ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN AND IMPLEMENTATION OF A TORQUE-BASED PREDICTIVE STEERING ASSISTANCE FOR HUMAN-CENTERED AND SAFE AUTOMATED DRIVING

Ph.D. THESIS

Ziya ERCAN

Department of Control and Automation Engineering

Control and Automation Engineering Programme

NOVEMBER 2017

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN AND IMPLEMENTATION OF A TORQUE-BASED PREDICTIVE STEERING ASSISTANCE FOR HUMAN-CENTERED AND SAFE AUTOMATED DRIVING

Ph.D. THESIS

Ziya ERCAN
(504112106)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Metin GÖKAŞAN
Co-advisor: Prof. Dr. Francesco BORRELLI

NOVEMBER 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNSAN-MERKEZLİ VE GÜVENLİ OTOMATİK SÜRÜŞ İÇİN TORK TABANLI ÖNGÖRÜMLÜ DİREKSİYON YARDIMCI SİSTEMİNİN TASARIMI VE GERÇEKLENMESİ

DOKTORA TEZİ

Ziya ERCAN
(504112106)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Metin GÖKAŞAN
Eş Danışman: Prof. Dr. Francesco BORRELLI

KASIM 2017

Ziya ERCAN, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 504112106 successfully defended the thesis entitled "DESIGN AND IMPLEMENTATION OF A TORQUE-BASED PREDICTIVE STEERING ASSISTANCE FOR HUMAN-CENTERED AND SAFE AUTOMATED DRIVING", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Metin GÖKAŞAN** Istanbul Technical University	.....
Co-advisor :	**Prof. Dr. Francesco BORRELLI** University of California, Berkeley	.....
Jury Members :	**Prof. Dr. Hakan TEMELTAŞ** Istanbul Technical University	.....
	Prof. Dr. Ata MUĞAN Istanbul Technical University	.....
	Prof. Dr. Ovsanna Seta ESTRADA Istanbul Technical University	.....
	Assoc. Prof. Dr. Tankut ACARMAN Galatasaray University	.....
	Asst. Prof. Dr. İlker ÜSTOĞLU Yıldız Technical University	.....

Date of Submission : **12 October 2017**
Date of Defense : **30 November 2017**

To my beloved family,

FOREWORD

I would like to express my sincere gratitude to my advisor Prof. Metin Gökaşan for the continuous support of my Ph.D study and research. He encouraged me to pursue my studies abroad and provided all means of help when I needed. I appreciate his effort on me while studying in Istanbul Technical University.

This dissertation would not have been possible without my co-advisor Prof. Francesco Borrelli who made my dream come true by accepting me as a visiting student in his laboratory. He supported my research all the time and provided me an amazing opportunity to witness and experience the research environment in UC, Berkeley.

The last but not least, I would like to thank professors Hakan Temeltaş and İlker Üstoğlu for being on my steering committee from the beginning and providing valuable feedback on this thesis. I would also like to acknowledge the support of Scientific and Technological Research Council of Turkey (TUBITAK) for granting a scholarship to me in order to conduct my research in UC, Berkeley.

This dissertation is flourished in the amazing city of Berkeley, CA with beautiful minds around me. I am grateful to the many members of Model Predictive Control lab. which is located in the sunlight-less halls of Etcheverry Hall basement. Since the first day I arrived, it is been a priceless experience and one of the best years of my life. This dissertation is partially outcome of an amazing collaboration with Ashwin Carvalho whom I learned a lot. We had amazing California Proving Ground trips and countless days spent by testing in Richmond Field Station. Fueled with coffee, I had my most efficient days in research while having a blast with my colleagues Ugo Rosolia and Jacopo Guanetti. My special gratitudes to Mathias Soppert and Andrea Reti for their priceless friendship and encouraging me to start playing basketball again. I would like to thank to Dr. Georg Schildbach and Dr. Stephanie Lefevre for their guidance and important feedback. I would also like to thank the Hyundai Motor Company for providing us the test vehicle in order to implement our control algorithms and all the visiting engineers for their friendship and help.

I would like to express my deep appreciation to my parents Zeynep&Ali Ercan and my brother Hasan Ercan who have been an incredible source of support and motivation through all my life. I could not bear this burden without their unconditional love and support. Finally, I would like to dedicate this effort to love of my life Özlem Bülbül whom I have started a new chapter of my life. She is always there for me and she will always be. Her support gave me a great motivation to carry on during though times.

November 2017 Ziya ERCAN

TABLE OF CONTENTS

Page

ABBREVIATIONS

ABS : Anti-lock Braking System
ACC : Adaptive Cruise Control
ADAS : Advanced Driver Assistance System
AEB : Autonomous Emergency Braking
AS : Automatic Steering
BSI : Blind Spot Intervention
CFTOC : Constrained Finite Time Optimal Control
COG : Center of Gravity
CPG : California Proving Grounds
DSAS : Driver Steering Assistance System
DVE : Driver-Vehicle-Environment
EFRA : Exponential Forgetting and Reseting Algorithm
EOM : Equations of Motion
EPAS : Electric Power Assisted Steering
ESC : Electronic Stability Control
EV : Ego Vehicle
FCWS : Forward Collision Warning System
HMI : Human-Machine Interface
HMM : Hidden-Markov Model
LDP : Lane Departure Prevention
LDWS : Lane Departure Warning System
LKAS : Lane Keeping Assistance System
LP : Linear Programming
LOA : Level of Automation
LOHA : Level of Haptic Authority
LQR : Linear Quadratic Regulator
MDPS : Motor Driven Power Steering
MPC : Model Predictive Control
NMPC : Nonlinear Model Predictive Control
NMS : Neuromuscular System
NP : Nonlinear Programming
PID : Proportional Integral Derivative
QP : Quadratic Programming
RLS : Recursive Least Squares
SVM : Support Vector Machines
TLC : Time-to-Lane Crossing
TTC : Time-to-Collision
TV : Target Vehicle
UCB : University of California, Berkeley
WHO : World Health Organization

LIST OF TABLES

Page

LIST OF FIGURES

Page

DESIGN AND IMPLEMENTATION OF A TORQUE-BASED PREDICTIVE STEERING ASSISTANCE FOR HUMAN-CENTERED AND SAFE AUTOMATED DRIVING

SUMMARY

Road traffic deaths continue to be a major global public-safety and health problem. According to the report of World Health Organization (WHO), globally more than 1.24 million people die every year due to road traffic crashes on the other hand between 20 to 50 million people suffer non-fatal injuries and disabilities.

The automotive industry and researchers have focused on developing the vehicle safety systems in order to mitigate the severity of injuries and even to avoid the collisions either by issuing warning signals or intervening the vehicle dynamics. With the advances in sensor, actuator and computing technologies, the automation is introduced in vehicle control through systems such as Anti-lock Braking System (ABS) and Electronic Stability Control (ESC) which are proven to increase vehicle safety and efficiency.

Although the human errors are the ultimate cause of the collisions, the humans are reluctant to give a complete authority to an automated driving technology because of the reliability issues. In order to support the drivers in such demanding tasks, the advanced driver assistance systems (ADAS) are designed to provide various functionalities while keeping the driver in the center of authority. One particular commercialized implementation of ADAS is the lateral vehicular safety systems such as lane departure prevention (LDP), lane keeping assistance system (LKAS) and blind spot intervention (BSI). The human-in-the-loop nature of these systems makes the design problem non-trivial and challenging since the influence of automation on the driver should be considered in terms of human factors thoroughly. There is a vast of human factor studies investigating the conflicts arising in human-machine systems and one approach has received considerable attention in academic research to fulfill the desired design guidelines.

Haptic shared control concept is build on the idea of two systems (e.g., human operator and automation) acting together on the manual control interface and performing the same task cooperatively. This framework aims to provide a bi-directional information flow between both systems while increasing the efficiency and reducing the workload of human operator. One of the most popular implementation in automotive research is the haptic guidance system which shares the steering control in order to support driver in lane keeping/changing and obstacle avoidance maneuvers. These systems communicate with the driver through feedback forces and so the demanding visual workload of the driver is relieved by utilizing the under-used haptic sensory channel. The various studies show that these systems are effective under routine driving tasks when the goals of both systems meet but there is still some topics which should be investigated in order to analyze the effectiveness of these systems during emergency maneuvers and/or conflict situations.

This thesis investigates the open issues in the literature of shared steering control problem by proposing an assistance system design based on the model predictive control (MPC) approach. MPC is a model-based optimal control routine which employs receding horizon technique. In this approach, an optimal input sequence is calculated by minimizing a cost function which depends on the future prediction of state trajectories that are calculated by using a mathematical model of the plant. This constrained optimal control problem is solved at every time step for a finite time horizon. Only the first input of the optimal sequence is applied to the plant and the optimization routine is solved at next time step. The design of the predictive controller could be separated into four elements: A prediction model, a user-defined objective function, the constraints and tuning of the associated weights and prediction horizon. We emphasis the trade-off between the control performance and the complexity of the optimization problem with respect to the prediction model that is used. Using a relatively simple but yet accurate model could decrease the complexity while not deteriorating the control performance.

In this thesis, we design and implement a driver steering assistance system (DSAS) which utilizes the torque-based steering actuator (i.e., a torque overlay is superimposed instead of angular position) of the vehicle in order to generate a torque guidance for safe and assisted automated driving. Most of the previous studies design the guidance forces through the combination of a preview controller, which calculates the optimal steering angle, and a proportional feedback loop. The tuning of these forces are heuristic and it may lead to conflicts and increase in physical workload even in routine driving tasks. The studies show the affects of neuromuscular system (NMS) properties of the driver's arms while interacting with the guidance forces. The adaptability of these properties is a challenging issue in designing such systems.

In order to over come these issues, we consider the NMS properties of the driver's arms coupled with the steering system and we propose a novel driver-in-the-loop steering model whose parameters are identified online by using a recursive least squares (RLS) scheme. This allows the controller to adapt the dynamic changes in the human-machine system and update the parameters of the prediction model in order to optimize a more suitable guidance force. We also design the objective function such that there is a trade-off between amount of utilized guidance forces and amount of deviation from the optimal trajectory. If the driver's and system's goals are matched (i.e., no conflict), the controller utilizes a suitable amount of guidance with respect to NMS properties. However when the goals do not match (i.e., conflict) as the controller increases the guidance torque and the driver opposes it by increasing the impedance, the controller decreases the guidance torque gradually since the input cost dominates the tracking cost in the objective function. On the other hand, if the driver's actions are most likely to cause a safety issue the system intervenes at all cost whether the driver accepts or opposes.

We first test the proposed assistance system through closed-loop simulations in lane departure and blind spot intervention scenarios. The results show that when the NMS properties are known correctly in the prediction model, the control performance increases with minimum intrusion and control effort. Next, the system is validated in a test vehicle which is instrumented with a production level electric power assisted steering (EPAS) system which is called motor driven power steering (MDPS) system. We performed the experiments with human drivers in lane departure scenarios. The

results show the effectiveness of the proposed system in adapting with respect to driver's response. The haptic guidance is generated according to how the driver interacts with the system's activity. A better control performance is achieved by updating the prediction model in the controller.

We also conducted an experimental study with five participants to validate the performance. The results show that the assistance system adapts the guidance torque systematically with respect to the interaction level. This approach could resolve conflicts between the driver and the controller during routine driving tasks when there exists disparity in control objectives.

İNSAN-MERKEZLİ VE GÜVENLİ OTOMATİK SÜRÜŞ İÇİN TORK TABANLI ÖNGÖRÜMLÜ DİREKSİYON YARDIMCI SİSTEMİNİN TASARIMI VE GERÇEKLENMESİ

ÖZET

Trafik kazaları kamu sağlığını ve huzurunu tehdit eden başlıca sorunlardan biri olmaya devam etmektedir. Dünya Sağlık Örgütü'nün yayınladığı raporlara göre trafik kazaları yüzünden küresel olarak her yıl 1.24 milyon insan hayatını kaybetmekte ve de 20 ila 50 milyon insan ölümcül olmayan yaralanmalara ve sakatlıklara maruz kalmaktadırlar.

Otomotiv endüstrisi ve araştırmacılar kazaların bu yıkıcı etkilerini azaltmak amacıyla uyarıcı işaretler üreten ya da aracın kontrolüne müdahale eden araç güvenlik sistemlerinin geliştirilmesi için çalışmalara odaklanmışlardır. Algılayıcılar, eyleyiciler ve de bilgisayar sistemlerindeki teknolojik gelişmeler sayesinde otomasyon sistemleri araç kontrolünde önemli bir rol oynamaya başlamıştır. Kilitlenme önleyici frenleme ve elektronik kararlılık denetimi gibi araç kontrol sistemleri aracın yol tutuş performansını ve dolayısıyla yol güvenliğini arttırdığı kanıtlanmıştır.

Kazaların nihai sebebi olarak insan hataları ilk sırada olmasına rağmen güvenilirlik meselelerinden dolayı insanlar otonom araçlara bütün kontrol yetkisini vermeğe gönüllü değillerdir. Araba sürmek gibi görsel ve bilişsel yönden külfetli olan bir eylemde sürücülere destek olmak için tasarlanan sürücü yardımcı sistemleri, kontrol otoritesini sürücüde tutarak sürücüyü bir çok yönden işlevsel olarak desteklemektedirler. Şeritten çıkmayı engelleyici sistem, şerit takip yardımı sistemi ve de kör nokta müdahale sistemi gibi yanal araç güvenlik sistemleri bu sistemlerin ticari olarak uygulanışına önemli örneklerden birkaçıdır. İnsanın da araç sistemin bir parçası olması sebebiyle bu tarz kontrol sistemlerinin tasarımı zorlaştırmakta ve ilgi çekici bir hale gelmektedir. Otomasyonun insan üzerindeki etkilerininin eksiksiz bir şekilde dikkate alınması, bu tarz insan-makina sistemlerinin tasarımı açısından büyük önem taşımaktadır. İnsan-makina sistemlerinde ortaya çıkan anlaşmazlıkları inceleyen birçok insan faktörleri çalışmaları vardır ancak istenilen tasarım ilkelerine uygun olan bir yaklaşım azımsanmayacak kadar bir ilgi görmüştür.

Temassal paylaşımlı kontrol yaklaşımı iki farklı sistemin (örneğin insan operatörü ve otomasyon sistemi) aynı fiziksel kontrol arayüzünü kullanarak beraber bir işi yapabilecek şekilde bu arayüzü paylaşmaları fikri üzerine inşaa edilmiştir. Bu sayede iki sistem arasında çift yönlü bir bilgi alışverişi kurularak kontrol verimliliğinin artması ve de operatörün iş yükünün azaltılması hedeflenmektedir. Temassal yönlendirme sistemleri bu çerçevede en ilgi gören otomotiv uygulama alanı olmuştur. Bu sistemler aracın direksiyon hakimiyetini insan operatörü ile paylaşarak şerit takibi, şerit değiştirmesi ve de engelden kaçma gibi manevralarda insanlara yardımcı olacak şekilde tasarlanmaktadırlar. Temassal yönlendirme sistemi, insan operatörü ile tensel kuvvet geri beslemesi yoluyla haberleşir ve bu sayede sürücünün görsel iş yükü hafifletilmektedir. Yapılan birçok çalışmada insan ve otomasyon arasındaki anlaşmazlığın olmadığı durumlarda bu sistemlerin çok yararlı olduğu ancak acil

manevra ve de anlaşmazlık durumlarında bu sistemlerin sürücülerle olan etkileşiminin etkilerinin incelenmesi gereklidir.

Bu tez kapsamında paylaşımlı direksiyon kontrolü konusunun literatürdeki açık problemleri incelenerek, bu problemleri çözmeye yönelik model öngörümlü kontrol yöntemini kullanan insan-makina sisteminin tasarımı önerilmiştir. Model öngörümlü kontrol yöntemi basık ufuk kontrol tekniğini kullanan model tabanlı bir optimal kontrol stratejisidir. Bu kontrol yönteminde, sistemin matematiksel bir modeli kullanılarak durum değişkenlerinin olası yörüngeleri kontrol giriş dizisinin fonksiyonu olarak ifade edilir. Nümerik optimizasyon yöntemleri kullanılarak tanımlanmış bir amaç fonksiyonunu minimize eden optimal giriş işareti dizisi hesaplanır. Bu kısıtlı optimizasyon problemi her ayrık zaman adımında sonlu bir zaman için çözülür. Hesaplanan optimal giriş dizisinin sadece ilk elemanı sisteme uygulanır ve optimizasyon rutini bir sonraki örnekleme periyodunda tekrar çözülür. Öngörülü kontrolcünün tasarımı dört farklı unsura ayrılabilir. Bunlar, öngörüm modeli, tasarlanmış bir amaç fonksiyonu, sistem kısıtları ve de ilgili ağırlık katsayıların ve öngörüm ufkunun belirlenmesidir. Kullanılan öngörüm modeline bağlı olarak kontrol performansı ve optimizasyon probleminin karmaşıklığı arasında bir değiş tokuş vardır. Basit ama yeterli bir doğrulukta model kullanmak optimizasyon probleminin karmaşıklığını azaltıp kontrol performansının bozulmamasını sağlayacaktır.

Bu tezde insan merkezli ve güvenli otomatik sürüş için sürücüyü temassal kuvvetler ile yönlendiren bir sürücü direksiyon yardımcı sisteminin tasarlanması ve gerçeklenmesi konusu işlenmiştir. Önerilen bu sistem eyleyici olarak talep edilen bir tork değerini üretebilen bir direksiyon mekanizması kullanmaktadır. Bu eyleyici sayesinde istenilen geri besleme kuvveti üretilip direksiyona uygulanmaktadır. Geçmişte yapılan çalışmaların birçoğunda yönlendirici kuvvetlerin hesaplanması için optimal direksiyon açısını hesaplayan bir öngörüm tabanlı kontrolcü ve orantısal bir geribesleme yöntemi kullanılmıştır. Bu geribesleme kuvvetlerinin ayarlanması genelde sezgisel yöntemlerle yapıldığı için sürücü ve sistem arasında anlaşmazlıklara yol açarak rutin sürüş görevlerinde dahil fiziksel iş yükünün artmasına sebebiyet vermektedir. Bu sebeple insanın sinir ve kaslara (nöromüsküler) ait özelliklerinin yönlendirici kuvvetlerle olan etkileşimi bazı çalışmalarda incelenmiştir. İnsanlar bir anlaşmazlık durumunda kol kaslarını kasarak yönlendirici kuvvetlerin tersi yönünde kuvvet üretebilmekte ya da bu kuvvetlere yol vermek için kol kaslarını gevşetebilmektedir. Ortaya çıkan bu değişken ve uyarlanabilir etkileşimi modelleyerek kontrol tasarımının içerisinde kullanmak, kontrol performansını ve de insan-makina etkileşiminin verimliliğini arttırmaktadır.

Bu sunulan çalışmada, insan kolunun nöromüsküler özellikleri ile direksiyon sisteminin dinamik özelliklerini birleştirerek sürücü-direksiyon bütünleşmiş sistemi modelini önerilmiştir ve bu modelin değişken katsayıları özçağrılı en küçük kareler yöntemi ile çevrimiçi hesaplanmıştır. Bu sayede önerilen kontrolcü insan-makina sisteminin zamana bağlı değişken özelliklerini öngörüm modelinin parametrelerini yenileyerek takip edebilmiş ve bunun sonucunda etkileşim seviyesine uygun bir yönlendirici kuvvet optimize etmesi sağlanmıştır. Tasarlanan masraf fonksiyonu içerisindeki amaç kriterleri kullanılacak yönlendirme kuvvetinin boyutu ve arzu edilen güzergahtan sapma miktarı arasında ödünleşim olacak şekilde tasarlanmıştır. Eğer sürücü ve sistemin hedefleri uyuşması durumlarında, yani herhangi bir anlaşmazlık olmaması durumu, kontrol sistemi insanın nöromüsküler özelliklerine uyacak şekilde uygun miktarda kuvvet uygulayacaktır. Ancak hedeflerin uyuşmaması durumunda

kontrolcü ilk olarak sürücüyü yönlendirecek miktarda kuvvet uygular ve sürücünün bu kuvvete karşı koymasıyla yavaş yavaş kuvvet miktarını azaltır. Bunun sebebi ise amaç fonksiyonunda giriş işareti masrafının güzergah takip etme masrafından daha büyük olmaya başlamasıdır. Diğer bir yandan, eğer sürücünün davranışları güvenlik zaafiyetine yol açacak şekilde ise sistem ne pahasına olursa olsun, yani sürücü kabul etsin ya da etmesin, araç dinamiğine müdahale edecektir.

Önerilen yardımcı sistemi ilk olarak kapalı çevrim simülasyonlarda şeritten ayrılma ve kör nokta müdahalesi gibi senaryolarda test ettik. Sonuçlar nöromüsküler özelliklerin öngörüm modelinde doğru ifade edilmesinin kontrol performansını arttırdığını göstermiştir. Daha sonra, bu tasarlanan sürücü yardımcı sistemi elektrik gücü yardımcılı direksiyon sistemine sahip olan bir deneysel araçta test edilerek onaylanmıştır. Şeritten ayrılma ve kör nokta müdahale gibi senaryolar ele alınarak insan sürücüler ile birlikte çeşitli deneyler yapılmıştır. Sonuçlar önerilen sistemin farklı sürücü tepkilerine göre uyum sağlamaktaki başarısını göstermektedir. Yönlendirici kuvvetler sürücünün sistemle olan etkileşimine bağlı olarak hesaplanmıştır. Daha başarılı bir kontrol performansı öngörüm modelini yenileyerek elde edilmiştir.

Bu deneysel çalışmalara ek olarak beş katılımcının yer aldığı bir deneysel çalışma yaptık. Bu çalışmanın amacı farklı bireylerin farklı senaryolarda sistem ile olan etkileşimini gözlemlemek ve de sistemin başarısını onaylayacak sonuçlar almaktır. Deney sonuçlarında göre önerilen sistem yönlendirici kuvvetleri sistematik olarak etkileşim seviyesine göre uyumlu bir şekilde uygulamaktadır. Bu yöntem kontrol hedeflerindeki farklılıklardan dolayı oluşan anlaşmazlık anlarının pürüzsüz bir şekilde çözülmesini sağlayabilir.

1. INTRODUCTION

1.1 Motivation

Road traffic deaths continue to be a major global public-safety and health problem. According to the report of World Health Organization (WHO), globally more than 1.24 million people die every year due to road traffic crashes on the other hand between 20 to 50 million people suffer non-fatal injuries and disabilities. These numbers remain unacceptably high and without new initiatives and precautions the total number of road traffic deaths and injuries is predicted to increase by 65% between 2000 and 2020. Around 1.9 million people would be affected each year by 2020 [6]. The automotive industry and researchers have focused on developing the vehicle safety systems in order to mitigate the severity of injuries and even to avoid the collisions by either issuing warning signals or intervening the vehicle dynamics. Earlier efforts on vehicular safety has focused on mitigating the severity of the collisions by using passive safety features such as seat belts, air bags, and laminated windshields. With the developments in technology, automation (i.e., assigning partial/full authority to machines in order to perform a task) is introduced in vehicle control through systems such as Anti-lock Braking System (ABS) and it is widely accepted. Therefore, the focus on road safety have been shifted from mitigating the collisions to preventing them by developing active safety systems.

One of the most successful application is Electronic Stability Control (ESC) system which marked a milestone in driving safety as they are commercialized and become a standard in vehicular technology. ESC system consists of an embedded control algorithm which uses vehicle's sensors and electronically controlled actuators (e.g., ABS) to maintain the stability and control of the vehicle in hazardous situations (e.g., preventing skidding in cases of understeer or oversteer) by applying the brakes on one or more wheels automatically. It is estimated that one-third of fatal crushes could be prevented by using this technology and by 2014 most of the European countries

approved ESC as a mandatory requirement for all new vehicles. Although these active safety features improve the handling of the vehicle in though situations, they are not capable of preventing any means of collision between vehicles which occur as a result of degradation in human driver's performance.

It is estimated that 90% of the road traffic crashes happen due to human errors while driving. In [7], it is proposed that the human errors are potentially induced by couple of factors such as :

1. **Distraction** : a flaw during perception stage (e.g., using mobile phones while driving).
2. **Poor Judgment** : a flaw in information processing during diagnostic stage (e.g., misinterpreting the distances and velocities between vehicles).
3. **Lack of Situational Awareness** : a flaw in information processing during the assessment stage (e.g., fail to anticipate the evolution of the situation).
4. **Operating Errors** : a flaw in motor functions to execute the necessary action (e.g., fail to apply enough breaking and/or steering).

While the human are still reluctant to give up the driving task to the automation, using automated features for the driving tasks is pointed out to be beneficial in improving the road safety by helping the humans in operating the vehicle. The idea of autonomous vehicles/self-driving cars emerge as an ultimate solution to prevent the collisions that are caused by human errors. These vehicles use a high level of automation to operate all aspects of driving task with respect to all traffic conditions (e.g., highway and urban). All the driving tasks are allocated to the automation and it has the absolute authority during driving.

Although there is an intense research effort on self-driving cars, the reliable operation of autonomous driving in unpredictable traffic environment still continues to be a non-trivial and challenging problem. Moreover, unsolved legal issues rise questions about the responsibility in case of a collision. Therefore, rigorous testing and validation is still needed before any means of mass deployment.

Advanced Driver Assistance Systems (ADAS) are emerged in the automotive industry for improving the road safety by supporting the drivers in hazardous situations. With

the advances in sensor, actuator and computing technologies, ADAS are claimed to be more feasible than the autonomous driving in practice and various safety features have been developed and commercialized recently. While some of the assistance systems are designed to introduce various levels of automation in driving tasks, some are developed to enhance the perception and awareness of the driver. ADAS could be categorized with respect to functionality as presented in [8].

1. **Perception and Informative Systems :** Enhancing the sensing capabilities of the driver and providing necessary information about the vehicle, its surroundings and even about the driver (e.g., adaptive light control, automotive night vision, smart heads up displays, blind spot monitor, driver monitor and etc.)

2. **Collaborative (Support) Systems :** Mitigating or avoiding the collisions by issuing a warning signal (e.g., lane departure warning system (LDWS), forward collision warning system (FCWS), and etc.) or intervening the driver/vehicle dynamics (e.g., autonomous emergency braking (AEB), lane keeping assistance system (LKAS) and etc.)

3. **Delegation (Automation) Systems :** Reducing the driver workload through full-automated driving features (e.g., adaptive cruise control (ACC), automatic steering (AS), highway autopilot and etc.)

The road map of driving automation is illustrated in Figure 1.1. This figure scales between zero automation (manual driving) to full automation (fully autonomous) driving modes. Any system between both ends could be considered as ADAS. As the level of automation increases, the ADAS takes more responsibility in control authority and the design gets more complicated. For example, an ADAS based on negative human-machine interface (HMI) could only issue warning signals to the drivers advising the maneuever is not feasible where on the other hand an ADAS based on positive HMI could advise the driver when and how to do the correct maneuver [1]. In this thesis, we are interested in a specific subgroup of advanced driver assistance systems which are designed to participate in the lateral control of the vehicle along with the driver.

Combining the advantages of the human operator and the automation is a challenge in designing human-machine systems and hence it is challenging to commercialize

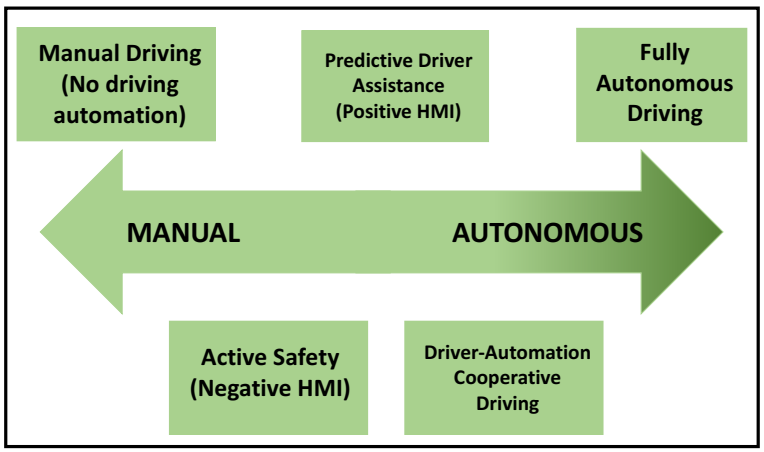


Figure 1.1 : The road map of automated driving [1].

ADAS with automated driving features such as lane keeping/changing and collision avoidance when both the driver and the system operate on the same control interface. A thorough, situation-based design of human-machine interface could yield a high acceptance rate when the influence of automated driving features on the driver is considered in the context of human factors [9].

A general approach is to incorporate *human-centered automation* concept which is widely accepted in designing human-machine systems. According to this concept, the human operator must always be actively in control and must maintain the final authority over the automation system [10]. A driver assistance system that is designed based on *human-centered automation* guidelines could perform information acquisition and situation analysis but critical decisions are taken by drivers and the system could intervene only if it is commanded by the driver [11].

However, there might be some safety-critical situations where an autonomous intervention is needed to maintain the safety without human initiation. The *situation-adaptive autonomy* is proposed in [10] which states that the level of automation (LOA) of a system could change dynamically and moreover the final authority could be traded between the human and the system depending on the situation. The experimental results of [11] show that the decision on being the final authority is likely to be context dependent.

