 2019.

[69] Toshiyuki Ohtsuka. A continuation/gmres method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.

[70] Carl T Kelley. *Iterative methods for optimization*. SIAM, 1999.

[71] Luigi Chisci, J Anthony Rossiter, and Giovanni Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37(7):1019–1028, 2001.

[72] DQ Mayne and W Langson. Robustifying model predictive control of constrained linear systems. *Electronics Letters*, 37(23):1422–1423, 2001.

[73] David Q Mayne, Erric C Kerrigan, EJ Van Wyk, and P Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.

[74] SV Raković, Eric C Kerrigan, David Q Mayne, and Konstantinos I Kouramas. Optimized robust control invariance for linear discrete-time systems: Theoretical foundations. *Automatica*, 43(5):831–841, 2007.

[75] Bijan Sakhdari and Nasser L Azad. Adaptive tube-based nonlinear mpc for economic autonomous cruise control of plug-in hybrid electric vehicles. *IEEE Transactions on Vehicular Technology*, 67(12):11390–11401, 2018.

[76] Sasa V Rakovic, Eric C Kerrigan, Konstantinos I Kouramas, and David Q Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[77] Ilya Kolmanovsky and Elmer G Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical problems in engineering*, 4(4):317–367, 1998.