Besides the authority allocation problem, there are still several human-automation interaction issues which should be considered thoroughly. First, the human operator

should be able to comprehend the capabilities, limits, objectives and current functionality of the system. This issue could be solved by maintaining a continuous information feedback from the system. Second, the automation should have a priori knowledge of the human operator. This issue could be solved by designing the system based on measurements and models of human operator in order to minimize the conflicts in objectives. All these human-machine interface issues could be met by the guidelines of *shared control* framework [12].

Shared control is emerged as a promising framework for designing an effective human-machine interactions and it received a growing research interest both in academia and industry. In this approach, the human operator continuously shares the control authority with the automation system in order to perform a task collaboratively. This framework enables the human operator to be in the locus of control while receiving continuous support from the automation system [2]. Although *shared control* has been applied to many fields and applications, in this thesis we focus on the lateral control of automobiles especially the design of driver steering assistance system (DSAS).

According to Abbink and Mulder, there are two distinct approaches in the literature to apply the *shared control* paradigm [2] which is illustrated in Figure 1.2.

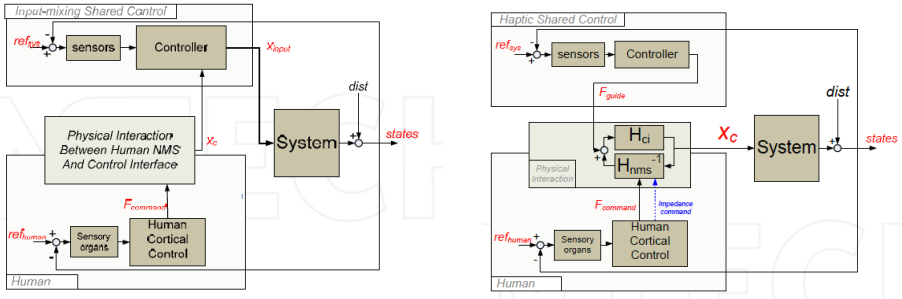


Figure 1.2 : The shared control approaches [2].

In *input-mixing shared control* approach, the control input of the plant is the combination of the manual control interface and an automated (motorized) interface. In this setup, the automation system could directly manipulate the control input with/without interacting with the human operator. For example in a steering control scenario, the total steering angle is the sum of the hand wheel (i.e., which is controlled by the driver) and the steering mechanism (i.e., controlled by the automation).

The steering mechanisms such as steer-by-wire [13, 14] and active front steering systems [15, 16] are generally used to implement these kind of DSAS. However, there is no mechanical coupling between the hand wheel angle and the tires since there is no direct connection. This could effect the steering feel of the driver and it could degrade the driving performance. Moreover, the driver does not receive any feedback from the system's activity without using an additional haptic interface such as an electric motor. And more importantly, the driver could not overrule the system's activity of manipulating the total steering angle.

On the other hand, *haptic shared control* allows both the human operator and the automation to exert forces on the control interface such that the position of the control interface is the input to the plant. In DSAS point of view, the driver and the DSAS shares the hand wheel and both exert torques in order to perform the steering task collaboratively. The controller of DSAS calculates the feedback forces which are generated by using an electric motor integrated to the steering column. The driver feels system's activity, the feedback forces acting on the hand wheel, and the system could also sense the activity of the driver. Therefore, both agents can interact and communicate continuously in a bi-directional way. This continuous interaction yields a smooth collaboration between the agents while performing a task together. The driver is able to judge the intentions of the system either reasonable or inappropriate and can act to yield or override. In order to provide such an authority to the driver, the system should exert its forces with a limited magnitude. The level of guidance to the driver varies continuously depending on the amount of generated feedback forces.

The study presented in [17] shows an integrated structure which takes advantage of both approaches. A steer-by-wire mechanism allows to superimpose an additional steering angle to the manual control interface (i.e., the hand wheel), on the other hand an electric motor allows to generate the feedback forces on the hand wheel. The driver feels the system's activity through the feedback forces. Although the driver can overrule the feedback forces in order to adjust the hand wheel angle but the system could still manipulate the total steering angle by using the steer-by-wire mechanism. The results show to yield a compromising performance.

1.2 Literature Review

Guiding the drivers by means of feedback forces in the hand wheel is shown to be a promising approach to realize a sound environment where the driver and DSAS could perform a steering task together. There are numerous studies reported in the literature about the design and implementation of DSAS which are based on the concept of shared steering control [9, 18–28].

A design and implementation of haptic steering wheel is introduced in [18] and in [20] for path following task and lane keeping task, respectively. The motivation of the authors is based on the fact that the demanding amount of visual workload while driving. Authors discuss the advantages of using haptic interface such as negotiation of the authority and bidirectional information flow. It is shown that using a haptic steering wheel improves path following performance and reduces the visual workload significantly. In both approaches, the magnitude of the haptic forces are determined by multiplying the steering angle error (i.e., $e_\delta = \delta_d - \delta$) with a proportional gain K_h (i.e., a spring stiffness). This gain is known as the Level of Haptic Authority (LOHA) and it shows the strength of the forces and is tuned such that the produced haptic forces are large enough to draw attention of the driver, but not too large to overcome driver's efforts and steer the hand wheel autonomously.

The research presented in [9] introduces a haptic steering interface for combined lane-keeping and collision avoidance tasks and points out the design issues such as the balance between comfort and safety. In such a collision avoidance scenario during lane keeping, the driver and the DSAS could act differently based on their planned avoidance maneuver. These differences in situation awareness and interpretation between both agents could cause driving intention conflicts. The authors propose using a switching-based torque characteristics (i.e., an adaptive K_h) depending on the driving situation. The results show that improvements are needed in path planning and haptic force generation algorithms in order to minimize conflicts.

There are interesting human factor studies performed with a fixed-based driving simulator associated with Abbink and Mulder [2, 21, 23, 29–32]. An experiment with twelve young drivers is performed in curve negotiation task in [21]. The haptic

forces are proportional to the lateral position error between a future reference path and an estimated look-ahead (future) position of the vehicle. Using a preview based force generation is shown to be effective for curve negotiation but the tuning of the gain and the look-ahead time constant remains to be an important issue. Moreover, the predictive behavior of controller which is obtained by using a predicted steering angle deviation is shown to be advantageous [33]. This is named as predictive haptic feedback.

The conflicts arise when the driver intendedly changes the lane and the system insists on lane keeping because both driving tasks have opposite objectives. The study in [23] extends the haptic guidance system presented in [21] to allow for intended lane change maneuvers by integrating a time-to-lane change (TLC) logic in path planning. The same issue is also considered in [27] and a variable gain for the system is proposed to maintain the cooperative state and smoothly integrate lane changing support.

The design of an haptic guidance system which supports the driver in evasive steering maneuver is discussed in [29]. By including time-to-contact (TTC) and a rule-based logic in the algorithm, driver's decision authority is preserved in obstacle avoidance scenarios with multiple escape paths. The haptic interface is designed to utilize both force and stiffness feedback to the driver. The authors in [30] point out the fact that previous studies have focused on the experiments based on a fine tuned haptic forces and investigate the effects of using an improperly tuned haptic guidance system in obstacle avoidance scenarios.

Another user study in presented in [32] proposes to use adaptive parameters (i.e., gain and look-ahead time) in the controller instead of constant. These parameters are configured individually for each driver in order to account for differences between the drivers. Although, the results show a better matching performance it is pointed out that more realistic and complicated design approaches should be considered.

There is a criticism of haptic shared control approach in [34] where the authors have discussed the drawbacks of using continuous haptic guidance. If the objective is to reduce workload and increase performance, the authors suggest to utilize full automated driving features instead of haptic shared control. Moreover, they conclude

that it is better not to assist drivers continuously in routine situations but it is more helpful to provide assistance only if the driver needs urgently.

In the aforementioned works, the haptic forces are designed in two stages. First, the amount of deviation from either an optimal steering angle or a future lateral trajectory is calculated. Calculating an optimal steering angle is generally based on road geometry and vehicle dynamics for path tracking. Second, the haptic force is proportional to the amount of deviation and the gain should be tuned to synthesis a proper and adequate level of forces. It is reported in [12] more research is needed to design more advanced techniques in calculating the haptic forces.

More control oriented approaches (e.g., optimal, robust, receding horizon) are also reported in the literature. One topic of discussion is whether the control input (optimization variable) should be chosen as the steering torque (guidance force) or steering angle. The study in [35] compares two full-automated lane keeping systems, design based on linear quadratic (LQ) control, using steering angle and torque, respectively. The results show that using steering torque as a control input allows for driver intervention and considers the steering dynamics in the control design.

A lane keeping assist system is introduced in [19] where the controller is based on LQ and steering torque as control input. However, the controller is designed based on vehicle model without considering driver dynamics. This approach could be problematic since the controller has no knowledge on the actions of the driver which might cause instability of the system.

While designing such an assistance system to share the control with the driver, it is essential to integrate a steering behavior model of the driver (action and intention) in the control framework. One of the first work that use a lane keeping driver model in the control structure is [36]. This driver model is based on lateral deviation and heading error at a look-ahead distance. On the other hand, the authors in [37, 38] use a model of two visual angles which are tracked by the driver. The assistance torque is modified based on the driver's activity during underload and overload intervals. A time-varying weight shapes the calculated assistance torque based on the need for assistance. This approach is similar to [19] but it differs in updating the weight with respect to driver's activity.

In the context of using driver model, both studies does not consider the neuromuscular system properties of driver's arms while holding the hand wheel. A cybernetic and two level driver models, with neuromuscular system properties, are integrated in a controller based on H2 optimal control theory for a human-centered driving assistance system [24,25], respectively. The neuromuscular system properties are used as time-invariant parameters in the driver model but the influence of neuromuscular adaptability to the haptic forces is not considered.

When driver steers the hand wheel and the system applies an assistance in the opposite direction (i.e., due to safety or a conflict in objectives), the haptic interface allows the driver to react in all possible ways. The driver can either accept or oppose the forces. Both actions actually modify the neuromuscular system properties of the driver's arms. A compliant driver agrees with the haptic forces and follows the system's actions on the other hand, a stiff driver disagrees with the haptic forces and overrides them [2,39]. The paper in [31] investigates the importance of including neuromuscular system properties while designing the haptic forces as a counterpart of the driver. Usually, the haptic forces are tuned heuristically without any systematic approach. The authors discuss that it is important to design the haptic forces based on the correct neuromuscular response of the driver and the results show that improperly tuned forces causes degrade in control performance. For example in a situation when the driver is stiff and the forces are designed based on compliant behavior, the forces will be too low for the driver.

Na and Cole investigate the cooperative control problem from a perspective of differential game [40–42]. The authors formulate a game theoretical approach to enhance the coordination between the driver and an active front steering system based assistance system during a collision avoidance maneuver.

The studies in [43, 44] extends the works of Na and Cole into shared control problem between the driver and the assistance system systematically as a differential game which results in a Nash equilibrium. The authors call this design approach as *cooperative shared control* since both driver and assistance system can exert torque on the hand wheel.

Model predictive control (MPC) is shown to be an attractive method for (semi) autonomous driving and it has been applied to vehicle path planning and control problems [15, 45–49]. In MPC (or receding horizon control), a constrained finite time optimal control (CFTOC) problem is solved at each time step and the first element of the resulting optimal input sequence (control input) is applied to the plant. The same process repeats when the measurements arrive for the next time step. MPC has been extensively used in control because of its capability to handle system nonlinearities, state and input constraints systematically. MPC uses a model description of the plant to predict future evolution of the state trajectories with respect to an open-loop control input sequence. The numerical optimization subroutine calculates the optimal input trajectory which minimizes a user defined objective function while satisfying all the constraints.

There are extensive studies in the literature which focus on design of driver assistance systems and shared steering control by using MPC framework. A predictive haptic feedback is designed in the context of shared control for obstacle avoidance [47]. MPC-based controller optimizes the steering angle input sequence for a collision free trajectory and predictive guidance is generated proportionally by angular deviation in the finite horizon. The system is realized in vehicle instrumented with steer-by-wire system.

A lane departure prevention system is designed in [16]. The controller calculates an additive steering angle which is assumed to be superimposed to driver's current steering action. This could be realizable with an active front steering system. But when using with an electric power assisted steering system, the authors use a low-level controller to generate the required torque. On the other hand, a torque-based steering assistance system is implemented to enhance the yaw stability of a vehicle in [48]. The steering dynamics and the torque behavior of the driver is considered in the MPC prediction model only in this study.

To sum up, the literature is vast with different control structures to calculate the requred haptic guidance force in a routine and emergency scenarios. But there are still a lot of open problems in perspective of haptic shared control when the human operator and the automation share the control interface and act together.

1.3 Purpose of the Thesis and Contributions

This thesis focuses on the design and implementation of a driver steering assistance system by using the haptic shared control approach. Therefore, we present the properties of the proposed system as :

1. The system and the driver can interact and communicate continuously in a bi-directional way by using a motorized steering wheel. The continuous interaction provides a smooth collaboration and bi-directional communication channel allows both agents to be aware of their actions.

2. The driver always remain in control and could judge the intentions of the system. If they are reasonable then the driver can yield to the system's actions. However if they are inappropriate, the driver can override the system's actions.

3. The proposed system should improve driving performance of the driver (i.e., reduce mental and physical workload) while ensuring the road safety.

The previous studies in the literature suggest to design the haptic guidance forces by using an advanced control method with more accurate models. The controller should capture the intent and steering behavior of the driver and produce the control law based on the mutual benefits and in order to minimize the conflicts. Especially one should consider using an adaptive modeling approach when dealing with human-in-the-loop control structures since the humans are tend to adapt their behaviors and their neuromuscular system properties. However, there exists couple of design issues to be addressed in order to maintain a high quality human-automation interface with good acceptance from the drivers. In order to achieve these properties, the following objectives should be met by the control framework.

Objective 1 : The feedback forces should adapt to the driver's intentions and it should be optimized with respect to driving situation. The controller should apply zero feedback force if the driver is driving safely in the predefined trajectory (manual driving). The controller should apply maximum feedback force if the driver is opposing and safety constraints are about to be violated (safety intervention). An optimum feedback force should be issued if the driver is out-of-loop (full-automated

driving). And finally, an optimum feedback force should be issued during guided navigation with respect to driver's neuromuscular response (haptic guidance).

Objective 2 : The control algorithm should reconsider its trajectory with respect to driver's intentions. There should not be any conflicts between the driver and the feedback forces if the driver's actions result in safe trajectories.

Objective 3 : The controller should be always active. Complicated rule-based switching logics should be avoided in order to simplify the control flow.

Objective 4 : The proposed control framework should be designed to be a unified control architecture for general driving situations which covers from manual to full automated driving, from lane keeping to lane changing.

Model Predictive Control (MPC) has been shown to be a promising control framework for designing such a human-automation system (see Section 1.2). Therefore we propose to use MPC framework as the control architecture of the proposed DSAS which shares the steering control task with a human driver while improving road safety and providing haptic guidance. Some of the design elements of the proposed driver steering assistance system (DSAS) is given below.

1. **Modeling :** The MPC framework employs a mathematical model of the process.
2. **Objective Function :** The goals of the DSAS should be reflected precisely in the objective function. Tuning of the associated weights enables distinctive control behavior therefore they should be tuned properly.
3. **Vehicle Safety :** The safety should be guaranteed by keeping the vehicle in the safe portion of the road and maintaining the lateral stability.
4. **Control Input :** The optimized control input should be the actual input to the steering actuator.

Those objectives could be accomplished by integrating the following ideas in MPC framework.

Modeling the driver-in-the-loop steering dynamics : The proposed controller uses the models of *Driver-Vehicle-Environment* system. Most of the previous studies only use the conventional vehicle model and does not include the model of the physical

interface (steering dynamics). Moreover, these studies consider the driver torque as a disturbance. In this thesis, we focus on modeling the interaction dynamics between the driver and the assistance system for a vehicle equipped with an MDPS system.

Estimation of driver's NMS properties : Since MPC depends on the mathematical models for predicting the future evolution of the state trajectories, the control performance depends on the prediction accuracy of future state trajectories. During the physical interaction, the driver adapts its neuromuscular properties when the feedback forces are acting. Therefore the system dynamics from the control input (feedback forces) to the steering system output (hand wheel angle) changes with respect to driver's response. In this thesis, we propose to integrate a recursive least squares algorithm in order to track the time-varying parameters of the driver-in-the-loop steering model. By updating the prediction model, we aim to increase the control performance while optimizing the guidance forces with respect to driver's preferences.

The safety constraints : The ultimate objective of the proposed driver assistance system is to keep the vehicle in the safe portion of the road while ensuring the lateral stability of the vehicle. Therefore, the safety corridor of the road should be determined for the finite horizon. These safety constraints are functions of road boundaries and the target vehicles that are traveling in the vicinity of the ego vehicle. In this thesis, we propose to formulate a convex set of constraints in terms of lateral vehicle position.

Driver intention estimation : The proposed system should be aware of the driver's intention while driving in the traffic. This minimizes the conflicts between the driver and the system in a guidance scenario. In this thesis, we integrate a driver intention estimation algorithm in the proposed control structure in order to modify the desired trajectory of the system. So when the desired trajectory of both agents match, the conflicts arising from trajectory tracking will be minimized.

Detect presence of driver : If the controller could detect whether the driver is in or out of the loop, then the controller can reconsider its actions for appropriate driving style such as full autonomous driving (when the driver is not holding the hand wheel) and semi-autonomous driving (when the driver is holding the hand wheel). In this thesis, the associated weight of the trajectory tracking term in objective function is adjusted with respect to the presence of the driver on the hand wheel. A larger weight on the

lateral tracking cost enforces the system for full autonomous driving. On the other hand a small weight is associated with assisted driving where the system does not enforce its trajectory to the driver and the system only intervenes to maintain safety (i.e., road departure prevention, blind spot prevention).

Designing the objective function : When the haptic forces are designed based on optimal control, it is important to reflect the goals of the assistance system in the objective function since the forces are calculated based on the minimization of this function. In our approach, we penalize the magnitude and rate of change of the control input in order to utilize a smooth and adequate level of haptic forces. Moreover, we penalize the lateral deviation from a predefined path in order to allow the system for acting along with the driver during a steering task. This is the base of the proposed approach and it could be extended to include yaw rate tracking and penalizing lateral acceleration. The system behavior depends on the objective function. Tuning a proper ratio between the lateral tracking weight and the input weight enables shared control properties.

Practical implementation : We design the assistance system for a commercial vehicle that is equipped with a motor driven power steering (MDPS) system which utilizes an overlay torque to the hand wheel. This type of steering actuator allows us to realize an assistance system based on haptic shared control framework.

The contributions of this thesis could be listed as follows:

- A predictive, driver-in-the-loop control framework for safe, assisted driving. It is simple to change the assistance behavior based on tuning of the weights.
- A modeling approach to represent the human-machine interactions taking place in the hand wheel while both the driver and assistance system share the steering task.
- An online parameter identification algorithm is integrated in the control framework in order to account for changes in driver's neuromuscular system properties.
- A torque-based optimization formulation to directly optimize the actual input of the plant.

- A real-time control implementation and experimental validation with a human drivers in a commercial vehicle that is equipped with a motor driven power steering (MDPS) system.
- A human factors study performed in a commercial vehicle rather than a fixed-base driving simulator.

1.4 Thesis Outline

This thesis is organized as follows.

In Chapter 2, we present the main idea and the components of model predictive control (MPC) framework. Since it is the control approach used in this thesis, we give a brief information of the optimization problem that is formed for automated driving purposes.

The formulation of the proposed driver assistance system starts from introducing the mathematical models of the *Driver-Vehicle-Environment* system in Chapter 3. A detailed information on the vehicle, steering, tire and environment models are presented since the model-based control design relies on these models in order to predict the state trajectories over prediction horizon. There is a trade off between model accuracy and computational complexity. Therefore, simple but yet accurate models are required to decrease computational complexity while having an acceptable prediction accuracy.

Instead of using state feedback models for representing the steering behavior of a driver, we use MPC framework to synthesis a steering control behavior in Chapter 4. The proposed driver model is used in closed-loop simulations to represent a human driver.

We introduce the formulation of the proposed assistance system structure in Chapter 5. The controller is based on MPC framework and it optimizes the assistance torque which is applied to the hand wheel in order to improve safety. We derive a model to describe the rotational dynamics of the hand wheel when both the driver and the controller share the steering task. This chapter presents the simulation and experimental results in blind spot intervention and road departure prevention scenarios.

The results show that changes in the neuromuscular system properties of the driver degrades the control performance and therefore an adaptive approach is needed.

We present the recursive parameter identification approach and the experimental results in Chapter 6. We use an open-loop torque reference (i.e., not calculated by a feedback controller) in order to excite the steering system and perform the identification online while the human driver is changing its response to the torque reference.

In Chapter 7, we extend the controller formulation presented in Chapter 5 by integrating the recursive least squares algorithm in order to update the parameters of the prediction model. We also conduct an experimental study in order to show the effectiveness of the adaptive assistance system in conflict scenarios.

We test the adaptive MPC formulation in a path tracking scenario for the autonomous vehicle control in Chapter 8. The main purpose of this chapter is to show the effectiveness of using an adaptive approach when there is modeling mismatch between the controller and the actual dynamics. We present the simulation results in varying road conditions.

Finally, we conclude the dissertation with a final discussion and a future directions sections in Chapter 9.

2. MODEL PREDICTIVE CONTROL

In this chapter, we present a powerful optimization framework named as “Model Predictive Control" (MPC) which is the main approach in designing the controller in this work. In Section 2.1, we give a brief introduction related to predictive control methods, we introduce some analogies to MPC in Section 2.2 and we present the structure of the MPC formulation in Section 2.3.

2.1 Introduction

Model Predictive Control (MPC) is an optimal control approach which explicitly integrates a mathematical model of the physical process to obtain a finite sequence of control policy by minimizing an objective function while satisfying input and state constraints. It is stated that MPC is the most general way of posing and approaching to a control problem in the time domain because of its capability to systematically handle system nonlinearities and constraints, while integrating concepts such as optimal control and multivariable control.

Rather than a specific control strategy, predictive control framework contains the following concepts [3].

1. A mathematical model of the process is employed in the control framework to make predictions of the future state trajectories of the system in finite time horizon.

2. An optimization procedure to calculate the optimal control policy by minimizing an objective function.

3. Receding control strategy is employed where the optimization problem is solved at each time instant and therefore the horizon is shifted one step towards the future. Only the first optimal input is applied to the system.

In MPC strategy, an optimization problem is constructed by using the mathematical model of the process in order to iterate the state trajectories as a function of

optimization variables (control inputs) for a finite time horizon. In general case, this optimization problem could be constrained such that the optimization variables should stay in a limited set of numbers which could be formed as nonlinear/linear functions. At each time step, an appropriate optimization algorithm solves the constructed optimal control problem and an optimal sequence of control inputs are calculated as a result. The open-loop state trajectories are calculated by using the mathematical model of the process and the sequence of optimal control inputs by starting from the measured/estimated states at the current time interval (as initial condition). However, only the first input of the optimal control sequence is applied to the plant and the rest of the elements are discarded. The feedback is introduced by solving the optimization problem at each time step when the full state measurement/estimate arrives. But this time the prediction horizon is shifted one step towards the future therefore introducing a receding horizon fashion. The resulting control law is nonlinear in terms of state feedback (i.e., $u(k) = f(x(k),k)$).

A general MPC plots are illustrated in Figure 2.1. At each time step (let $t_0 = k$), an open loop optimal control problem is solved for a finite time horizon ($t_{k+p} = k+p$) and open-loop state trajectory is calculating by using the optimal input sequence. In this figure, it is seen that the open-loop state trajectory approaches to the desired reference in the finite horizon but most importantly the closed-loop state trajectory behavior should be investigated in order to analyze the stability of the closed loop system.

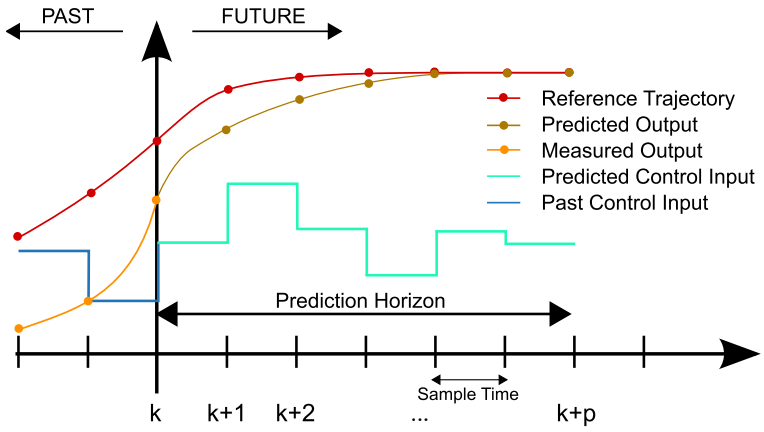


Figure 2.1 : The model predictive control strategy.

The plots in Figure 2.2 shows the open-loop and closed-loop trajectories of the output and the control input. At the current time $t = 5$s, the optimization problem is solved and an optimal input sequence is calculated. The future predictions of the output is

obtained by using the process model and the optimal input sequence in an open-loop fashion. We illustrate the actual future values (i.e., the closed-loop behavior) of the output and the control input with the open-loop trajectory at the current time step. The open-loop behavior and the closed-loop behavior should coincide when an infinite horizon optimal control problem is solved. However when the optimal control problem is solved for finite horizon, the open-loop state trajectory and the actual closed-loop state trajectory is not guaranteed to coincide which might cause stability problems. Therefore it is important to check the control system's open-loop state trajectory for any abnormal discrepancy from closed-loop state trajectory.

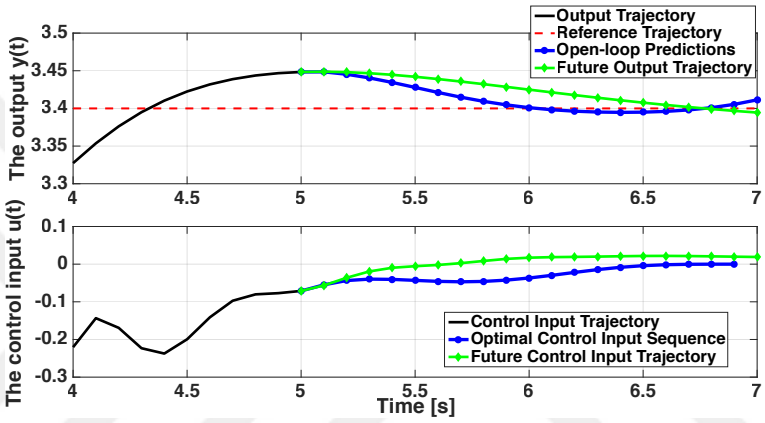


Figure 2.2 : The plots of the output and control input trajectories.

In Linear Quadratic Regulator (LQR), an infinite horizon optimal control problem is solved by using linear model of the process without any constraints. The resulting control law is linear and a constant state feedback gain is calculated by solving the algebraic Riccati equation. MPC differs itself from such optimal control techniques by using a finite-time horizon which comes with the ability to handle constraints and nonlinearities. However, this also affects obtaining the stability, robustness and feasibility proofs in all conditions. Even if there is an optimal solution, it does not imply that closed-loop system is stable. While this poses a problem in the control community, Mayne et. al. presented general sufficient conditions to design a stabilizing controller in constrained MPC cases [50].

MPC comes in handy when dealing with a variety of processes (e.g., long delay times, unstable and non-minimum phase systems), multi variable inputs and systematical inclusion of the constraints during design, the derivation of the controller is more complex than approaches such as Proportional-Integral-Derivative (PID) and the

computational time to solve the optimization problem could be high, especially when the model is nonlinear and there are constraints. But most importantly, an appropriate model which precisely captures the process dynamics should be available.

2.2 MPC Analogies

The following examples are given to provide an intuition for understanding model predictive control strategy. Actually the humans use a decision and control strategy that is very similar to what MPC use. For example, an analogy of MPC to playing chess is illustrated in Figure 2.3. In this analogy, the opponent is the *plant* and we are the *controller*. At each time step, we predict the game plan of the opponent and plan a strategy based on our prediction. However this strategy changes every time when the opponent executes an unpredicted move. This is the introduction of the feedback.

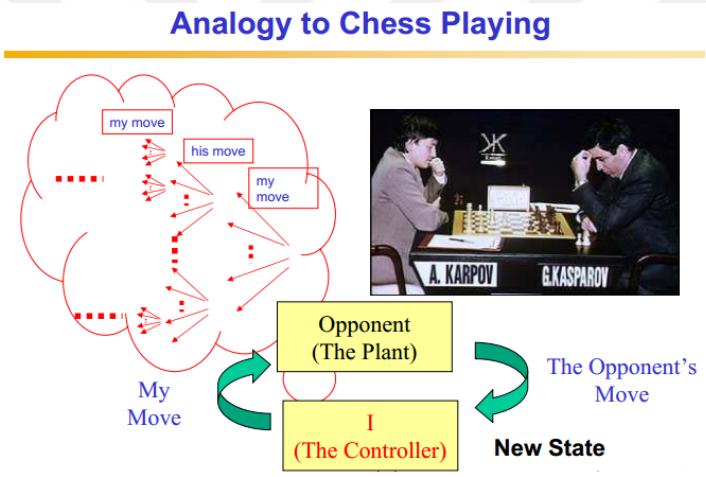


Figure 2.3 : The MPC strategy is very similar to playing chess.

Another analogy and comparison, illustrated in Figure 2.4, could be given while driving a vehicle. The driver knows the desired reference trajectory for a finite control horizon (visual perception range) and by considering the vehicle characteristics, the driver decides the control actions (i.e., steering, throttle, brakes) to follow the desired trajectory. This control strategy is very similar to MPC strategy. On the other hand when we consider the PID style of driving car since the control law is obtained based on past errors, it would be equal to driving the car just looking to the rear mirror. From these examples, it is seen that MPC strategy is a part of our everyday life since it integrates essential concepts such as prediction, optimization, feedback and system simulation.

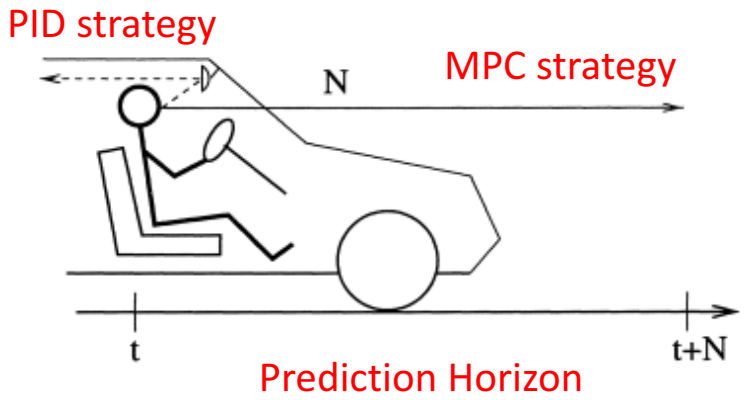


Figure 2.4 : The difference in MPC and PID when driving a vehicle [3].

2.3 The MPC Framework

We consider the following example for explaining the MPC framework in controlling the vehicles. A control system for the automated driving is illustrated in Figure 2.5. A high level trajectory generator uses the measurements from sensors and calculates a path for the controller to follow. The MPC control algorithm uses the vehicle model integrated in the optimal control problem. An optimization routine is used to solve this problem and the optimal control input is applied to the vehicle actuators.

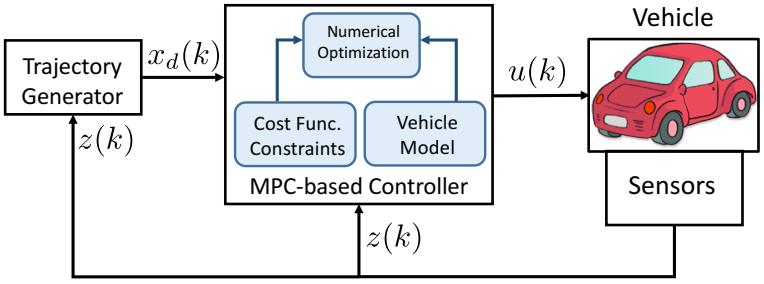


Figure 2.5 : The structure of MPC based control framework.

In the most general case, assuming there exists constraints, this optimal control problem is called “The Constrained Finite Time Optimal Control" (CFTOC) problem. Assuming a full-state measurement is available, the CFTOC problem given in (2.1) is solved at each discrete time step k by using an optimization routine in order to obtain an optimal control input sequence $\boldsymbol{U}_k = \{\boldsymbol{u}_{0|k}, \boldsymbol{u}_{1|k}, \ldots, \boldsymbol{u}_{N-1|k}\}$ for the finite prediction horizon N. Only the first control input is applied to the vehicle actuators, and this optimization routine is repeated at the next time step using new state measurements as

initial condition.

$$\min_{\boldsymbol{U}_k} \quad J(\boldsymbol{X}_k, \boldsymbol{U}_k) \tag{2.1a}$$

$$\text{s.t. } \boldsymbol{x}_{i+1|k} = f_d(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}), \quad i = 0 \dots N-1, \tag{2.1b}$$

$$\boldsymbol{x}_{i|k} \in \mathscr{X}, \quad i = 0 \dots N-1, \tag{2.1c}$$

$$\boldsymbol{u}_{i|k} \in \mathscr{U}, \quad i = 0 \dots N-1, \tag{2.1d}$$

$$\boldsymbol{x}_{i+N|k} \in \mathscr{X}_f, \tag{2.1e}$$

$$\boldsymbol{x}_{0|k} = \boldsymbol{x}(k), \tag{2.1f}$$

where the objective function is given in (2.3). The state dynamics is formulated as equality constraints in (2.1b) since the state trajectories are calculated by using the prediction model. The system is also subject to state and input constraints as given in (2.1c) and (2.1d). Moreover the last predicted state is restricted to be in the terminal set which is imposed by the constraint in (2.1e). Finally, the current time step state measurement is used as the initial conditions to the optimization problem in (2.1f). The detailed explanation of CFTOC problem is discussed in the following sections.

2.3.1 The prediction model

The prediction model plays an essential role in the controller since it is used in predicting the future state trajectory in the finite horizon. This model should capture the process dynamics accurately in order to obtain precise predictions which improves the control performance. We discretize the continuous-time dynamics (i.e., differential equations) given in (2.2a) with a fixed sampling time T_s in order to obtain the discrete-time model of the process which is given in (2.2b) where $t = kT_s$.

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t),\ \boldsymbol{u}(t)), \tag{2.2a}$$

$$\boldsymbol{x}_{i+1|k} = f_d(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}), \tag{2.2b}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^m$ is the control input vector and $f_d(.) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the state update function. We use a double subscript notation separated by a vertical bar in order to denote the future predictions of a variable. For example $\boldsymbol{x}_{i|k}$ denotes the state prediction at i-th time step originated from the state measurement obtained at time step k. So the state measurement at time step k could be written as $\boldsymbol{x}(k) = \boldsymbol{x}_{0|k}$. The future predictions are calculated by using the discrete-time

model and the optimal control input sequence $\boldsymbol{U}_k = \{\boldsymbol{u}_{0|k}, \boldsymbol{u}_{1|k}, \ldots, \boldsymbol{u}_{N-1|k}\}$ where $\boldsymbol{U}_k \in \mathbb{R}^{Nm}$.

The discrete-time model $f_d(.)$ could be linear time-invariant (LTI) such as $\boldsymbol{x}(k+1) = A_d\boldsymbol{x}(k) + B_d\boldsymbol{u}(k)$ or linear time-varying (LTV) such as $\boldsymbol{x}(k+1) = A_d(k)\boldsymbol{x}(k) + B_d(k)\boldsymbol{u}(k)$. In these cases, the predictive framework is called “Linear MPC". On the other hand, if the discrete-time model is nonlinear then it is called “Nonlinear MPC".

2.3.2 The objective function

The objective function (or cost function) $J(\boldsymbol{X}_k, \boldsymbol{U}_k) : \mathbb{R}^{(N+1)n} \times \mathbb{R}^{Nm} \to \mathbb{R}^+$ reflects the control objectives such as minimum control utilization and tracking the state references and it is generally defined as :

$$J(\boldsymbol{X}_k, \boldsymbol{U}_k) = \sum_{i=0}^{N-1} q\big(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}\big) + p\big(\boldsymbol{x}_{N|k}\big), \tag{2.3}$$

where $q(.) \in C^1$ and $q(.) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+$ is the stage cost term. The class C^1 denotes all differential functions whose first derivative is continuous. The terminal cost term is denoted by $p(.) : \mathbb{R}^n \to \mathbb{R}^+$. The set of the state predictions are denoted as $\boldsymbol{X}_k = \{\boldsymbol{x}_{0|k}, \boldsymbol{x}_{1|k}, \ldots, \boldsymbol{x}_{N|k}\}$ where $X_k \in \mathbb{R}^{(N+1)n}$.

The optimal control input sequence $\boldsymbol{U}_k$ is computed by optimizing the objective function while satisfying the constraints.

2.3.3 The constraints

One of the powerful side of MPC is to integrate the constraints beforehand. The optimizer calculates the optimal control action while considering the input and state bounds. Since the constraints are handled during optimization, there is no need to involve nonlinear elements such as saturation and slew rate limiter which might affect the stability and degrades the control performance. Also in practice, most of the processes are subject to constraints. Most importantly, actuators have finite bandwidth and have limited capacity. Therefore, it is essential to consider these shortcomings before calculating the control input.

In this thesis, we assume that the state and control inputs lie in a hyperspace which is defined by a polyhedron. In general, a polyhedron $\mathscr{P}$ in $\mathbb{R}^n$ describes the intersection of a finite set of closed half-spaces in $\mathbb{R}^n$. The mathematical expression could be

written as:

$$\mathscr{P} = \{x \in \mathbb{R}^n : Ax \leq b\}. \tag{2.4}$$

Therefore the state and input constraints given in (2.1c) and (2.1d) are polyhedra where $\mathscr{X} \subseteq \mathbb{R}^n$ and $\mathscr{U} \subseteq \mathbb{R}^m$. Also the terminal state constraint is defined by a polytope which is a bounded polyhedron where $\mathscr{X}_f \subseteq \mathbb{R}^n$.

If the control design does not consider the constraints, then the polyhedra are equal to the set of all real numbers with appropriate dimensions (e.g. $\mathscr{X} = \mathbb{R}^n$, $\mathscr{U} = \mathbb{R}^m$ and $\mathscr{X}_f = \mathbb{R}^n$).

2.3.4 Solving the optimal control problem

If we focus on the simplest case where the cost function is quadratic, there is no constraints and the prediction model is linear, the analytical solution exists. For example consider the unconstrained optimal control problem given below:

$$\min_{\boldsymbol{U}_k} \quad J(\boldsymbol{X}_k, \boldsymbol{U}_k) = \sum_{i=0}^{N-1} \left(||\boldsymbol{x}_{i+1|k} - \boldsymbol{r}_{i+1|k}||_T^2 + ||\boldsymbol{u}_{i|k}||_Q^2 \right) \tag{2.5a}$$

$$\text{s.t. } \boldsymbol{x}_{i+1|k} = A\boldsymbol{x}_{i|k} + B\boldsymbol{u}_{i|k}, \quad i = 0 \ldots N-1, \tag{2.5b}$$

$$\boldsymbol{x}_{0|k} = \boldsymbol{x}(k), \tag{2.5c}$$

where the cost function given in (2.5a) is quadratic, the discrete-time process dynamics given in (2.5b) is linear and the optimization problem is unconstrained. In this case, the optimal control input sequence could be calculated analytically by taking the derivative of the cost function with respect to the control input sequence and equating it to zero. As $N \to \infty$, the solution of MPC approaches to the solution of linear quadratic regulator (LQR). But this does not work when we can not simply take the derivative.

2.3.5 The solvers

When there are constraints in the optimal control problem an optimization routine is needed in order to solve for the optimal variables iteratively. This optimization algorithms are called as solvers or optimizers. There are a lot of open-source or commercial algorithms (solvers) which are specified by the structure of the optimization problem. Either linear programing (LP) or quadratic programming (QP) solvers should be used if the cost function is linear or quadratic with respect to the

optimization variables, assuming that the process dynamics is linear. On the other hand, when the process dynamics are nonlinear then the optimization problem becomes nonlinear programming (NP). The readers can refer to online resources for the list of all available optimization software.

2.3.6 The tuning parameters

The tuning parameters of MPC framework could be given as the weights of the individual cost terms (w_i) in the objective function and the prediction and control horizons, N_p and N_c, respectively. The weights are the tuning parameters which emphasizes the priority of the control objectives. If a relatively large weight is associated with a cost term, then the optimizer will try to minimize that term first in order to minimize the total cost.

On the other hand, the selection of prediction and control horizon is very essential for the control performance. Both parameters affect the number of optimization variables and therefore affect the size of the optimization problem. The prediction horizon N_p defines how far the controller will plan in order to decide a control law. Usually a large horizon shows good convergence and stability properties and moreover if we assume an infinite horizon without any constraints, the MPC formulation is identical to LQR. However in the presence of constraints, a large horizon makes the optimization problem complex and it might take longer duration for the solver to compute the optimization parameters. On the other hand, a short horizon results in a short-sighted control strategy which degrades the control performance in terms of stability. Therefore a suitable prediction horizon should be selected based on the control problem and computational power.

The control horizon N_c is usually selected as the same size as the prediction horizon ($N_c = N_p = N$). However in order to reduce the size of the optimization problem, a common practice is to fix the elements of the control input sequence after a certain defined size of control horizon. For example, one can set $N_p = 20$ and $N_u = 10$. In this case, the remaining ten elements of control input sequence could be assumed to be equal to the tenth element of control sequence. This way, the size of the optimization problem is reduced to 30 (assuming that the model has one state).

2.4 Discussions and Conclusion

In this chapter, we briefly introduce a popular control framework called as model predictive control. MPC offers a good amount of advantages especially when dealing with constraints in the system. Since it uses a mathematical model of the system, the control inputs are optimized with respect to predicted state trajectories which minimizes the predefined objective function. The type of optimization problem changes with respect to the relation between the states and the control inputs. Nonlinear optimization problems are more complicated and needs a lot of computational power. Also the stability and feasibility of the closed-loop control system should be considered when using a nonlinear MPC approach. It is hard to guarantee stability since the feedback control law has a nonlinear nature.

3. MODELING OF DRIVER-VEHICLE-ENVIRONMENT SYSTEM

3.1 Introduction

In this work, we are interested in designing a controller for the “Driver-Vehicle-Environment" (DVE) system which is given in Figure 3.1. The controller should interact with all the elements of DVE system where all the elements are in interaction. We need mathematical models of each element in DVE system for control and simulation purposes.

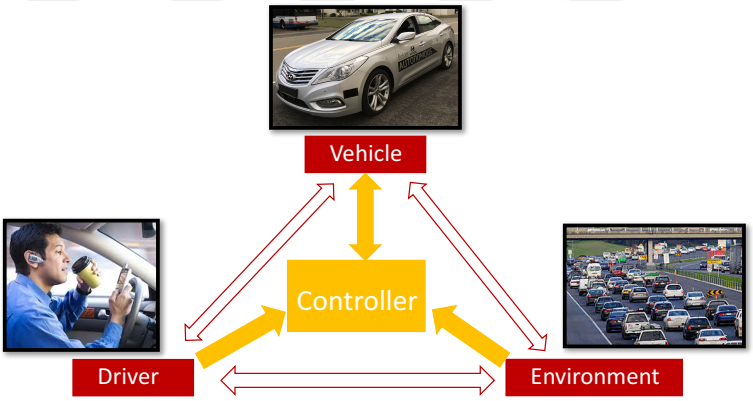


Figure 3.1 : The Driver-Vehicle-Environment system.

The DVE is illustrated as block diagrams and information flow charts in Figure 3.2. The ego vehicle (EV), where the controller action T_c is implemented, is represented by the “Steering System", “Vehicle Body Dynamics" and the “Ego Vehicle Sensors" blocks. The human-machine interactions take place in the hand wheel where both the driver and the assistance system could exert torques to steer the vehicle. The “Environment" block represents the target vehicles (TVs) that are traveling in the vicinity of the EV and it represents the road geometry and its properties. The “Driver" block represents the human driver, both cognitive and muscular properties, which uses the visual information from the environment to follow a predefined trajectory by applying a torque T_d and receiving a torque feedback (steering feel) T_s from the steering system. The “Ego Vehicle Sensors" block incorporates various sensors to

provide measurements from EV, TVs and the road geometry to the controller. The proposed controller is implemented in the “MPC Controller" block which uses the models of DVE system and the related measurements to optimize a guidance torque T_c.

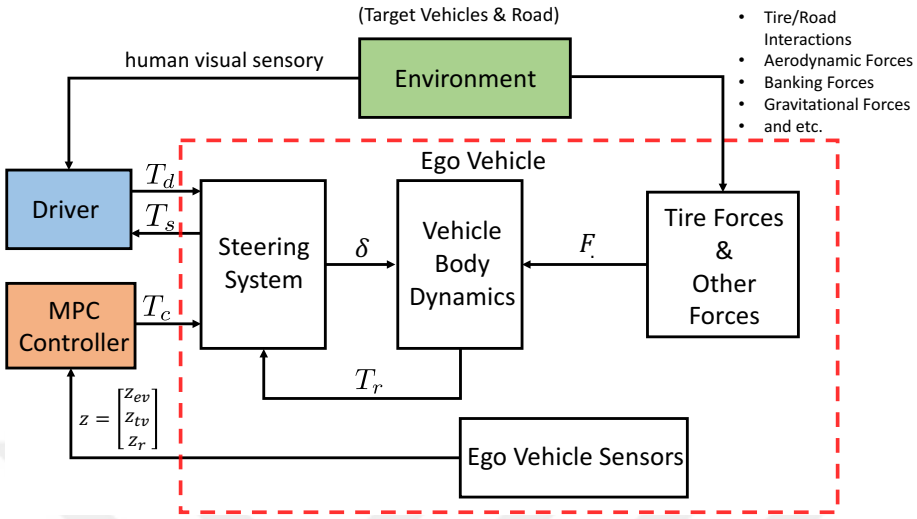


Figure 3.2 : The Driver-Vehicle-Environment (DVE) system as a block diagram.

3.2 Vehicle Dynamics Models

In this section, we present the mathematical models that describe the motion of the ego vehicle (“Vehicle Body Dynamics" block in Figure 3.2).

3.2.1 Four wheel vehicle model

The four wheel nonlinear model is an advanced model to describe the rigid body dynamics of the vehicle while capturing the most relevant dynamics associated to automated driving in highway at high speeds. This model accounts for the tire-road interactions that occurs in each tire, hence it captures the effect of individual tire forces on the vehicle dynamics.

We use the following assumptions in order to simplify the vehicle dynamics so that only the relevant dynamics (e.g., longitudinal, lateral and yaw) are captured by the model.

Assumption 1. *We ignore the roll and pitch dynamics during acceleration and braking. Also the road is assumed to be flat since we consider highway driving scenarios.*

The following set of nonlinear differential equations are used to describe the motion of the vehicle in an inertial coordinate frame (X,Y) :

$$m\ddot{x} = m\dot{\psi}\dot{y} + \sum_{i\in\{f,r\}}\sum_{j\in\{l,r\}} F_{x_{i,j}}, \tag{3.1a}$$

$$m\ddot{y} = -m\dot{\psi}\dot{x} + \sum_{i\in\{f,r\}}\sum_{j\in\{l,r\}} F_{y_{i,j}}, \tag{3.1b}$$

$$I_z\ddot{\psi} = \left(l_f \sum_{j\in\{l,r\}} F_{y_{f,j}}\right) - \left(l_r \sum_{j\in\{l,r\}} F_{y_{r,j}}\right) + \frac{l_w}{2}\left(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}\right), \tag{3.1c}$$

$$\dot{X}_v = \dot{x}\cos(\psi) - \dot{y}\sin(\psi), \tag{3.1d}$$

$$\dot{Y}_v = \dot{x}\sin(\psi) + \dot{y}\cos(\psi), \tag{3.1e}$$

where the parameters are the vehicle mass m and moment of inertia about the z-axis I_z. The distances from the CoG to front and rear axles are denoted as l_f and l_r, respectively and also l_w denotes the track width of the vehicle. The four-wheel vehicle model is illustrated in Figure 3.3.

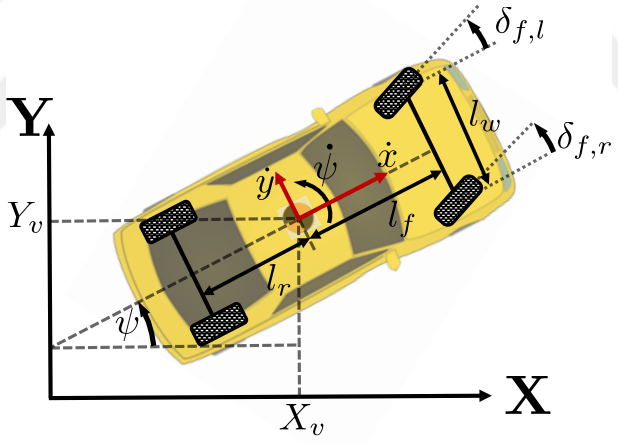
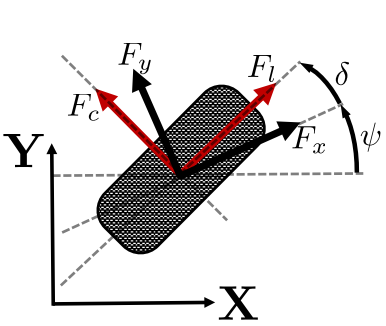


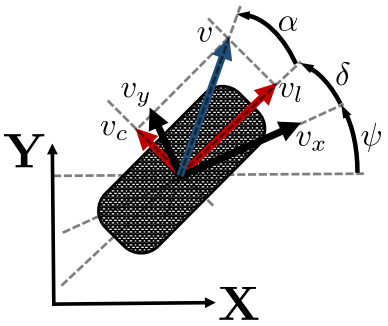
Figure 3.3 : The notation used in the four-wheel vehicle model.

The state variables are given as follows. The vehicle body axes are originated from the center of gravity (CoG) of the vehicle where $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral velocity of the vehicle at the CoG, respectively. The yaw rate $\dot{\psi}$ describes the rotational speed of the vehicle about z-axis. The yaw angle ψ is defined as the misalignment angle between the vehicle body axes x-y and the inertial axes X-Y. The position of the vehicle in inertial coordinate frame could be written as (X_v, Y_v).

The longitudinal and lateral forces in the vehicle-body axes are denoted by $F_{x_{i,j}}$ and $F_{y_{i,j}}$, respectively. The following rotational transformation is used to calculate the tire



(a) The tire forces both in tire axes and vehicle body axes.



(b) The tire velocites both in tire axes and vehicle body axes.

Figure 3.4 : The tire forces and velocities illustrated in tire axes and vehicle body axes.

forces acting on the vehicle body axes.

$$F_{y_{i,j}} = F_{l_{i,j}} \sin(\delta_{i,j}) + F_{c_{i,j}} \cos(\delta_{i,j}), \tag{3.2a}$$

$$F_{x_{i,j}} = F_{l_{i,j}} \cos(\delta_{i,j}) - F_{c_{i,j}} \sin(\delta_{i,j}), \quad i \in \{f,r\},\ j \in \{r,l\}. \tag{3.2b}$$

where $F_{l_{i,j}}$ and $F_{c_{i,j}}$ denotes the longitudinal (traction) and lateral (cornering) tire forces resolved in the tire axes, respectively and the steering angle of a tire is denoted by $\delta_{i,j}$. The first subscript i describes the longitudinal position of the tire with respect to CoG (e.g., front / rear) and the second subscript j describes the lateral position of the tire with respect to CoG (e.g., left / right). The relation between the tire and vehicle axes is illustrated in Figure 3.4(a).

We use the following assumption related to the steering of the vehicle.

Assumption 2. *The vehicle is only front steerable and the steering angles at the right and left wheels of each axle are assumed to be the same. Therefore* $\delta_{f,l} = \delta_{f,r} = \delta$ *and* $\delta_{r,l} = \delta_{r,r} = 0$.

The tire forces F_l and F_c are friction forces from the ground and generated by the tire-road interactions. These forces could be modeled as complex functions of several factors such as slip angle (α), slip ratio (σ), friction coefficient (μ) and normal force (F_z). We discuss the mathematical tire force models in Section 3.4. A general nonlinear function could be written as follows:

$$F_{l_{i,j}} = f_l(\alpha_{i,j}, \sigma_{i,j}, \mu, F_{z_{i,j}}), \tag{3.3a}$$

$$F_{c_{i,j}} = f_c(\alpha_{i,j}, \sigma_{i,j}, \mu, F_{z_{i,j}}), \quad i \in \{f,r\},\ j \in \{r,l\}. \tag{3.3b}$$

The normal force of each tire $F_{z_{i,j}}$ is simplified by using the Assumption 3.

Assumption 3. *The normal force of each tire is assumed to be constant and calculated by the steady-state weight distribution of the vehicle. This corresponds to zero longitudinal and lateral acceleration acting at the CoG.*

Hence the normal forces are defined by:

$$F_{z_{f,j}} = \frac{l_r mg}{2(l_f + l_r)}, \quad F_{z_{r,j}} = \frac{l_f mg}{2(l_f + l_r)}, \quad j \in \{r,l\}, \tag{3.4}$$

where g is the gravitational force.

The velocity vectors of a tire and its associated slip angle is illustrated in Figure 3.4(b). The lateral and longitudinal velocities in the tire axes are denoted by v_c and v_l, respectively. Also v_x and v_y denote the longitudinal and lateral tire velocities in the vehicle body axes. The transformation between those two axes are written as:

$$v_{l_{i,j}} = v_{x_{i,j}} \cos(\delta_{i,j}) + v_{y_{i,j}} \sin(\delta_{i,j}), \tag{3.5a}$$

$$v_{c_{i,j}} = -v_{x_{i,j}} \sin(\delta_{i,j}) + v_{y_{i,j}} \cos(\delta_{i,j}), \quad i \in \{f,r\},\ j \in \{r,l\}. \tag{3.5b}$$

where the velocity components in the vehicle body axes are given by:

$$v_{y_{f,j}} = \dot{y} + l_f \dot{\psi}, \quad v_{y_{r,j}} = \dot{y} - l_r \dot{\psi}, \quad j \in \{r,l\} \tag{3.6a}$$

$$v_{x_{i,l}} = \dot{x} - \frac{l_w}{2}\dot{\psi}, \quad v_{x_{i,r}} = \dot{x} + \frac{l_w}{2}\dot{\psi}, \quad i \in \{f,r\}. \tag{3.6b}$$

The slip angle of a tire α, see Figure 3.4(b), is described by the angular difference between the orientation of the tire (i.e., the physical direction) and the velocity vector of the tire (i.e., actual direction). In other words, it is the angle between the velocity vector v and its projection in longitudinal axis of the tire (i.e., v_l). This phenomena occurs because of the deformation of the treads in the contact patch of the tire.

The slip angle of each tire is defined as:

$$\alpha_{i,j} = \tan^{-1}\left(\frac{v_{c_{i,j}}}{v_{l_{i,j}}}\right). \tag{3.7}$$

The longitudinal slip of a tire is the difference between the actual longitudinal velocity v_l and the equivalent rotational velocity $r_w\omega_t$ of the tire. The angular velocity of a tire is denoted by ω_t and the effective radius of the tire is denoted by r_w.

The slip ratio of a tire σ is defined as:

$$\sigma_{i,j} = \begin{cases} \frac{r_w \omega_{t_{i,j}} - v_{l_{i,j}}}{v_{l_{i,j}}}, & \text{if } v_{l_{i,j}} \geq r_w \omega_{t_{i,j}} \text{ and } v_{l_{i,j}} \neq 0, \\ \frac{r_w \omega_{t_{i,j}} - v_{l_{i,j}}}{r_w \omega_{t_{i,j}}}, & \text{if } v_{l_{i,j}} < r_w \omega_{t_{i,j}} \text{ and } \omega_{t_{i,j}} \neq 0, \quad i \in \{f,r\},\ j \in \{r,l\}. \end{cases} \tag{3.8}$$

The slip ratio gives information about the longitudinal behavior of the tire. The first case of Equation (3.8) corresponds to the behavior during braking on the other hand the second case corresponds to the behavior during acceleration where $\sigma \in [-1,1]$. For example consider the case when $\sigma = -1$. It is only possible when $\omega_t = 0$ which corresponds to the situation where the tires are locked. Also in the case where $\sigma = 1$ corresponds to the situation where the tires are spinning (i.e., $v_l = 0$).

For completeness, we present the wheel rotation dynamics which is given by (3.9).

$$J_w \dot{\omega}_t = -b_w \omega_t - T_b - r_w F_x + T_w, \tag{3.9}$$

where J_w is the wheel inertia and b_w is the wheel damping. The engine torque distributed to the wheel is denoted by T_w and the braking torque applied to the tire is T_b. The free body diagram is illustrated in Figure 3.5.

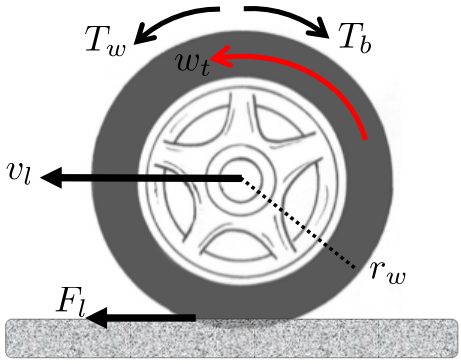


Figure 3.5 : The free body diagram for the rotational wheel dynamics.

Remark 1. For simplification, one can assume the existence of a low-level controller that regulates the wheel braking torque T_b and the wheel engine torque T_w of each tire in order to track the desired slip ratios.

3.2.2 Bicycle model

The bicycle model is derived from the four wheel vehicle model which assumes a lumped structure for the two front and rear tires as illustrated in Figure 3.6. Using

bicycle model is a preferred method to design a model-based controller since it is less complicated and captures the most important dynamics.

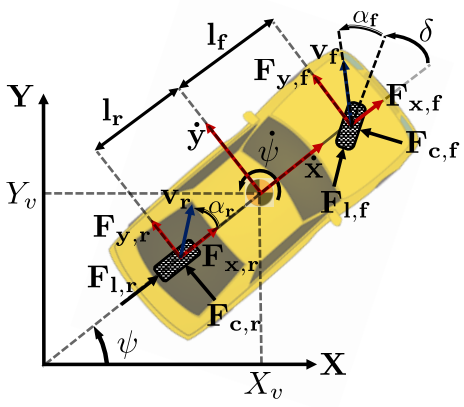


Figure 3.6 : The notation used in the bicycle model.

We can derive the equation of motion (EoM) along the vehicle body axes by applying Newton's law of motion to the CoG. We used the Assumption 1 & 4 while deriving the bicycle model.

Assumption 4. *Since the main focus is on active safety applications, we assume that the external longitudinal forces such as aerodynamic drag and rolling resistance are compensated by using a low-level controller therefore it enables achieving the desired longitudinal acceleration.*

The differential equations in (3.10a)-(3.10e) describe the motion of the vehicle in inertial coordinate system while capturing longitudinal, lateral and yaw dynamics.

$$m\ddot{x} = m\dot{\psi}\dot{y} + 2(F_{x_f} + F_{x_r}), \tag{3.10a}$$

$$m\ddot{y} = -m\dot{\psi}\dot{x} + 2(F_{y_f} + F_{y_r}), \tag{3.10b}$$

$$I_z\ddot{\psi} = 2(l_f F_{y_f} - l_r F_{y_r}), \tag{3.10c}$$

$$\dot{X}_v = \dot{x}\cos(\psi) - \dot{y}\sin(\psi), \tag{3.10d}$$

$$\dot{Y}_v = \dot{x}\sin(\psi) + \dot{y}\cos(\psi), \tag{3.10e}$$

where the parameters are the vehicle mass m, moment of inertia about the z-axis I_z and distances from the CoG to front and rear axles, denoted as l_f and l_r, respectively. The state variables are already defined in Section 3.2.1. The lateral and longitudinal tire forces in the vehicle body axes are denoted by F_{y_i} and F_{x_i}, respectively where $i \in \{f, r\}$

denotes the position of the tire. The lateral and longitudinal tire forces in the tire axes are denoted by F_{c_i} and F_{l_i}.

Assumption 5. *We consider a front steerable vehicle (i.e., $\delta_f = \delta$, $\delta_r = 0$) where δ_f and δ_r denote the steering angle of the front tires and rear tires, respectively.*

By using the Assumption 5, the tire forces in the vehicle body frame are calculated as:

$$F_{y_f} = F_{l_f}\sin(\delta) + F_{c_f}\cos(\delta), \quad F_{y_r} = F_{c_r}, \tag{3.11a}$$

$$F_{x_f} = F_{l_f}\cos(\delta) - F_{c_f}\sin(\delta), \quad F_{x_r} = F_{l_r}. \tag{3.11b}$$

Assumption 6. *The slip angles are assumed to be small (i.e., less than four degrees) under normal driving conditions.*

By using the Assumption 6, the slip angles are approximated as:

$$\alpha_f = \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}} - \delta, \qquad \alpha_r = \frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}. \tag{3.12}$$

We also use Assumption 3 for the normal forces in the bicycle model and they are defined as:

$$F_{z_f} = \frac{l_r mg}{(l_f + l_r)}, \quad F_{z_r} = \frac{l_f mg}{(l_f + l_r)}. \tag{3.13}$$

3.2.3 The vehicle motion in a road-aligned coordinate frame

While developing a controller for automated driving purposes in highways, it is useful to describe the kinematic states (e.g., position and orientation) of the vehicle relative to the road as illustrated in Figure 3.7.

The road-aligned coordinate frame is defined by the axes x_r and y_r which rotate with respect to the road geometry. The following differential equations describe the motion of the vehicle in a road-aligned coordinate frame.

$$\dot{e}_\psi = \dot{\psi} - \kappa\dot{s}, \tag{3.14a}$$

$$\dot{e}_y = \dot{x}\sin(e_\psi) + \dot{y}\cos(e_\psi), \tag{3.14b}$$

$$\dot{s} = \frac{1}{1 - \kappa e_y}\Big(\dot{x}\cos(e_\psi) - \dot{y}\sin(e_\psi)\Big), \tag{3.14c}$$

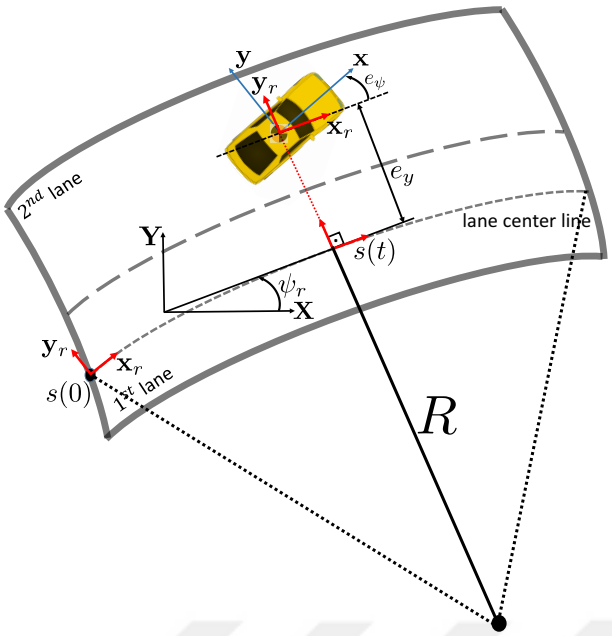


Figure 3.7 : The position and orientation of the vehicle with respect to the road.

where the spatial coordinates e_y and e_ψ denote the lateral position of the vehicle with respect to the lane center line and the orientation of the vehicle with respect to the tangent at the lane center line, respectively. s denotes the longitudinal position of the vehicle along the lane center line. The curvature of the lane markings is denoted as κ which is a function of road radius R such that $\kappa = \frac{1}{R}$. The orientation of the road with respect to the tangent line at s(t) is denoted as ψ_r (heading of the lane center line) where $\psi_r = \kappa \dot{s}$.

3.3 The Vehicle Model For Lateral Control Purposes

In this section, we derive a vehicle model for lateral control purposes. This model is simple and accurate enough for real-time model predictive control. The vehicle motion in the road aligned coordinate frame could be defined by using a bicycle model introduced in Section 3.2.2 and the kinematic equations given in (3.14). Finally, we assume that a low-level acceleration controller is available such that a desired longitudinal acceleration a_x is achieved by commanding it to the low-level controller.

The vehicle motion in road aligned coordinate frame is illustrated in Figure 3.8. The equations of motion of the vehicle in a road aligned coordinate frame is given in (3.15).

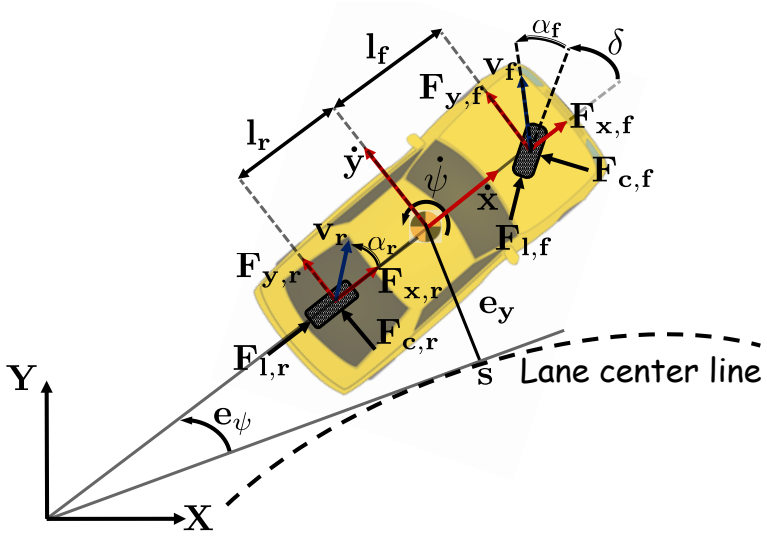


Figure 3.8 : The notations used in bicycle model in a road aligned coordinate frame.

$$\ddot{x} = a_x \tag{3.15a}$$

$$m\ddot{y} = -m\dot{\psi}\dot{x} + 2(F_{y_f} + F_{y_r}), \tag{3.15b}$$

$$I_z\ddot{\psi} = 2(l_f F_{y_f} - l_r F_{y_r}), \tag{3.15c}$$

$$\dot{e}_\psi = \dot{\psi} - \kappa\dot{s}, \tag{3.15d}$$

$$\dot{e}_y = \dot{x}\sin(e_\psi) + \dot{y}\cos(e_\psi), \tag{3.15e}$$

$$\dot{s} = \frac{1}{1-\kappa e_y}\Big(\dot{x}\cos(e_\psi) - \dot{y}\sin(e_\psi)\Big) \tag{3.15f}$$

where the vehicle body parameters are the vehicle mass m and the moment of inertia about the z-axis I_z. The distances from the CoG to front and rear axles are denoted as l_f and l_r.

The augmented continuous-time prediction model is compactly written as :

$$\dot{\boldsymbol{\xi}}(t) = f\big(\boldsymbol{\xi}(t), u(t), \boldsymbol{v}(t)\big), \tag{3.16}$$

where the state vector is $\boldsymbol{\xi} = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s]^\intercal$, the control inputs of the model is denoted by $u = \delta$, the exogenous inputs of the model is denoted as $v = [\kappa, \mu, a_x]^\intercal$.

3.4 Tire Models

The tire forces are generated as a result of tire-road interactions, which is caused by the deformation of the tire in the contact patch, during acceleration, braking and cornering. They are the most important factors that influence the vehicle dynamics. Therefore it

is essential to have knowledge on tire forces to improve the handling of the vehicle while designing an active safety system.

In literature, there have been an extensive study on modeling the tire-road interactions in order to capture the generated tire forces. In the next sections, we discuss the tire models that are used in this work.

For simplicity, we assume that the road-tire friction coefficient μ of each tire is equal $\big($i.e., $\mu_{i,j} = \mu,\ i \in \{f,r\},\ j \in \{r,l\}\big)$.

3.4.1 Pacejka tire model - magic formula

A complex and semi-empirical tire model is introduced in [51] which is known as "The Magic Tire Formula". This model provides a general formulation to calculate the tire forces accurately in wide range of operating conditions. Because of its semi-empirical structure, the parameters of the Pacejka tire model should be estimated from the experimental tests hence it provides a great accuracy in practice. Therefore, it is common to employ this model in high fidelity simulations.

Here, we consider the simple case where either lateral or longitudinal tire force is generated from the associated input variable. The simplified Pacejka tire model is defined as:

$$Y(X) = y(x) + S_v, \tag{3.17}$$

where

$$y(x) = D\sin\Big(C\tan^{-1}\big(Bx - E\big(Bx - \tan^{-1}(Bx)\big)\big)\Big), \tag{3.18a}$$

$$x = X - S_h. \tag{3.18b}$$

The output Y is the longitudinal (lateral) tire force when the input X is the slip ratio σ (slip angle α). The parameters of the Magic Formula is described as follows. B represents the stiffness factor, C denotes the shape factor, D is the peak value and E is the curvature factor. S_h and S_v denote the horizontal and vertical shift.

These parameters capture the behavior of the Magic Formula and they are acquired from the experimental tests. As seen from (3.17)-(3.18), this tire model does not depend on the road-tire friction coefficient μ and the vertical load of the tire F_z. These

variables are captured implicitly in the model parameters. Therefore, a look up table should be used in different driving situations.

3.4.2 Fiala tire model

The Pacejka model, which is presented in Section 3.4.1, depends on experimental data to calculate its parameters. However, sometimes an analytical tire model is desired because they are easy to comprehend and physically intuitive. The analytical tire model expresses the generated tire forces as a function of the variables such as slip angle α, lateral slip ratio σ, tire normal force F_z, tire-road friction parameter μ.

The Fiala tire model analytically represents the lateral tire forces by using the elastic foundation model of the tire which depends on the stiffness and elastic properties of the tire. The amount of lateral tire force that could be generated depends on the normal force F_z and the tire-road friction parameter μ. In absence of longitudinal tire force (e.g., $F_x = 0$), the maximum lateral force that can be generated is given by $F_y^{\max} = \mu F_z$. When there is an applied longitudinal force, the amount of maximum lateral force that can be generated decreases. Since the tire forces should satisfy the friction circle where $\sqrt{F_x + F_y^{\max}} = \mu F_z$. This relation is called friction circle and it is illustrated in Figure 3.9.

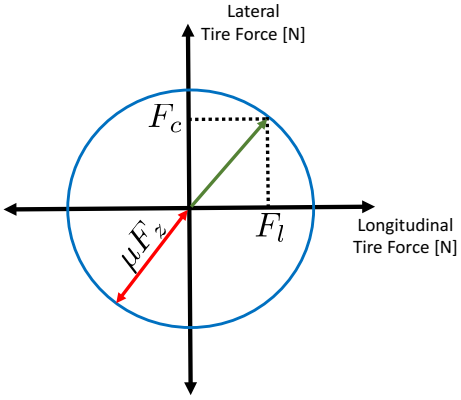


Figure 3.9 : The illustration of the friction circle.

Longitudinal Tire Force : We consider a simple model where the required longitudinal tire force is a linear function of normal force F_z. This model is given as:

$$F_l = \beta \mu F_z, \tag{3.19}$$

where β is the braking&throttle ratio where $\beta \in [-1,1]$. The case $\beta = -1$ corresponds to full braking and where on the other hand the case $\beta = 1$ corresponds to full throttle.

Also for control purposes, β could be used as the control input of the vehicle.

Remark 2. We assume there exists a low-level controller that regulates the individual wheel braking torques T_b and the wheel engine torques T_w in order to track the desired value of β.

Lateral Tire Force : Here, we consider a simplified version of Fiala tire model, which is presented in [52], where the lateral forces F_c are calculated by the following equation where normal force distribution on the contact patch of the tire is assumed to be parabolic.

$$F_c = \begin{cases} -\eta \mu F_z \phi S\big(3 - 3|\phi S| + (\phi S)^2\big) & \text{if } |S| \leq \phi^{-1} \\ -\eta \mu F_z \text{sgn}(S) & \text{otherwise,} \end{cases} \tag{3.20}$$

where $S = \tan(\alpha)$ is the tangent of the slip angle and ϕ is a function of tire properties, tire-road friction coefficient μ and the normal force F_z and it is calculated by:

$$\phi = \frac{2c_p\rho^2}{3\eta\mu F_z}, \tag{3.21}$$

where c_p is the stiffness coefficient of the tire tread in unit length and ρ is the half of the tire contact patch.

Remark 3. The cornering stiffness of a tire could be calculated as a function of tire properties as $C_\alpha = 2c_p\rho^2$.

The parameter η accounts for the effect of lateral-longitudinal force coupling and it is defined as:

$$\eta = \frac{\sqrt{\mu^2 F_z^2 - F_l^2}}{\mu F_z}, \tag{3.22}$$

where the case $\eta = 1$ corresponds to zero-longitudinal tire force case. The case $\eta = 0$ occurs when $F_l = \mu F_z$. There is a relationship between η and β as $\eta = \sqrt{1-\beta^2}$.

The Self-Aligning Moment : The self-aligning moment M_z (or the self-aligning torque T_{sat}) is the moment generated by the tire-road interactions along the z-axis of the tire. It is an important factor while driving a vehicle since it gives a torque feedback to the driver through the steering mechanism. The self-aligning moment occurs because the distance between the center of the contact patch and the point where the equivalent

lateral force acts. This is caused by the non-uniform lateral force generation along the contact patch. This distance is known as pneumatic trail t_p.

The self-aligning moment is calculated by:

$$M_z = \begin{cases} -\eta \mu F_z \rho \phi S\big(1 - 3|\phi S| + 3(\phi S)^2 - |\phi S|^3\big) & \text{if } |S| \leq \phi^{-1} \\ 0 & \text{otherwise.} \end{cases} \tag{3.23}$$

The pneumatic trail is given by:

$$t_p = \frac{M_z}{F_c}. \tag{3.24}$$

3.4.3 Linear tire model

The experimental results show that there is a linear relation between the lateral tire forces and the slip angles when the slip angles are small. Moreover, when the slip ratio is small (e.g., less than 0.1 on dry surface) and under normal driving conditions, the longitudinal tire forces is found to be proportional to the slip ratio [53]. Because of its simplicity, it is a common approach to use this model in the control design.

The linear tire model could be derived by approximating the Magic Formula tire model as:

$$Y = (BCD)\, X, \tag{3.25}$$

where the proportional term BCD represents the cornering stiffness C_α or the longitudinal stiffness C_σ.

On the other hand, one can also derived an analytic linear tire model by assuming small slip angle which is introduced in Assumption 6.

$$F_c = C_\alpha \alpha, \tag{3.26}$$

where $C_\alpha = 2c_p\rho^2$.

In the small slip angle case, the self-aligning moment is also calculated by:

$$M_z = \frac{\rho}{3} F_c. \tag{3.27}$$

3.5 Steering System Models

In this section, we focus on modeling the steering dynamics of a vehicle which is represented by “Steering System" block in Figure 3.2. We will consider two different

dynamic models which describe the steering characteristics of a vehicle which is instrumented with motor driven power steering (MDPS). The first model is introduced in Section 3.5.1 which is used in the prediction model of the controller in experimental studies. The other model is a generic steering model which characterizes the rotational dynamics of a vehicle. We use the latter model for simulations.

3.5.1 Modeling of MDPS system

MDPS is a type of electric power assisted steering (EPAS) technology which uses an electric motor to reduce the steering effort of driver. MDPS system consists of a hand wheel and a steering column, a torque sensor (a torsion bar and electronics), an electric motor and control unit, a reduction gear and a rack&pinion mechanism. The steering system of the vehicle is illustrated in Figure 3.10. The MDPS system

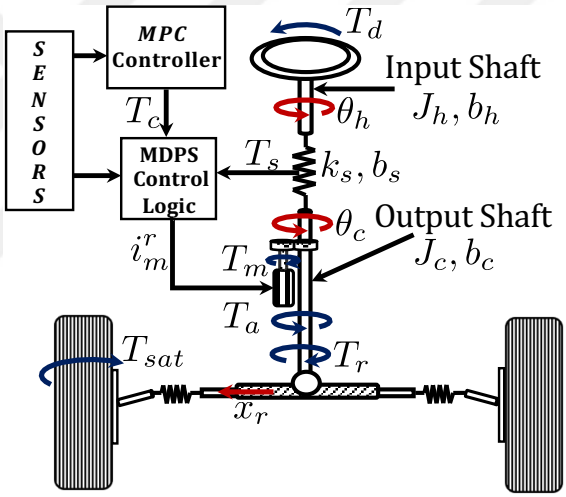


Figure 3.10 : The steering mechanism of a vehicle equipped with MDPS system.

works as follows. When the driver applies torque T_d on the steering wheel, the torsion bar twists which causes an angular difference in both ends of the torsion bar. The torque sensor measures this angular displacement and outputs the column torque measurement T_s which indicates the steering request of the driver. During a regular operation, MDPS control logic, which is illustrated in Figure 3.11, calculates a torque assistance T_a which reduces the steering effort of the driver while enhancing the steering (road) feel by compensating undesired dynamics through low-level control strategies. The MDPS control logic can also superimpose a torque overlay T_c which enables the implementation of additional safety and comfort features such as lane

keeping assist. The functionality of MDPS is preserved for driver's convenience since the basic assist functions are not disabled while using the torque overlay function [54].

The motor torque reference is calculated as $T_m^r = T_a^r/N_m$ by the MDPS control logic where the torque assistance is calculated as $T_a^r = f(T_s, T_c, \theta_s, v_x)$. The motor current reference is calculated by $i_m^r = k_t^{-1} T_m^r$ where k_t is the torque constant. Finally, the required pulse-width modulation (PWM) signals are generated to achieve the desired motor torque T_m and the assist torque $T_a = N_m T_m$ is applied to the steering column through a reduction gear with a gear ratio of N_m.

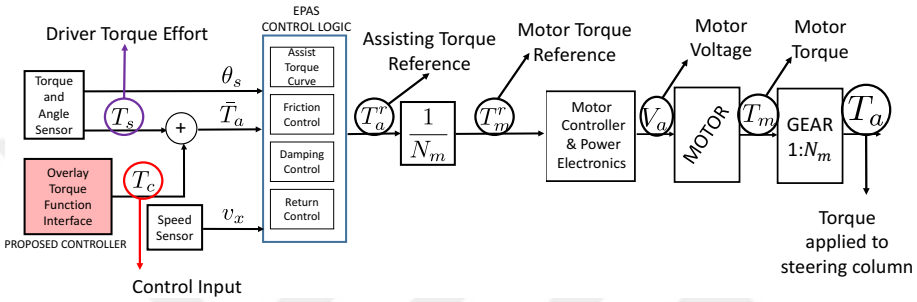


Figure 3.11 : The MDPS control logic.

Remark 4. We assume that the motor torque controller has a high bandwidth so the motor torque reference T_m^r is generated accurately with a small time constant.

A linearized free-body diagram of the steering system is illustrated in Figure 3.12 and it is represented by two lumped inertia models separated by a torsion bar with stiffness and viscous damping coefficients k_s and b_s, respectively.

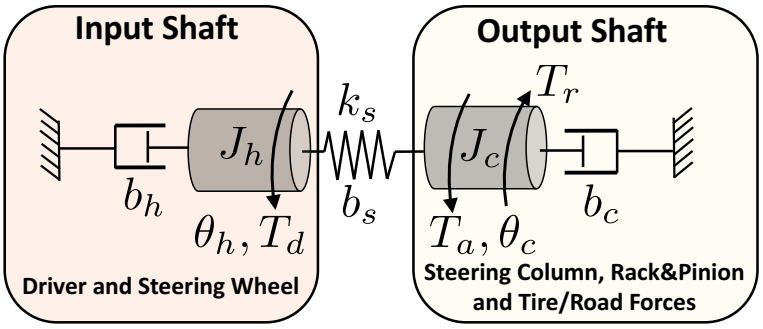


Figure 3.12 : The free-body diagram of the steering mechanism. The system is separated into two lumped inertia models because of the existence of a stiff torsion bar which couples the input and output shaft dynamics.

Assumption 7. *The nonlinear dynamics such as Coulomb friction, dead-zones and saturation limits are neglected in the models. The dynamics of the electric motor is also*

neglected and the rack&pinion dynamics are assumed to be captured in the equivalent parameters.

The driver exerts torque T_d to the hand wheel modeled by inertia J_h and viscous damping b_h. The linearized equation of motion of the input shaft dynamics is described as:

$$J_h\ddot{\theta}_h = -b_h\dot{\theta}_h - T_s + T_d, \tag{3.28}$$

where $T_s = k_s(\theta_h - \theta_c) + b_s(\dot{\theta}_h - \dot{\theta}_c)$, where θ_h and θ_c are the hand wheel and the steering column angles, respectively. The angular velocities are denoted by $\dot{\theta}_h = \omega_h$ and $\dot{\theta}_c = \omega_c$.

The output shaft dynamics describe the motion of the steering mechanism from the lower end of torsion bar to the tires. This includes the steering column, the electric motor and the gearbox, the rack&pinion mechanism and the tires. The pinion transforms the angular position of the steering column to linear displacement of the rack (i.e., $x_r = \theta_c r_p$ where r_p is the radius of the pinion) which enables the tires to move.

The linearized equation of motion of the output shaft dynamics is described as:

$$J_c\ddot{\theta}_c = -b_c\dot{\theta}_c + T_s + T_a - T_r, \tag{3.29}$$

where J_c and b_c are the equivalent inertia and damping of the output shaft. The assist torque T_a, which is defined previously, and the road reaction torque at the pinion $T_r = F_r r_p$ act on the output shaft inertia.

The road reaction force acting on the rack F_r is caused by the combined effect of mechanical trail t_m and the pneumatic trail t_p. We introduced the pneumatic trail and the resulting self-aligning torque T_{sat} in Section 3.4.2. The mechanical trail depends on the geometry of the front axle that connects the front tire to the rack which is illustrated in Figure 3.13.

The following angles are important and they determine the steering axis of a tire. The camber angle γ_p is defined by the angle between the steering axis and the vertical axis of the tire when viewed from in front of the tire. On the other hand, the caster angle γ_c is the angular displacement of the steering axis from the vertical axis of the tire. The

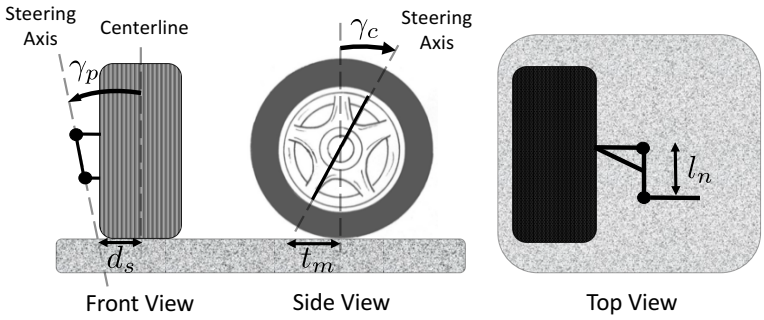


Figure 3.13 : The geometry of the front axle.

ground distances between the steering axis and the center of the tire are defined by the scrub radius d_s and the mechanical (geometric) trail t_m [55].

The road reaction force on the rack F_r is given by:

$$F_r = \frac{\cos(\gamma_p)\cos(\gamma_c)}{l_n} M_s, \tag{3.30}$$

where l_n is the length of the knuckle arm and M_s is the reaction torque along the steering axis which is a function of lateral forces $F_{c_{f,j}}$, longitudinal forces $F_{l_{f,j}}$ and the self-aligning torque M_z and it is calculated by:

$$M_s = -\cos(\gamma_p)\cos(\gamma_c)\Big(d_s(F_{l_{f,l}} - F_{l_{f,r}}) + t_m(F_{c_{f,l}} + F_{c_{f,r}}) + M_z\Big). \tag{3.31}$$

Remark 5. If the γ_c and γ_p are assumed to be small, then the scrub radius d_s and mechanical trail t_m are also small. Using this assumption, the equations (3.30) and (3.31) can be simplified as:

$$M_s = M_z = t_p F_c, \qquad F_r = \frac{M_z}{l_n} = \frac{t_p F_c}{l_n}. \tag{3.32}$$

The steering system dynamics are described by the set of differential equations given in (3.28) and (3.29). The system has two degrees-of-freedom (DOF) (i.e., θ_h and θ_c). The exogenous input of the steering dynamics is the driver torque T_d. The control input is the overlay torque input T_c. The torque assistance T_a is calculated by a complicated control framework (i.e., MDPS control logic in Figure 3.11) which involves nonlinear elements such as look-up tables, rule-based algorithms, and low-level controllers (e.g., friction, return and damping control). More detailed information related to the control algorithms of MDPS-like systems could be found in [55, 56].

For control purposes, we are only interested in the input shaft dynamics represented by (3.28) since the low-level control of MDPS system is handled by the MDPS control

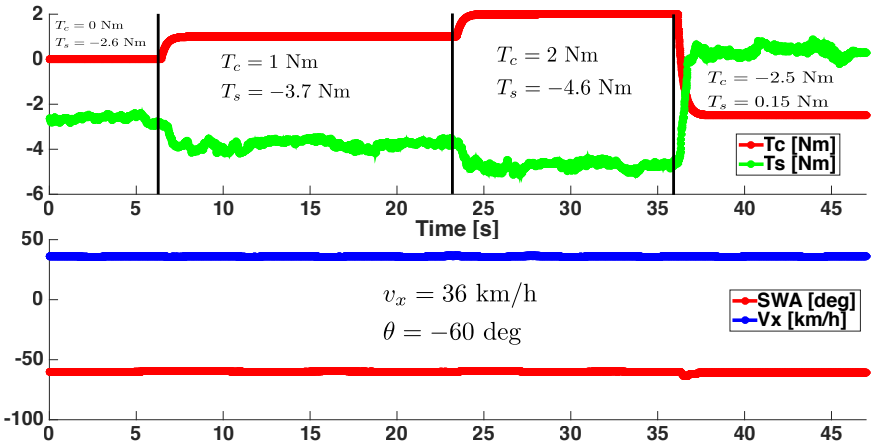


Figure 3.14 : The measurements from constant radius cornering experiments.

logic which can not be overriden. Therefore we use the following assumption to simplify the steering dynamics in (3.28) by neglecting the output shaft dynamics.

Assumption 8. *We assume that the torque assistance T_a in* (3.29) *controls the output shaft dynamics such that the driver receives torque feedback from the road (steering feel) T_{rd} when MDPS is operating in normal mode (i.e., $T_c = 0$), where in this case $T_s = T_{rd}$. On the other hand, when the overlay torque function is utilized (i.e., $T_c \neq 0$) we assume that the steering feel of the driver is described by $T_s = T_{rd} - T_c$.*

The simplified steering system model is given by:

$$J_h \ddot{\theta}_h = -b_h \dot{\theta}_h - (T_{rd} - T_c) + T_d. \tag{3.33}$$

We performed a constant radius cornering experiment with the test vehicle in order to investigate the affect of the overlay torque on the steering feel of the driver. The measurements are illustrated in Figure 3.14. The experiment is conducted at a constant longitudinal velocity $v_x = 36$ km/h and a constant hand wheel angle $\theta_h = -60$ deg. From the experimental results it is seen that under normal MDPS operation at $t \in [0,6]$ s, the torque sensor measures around $T_s = -2.6$ Nm. Between the time intervals $t \in [6,36]$ we commanded an overlay torque input with magnitudes 1 and 2 Nm consecutively in the opposite direction to the driver's steering torque. In order to keep the hand wheel angle constant, the driver opposed to the overlay torque which increased the steering torque measurement $T_s = -3.7$ Nm and $T_s = -4.6$ Nm.

On the other hand when the overlay torque is in the same direction with the driver's steering torque, the driver uses the overlay torque to decrease its workload. At the time

interval $t \in [36, 45]$ s we applied $T_c = -2.5$ Nm and the resulting measurement is $T_s = 0.15$ Nm. Therefore these experimental results justify the Assumption 8. The derived model is an approximation of the general dynamics under steady state conditions.

3.5.2 Generic steering system model

Apart from the model presented in Section 3.5.1, which is derived for experimental purposes, we derive the following steering system model for simulation purposes. We assume that the alignment torque is transfered to the hand-wheel without any modification through rack&pinion mechanism. For clarity, let $\theta_h = \theta$.

$$J_{ss}\ddot{\theta} = -b_{ss}\dot{\theta} - k_{ss}\theta + T_d + T_c - T_r, \tag{3.34}$$

where J_{ss}, b_{ss} and k_{ss} represent the equivalent inertia, damping and stiffness of the steering system, respectively. All these impedance variables are derived with respect to the hand wheel's axis of rotation. The torques acting on the hand wheel are the driver torque T_d, the guidance torque T_c (control input) and the equivalent self-aligning torque at the hand wheel T_r.

3.6 Modeling the Environment

This section considers the models for the "Environment" block in Figure 3.2. The term *Environment* includes the target vehicles (TVs), which are the vehicles traveling around the EV, and the road where all vehicles are traveling on.

3.6.1 Target vehicle dynamics

We use the intelligent driver model (IDM) to represent the longitudinal dynamics of TVs [57]. The acceleration of an individual TV is calculated by:

$$\dot{v}_t = a_{\max}\left[1 - \left(\frac{v_t}{v^d}\right)^{a_{exp}} - \left(\frac{d^*(v_t, \Delta v_{p,t})}{\Delta s_{p,t}}\right)^2\right], \tag{3.35a}$$

$$d^*(v_t, \Delta v_{p,t}) = \begin{cases} d_0 + t_{gap}v_t + \frac{v_t\,\Delta v_{p,t}}{2\sqrt{a_{\max}a_c}} & \text{if there is a PV} \\ 0 & \text{otherwise,} \end{cases} \tag{3.35b}$$

where the relative velocity is defined as $\Delta v_{p,t} \triangleq v_p - v_t$ and v_p is the velocity of preceding vehicle (PV), usually it is the EV. The relative distance (actual gap) is denoted as $\Delta s_{p,t} \triangleq s_p - s_t$ where s_p and s_t denote the longitudinal positions of PV

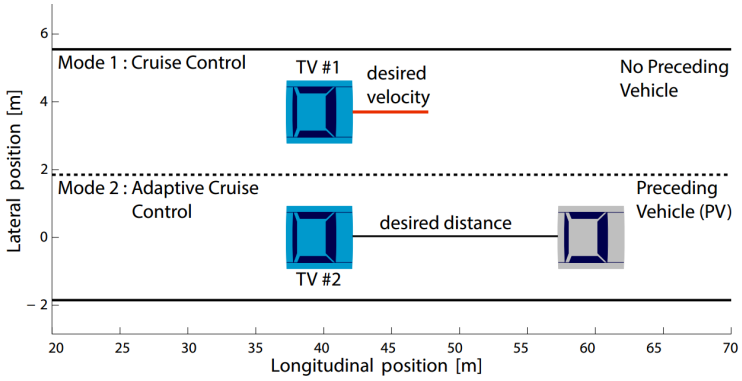


Figure 3.15 : The behavior modes of IDM algorithm.

and TV, respectively. The IDM has two modes: free-flow and car-following which is illustrated in Figure 3.15. If there is no PV (i.e., the free-flow mode), the algorithm calculates the required longitudinal acceleration to track the desired velocity. This is determined by the maximum acceleration parameter $a_{\max}$, current velocity v_t, the desired velocity v^d and a fixed acceleration exponent a_{exp}. The desired speed is assumed to be known for the TVs. On the other hand when there is PV (i.e., the car-following mode), the algorithm calculates the required longitudinal acceleration to track the desired distance. The affect of a PV is described by the ratio of the desired gap $d^*(.)$ and the actual gap $\Delta s_{p,t}$.

Remark 6. We use fixed parameters in IDM to describe the motion of the target vehicles but in practical situations one should estimate those parameters by observing the motion of the target vehicles in traffic.

3.6.2 The road model

The road model could be described by its geometry and its surface type. The curvature of the road is denoted by κ and it is the inverse of road radius $\kappa = \frac{1}{R}$. For a given straight road, the road radius is infinite therefore the curvature is zero. Usually, clothoid spirals are used to describe the road geometry since they provide smooth transition from a circular road to straight road [53]. Also the length of the lane w_{lane} and the number of lanes in the same direction are another geometric properties of the road.

On the other hand, the type of the road surface (e.g., dry asphalt, packed snow) is very important for determining the generated tire forces since it affects the tire-road

interactions. The road friction coefficient μ is a parameter which characterizes the frictional relations between the road and the tire contact patch.

3.7 Discussions and Conclusions

In this chapter, we introduce the mathematical models that are used in simulations and experiments. Because of the scope of this thesis, we are only interested in the lateral dynamics of the vehicle. Therefore we didn't include the longitudinal dynamics or suspension dynamics in the control design. For longitudinal control applications, a thorough modeling on longitudinal tire forces, vehicle dynamics and environment forces is important.

Since the control strategy is chosen as model predictive control, the control performance heavily depends on the mathematical models of the process. More complex models are more accurate than simple models but using a complex model in MPC framework is not desired since it makes the optimization problem very hard to solve. Generally, the four-wheel vehicle model is used to simulate the vehicle dynamics in simulations where on the other hand, the nonlinear bicycle model is used in the MPC controller. The complex tire models, such as Pacejka and Fiala, could be used for simulating the actual tire forces. However, for control purposes we use linear tire model as an approximation to the actual tire forces in order to simplify the problem.

4. DRIVER STEERING CONTROL

4.1 Introduction

The driver assistance systems (DAS) are designed to work with the human driver, therefore it is important to realize DAS while considering its interaction with the human driver. But before commercializing a DAS feature, these systems should be rigorously tested and validated. However, it is not desired to test the system with numerous human drivers since it would not be cost and time effective. Therefore, it is a good practice to use a realistic driver model in order to reduce development time and cost by performing a simulation-based studies.

Human drivers are subject to a demanding visual tasks while driving and therefore most of the studies in the literature consider modeling the driver steering behavior as a function of visual information. The idea to model the human steering task into two components as an open-loop anticipatory mode and a closed loop compensatory mode is first proposed in [58]. A more developed recent version is presented in [59]. More complex approaches use the power of optimal control theory with a road preview information (lateral preview control) to represent the anticipatory and compensatory control actions of the driver [60–62]

Although these models only depend on visual road preview, it is shown that the drivers also integrate their neuromuscular system (NMS) properties such that steering feel (steering torque feedback) of the road is essential to comprehend the vehicle-road interactions and thus important for handling of the vehicle [63]. Optimal preview control based NMS driver models are presented in [64–67] where an optimal preview control based path-following controller is integrated with the NMS of driver's arm. On the other hand, a more complex version of two-point driver model which is integrated with NMS of driver's arm is introduced in [68, 69].

A human driver uses a knowledge and experience based model of the vehicle dynamics along with the sensory channels providing information on visual road preview and

steering feel in order to predict the future trajectory of the vehicle and determines the necessary control action (e.g., steering torque/angle) that minimizes the deviation between current and desired trajectory. This process is similar to the principles of MPC framework and there are numerous research about modeling the driver steering behavior using MPC framework in the literature [70–72]. These works generally use the MPC framework in representing the “decision module" of the driver and in result the desired steering angle is optimized as the control input and a neuromuscular system model is used separately as the “execution module" that accounts for the neuromuscular dynamics in order to calculate the steering torque of the driver.

4.2 Objectives and Contributions

The objective of this chapter is to formulate an MPC-based driver steering control framework which represents the “Driver" block in Figure 3.2 for closed-loop simulation purposes. In order to evaluate the proposed ADAS in simulations, we need to simulate the human driver’s steering behavior while the driver is interacting with the feedback forces. The driver-automation interactions should be captured in the “Driver Steering Control" in order to have a realistic behavior.

The driver steering control block is illustrated in Figure 4.1 where the proposed control strategy is shown in dotted line. This “simulated human driver" applies T_d to the steering system and receives a torque feedback through the hand wheel.

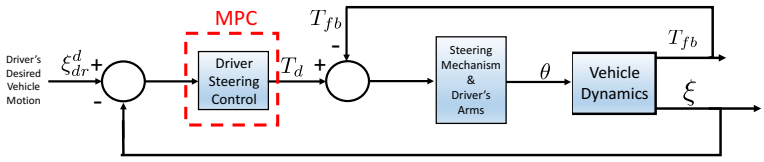


Figure 4.1 : The driver steering control structure.

The contribution is integration of the decision and execution module in MPC framework and formulate one optimization problem to calculate the driver torque T_d that is generated by the muscles. This approach is a realistic way to mimic a human-like steering behavior in routine driving tasks such as lane keeping/changing while the driver is also interacting with the feedback forces. The proposed approach

also reflects the neuromuscular response of driver's arms when the driver is giving way or opposing these forces on the hand wheel.

4.3 Controller Architecture

4.3.1 Prediction model

The Vehicle Model : We use the vehicle model introduced in Section 3.3 which is compactly written in equation (3.16).

Tire Model : For simplicity, we use the linear tire model introduced in Section 3.4.3 where the slip angles are assumed to be small and they are given in (3.12).

Steering Dynamics : We use the generic steering dynamics model presented in Section 3.5.2. As the driver holds the hand wheel, the mechanical impedance properties of the driver's arms and the steering system are integrated as one lumped inertia model. Therefore the coupled steering dynamics could be given as follows :

$$\big(J_{ss}+J_d(t)\big)\ddot{\theta} = -\big(b_{ss}+b_d(t)\big)\dot{\theta} - \big(k_{ss}+k_d(t)\big)\theta + T_d - T_{fb}, \tag{4.1}$$

where $J_d(t)$, $b_d(t)$ and $k_d(t)$ represent the inertia, damping and stiffness of driver's arms arising from properties of limbs, tendons and muscles. The driver receives a torque feedback from the hand wheel $T_{fb} = T_{rd} - T_c$. The human driver also can increase the stiffness and damping of the arms when opposing the torque intervention and the inertia depends on the position of the arms.

Remark 7. The mechanical impedance properties of driver's arms could be estimated on-line by using recursive least squares [73]. In this paper, we assume to have the estimated values available to be used by the proposed control framework.

In this work, we consider the following driving behavior modes of the driver.

1. **Impedance Mode :** The driver opposes the guidance forces by co-contracting the arm muscles. The driver only resists the motion of the hand wheel and increases the equivalent impedance of the coupled dynamics without actively steering the vehicle.

2. **Driving Mode :** The objective of the driver is to accomplish a predefined task, such as lane keeping/changing, while interacting with the guidance forces.

We discretize the continuous dynamics by using a forward difference method at a fixed sampling time. The plant model for the “Driver" block is compactly written as:

$$\boldsymbol{\xi}(k+1) = f_d(\boldsymbol{\xi}(k), u(k), \boldsymbol{v}(k)), \tag{4.2}$$

where the state vector is $\boldsymbol{\xi} = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s, \theta, \dot{\theta}]^\intercal$, the control input is $u = T_d$ and $\boldsymbol{v} = [\kappa, T_{fb}, e_y^d]^\intercal$ is the exogenous input vector where e_y^d is the predefined lateral trajectory of the driver.

4.3.2 Safety constraints

We can express the time-varying safety constraints in prediction horizon N as:

$$e_{y,\min} \leq e_y(i) \leq e_{y,\max}, \qquad i = 1 \ldots N. \tag{4.3}$$

We also impose slip angle constraints on the vehicle dynamics in order to enforce the slip angles to stay in the linear region of tire forces.

$$\alpha_{\min} \leq \alpha_j(i) \leq \alpha_{\max}, \quad j \in \{f, r\}, \quad i = 1 \ldots N. \tag{4.4}$$

The safety constraints in (4.3) - (4.4) are compactly written as:

$$h^{\xi}(\boldsymbol{\xi}_{i+1}, u_i) \leq \mathbf{0} \quad i = 0 \ldots N-1, \tag{4.5}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

4.3.3 Driver torque constraints

We enforce the following magnitude and the rate constraints on the control input $u = T_d$ which characterizes the physical limitations of human driver's arm dynamics and torque generation.

$$T_{d,\min} \leq T_d(i) \leq T_{d,\max}, \tag{4.6a}$$

$$\Delta T_{d,\min} \leq \Delta T_d(i) \leq \Delta T_{d,\max}, \quad i = 1 \ldots N. \tag{4.6b}$$

The driver torque constraints (4.6) could be compactly written as:

$$h^u(u_i, \Delta u_i) \leq \mathbf{0}, \quad i = 0 \ldots N-1, \tag{4.7}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

4.3.4 Optimal control problem

The following numerical optimization problem in (4.8) is solved at each time step to determine the optimal driver torque ($u(k) = T_d$) that minimizes the objective function in (4.8a) and satisfies the constraints given in (4.8c)-(4.8d) by using the prediction model in (4.8b).

$$\min_{\boldsymbol{u}(k),\varepsilon} \sum_{i=0}^{N-1} \left(||u_{i|k}||^2_{Q_d} + ||\Delta u_{i|k}||^2_{R_d} \right) + \sum_{i=1}^{N} \left(||\boldsymbol{\xi}_{i|k} - \boldsymbol{\xi}^d_{i|k}||^2_{S_d} \right) + M\varepsilon \tag{4.8a}$$

$$\text{s.t. } \boldsymbol{\xi}_{i+1|k} = f_d(\boldsymbol{\xi}_{i|k}, u_{i|k}, \boldsymbol{v}_{i|k}), \tag{4.8b}$$

$$h^{\boldsymbol{\xi}}(\boldsymbol{\xi}_{i+1|k}, u_{i|k}) \leq \varepsilon, \quad \varepsilon \geq \mathbf{0} \tag{4.8c}$$

$$h^{u}(u_{i|k}, \Delta u_{i|k}) \leq \mathbf{0}, \tag{4.8d}$$

$$\boldsymbol{\xi}_{0|k} = \boldsymbol{\xi}(k), \qquad u_{-1|k} = u(k-1), \quad i = 0 \ldots N-1, \tag{4.8e}$$

where $\boldsymbol{\xi}^d(i)$ is the desired state trajectory of the driver for finite prediction horizon N. This trajectory is provided by a high-level decision module which models the intention of the driver in a given situation. For simplicity, we only consider lane changing/keeping as driving modes by using a desired lateral position reference e^d_y which is generated at predefined time intervals. By setting the lateral tracking weight to a positive value will enforce the driver to execute the desired maneuver (i.e., driving mode). On the other hand, if it is zero then the driver passively opposes the guidance forces (i.e., impedance mode).

4.4 Discussions and Conclusions

This chapter introduces a driver steering control approach based on MPC framework which is used to simulate a steering behavior of driver in simulations. The driver model consists of an internal model of vehicle and steering dynamics as well as the neuromuscular system properties of driver's arms and the road preview. This formulation enables a realistic representation of human driver steering behavior because of the similarity between how humans operate the vehicle and how a MPC based controller is formulated. We use the proposed driver control structure in driver-in-the-loop simulations. These simulations are presented in Chapter 5.

5. A NOVEL TORQUE-BASED PREDICTIVE STEERING ASSISTANCE SYSTEM

5.1 Introduction

In this chapter, we are interested in the design of a driver steering assistance system (DSAS) by using a powerful optimization method called model predictive control (MPC). The proposed DSAS utilizes the steering actuator of the vehicle in order to generate a desired feedback forces (haptic force/guidance or assistance torque) which is exerted to the hand wheel while the driver is in-the-loop. The purpose of such a system could vary depending on the design criteria but common applications are lane keeping support, lane departure prevention, blind spot prevention and etc.

In our design, we specifically consider a torque-based steering actuator such as motor driven power steering system (MDPS) which supports the driver by electric powered steering assistance (EPAS) properties and also it could exert an overlay torque on top of the steering assistance. The proposed DSAS uses the torque overlay function in order to issue the feedback forces. MDPS structure allows both driver and the assistance system to perform a cooperative steering task while operating on the same physical control interface, the hand wheel. We also consider the preservation of the steering column which allows a direct mechanical relation between the hand wheel and the tire positions. When driver grips the hand wheel, this mechanical coupling enables the driver to receive a torque feedback both from the tire-road interactions and from the assistance system. This property is important in driving since it preserves natural steering feel (torque feedback) and the driver is aware of system's activity.

The previous studies in the literature focus on formulating the DSAS without considering both steering and driver's neuromuscular dynamics [27, 32, 47, 74, 75]. In these studies the haptic forces are calculated in two steps. First step is the calculation of a desired steering angle which fulfills the given objectives (e.g., lane keeping/changing, obstacle avoidance). The controller is usually based on a preview driver model.

Second step is the calculation of the haptic guidance which is proportional to the either deviation from an optimal steering or a desired lateral trajectory. However, the tuning of the haptic guidance is heuristic and requires subjective and complex steering feel calibration.

When using model-based control approaches, such as MPC, to design the assistance system, sometimes it is favorable to use a simple model of the process and therefore ignoring the steering dynamics. However, it is discussed that steering dynamics affect the control of the vehicle dominantly in the practical challenges [76].

On the other hand, a lane departure avoidance system is proposed using a predictive controller to optimize a steering angle in [16]. This formulation could be directly implemented in a vehicle with active front steering system since it requests an incremental steering angle as an input. However, when this assistance system is implemented in a vehicle with MDPS, the following problems arise.

1. **Control Input Mismatch :** The optimal input is not applied to the plant. This corresponds to a model mismatch since the dynamics of mapping from the optimal input to actual input is not considered.

2. **A need for a low-level controller :** A low-level controller (e.g., P, PI/D) should be designed to calculate the actual input (torque). This controller needs to be tuned so that the magnitude and rate of the corrective torque feels comfortable to the driver.

3. **External Nonlinearities :** An external nonlinear functions such as saturation and slew-rate should be use to limit the calculated input with respect to physical limitations of the actuator.

All these studies in the literature show us the importance of formulating the control architecture depending on the actual control input of the plant. Also a systematical way to design the assistance torque is essential to prevent subjective calibration process. In order to meet these points, we focus on formulating an MPC-based controller which integrates the steering dynamics into its prediction model while considering the neuromuscular system properties of the driver's arms. The proposed control approach directly optimizes the actual input of the system (guidance torque) by minimizing an objective function while satisfying the constraints on control input and state variables.

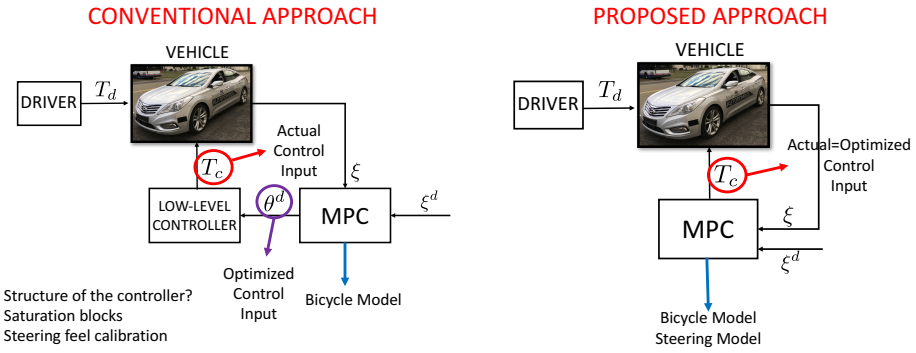


Figure 5.1 : The comparison of both control approaches.

The proposed and conventional control approaches are illustrated in Figure 5.1. It is seen that in conventional approach the optimized control input does not match the actual control input. Therefore an additional control structure should be used to transform the optimized input to the actual input. This might degrade the control performance by introducing additional dynamics and nonlinearities.

In this work we focus on the safety intervention scenarios where the proposed assistance system applies a guidance torque in order to improve safety of the driver. The novelty of the proposed controller arises from the formulation of the human-machine interactions in the control framework. During a shared steering task, both the driver and the controller aims to achieve their optimal control performance while interacting at the hand wheel through exerted forces (torques). This is illustrated in Figure 5.2. Since the driver is holding the hand wheel, the neuromuscular properties of the driver's arms comes into play. Therefore these interactions should be captured by the controller. This enables the controller to optimize the guidance torque systematically with respect to driver's response to system's activity instead of tuning it depending on the subjective steering feel of different drivers. This chapter is

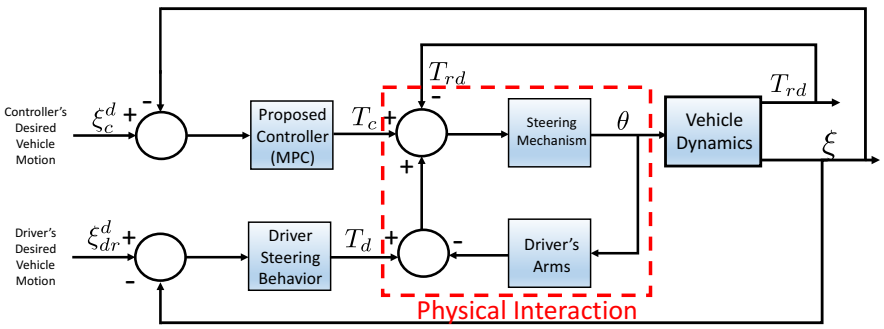


Figure 5.2 : The driver and the controller both acting on the hand wheel and share the control of the vehicle.

partly published in the following papers of the authors [77, 78]. The contributions of this chapter are given as follows:

- A modeling approach to formulate the neuromuscular response of the driver during physical interaction with the guidance torque.
- A predictive, driver-in-the-loop control framework for safe, assisted driving. The proposed controller optimizes the actual control input (i.e., actuator torque input) rather than desired steering angle.
- A real-time control implementation and experimental validation of the proposed framework with a human driver in a commercial vehicle that is equipped with a motor driven power steering (MDPS) system.

5.2 The Controller Architecture

In this section, we are interested in designing a predictive controller for the "Driver-Vehicle-Environment" (DVE) system which is introduced in Chapter 3.1. We present the prediction model employed in MPC which represents the dynamics of DVE system in Section 5.2.1. The constraint formulation is given in Section 5.2.2. Finally, the optimal control problem is introduced in Section 5.2.3.

5.2.1 Prediction model

The Vehicle Model : We use the vehicle model introduced in Section 3.3 which is compactly written in equation (3.16). This model has six states and one control input which is the steering angle (δ).

Tire Model : For the lateral tire forces we use the linear tire model introduced in Section 3.4.3 where the slip angles are assumed to be small and they are given in (3.12).

The Driver-in-the-loop Steering System Model : We focus on modeling the human driver steering behavior, in terms of driver torque T_d, during conflicts with the system's activity. In order to minimize the conflicts and provide less intrusive intervention, the controller should consider the steering behavior of the driver (e.g., driver intention). On the other hand when the system intervenes to ensure safety, the driver could increase

the damping and stiffness of the arms to oppose the guidance torque [2, 4]. Therefore for an effective safety intervention, the neuromuscular response of the driver should be also considered in the control framework.

We consider a motor driven power steering (MDPS) system where an electric motor is mounted on the steering column. The proposed DSAS uses this actuator to provide feedback forces (guidance torque/haptic guidance) T_c to the driver through the hand wheel. We consider the simplified dynamic model of MDPS system given in (3.33) and integrate the mechanical impedance properties of driver's arms [64].

When the driver grips the hand wheel, the mechanical impedance properties of the steering system and the driver's arms are coupled at the contact point. This is illustrated in Figure 5.3. In normal driving (i.e., without safety intervention), the driver exerts a

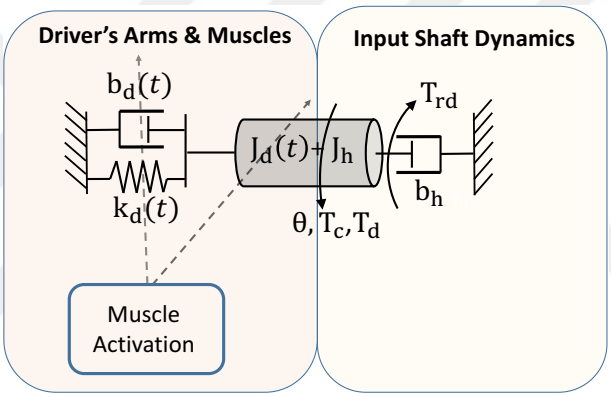


Figure 5.3 : The integrated driver and steering system free body diagram.

steering torque T_d to follow a desired trajectory. The integrated model of the driver's arms and the steering system is given below.

$$(J_h + J_{d,p})\ddot{\theta} = -(b_h + b_{d,p})\dot{\theta} - k_{d,p}\theta + T_d - T_{rd}, \tag{5.1}$$

where $J_{d,p}$, $b_{d,p}$ and $k_{d,p}$ describe the inertia, damping and stiffness of the driver's arms in relaxed condition (i.e., low co-contraction level). Those parameters are assumed to be constant during normal driving.

We use the approach presented in [48] to model T_d as a sum of feedback (FB) and feedforward (FF) torque components as:

$$T_d = T_d^{fb} + T_d^{ff}, \tag{5.2}$$

where the FF component $T_d^{ff} = T_{rd} + k_{d,p}\theta$ compensates for the self-aligning torque and the centering behavior of the steering dynamics. The FB component $T_d^{fb} =$

$K_d(\theta^d - \theta)$ represents the torque that is generated to track a desired steering angle θ^d and it is a proportional controller with the gain K_d. The modeling of the driver torque behavior is illustrated in Figure 5.4.

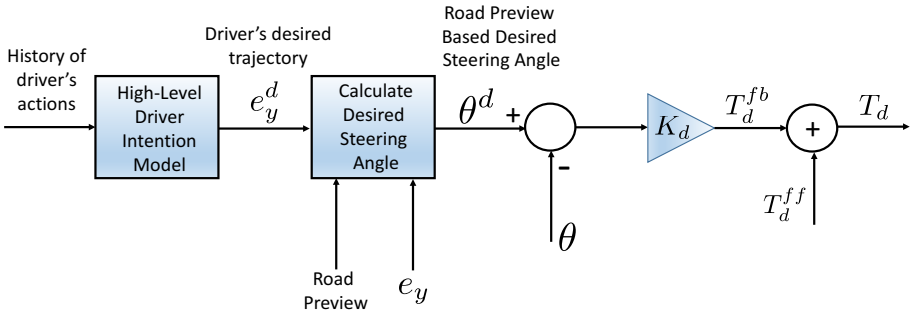


Figure 5.4 : The modeling of driver torque behavior.

The desired steering angle is a function of vehicle states and road geometry [16]. Assuming the road is straight, it is modeled as:

$$\theta_d = K_y(e_y^d - e_y) - K_\psi(e_\psi), \tag{5.3}$$

where e_y^d is the desired lateral distance which is calculated by using a driver intention (e.g. lane keeping/changing) algorithm such as proposed in [79]. If the driver is estimated to be in lane keeping mode, then e_y^d is the center line of the current lane. Otherwise if the driver intents to change the lane to the left or right, e_y^d is the center line of the destination lane.

The experimental results presented in [4] show that the damping and stiffness of driver's arms increase as the driver opposes the torque intervention by co-contracting the muscles (i.e., tensed condition or high co-contraction level). We consider the muscle activation of the driver's arms in the prediction model by augmenting it with the mechanical impedance properties of the arms in relaxed condition. Therefore, the driver-in-the-loop steering dynamics is represented by the following equation of motion:

$$J_{eq}\ddot{\theta} = -b_{eq}(t)\dot{\theta} - k_{eq}(t)\theta + T_d + T_c - T_{rd}, \tag{5.4}$$

where $J_{eq}(t) = J_h + J_{d,p}$, $b_{eq}(t) = b_h + b_{d,p} + b_{d,a}(t)$ and $k_{eq}(t) = k_{d,p} + k_{d,a}(t)$ are the equivalent inertia, damping and stiffness of the combined driver-steering system dynamics.

Remark 8. The active damping and stiffness of the driver's arms, $b_{d,a}(t)$ and $k_{d,a}(t)$, are zero either when there is no guidance from the assistance system or when driver does not oppose the torque intervention.

The augmented continuous-time prediction model, which consists of the vehicle model and driver-in-the-loop steering model, is compactly written as :

$$\dot{\boldsymbol{\xi}}(t) = f\big(\boldsymbol{\xi}(t), u(t), \boldsymbol{v}(t)\big), \tag{5.5}$$

where the state vector is $\boldsymbol{\xi} = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s, \theta, \dot{\theta}]^\intercal$, the control input is $u = T_c$ and the exogenous input vector is $\boldsymbol{v} = [e_y^d, \kappa, a_x]^\intercal$.

We discretize the continuous-time model in (5.5) by using a forward difference method at a fixed sampling time Δt_c. The discrete-time prediction model is compactly written as:

$$\boldsymbol{\xi}(k+1) = f_d\big(\boldsymbol{\xi}(k), u(k), \boldsymbol{v}(k)\big), \tag{5.6}$$

where k denotes the current discrete-time step such that $t = k\Delta t_c$.

5.2.2 Constraints

The main objective of the controller is to maintain safety of the EV by keeping the vehicle in the safe portion of the road while improving the lateral stability. Therefore, we define the following safety constraints imposed on the vehicle states.

5.2.2.1 Lateral position constraints

We compute the safety intervals between EV and TVs by defining a "safety box" around TVs as illustrated in Figure 5.5. If the TV is in the target lane (i.e., the lane that EV intents to travel), then the front and rear safety distances $\big(d_{f,j}(i)$ and $d_{r,j}(i)\big)$ of the j-th TV at prediction index i are calculated as:

$$d_{\mathrm{f,j}}(i) = d_0 + t_{\mathrm{gap}} \max(0, \Delta v_{te,j}(i)), \tag{5.7a}$$

$$d_{\mathrm{r,j}}(i) = d_0 - t_{\mathrm{gap}} \min(0, \Delta v_{te,j}(i)), \qquad i = 1 \ldots N, \tag{5.7b}$$

where d_0 is a fixed safety distance, $\Delta v_{te,j}(i) \triangleq v_{t,j}(i) - v_e(i)$ is the relative longitudinal velocity between j-th TV and EV at prediction index i, t_{gap} is the time constant which is a design parameter that relates the size of safety box to relative velocity.

In the proposed control framework, we use the TV model introduced in Section 3.6.1 to predict the future trajectories of TVs and calculate the safety boxes for each predicted positions of TV. The collision avoidance constraints are formulated as upper and lower bounds on the lateral position of the EV in the prediction horizon using (5.8). Here, we consider a highway with two lanes where the traffic flow is in the same direction (see Figure 5.5). The maximum allowable lateral distance towards rightmost road boundary $e_{y,\min}$ is calculated by Eq. (5.8a). On the other hand, the maximum allowable lateral distance towards the leftmost road boundary $e_{y,\max}$ depends on a simple rule. We denote the size of the safety box of j-th TV at prediction index i by $S_{t,j}(i) = [s_{t,j}(i) - d_{r,j}(i), s_{t,j}(i) + d_{f,j}(i)]$. If the longitudinal position of EV at prediction index i, denoted by $s_e(i)$, stays in the safety box (i.e. $s_e(i) \in S_{t,j}(i)$) then $e_{y,\max}$ is the left boundary of the rightmost lane. If there is no TV on the leftmost lane, hence $s_e(i) \notin S_{t,j}(i)$, then $e_{y,\max}$ is the left boundary of the leftmost lane.

$$e_{y,\min}(i) = 0.5(w_c - w_{lane}), \tag{5.8a}$$

$$e_{y,\max}(i) = \begin{cases} 0.5(w_{lane} - w_c) & \text{if } s_e(i) \in S_{t,j}(i), \\ 1.5w_{lane} - 0.5w_c & \text{otherwise}, \end{cases} \tag{5.8b}$$

where w_{lane} and w_c denote the width of lane and the EV.

We can express the time-varying safety constraints in prediction horizon N as:

$$e_{y,\min}(i) \leq e_y(i) \leq e_{y,\max}(i), \qquad i = 1 \ldots N. \tag{5.9}$$

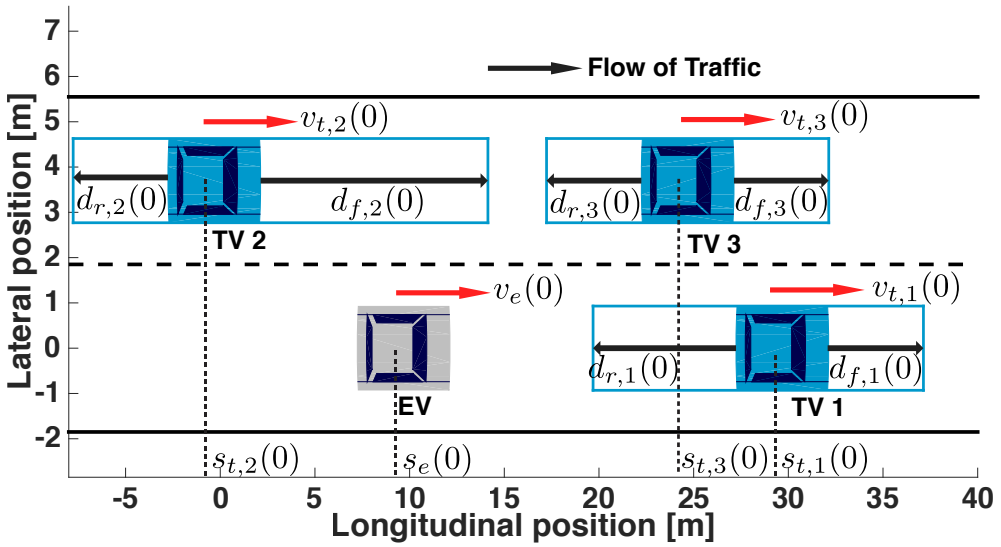


Figure 5.5 : The safety boxes of target vehicles traveling around ego vehicle.

Remark 9. The size of the safety box is associated to the conservativeness of the intervention by the controller as the safety constraints are calculated based on safety box.

5.2.2.2 Slip angle constraints

We also impose slip angle constraints on the vehicle dynamics in order to enforce the slip angles to stay in the linear region of tire forces.

$$\alpha_{\min} \leq \alpha_j(i) \leq \alpha_{\max}, \quad j \in \{f,r\}, \quad i = 1 \dots N, \tag{5.10}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ denote the upper and lower bounds on slip angle. $\alpha_{j,i}$ denotes the slip angle prediction of tire j at prediction index i.

The safety constraints in (5.9) - (5.10) are compactly written as:

$$h^{\xi}(\boldsymbol{\xi}_{i+1}, u_i) \leq \mathbf{0} \quad i = 0 \dots N-1, \tag{5.11}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

5.2.2.3 Control input constraints

We enforce the following magnitude and the rate constraints on the overlay torque (i.e., actuator input) T_c in order to have a smooth and adequate level of torque guidance considering the comfort of the driver:

$$T_{c,\min} \leq T_c(i) \leq T_{c,\max}, \tag{5.12a}$$

$$\Delta T_{c,\min} \leq \Delta T_c(i) \leq \Delta T_{c,\max}, \quad i = 0 \dots N-1, \tag{5.12b}$$

where $T_{c,j}$ and $\Delta T_{c,j}$, $j \in \{\min, \max\}$ denote the boundaries on the magnitude and rate of control input, respectively.

The control input constraints (5.12) could be compactly written as:

$$h^{u}(u_i, \Delta u_i) \leq \mathbf{0}, \quad i = 0 \dots N-1, \tag{5.13}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

5.2.3 Optimal control problem

The constrained finite time optimal control (CFTOC) problem in (5.14) is solved at each discrete time step k to compute an optimal control input sequence $\boldsymbol{u}_k$. Only the first control input is applied to the vehicle, and this optimization routine is repeated at the next time step using new state measurements as initial conditions.

$$\min_{\boldsymbol{u}_k, \boldsymbol{\varepsilon}} \sum_{i=0}^{N-1} \left(\|u_{i|k}\|_Q^2 + \|\Delta u_{i|k}\|_R^2 \right) + W\boldsymbol{\epsilon} \tag{5.14a}$$

$$\text{s.t. } \boldsymbol{\xi}_{i+1|k} = f_d(\boldsymbol{\xi}_{i|k}, u_{i|k}, \boldsymbol{v}_k), \tag{5.14b}$$

$$h^{\xi}(\boldsymbol{\xi}_{i+1|k}, u_{i|k}) \leq \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \geq \mathbf{0} \tag{5.14c}$$

$$h^{u}(u_{i|k}, \Delta u_{i|k}) \leq \mathbf{0}, \tag{5.14d}$$

$$i = 0 \ldots N-1$$

$$\boldsymbol{\xi}_{0|k} = \boldsymbol{\xi}(k), \qquad u_{-1|k} = u(k-1), \tag{5.14e}$$

where $\boldsymbol{\xi}_{i|k}$ denotes the i-th state prediction at time step k, obtained by applying the optimal control input sequence $\boldsymbol{u}_k = \{u_{0|k}, u_{1|k}, \ldots, u_{N-1|k}\}$ to the discrete-time dynamics (5.14b) starting from the measured state at the current time step (5.14e). The cost function in (5.14a) reflects control objectives such as utilizing smooth and minimal control action and tracking the desired lateral trajectory. We penalize the magnitude and rate of control input using corresponding weight matrices Q and R, respectively. The input constraints are given in (5.14d). We introduce slack variable vector $\boldsymbol{\varepsilon}$ for the safety constraints (5.14c) in order to keep the optimization problem feasible in case of constraint violations due to model mismatch. A large matrix W strongly penalizes constraint violation.

5.3 Simulation Environment and Results

In this section, we validate the performance of the *proposed controller* in a closed-loop simulation in Matlab/Simulink. The plant dynamics of the *Ego Vehicle* (EV) (i.e., “Vehicle Dynamics" and “Steering System" blocks in Figure 3.1) is simulated by using the four-wheel nonlinear dynamic model and the generic steering model presented in Section 3.2.1 and 3.5.2, respectively. The lateral tire forces are generated by the Fiala (brush) model in (3.20) and the longitudinal tire forces are

assumed to be zero (i.e., constant longitudinal velocity) because only lateral dynamics are considered. The parameters of the EV is given in Table B.1. The *Target Vehicles* (TVs) are simulated by using the model presented in Section 3.6.1. The parameters of the TV is given in Table 5.1.

Table 5.1 : Target vehicle model parameters.

Parameter	Value	Parameter	Value
acceleration exponent (a_{exp})	4	max. acceleration ($a_{\max}$)	1.50 m/s^2
desired speed (v^d)	35 m/s	comfort deceleration (a_c)	1.15 m/s^2
min. gap to PV (d_0)	5.0 m	time gap to PV (t_{gap})	1.5 s

The road is straight and has two lanes in each direction which has a width of $w_{lane} = 3.4$ m. Because of the straight road assumption, the curvature of the road is zero (i.e., $\kappa = 0$) for the finite time horizon. Both EV and TV is simulated at a sampling interval of $\Delta t_{tv} = 10^{-4}$ s.

We simulate a human driver steering torque behavior (i.e., “Driver" block in Figure 3.2) by using the MPC formulation introduced in Chapter 4 at a sampling interval of $\Delta t_d = 0.1$ s and a prediction horizon of $N = 20$. The parameters of the MPC-based driver steering controller is given in Table 5.2. The vehicle model parameters used in this formulation is also given in Table B.1.

Table 5.2 : The parameters of the MPC-based driver steering controller.

Parameter	Value	Parameter	Value
(Q_d, R_d, M)	(20, 5, 10^4)	S_d	**diag**(0,0,0,0,5,0,0,0)
$J_{d,p}$	0.064 kgm^2	$b_{d,p}$	0.56 Nm/rad/s
$k_{d,p}$	3.8 Nm/rad	$(e_{y,\min}, e_{y,\max})$	(-0.67, 4.07)
$(T_{d,\min}, T_{d,\max})$	(-6, 6) Nm	$(\Delta T_{d,\min}, \Delta T_{d,\max})$	(-12, 12) Nm/s

The proposed DSAS is based on MPC framework, introduced in Section 5.2.1 and the optimal control problem is solved at a sampling interval of $\Delta t_c = 0.1$ s. We assume that a various set of sensors are available to provide all necessary measurements to the controller since the control algorithm is based on full state feedback. The design parameters of the proposed DSAS are given in Table 5.3.

5.3.1 Blind spot intervention

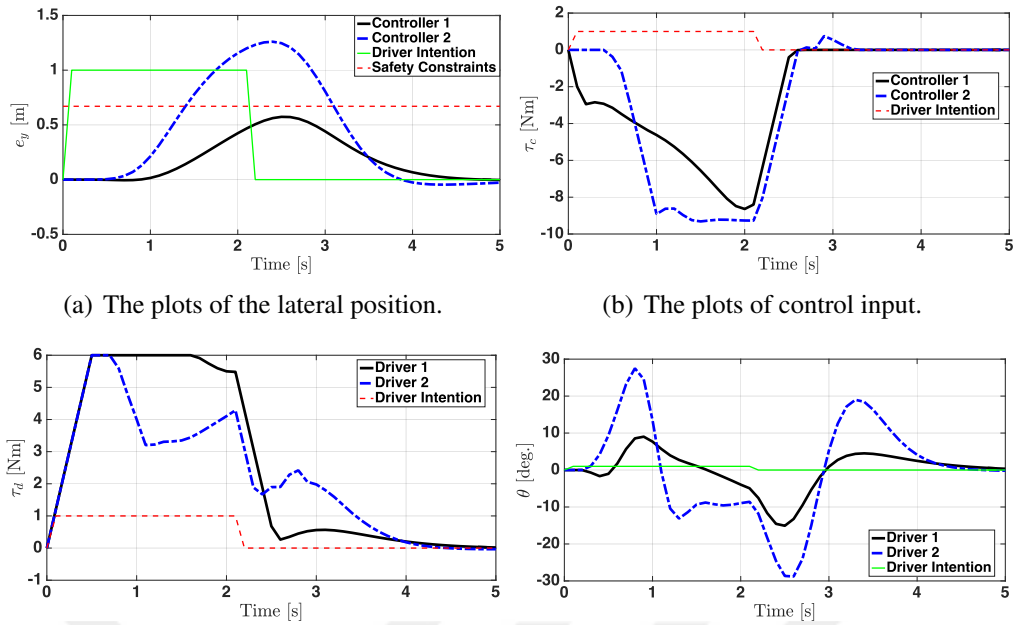
We consider a scenario where we simulate a distracted driver behavior by initiating a lane change maneuver when there is a fast approaching TV in the destination lane. Particularly, we focus on the conflict between the agents (i.e., the driver and the DSAS) during the torque intervention. In this situation both the agents are in an uncooperative state and both exert torque in opposite direction. This state could occur when the driver's actions could cause a hazardous situation and the driver does not respond to guidance forces.

Table 5.3 : The design parameters of the DSAS controller for the simulations.

Parameter	Value	Parameter	Value
(Q,R)	(40, 5)	M	10^4
$(T_{c,\min}, T_{c,\max})$	(-10, 10) Nm	$(\Delta T_{c,\min}, \Delta T_{c,\max})$	(-20, 20) Nm/s
$(\alpha_{\min}, \alpha_{\max})$	(-4, 4) deg.	K_d	4
K_y	0.32	K_ψ	12.10

In order to show the effectiveness of the proposed approach, we define two Driver-Vehicle-Environment (DVE) systems where each system has the same dynamics except the controller formulation. We compare the performance of the proposed controller ("Controller 1"), that is based on the driver-in-the-loop steering model with a controller ("Controller 2") that does not consider the steering torque behavior of the driver in its prediction model. In the following simulation, the EV travels at constant longitudinal speed of 25 m/s in the right lane and the TV travels at 35 m/s in the destination lane (i.e., the leftmost lane). The EV is initially positioned in front of the TV with a distance of 25 m.

The results of both DVE systems are illustrated in Figure 5.6 and the snapshots from the animation of the simulated DVE systems is given in Figure 5.7. In the first phase, we set $e_y^d(t) = 3.4$ m for $t \in [0.1, 2.1]$ s in order to initiate a lane change maneuver, as illustrated in Figure 5.6(a). When the driver intention is in high state, it corresponds to lane change intention of the driver. During this time interval, the assistance system intervenes since it is considered as a high risk lane change maneuver and we assume that the driver opposes the guidance torque by increasing the damping and stiffness of its arm muscles in order to generate more torque to restrict the motion of the hand



(a) The plots of the lateral position.

(b) The plots of control input.

(c) The plots of driver torque.

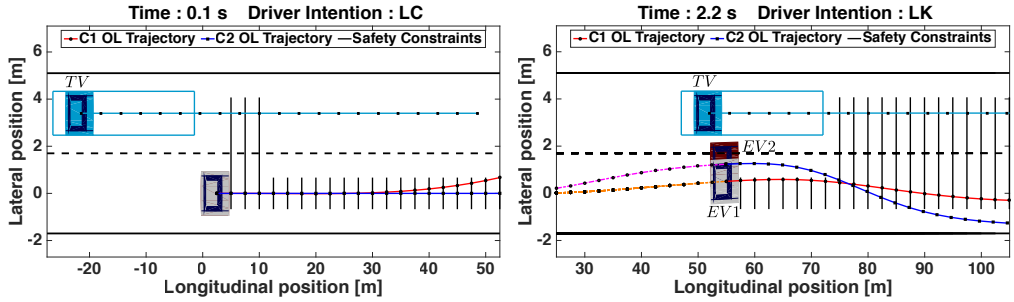
(d) The plots of hand wheel angles.

Figure 5.6 : The simulation results for both Driver-Vehicle-Environment systems in Blind Spot Intervention scenario.

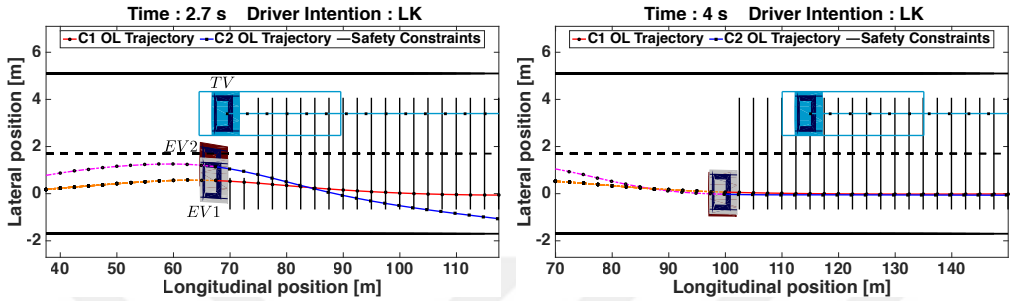
wheel. The high co-contraction level of the muscles are represented by the damping $b_{d,a}(t) = 0.036|T_c|$ [Nms/rad] and the stiffness $k_{d,a}(t) = 2.9|T_c|$ [Nm/rad]. These values were given with respect to the results of [4].

The driver torque plots given in Figure 5.6(c) show that both drivers apply maximum torque to perform the lane change while opposing the guidance forces illustrated in Figure 5.6(b). While “Driver 1" keeps the maximum effort to steer the vehicle, “Driver 2" decreases the torque as the hand wheel is around 30 deg. This is caused by the input and tracking cost terms in the MPC formulation. If the input cost is getting higher than the tracking cost then the controller calculates an optimal strategy to yield the lowest total cost.

The “Controller 1" intervenes earlier than “Controller 2" since an estimate of the driver intention (e.g., lane change/keep) is employed in the model. The proposed controller lets the driver to steer the vehicle close to the safety constraints without violating them. As the vehicle gets close to the boundary, the magnitude of torque intervention increases adequately to prevent any constraint violation. On the other hand, without using a driver model decreases the control performance since the controller could not exhibit an optimal behavior due to model mismatch between the prediction model and the actual dynamics. This results in constraint violation and high control input



(a) The snapshot of the animation at $t = 0.1$ s. (b) The snapshot of the animation at $t = 2.2$ s.



(c) The snapshot of the animation at $t = 2.7$ s. (d) The snapshot of the animation at $t = 4.0$ s.

Figure 5.7 : The figures show the instantaneous positions of EV and TV in the Blind Spot Intervention scenario.

utilization to recover the system back to safe operating region. Also the hand wheel angle plots in Figure 5.6(d) show that “Driver 1" could steer the hand wheel up to 30 deg. while performing lane change.

In the second phase we assume that driver accepts the torque intervention. That is achieved by setting $e_y^d(t) = 0$ m for $t > 2.1$ s.

Remark 10. In reality the drivers would act distinctively during the intervention (e.g., when to accept it), in this simulation we simply assume both drivers exhibit the same decision and execution functionalities.

The position of the EVs and the TV is illustrated in Figure 5.7(b). Driver decreases the impedance properties of its arms and transitions back to a low co-contraction level such in a normal driving case (i.e., $b_{d,a} = k_{d,a} = 0$). As the driver agrees with the system, the guidance torque vanishes gradually and the driver performs the necessary maneuver and steers back to the original lane as illustrated in Figures 5.7(c) and 5.7(d). We would like to notify the readers that we could only plot the safety box around the TV at the current time step. The safety boxes around the finite-time predictions of the TV are not considered for the sake of clarity of the figures. Although it looks like the

safety constraints are overlapping with the safety box, in fact one should compare each element of the safety constraints with the safety boxes around predicted TV positions.

These simulation results show that the proposed controller achieve the control objectives such as less control input utilization and satisfying the safety constraints. If the driver model is not considered in the control framework, the controller fails to utilize the required guidance torque on time and this causes a late intervention with higher magnitude. As a result, the controller is not able to satisfy the safety constraints because of the inaccurate dynamics used in the prediction model. This results show the importance of accurate modeling in the MPC framework in order to achieve a desired control performance.

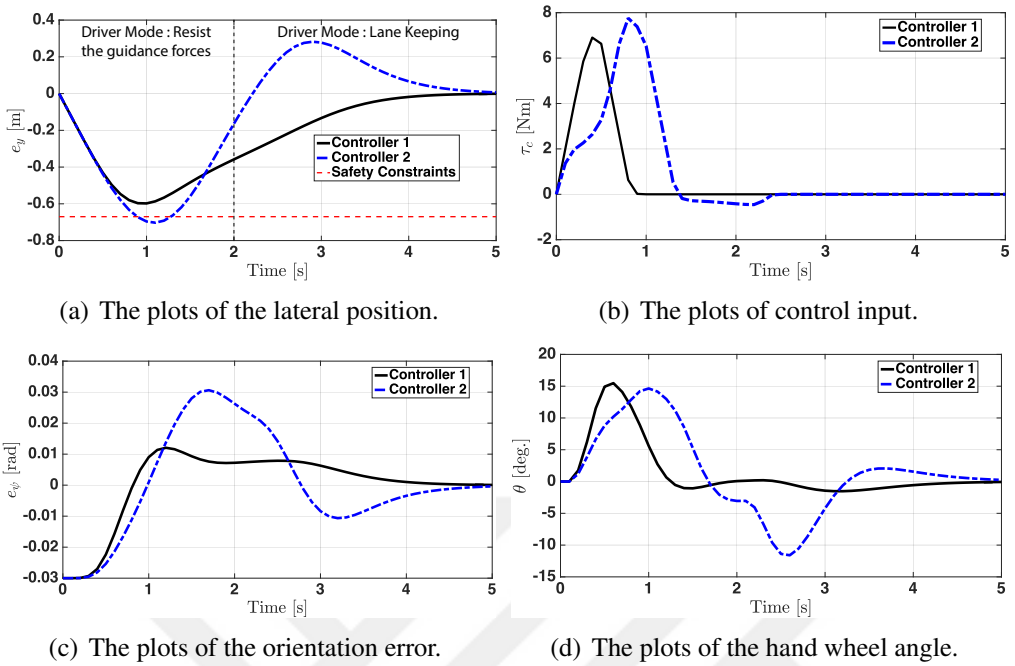
5.3.2 Road departure prevention

In this scenario, we consider a situation where the vehicle is traveling to the road boundary while the driver opposes the guidance forces passively by holding the hand wheel firmly and increasing the damping and stiffness of the arms. Therefore the total impedance of the coupled dynamics changes drastically when driver interacts with the guidance forces.

The simulation results are illustrated in Figure 5.8. The EV travels at 30 m/s and initially positioned at $e_y = 0$ m with $e_\psi = -0.03$ rad. The simulated driver is in impedance mode (see Section 4.3.1) between $t \in [0,2]$ s. This corresponds to a pure impedance behavior from the driver by opposing the motion of the hand wheel resulting from the applied guidance torque. After $t = 2$ s, the driver mode is switched to lane keeping mode to mimic a driver behavior shift from distracted to attentive.

We compare the results of the controllers which are introduced in Section 5.3.1. The lateral position and the heading error plots of both DVE systems are illustrated in Figure 5.8(a) and 5.8(c). It is seen that the proposed approach (i.e., “Controller 1") satisfies the safety constraints where the controller without the driver model (i.e., “Controller 2") violates them. The proposed approach also achieves a better response in terms of less overshoot and faster settling time in both lateral position and orientation error. The applied torque from the both controllers are shown in Figure 5.8(b) suggesting an earlier intervention with less control input utilization from the proposed approach. The torque intervention of “Controller 1" decays to zero around $t = 0.9$ s

which is 0.5 s earlier than “Controller 2". Finally we illustrate the hand wheel angle plots in Figure 5.8(d). It is seen that the resulting hand wheel behavior is also smooth.



(a) The plots of the lateral position.

(b) The plots of control input.

(c) The plots of the orientation error.

(d) The plots of the hand wheel angle.

Figure 5.8 : The simulation results for both controllers in Road Departure Prevention scenario.

5.4 Experimental Results

5.4.1 Experimental setup

Computations are performed on a dSPACE MicroAutoBox II embedded computer running at 900 MHz. Besides the built-in sensors of the vehicle, a production level forward-looking camera measures the lane boundaries, curvature and vehicle heading with respect to the lane. The steering actuator is an MDPS system which takes a torque request (T_c) as an input and generates an overlay torque by using the electric motor. The sensors, embedded computer and actuators communicate through a CAN bus. The online optimization problem is solved using the general purpose nonlinear solver NPSOL [80]. During the experiments, the parameters given in Table B.1 and Table 5.4 are used in the control framework. The nonlinear optimization problem presented in Section 5.2.3 is solved at a sampling interval of $\Delta t_c = 0.2$ s in real time. The prediction horizon is selected as $N = 10$ for computational reasons.

Assumption 9. *We introduce the following assumptions for the experiments.*

Table 5.4 : The design parameters of the controller for the experiments.

Parameter	Value	Parameter	Value
(Q,R)	(10, 10)	M	10^4
$(T_{c,\min}, T_{c,\max})$	(-5, 5) Nm	$(\Delta T_{c,\min}, \Delta T_{c,\max})$	(-10, 10) Nm/s
$(\alpha_{\min}, \alpha_{\max})$	(-4, 4) deg.	J_{eq}	0.1 kgm^2
b_{eq}	2.5 Nms/rad	k_{eq}	6.25 Nm/rad

1. *The mechanical impedance parameters (i.e., J_{eq}, b_{eq} and k_{eq}) used in the prediction model of the proposed controller is assumed to be constant.*

2. *The driver does not steer actively but only accepts/opposes the haptic forces in the hand wheel. Therefore the driver is assumed to be in Impedance Mode (see Section 4.3.1).*

3. *Due to safety measures, we use virtual vehicles to simulate the TVs around the EV instead of using actual vehicles.*

4. *The ego vehicle is in speed control mode therefore the longitudinal speed is constant throughout the experiments.*

5. *The driver does not change its neuromuscular system properties during the conflict.*

5.4.2 Blind spot intervention

The objective of the test is to evaluate the system's ability to prevent high risk lane changes. We simulated a TV around the test vehicle (next called EV). At time $t = 2s$, driver of the EV simulated an inattentive driver behavior by initiating a lane change maneuver while there is a fast TV in the target lane. The measured lateral position of the EV along with the open-loop (OL) predictions from the controller are illustrated in Fig. 5.9. The results show that the controller prevents the vehicle from departing the lane.

The predictive controller assessed the risk of collision by predicting the positions of the EV and TV and intervened only if it is safety critical. The corrective torque is illustrated in Figure 5.10. The corrective torque is zero when the predicted driver actions result in a safe trajectory. However, if the predicted trajectory violates the

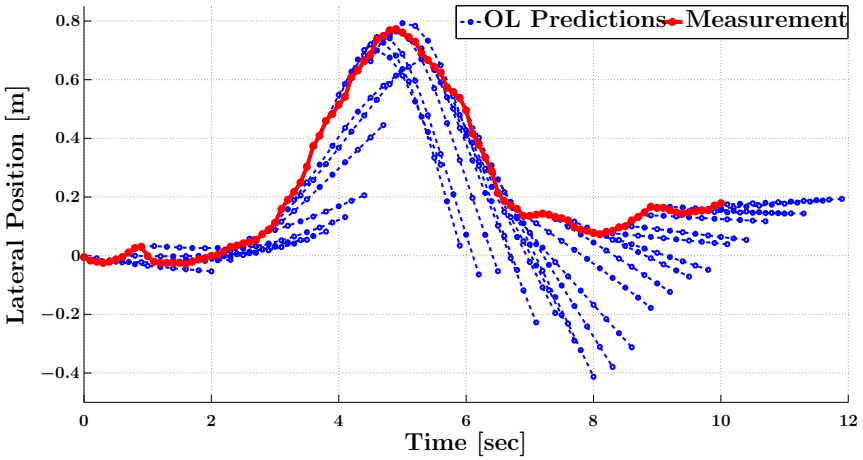


Figure 5.9 : The lateral position and the open-loop predictions of the EV during the experiment.

safety constraints, an intervention is triggered to guide the driver back to the safe region of the road.

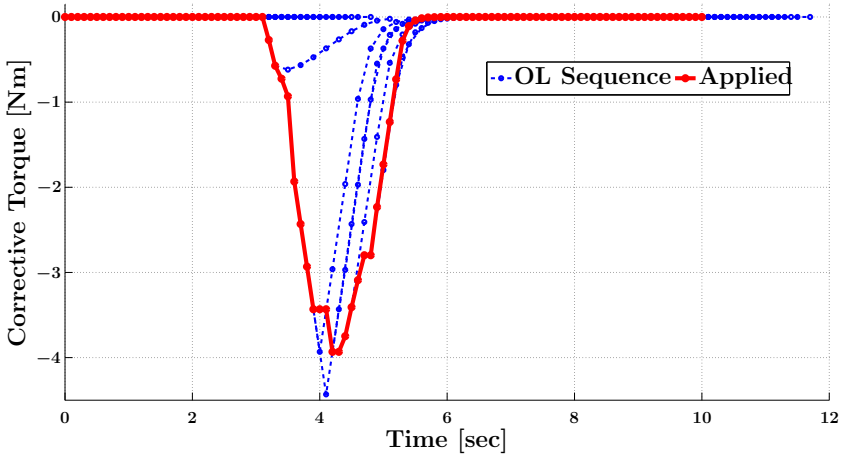


Figure 5.10 : The corrective torque sequence during the experiment.

At time $t = 3.2s$, the predictive controller started applying a clockwise (i.e., negative in magnitude) corrective torque to steer the vehicle to right. In Figure 5.11, we illustrate a snapshot of the experiment at $t = 3.8s$. It is observed that without intervention, the controller predicted the vehicle would leave the lane therefore violating the safety constraints. By applying the corrective torque, the controller updated its predictions to satisfy the safety constraints.

Starting from time $t = 3.8s$, driver started to steer in clockwise direction as a result of corrective torque. The steering wheel angle measurements and the open-loop

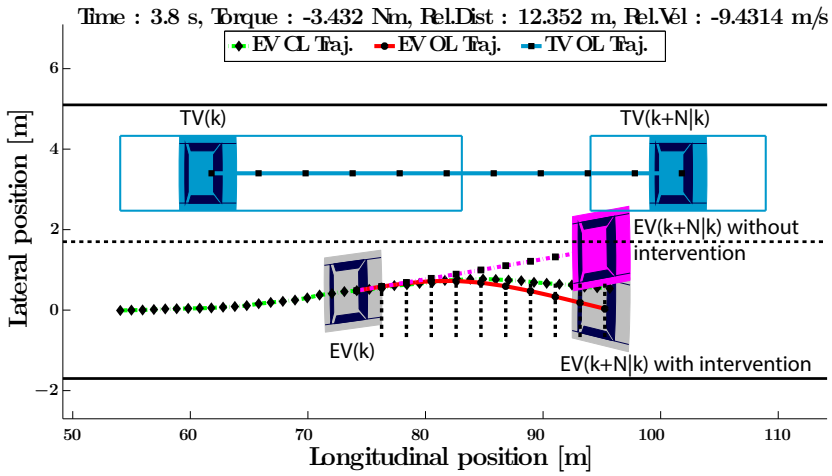


Figure 5.11 : The snapshot at time $t = 3.8s$. This figure shows the open-loop (OL) predictions of EV and TV along with the actual trajectories.

predictions are given in Figure 5.12. The corrective torque decayed to zero when driver steered back to safety portion of the road. Because of the rate constraints, the corrective torque is applied gradually which provides a smooth transition instead of rapid changes in corrective torque dynamics.

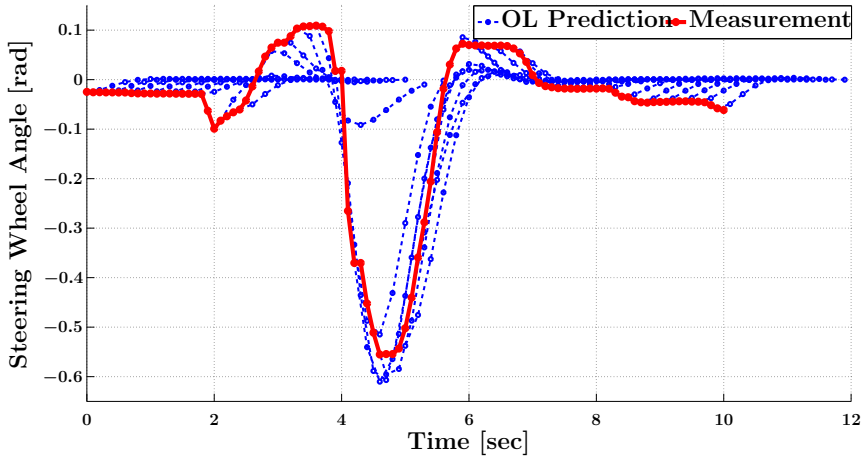


Figure 5.12 : The steering wheel angle of EV during the experiment.

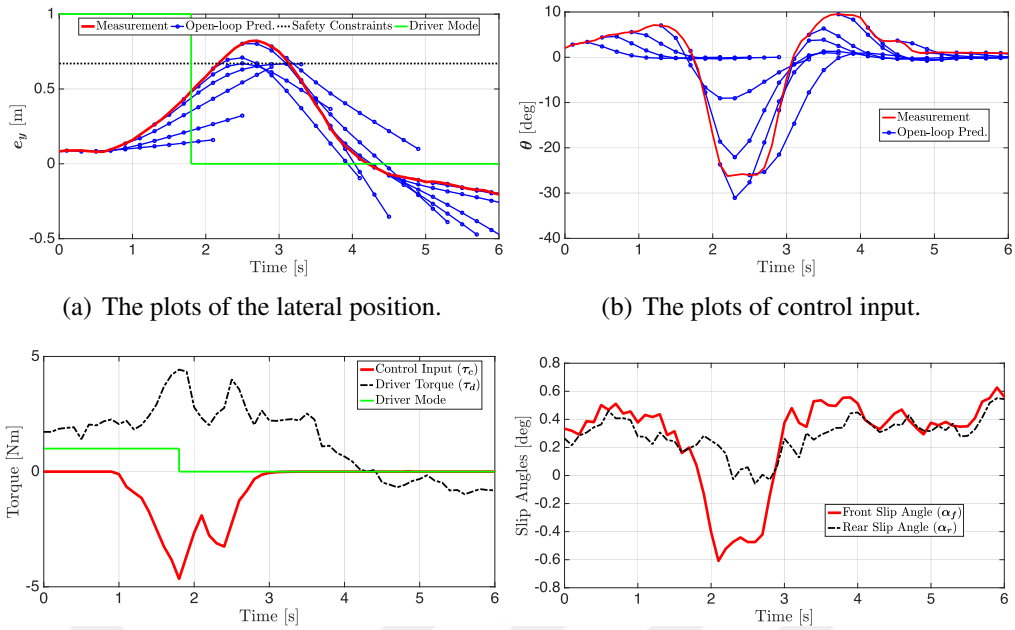
5.4.3 Lane departure prevention

In this section, we discuss the experiment results of the proposed DSAS in a lane departure prevention scenario. We ask the driver to perform a distracted steering behavior (e.g., not looking to the road) while griping the hand wheel. As the vehicle approaches to the lane boundary, the assistance system intervenes by applying a guidance torque in the opposite direction to prevent the lane departure. We conduct

two experiments with distinct driver behavior while the test vehicle is traveling at a constant longitudinal speed of 10 m/s. For the first experiment, we ask the driver to grip the hand wheel, keep the arms in the relax position and behave naturally to the torque intervention. On the other hand for the second experiment, we ask the driver to grip the hand wheel firmly and oppose to the torque intervention by just resisting the motion of the hand wheel for couple of seconds. This behavior corresponds to the impedance mode of the driver.

The results of the first experiment is illustrated in Figure 5.13. The controller intervenes around $t = 1$ s when the vehicle approaches to the safety constraints as seen from the Figure 5.13(c). The high level of driver mode, shown in Figure 5.13(a), indicates that the driver is in impedance mode and resists the torque from the system for about 1 second and lets the guidance forces to steer the vehicle back to the lane centerline after $t = 1.9$ s . There is a fluctuation in the guidance torque that is caused by the constraint violation. But as the predictions satisfy the constraints, the guidance torque decays to zero around $t = 3$ s and the driver returns to the safety portion of the road. It is also seen that the slip angles stay in the linear region of tire forces in Figure 5.13(d). The open-loop predictions of the hand wheel angle is illustrated in Figure 5.13(b). It is seen that the predictions approach to zero at the end of the horizon because of the spring term in the model. Since we assume that the driver is in impedance mode for the experiments, the model fails to capture the steering dynamics when the driver actively steers. The maximum constraint violation is around 0.15 m and it is attributed to the inaccurate model of the process used in the controller. The unmodeled time delays of the process might also affect the control performance in real time.

The results of the second experiment is illustrated in Figure 5.14. The controller intervenes around $t = 1$ s when the vehicle approaches to the safety constraints as seen from the Figure 5.14(c). The driver opposes as much as possible to the torque intervention while the vehicle starts violating the safety constraints (see Figure 5.14(a)). It is seen that the system is able to steer the hand wheel up to -15 deg while the controller applies maximum torque of 5 Nm between $t \in [2.4, 4.1]$ s. If we assume the derivatives of hand wheel angle are zero for this time interval, then the equivalent stiffness of the actual system could be found as $k_{eq}^{actual} = 19.1$ Nm/s



(a) The plots of the lateral position.

(b) The plots of control input.

(c) The plots of the orientation error.

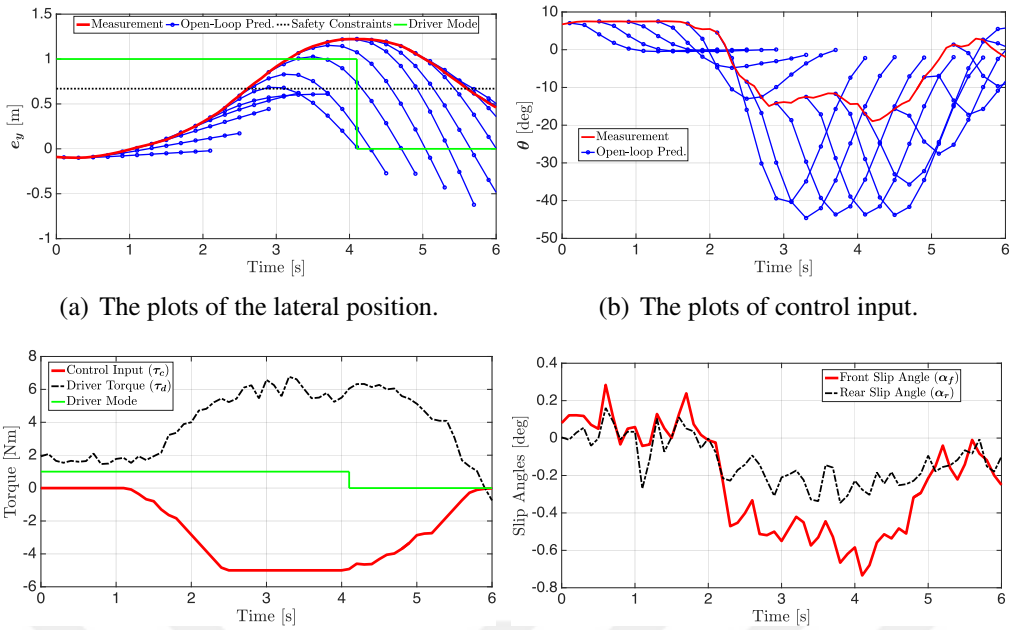
(d) The plots of the hand wheel angle.

Figure 5.13 : The results of first experiment with compliant driver behavior.

which shows the mismatch in the prediction model as we set $k_{eq} = 6.25$ Nm/s. After $t = 4.1$ s, the guidance torque gradually decays to zero as the vehicle steers back to the original lane. It is also seen that the slip angles stay in the linear region of tire forces in Figure 5.14(d). We would like to point out the open-loop predictions of the hand wheel angle as illustrated in Figure 5.14(b). This plot clearly show the inaccurate predictions caused by the mismatch between the prediction model and the actual dynamics. The maximum constraint violation is around 0.55 m and it is attributed to inaccurate modeling of the process. We also think that the magnitude of the guidance torque might be small for a human driver to override the system's actions easily. The constraint violation might be prevented if the system could apply a higher magnitude of intervention such as ± 8 Nm.

5.5 Discussions and Conclusions

We propose a predictive, human-in-the-loop control framework for preventing hazardous situations caused by inattentive driver behavior such as blind spot collisions and lane departures. We focus on modeling the steering dynamics such that we could use the native input of the steering actuator (i.e., overlay torque) as a direct optimization variable in predictive controller. This has the following advantages:



(a) The plots of the lateral position.

(b) The plots of control input.

(c) The plots of the orientation error.

(d) The plots of the hand wheel angle.

Figure 5.14 : The results of second experiment with stiff driver behavior.

- Compared to an optimal steering angle approach, a low-level controller is no longer needed to calculate the torque input from a desired corrective steering angle.
- The actuator limits can be systematically handled in the predictive controller.

As both the driver and controller act on the hand wheel, it is important to account for the reaction of the driver. Our approach enables this by embedding the driver torque dynamics into the steering system dynamics and formulating a driver-in-the-loop steering model that is employed in the control framework.

We propose a simulation setup which uses an MPC framework for simulating the steering behavior of the driver which is introduced in Chapter 4. The results show that the performance of the controller is depended on the accuracy of the model that is used by the predictive controller. We implement the proposed control framework on our test vehicle and validate the performance with a human driver. The experimental results demonstrate the effectiveness of the controller in preventing high risk lane change maneuvers and road departures. However, it is observed that the parameters of the driver-in-the-loop steering model is subject to change with respect to driver's interaction with feedback forces. In order to enhance the control performance and guidance, these parameters should be estimated in an online fashion. The results show that mismatch between the prediction model and the actual dynamics result in

poor prediction performance and therefore poor control quality. This motivates us to integrate a recursive online parameter identification algorithm to track the time-varying parameters of the driver-in-the-loop steering dynamics in the next chapter.

6. IDENTIFICATION OF DRIVER-STEERING SYSTEM DYNAMICS

6.1 Introduction

During a physical interaction, the driver can change these mechanical impedance properties (i.e., damping and stiffness) with respect to acceptance of the feedback forces. These properties could be increased/decreased in order to resist/give way to the feedback forces when these properties are compared to the relax state of human neuromuscular properties. The model mismatch between the system and the human driver causes inaccurate state predictions which prevents the system to optimize the amount of feedback forces. These time-varying changes should also be considered in the control framework in order to increase the performance and driver's acceptance. This chapter investigates the identification of driver-steering system dynamics (i.e., the mechanical impedance properties) for the steering task under torque disturbance.

In the literature, there are studies about the identification of the mechanical impedance properties of the driver's arms while holding the hand wheel with respect to the open-loop test input (i.e., torque disturbance) in [2, 4, 63, 73, 81].

The authors in [4] use a hand wheel that is integrated with an electric motor as the interface and an experimental study is conducted with eight participants. The objective of this study is to investigate the effect of the NMS of the driver's arms against a torque disturbance while the driver is holding the hand wheel. The experimental procedure that is presented in [4] is illustrated in Figure 6.1.

The participants are asked to hold the hand wheel in three different conditions while a torque signal is actively exciting the system. A single-degree of freedom mass-spring-damper model is chosen and the parameters of this model is identified using frequency domain techniques. It is found that the stiffness and damping of the arms increase when opposing to the torque disturbance.

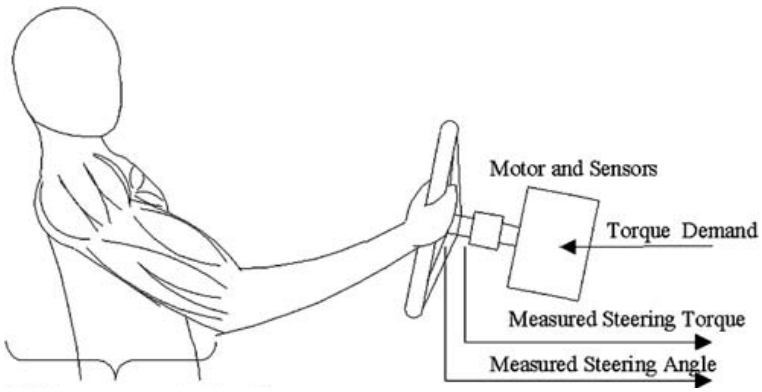


Figure 6.1 : The experimental procedure that is presented in [4].

A fixed-base driving simulator with an actuated hand wheel is used in order to quantify the adaptability of NMS as a function of task and hand placement [2]. The disturbance torque is selected to be a randomized multi-sine signal. It is concluded that identification of NMS properties is useful to understand the driver's response to feedback forces, and it may be a basis to design these forces without involving trial-and-error tuning.

On the other hand, a instrumented test vehicle is used in [81] for identifying the driver's arms' admittance during steering. In this study, the participants are asked to drive the vehicle in a designated route with different longitudinal speeds while an open loop torque disturbance is applied on the hand wheel. The frequency response function is identified from the measurements of hand wheel angle and driver's torque.

Another study with a driving simulator shows the results of an experimental study where the human mechanical impedance properties are identified according to the limb posture (i.e., position of hand wheel) and muscle co-contraction (i.e., steering torque). Again a mass-spring-damper model is used for fitting the parameters.

Finally, a recursive method for identifying the NMS properties during steering of the vehicle is proposed in [73]. The authors discuss the previous studies for using a time invariant models since these properties could vary depending on several factors such as position, activation level and etc. A time domain identification approach is preferred over frequency domain approaches. A recursive least squares algorithm is used in order to estimate the parameters of a linear model.

In this chapter, our objective is to use a recursive least squares algorithm in order to estimate the parameters of a linear regression model that captures the input-output

relations of the steering dynamics while the driver is holding the hand wheel. Specifically, we focus on the identification of the mechanical impedance properties of the driver-in-the-loop steering model (i.e., J_{eq}, b_{eq}, k_{eq}) in (5.4) online, since they vary as the driver interacts with the guidance torque of the assistance system. In order to enhance the control performance of the proposed driver assistance system, it is important to update the parameters while the controller is operating. We also compare the performance of two derived regression model under different driver behaviors.

6.2 Parameter Identification Architecture

We use the following assumption related to the torque behavior of the driver.

Assumption 10. *During the parameter identification, we assume that the driver only acts like an impedance source while holding the steering wheel. Therefore, the driver does not apply torque to steer actively but the driver increases/decreases the total impedance of the system seen from the control input.*

The effect of driver's response to the overall system's dynamics is illustrated in Figure 6.2. The driver can increase the overall impedance of the transfer function (from input T_c to output θ) by using the muscles. On the other hand, the driver can also decrease the overall impedance by giving way to the torque input. Because of this time-varying interaction between the driver and the feedback forces, we need an algorithm which is able to track these while the system is operating.

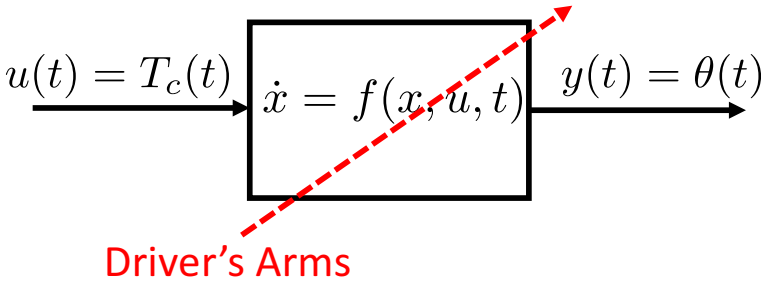


Figure 6.2 : The driver's response to the feedback forces changes the overall system's dynamics.

The parameter identification architecture is illustrated in Figure 6.3 where the input of the MDPS system is the overlay torque $u_k = T_c(k)$ and the output of the system is denoted by y_k and k denotes the current discrete-time step (i.e., $t = k\Delta t$), Δt is the fixed sampling period.

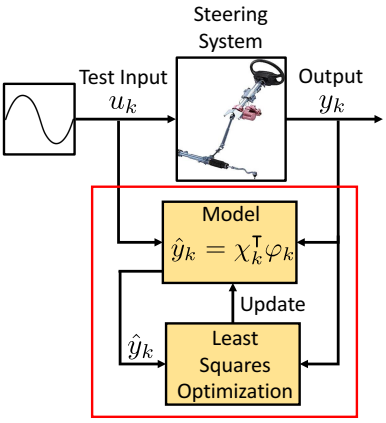


Figure 6.3 : System identification architecture.

6.3 Linear Regression Model

We propose the following regression models for the parameter identification. These models are linear with respect to the parameters therefore we can use linear parameter identification methods such as recursive least squares method.

6.3.1 Proposed regression model - I

Let $\theta_h(k) = \theta_k$. We form our linear regression model based on discretization of the continuous-time dynamics given as:

$$J_{eq}(t)\ddot{\theta} = -b_{eq}(t)\dot{\theta} - k_{eq}(t)\theta + T_c \tag{6.1}$$

where $J_{eq}(t) = J_h + J_d(t)$, $b_{eq}(t) = b_h + b_d(t) + b_r(t)$, $k_{eq}(t) = k_d(t) + k_r(t)$ describe the equivalent inertia, damping and stiffness of the driver-in-the-loop steering model.

The first-order and the second-order backward difference approximations to the derivative at time $t = k\Delta t$ are

$$\frac{d\theta}{dt} \cong \frac{\theta_k - \theta_{k-1}}{\Delta t}, \quad \frac{d^2\theta}{dt^2} \cong \frac{\theta_k - 2\theta_{k-1} + \theta_{k-2}}{(\Delta t)^2}. \tag{6.2}$$

By using the approximations in (6.2) and defining $u_k = T_c(k)$, we can obtain the discrete-time model of the continuous-time dynamics as:

$$\theta_k = \varphi_{1,k}\ \theta_{k-1} + \varphi_{2,k}\ \theta_{k-2} + \varphi_{3,k}\ u_{k-1}, \tag{6.3}$$

where the true discrete-time model parameters φ_i are functions of the mechanical impedance properties $\big($i.e., $\varphi_{i,k} = g_i(J_{eq}(k),\ b_{eq}(k),\ k_{eq}(k), k)\big)$ and they are given as:

$$\varphi_1 = \frac{2J_{eq} + \Delta t b_{eq}}{den}, \quad \varphi_2 = -\frac{J_{eq}}{den}, \quad \varphi_3 = \frac{(\Delta t)^2}{den}, \tag{6.4}$$

where the denominator is $den = J_{eq} + \Delta t b_{eq} + (\Delta t)^2 k_{eq}$.

The proposed regression model is given by:

$$\hat{y}_k = \hat{\varphi}_{0,k-1}\, x_{0,k} + \hat{\varphi}_{1,k-1}\, x_{1,k} + \hat{\varphi}_{2,k-1}\, x_{2,k} + \hat{\varphi}_{3,k-1}\, x_{3,k}, \tag{6.5}$$

where we define $y_k = \theta(k)$ and a bias term φ_0 is integrated in order to account for unmodeled input disturbances.

The proposed regression model is compactly written as $\hat{y}_k = \boldsymbol{x}_k^{\intercal}\hat{\boldsymbol{\varphi}}_{k-1}$ where the regression vector is given as $\boldsymbol{x}_k = [1,\ y_{k-1},\ y_{k-2},\ u_{k-1}]$.

6.3.2 Proposed regression model - II

We form our linear regression model based on discretization of the continuous-time dynamics given as:

$$J_{eq}(t)\dot{\omega} = -b_{eq}(t)\omega - k_{eq}(t)\theta + T_c, \tag{6.6}$$

where we define the angular velocity of the hand wheel as $\omega = \dot{\theta}$.

By using the approximations in (6.2) and defining $u_k = T_c(k)$, we can obtain the discrete-time model of the continuous-time dynamics as:

$$\omega_k = \hat{\varphi}_{1,k}\ \omega_{k-1} + \hat{\varphi}_{2,k}\ \theta_{k-1} + \hat{\varphi}_{3,k}\ u_{k-1}, \tag{6.7}$$

where the true discrete-time model parameters φ_i are functions of the mechanical impedance properties $\big($i.e., $\varphi_{i,k} = g_i(J_{eq}(k),\ b_{eq}(k),\ k_{eq}(k), k)\big)$ and they are given as:

$$\varphi_1 = \frac{J_{eq} - b_{eq}\Delta t}{J_{eq}}, \quad \varphi_2 = -\frac{k_{eq}\Delta t}{J_{eq}}, \quad \varphi_3 = \frac{\Delta t}{J_{eq}}. \tag{6.8}$$

The proposed regression model is given by:

$$\hat{y}_k = \hat{\varphi}_{0,k-1}\, x_{0,k} + \hat{\varphi}_{1,k-1}\, x_{1,k} + \hat{\varphi}_{2,k-1}\, x_{2,k} + \hat{\varphi}_{3,k-1}\, x_{3,k}, \tag{6.9}$$

where we define $y_k = \omega(k)$ and a bias term $\hat{\varphi}_0$ is integrated in order to account for unmodeled input disturbances. The proposed regression model is compactly written as $\hat{y}_k = \boldsymbol{x}_k^{\intercal}\hat{\boldsymbol{\varphi}}_{k-1}$ where the regression vector is given as $\boldsymbol{x}_k = [1,\ \omega_{k-1},\ \theta_{k-1},\ u_{k-1}]$.

6.4 Exponential Forgetting and Reseting Algorithm

“Exponential Forgetting and Resetting Algorithm" (EFRA) is proposed in [82]. EFRA bounds the trace of the covariance matrix from above and below in order to maintain the stability of the algorithm (e.g., preventing covariance blow-up) while preserving the tracking ability (i.e., preventing zero gain). The EFRA algorithm step are given as:

$$\hat{y}_k = \boldsymbol{x}_k^\intercal \hat{\boldsymbol{\varphi}}_{k-1} \tag{6.10a}$$

$$K_k = \frac{P_{k-1}\boldsymbol{x}_k}{\lambda + \boldsymbol{x}_k^\intercal P_{k-1}\boldsymbol{x}_k} \tag{6.10b}$$

$$\hat{\boldsymbol{\varphi}}_k = \hat{\boldsymbol{\varphi}}_{k-1} + \alpha K_k\big(y_k - \hat{y}_k\big), \tag{6.10c}$$

$$P_k = \frac{1}{\lambda}\big(I - K_k\boldsymbol{x}_k^\intercal\big)P_{k-1} + \beta I - \gamma P_{k-1}^2, \tag{6.10d}$$

where the recursive algorithm starts from an initial guess of P_0 and $\hat{\boldsymbol{\varphi}}_0$. At each time step, when a new measurement y_k, and therefore a new regression vector $\boldsymbol{x}_k$, is acquired, the filter gain K_k is calculated in (A.12b). P_k and $\hat{\boldsymbol{\varphi}}_k$ are updated in (A.12c)-(A.12d) where λ is the forgetting factor and the filter gain is denoted by α. The parameters β and γ are used for covariance resetting and normalization.

6.5 Experimental Procedure

We perform experiments with our test vehicle having an MDPS system which is introduced in Section 3.5.1. The experiments are conducted in Hyundai-Kia California Proving Grounds. We request different overlay torque waveforms (e.g., step and sine) from MDPS control logic where the overlay torque is generated by the electric motor of MDPS system. Since the overlay torque (control) input T_c is not calculated by a feedback controller, it is called as the open-loop test input. This section considers the system identification experiments where the open-loop test input is the source of motion for the steering dynamics.

We consider the following three cases to identify the mechanical impedance properties of the driver-in-the-loop steering model where the overlay torque (control) input is the source of motion.

Case 1 - Hands Off: The aim of this case is to identify the response of the steering system to the control input T_c in order to model the steering dynamics for autonomous driving purposes (i.e., $T_d = 0$).

Case 2 - Compliant Driver: We ask the driver to hold the hand wheel, without actively steering (i.e., $T_{d,s} = 0$), and let the control input T_c to steer the vehicle.

Case 3 - Stiff Driver: The driver holds the hand wheel firmly and resists the motion by increasing the impedance through muscle activation without actively steering (i.e., $T_{d,s} = 0$). These test cases are illustrated in Figure 6.4.

Figure 6.4 : The snapshot from the experimental procedure.

We conducted various experiments and logged the measurements at a sampling frequency of $f_s = 100$ Hz. The RLS algorithms for system identification is executed at at 10 Hz.

6.5.1 Dataset 1

We performed experiments for the proposed test cases at a constant longitudinal velocity of 15 m/s. We commanded an overlay torque sine sweep reference, $T_c(t) = 3\sin(2\pi f_c t)$ Nm where $f_c = [0.2, 0.5]$ Hz, to MDPS control logic. The input-output signal measurements of *Dataset 1* are illustrated in Fig. 6.5 (e.g., The hand wheel angle measurement of *Case i* is entitled as *Output i*).

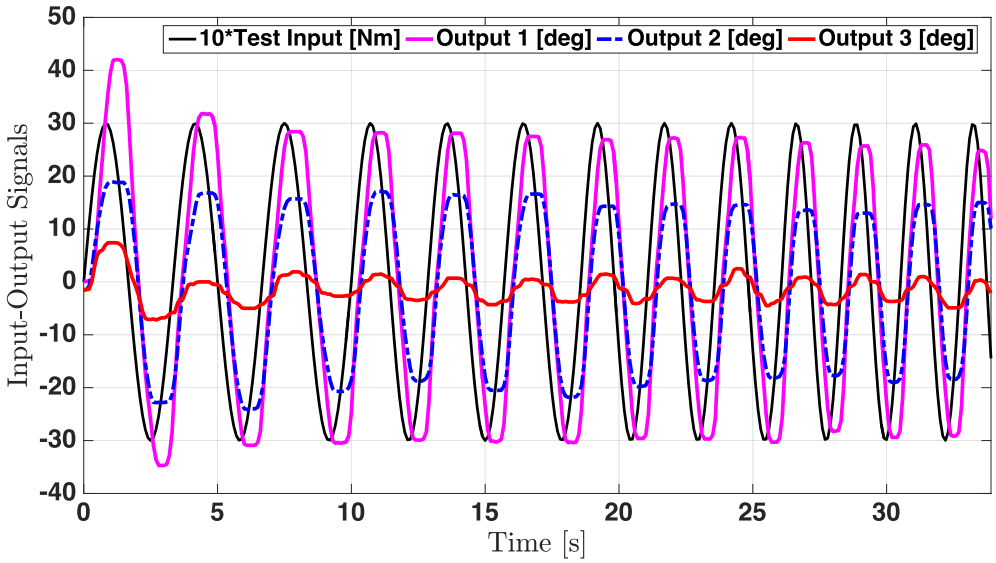
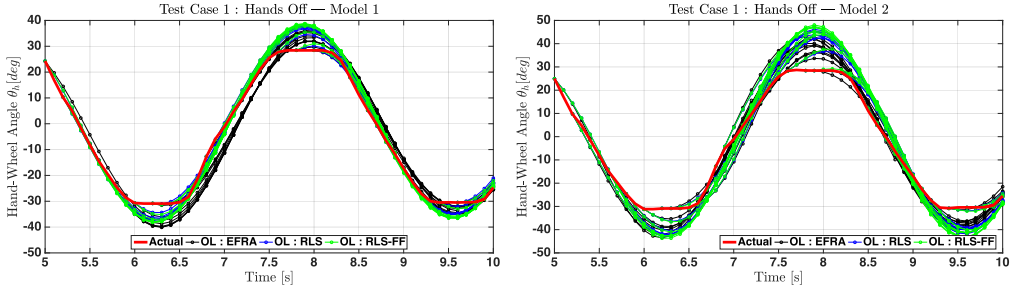


Figure 6.5 : The input-output signal measurements of *Dataset 1* for the proposed test cases.

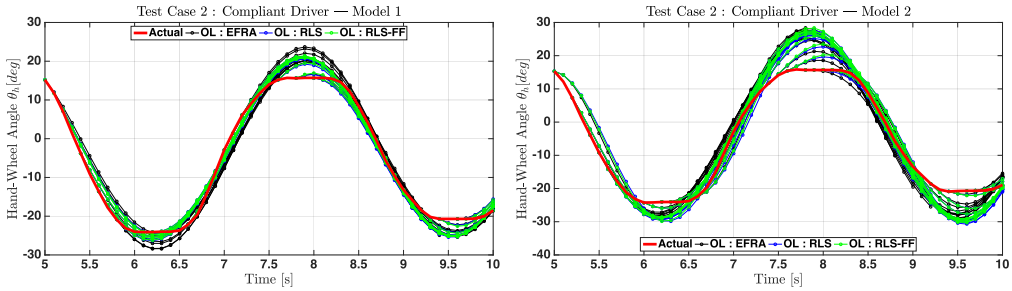
One of the objective of this section is to compare the performances of recursive algorithms, which are presented in Appendix A, by using the regression models introduced in Section 6.3 and to consider the best model and the algorithm to be used in the controller. We compare three algorithms, namely RLS given in Eq. (A.9), RLS-FF given in Eq. (A.11) and EFRA given in Eq. (A.12). We also compare two linear models where *Model 1* is given in Eq. (6.5) and *Model 2* is given in Eq. (6.9). The initial conditions and the parameters are set as follows. $\hat{\varphi}(0) = \mathbf{0}$, $\lambda = 0.98$, $P(0) = 100$ and $\gamma = \beta = 0.005$.



(a) (b)

Figure 6.6 : The open-loop (OL) predictions are calculated by using *Model 1* and *Model 2* for *Case 1*.

We illustrate the open-loop (OL) predictions of hand wheel angle which are calculated by using *Model 1* and *Model 2* for *Case 1*, *Case 2*, and *Case 3* in Figure 6.6, Figure 6.7, and Figure 6.8, respectively. The figures show that using *Model 1* provides better predictions in the finite horizon.



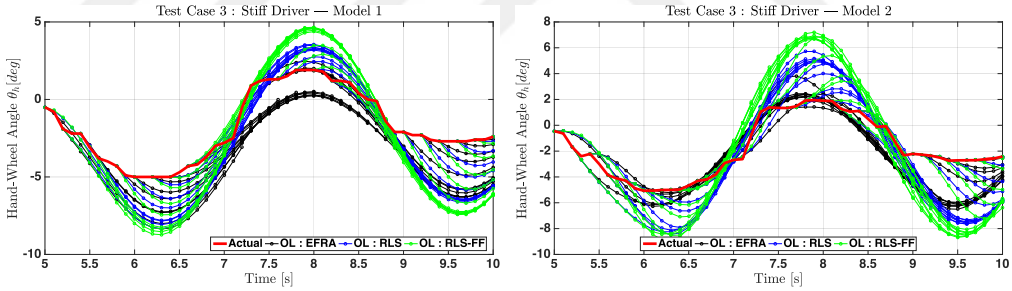
(a) (b)

Figure 6.7 : The open-loop (OL) predictions are calculated by using *Model 1* and *Model 2* for *Case 2*.

In order to compare the prediction performance of the regression models, we calculate the mean square error (MSE) of hand wheel angle predictions by using:

$$MSE(j) = \frac{1}{M}\sum_{k=1}^{M}(y_{j+k} - \hat{y}_{j|k})^2, \qquad j = 1 \ldots N, \tag{6.11}$$

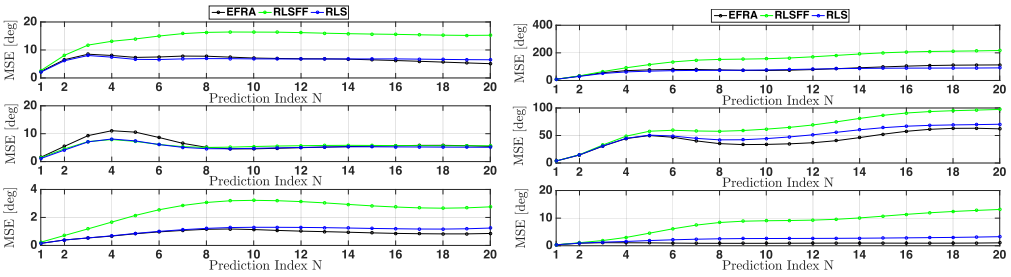
where N and M denote the prediction horizon and the size of the data, respectively. y_{j+k} denotes the output value at time step $k+j$. $\hat{y}_{j|k}$ denotes the output prediction at index j starting from current time step k which is calculated by using the estimated parameters $\hat{\varphi}_k$ along with the future control inputs.



(a) (b)

Figure 6.8 : The open-loop (OL) predictions are calculated by using *Model 1* and *Model 2* for *Case 3*.

We illustrate the prediction performance of each regression model in terms of MSE in Fig. 6.9. It is clear that *Model 1* is more accurate than *Model 2* in terms of prediction quality. Both *EFRA* and *RLS* methods show similar error characteristics with *Model 1*. On the contrary *RLS-FF* method shows a poor performance in *Case 1*. This is related to the covariance matrix which converges to zero for the *RLS-FF* method. For this reason the filter gain also converges to zero. The traces of the covariance matrix P_k is illustrated in Figure 6.11. It is seen that the trace of P_k for the *RLS-FF* method converges to zero for all the three cases. This might degrade the tracking performance of the recursive algorithm.



(a) (b)

Figure 6.9 : The mean squared error (MSE) values of OL predictions which are calculated by both *Model 1* and *Model 2* for all the cases.

In order to comprehend the performance of parameter identification, we show the results of mechanical impedance values which are calculated by solving J_{eq}, b_{eq} and k_{eq} from estimated discrete-time parameters φ_i of *EFRA* method by using *Model 1* in Figure 6.10. As expected, the mechanical impedance properties have the lowest impedance values in *Case 1 - Hands Off* and have the highest impedance values in *Case 3 - Stiff Driver*. Therefore the results show consistency with the cases and validates the effectiveness of the recursive parameter identification algorithm with the proposed discrete-time linear model.

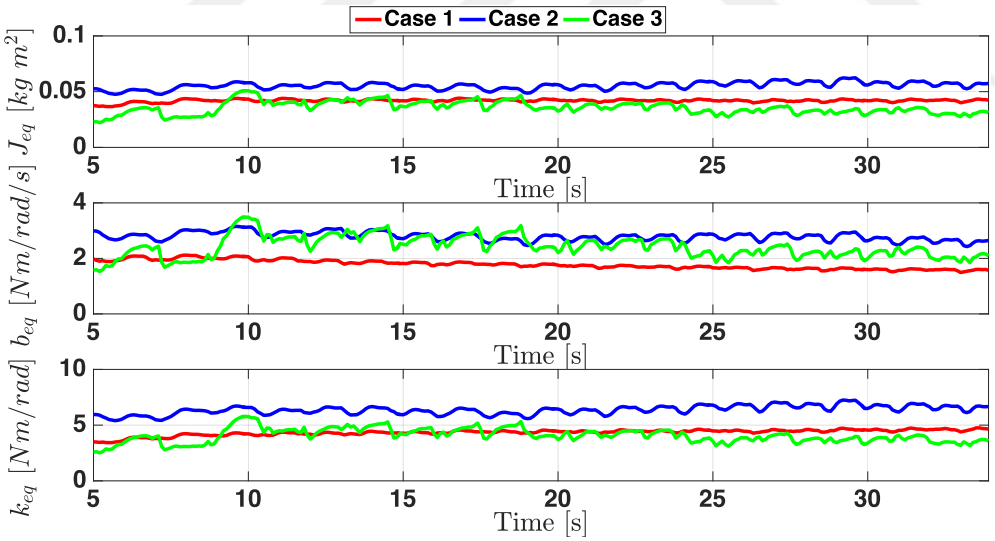


Figure 6.10 : The calculated mechanical impedance properties from the identified parameters of *Model 1* by using the algorithm EFRA. The plots of *Case 3* are scaled down by a factor of 10.

The average values for the estimated inertia, damping and stiffness of the driver-steering system is illustrated in Table 6.1. The identification results are consistent with respect to the test cases. However, since the real impedance values are unknown we can not justify the correctness of the numeric values.

Table 6.1 : Estimated mechanical impedance properties for dataset 1.

Test Case	Inertia [kgm^2]	Damping [Nm/rad/s]	Stiffness [Nm/rad]
Hands Off	0.03	2	4
Compliant Driver	0.05	3.5	6
Stiff Driver	0.3	25	30

6.5.2 Dataset 2

The dataset introduced in previous section includes three different cases which are consistent from beginning to end (i.e., the behavior of the driver is the same throughout the experiment). However, we would like to experiment on the situations where the behavior of the driver changes with respect to time. For example the driver can take its hands off and then complies/resists the motion of the hand wheel afterwards.

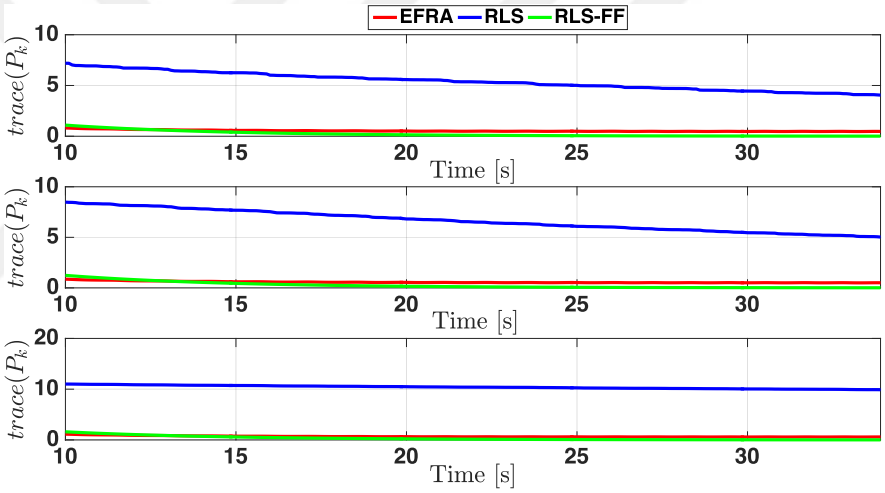


Figure 6.11 : The trace of the covariance matrix for the three algorithms calculated for three cases.

We performed an experiment at a constant longitudinal velocity of 10 m/s while commanding an overlay torque sine sweep reference, $T_c(t) = 3\sin(2\pi f_c t)$ Nm where $f_c = [0.2, 0.5]$ Hz, to MDPS control logic. The input and output signals of *Dataset 2* is illustrated in Figure 6.12. It is seen that the driver interacted with the haptic forces at different levels. For example at the time interval $t \in [0, 10]s$, the driver was compliant with the haptic forces but at $t \in [24, 32]s$ the driver opposed the motion and etc.

The mechanical impedance properties are calculated from the identified parameters, of the three recursive algorithms, and they are illustrated in Figure 6.13. For the sake of simplicity, we only use the results of *Model 1*. It is seen that the *EFRA* method

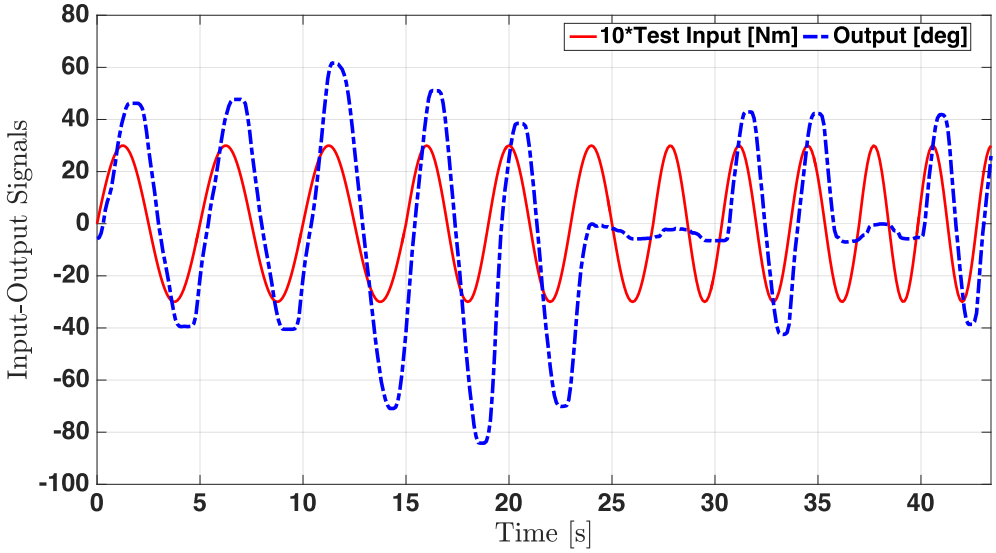


Figure 6.12 : The input-output signal measurements of *Dataset 2*.

successfully adapt the changes in the dynamics rather than the methods of *RLS* and *RLS-FF*. There is a slight change in the process dynamics at $t = 10\ s$ where the driver grabs the hand wheel and applies torque in the same direction with the haptic forces. This explains the increase in the hand wheel angle measurements between $t \in [10, 24]\ s$ since an exogenous torque is applied by the driver. This change in process dynamics is reflected in the mechanical impedance parameters as they decrease in the time interval $t \in [10, 24]\ s$. Moreover, the driver opposes the haptic forces between $t \in [24, 30]\ s$.

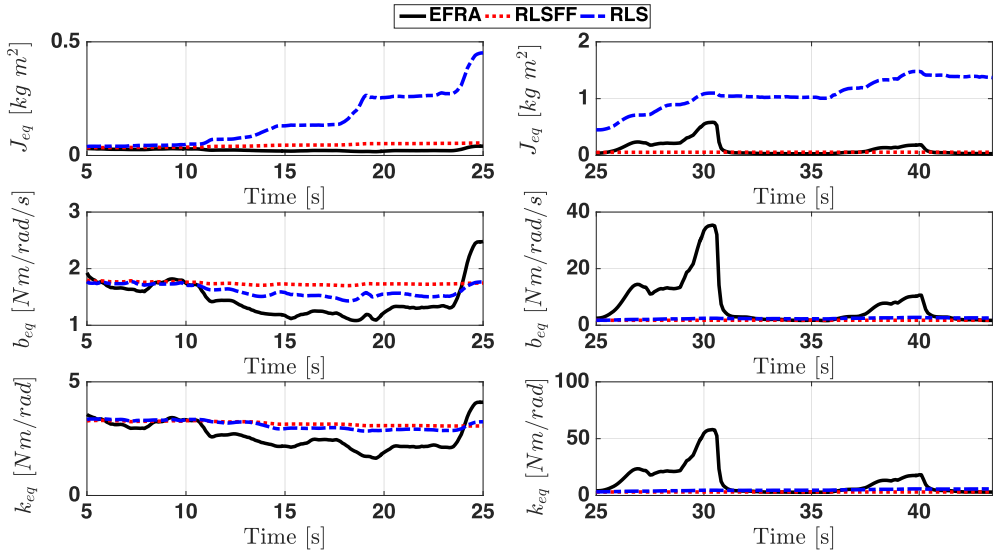


Figure 6.13 : The calculated mechanical impedance properties for *Dataset 2*.

The position of the hand wheel is close to the origin since the driver grabs it firmly. This abrupt change in the dynamics is reflected in the identified impedance parameters

as they increase significantly. However, the recursive algorithms except *EFRA* fails to track these abrupt changes in the process dynamics as seen from the Figure 6.13.

The advantage of *EFRA* method is more clear with the results of *Dataset 2*. Although the other methods show more or less the same performance as *EFRA* for *Dataset 1*, they fail to converge to correct values when there is abrupt changes in the dynamics. Therefore, the effectiveness of *EFRA* method is validated with the results of the second dataset. We compare the performance of *RLS* and *EFRA* methods. Starting from $t = 25\ s$ when the driver opposes the haptic forces, it is seen that the predictions of *EFRA* stay close to the actual values since *EFRA* is able to track the changes in the parameters better than *RLS* method.

6.6 Discussions and Conclusions

The main objective of this section is to use recursive parameter identification algorithms for linear, data-driven models in order to identify the unknown and time-varying parameters of the driver-in-the-loop steering system. The recursive least squares method is chosen in order to reduce the computational complexity while the parameter identification runs online at each sampling interval. Therefore we introduce three different algorithms and propose two linear discrete time models which are based on the previous time-step input-output values. These data-driven models are derived from the continuous-time dynamics so there is a one-to-one transformation from the identified discrete-time parameters to physical mechanical impedance values (parameters of the differential equation). This gives us some insight about the results since we can discuss the estimated mechanical impedance parameters in an intuitive way.

We conduct experiments with various interaction levels of driver and the feedback forces. For example the hands-off case refers to no interaction. On the other hand, the compliant driver case refers to positive interaction (i.e., agreeing) while the stiff driver case refers to negative interaction (i.e., opposing). We also conduct an experiment with alternating interaction level of the driver to simulate abrupt changes in the dynamics. As a result, it is shown that *EFRA* methods shows superior performance in tracking of the actual parameters of the system when there is abrupt changes in the process.

In these cases, the covariance resetting and forgetting factor properties of EFRA algorithm has provided a better tracking capability for the algorithm.

The results of this chapter shows that by using a linear data-driven model and a recursive parameter identification algorithm we can identify the driver-in-the-loop system. This gives us encourage to integrate the RLS algorithm in the control framework in order to update the parameters of the prediction model which is employed in MPC. But this time the system will be excited with the control law provided from the feedback controller instead of using an open-loop test signal.

7. ADAPTIVE AND HUMAN-CENTERED STEERING ASSISTANCE SYSTEM FOR SHARED AND AUTOMATED DRIVING

7.1 Introduction

One of the most challenging design issue in implementing the haptic shared control system is the calculation of the feedback forces which yields a good acceptance from the driver and therefore maintains a healthy interaction between the driver and the assistance system. Generally, the design of these forces are heuristic and they are designed by trial-and-error methods which results in a subjective tuning. The feedback forces should be adequate enough to guide the driver and they should not exceed a certain amount that could be inconvenient for the driver's comfort. Therefore while designing such an assistance system, it is important to account for the human-machine interactions since the driver feels the feedback forces through the hand wheel.

The previous studies show that designing the feedback forces based on neuromuscular system (NMS) properties of the driver could be a promising approach [2, 81] for implementing shared control systems. However, the NMS properties of the driver could vary while interacting with the guidance forces during a shared steering task. It is concluded in [2] that it is difficult to design the guidance forces if the neuromuscular impedance of the driver is not known and it causes a mismatch in neuromuscular level which results a decrease in performance and increases the effort of the driver [31]. Therefore one should estimate the NMS characteristics of the driver's arms while interacting with the feedback forces such that these forces could recalculated with the estimated properties.

We present the identification results of human mechanical impedance properties with respect to different conditions which shows the adaptability of the NMS properties in Chapter 6. The results show that the transfer function from the control input (feedback forces) to output (hand wheel angle) could be captured by using a regression model whose parameters are identified online by a recursive least squares

algorithm. Encouraged from these preliminary results, we propose to integrate the parameter identification scheme into the predictive control algorithm presented in Chapter 5 and update the driver-in-the-loop steering model parameters online. The preliminary experimental results in Chapter 5 show that the neuromuscular system (NMS) properties of the driver model plays an important role in the predictive control framework since the feedback forces are optimized by using the prediction model.

It is important for the assistance system to have an information related to the driver's presence on the hand wheel. Generally, most of the production level systems use capacitive sensors instrumented inside the hand wheel so that when the driver grips the hand wheel, the capacitive sensor measures the pressure and therefore indicating if the driver is in the loop or not. For example in a lane keeping assistance system, the system does not expect the driver to take hands off the hand wheel during the operation. Therefore, the system alerts the driver to hold the hand wheel.

However, in this thesis we propose to design an assistance system which enables full autonomous driving without switching between independent controllers. In order to allow for a seamless transfer of the control authority, we propose to use the hands on/off information in the predictive control framework such that when the driver is in the loop, a small value is assigned to the trajectory tracking weight and when the driver is not in the loop, a high value is assigned. This allows the system to have both ADAS and full autonomous driving capabilities without enforcing the driver.

The success of the haptic shared control systems depends on the thorough design of the control algorithms that accounts for the intention of the driver (i.e., to be less intrusive). If the system is not aware of the intentions of the driver (i.e., lane keeping/changing) then conflicts might arise which degrades and distorts the shared driving properties. In order to avoid these conflicts, the intention of the driver should be estimated from the driving history of the human driver by using machine learning methods. For example in [79, 83], a high-level driver mode estimation algorithm based on Hidden-Markov Model (HMM) is proposed for the longitudinal speed control of the vehicle. This data-driven model learns the longitudinal driving profile of the driver and estimates the acceleration input of the driver. A personalized adaptive cruise control system is achieved by using this driver model. The same authors propose to use the same

learning methods for estimating the steering intentions of the driver in [84]. Moreover in [85], a lane change intention estimation algorithm is proposed by using support vector machines (SVM) for lateral control purposes.

7.2 Contributions

The novelty of this work lies in the formulation of the predictive controller which integrates the aforementioned concepts in a unified optimal control framework. Firstly, an online parameter identification scheme based on recursive least squares is used in order to update the prediction model of the proposed controller. This allows the controller to track the changes in the properties of the neuromuscular system of the driver's arms and therefore an optimal guidance torque (feedback force) is calculated with respect to the level of interaction using the updated model. Secondly, we integrate a high-level driver intention estimation algorithm which is introduced in [84]. This module allows the controller to understand the lateral control behavior of the driver and to generate a desired lateral position trajectory for the controller. As we discussed before, it is essential for the controller to comprehend the intentions of the driver in order to minimize the conflicts in the objectives. Finally, we integrate a machine learning approach to detect the driver's presence on the hand wheel. This allows the controller to assign its priorities in the objective function. When the driver takes its hands off the wheel, the most important objective of the controller is to track the desired path autonomously. On the other hand, when the driver is in the loop, the controller should not enforce the driver to follow its desired path but guides the driver gently and intervenes only if there is hazardous situation.

A novelty in the context of experiments is that we conduct an experimental study with five participants to test the performance of the proposed assistance system. While most of the presented work in literature [2,4,24] is implemented by using a driving simulator, in this paper we performed experiments by using a test vehicle with an MDPS system. Also the proposed controller is implemented and tested in real world environments with human drivers. The results show the capabilities of the proposed system from guidance to full autonomous driving modes while providing safe automated driving.

The results of this chapter is published in Transactions of Human-Machine System [86] and the author is still working on publishing the unfinished material.

7.3 The Controller Architecture

The controller of the assistance system is built on a nonlinear model predictive control (NMPC) framework. The control structure is illustrated in Figure 7.1. The

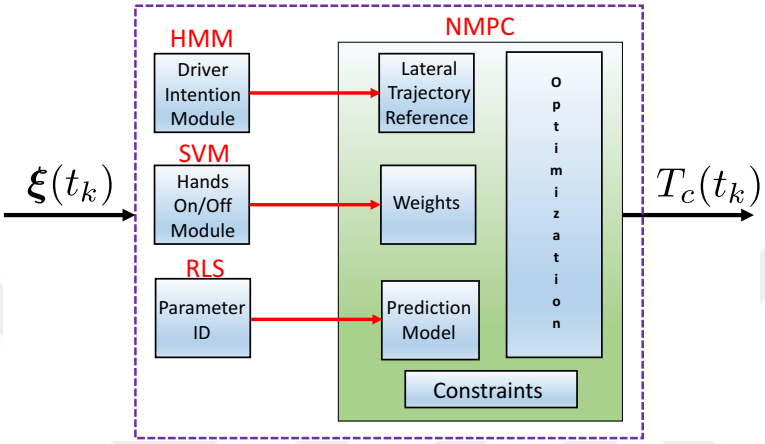


Figure 7.1 : The proposed control structure.

prediction model is introduced in Section 7.3.1 and the parameter identification method is given in Section 7.3.2. The constraint formulation is summarized in Section 7.3.5 and the optimal control problem is formulated in Section 7.3.6.

7.3.1 Prediction model

The Vehicle Model : We use the vehicle model introduced in Section 3.3 which is compactly written in equation (3.16). This model has six states and one control input which is the steering angle (δ).

Tire Model : For the lateral tire forces we use the linear tire model introduced in Section 3.4.3 where the slip angles are assumed to be small and they are given in (3.12).

Data-Driven Model for Steering Dynamics : In this formulation, we are interested in using a data-driven model to capture the steering dynamics from control input T_c to the output θ. Therefore we need some simplifications and assumptions to model the dynamics as a single input single output (SISO) system which is suitable for using simple machine learning approaches.

We start with modeling the torque feedback of the road through the hand wheel T_{rd} as a function of states such as θ and $\dot{\theta}$. In [56], it is shown that T_{rd} is proportional to the hand wheel angle θ. Also the study presented in [63] formulates the T_{rd} as a linear function of slip angle α_f, hand-wheel angle θ and the angular velocity $\dot{\theta}$.

We use the following model to represent the torque feedback of the road through the hand wheel T_{rd}:

$$T_{rd} = b_r(t)\dot{\theta} + k_r(t)\theta, \tag{7.1}$$

where the values of $k_r(t)$ and $b_r(t)$ could be configured by the MDPS control logic to adjust the steering road feel of the driver. We assume that these values are constant in normal driving mode.

We formulate the following continuous-time dynamic model which is derived for experimental purposes for MDPS.

$$\big(J_h + J_{d,p}\big)\ddot{\theta} = -\big(b_h + b_{d,p} + b_r(t)\big)\dot{\theta} - \big(k_{d,p} + k_r(t)\big)\theta + T_d + T_c. \tag{7.2}$$

We model the mechanical impedance of the driver's arms while opposing the system's activity as :

$$-T_d = J_{d,a}(t)\ddot{\theta} + b_{d,a}(t)\dot{\theta} + k_{d,a}(t)\theta, \tag{7.3}$$

where $J_{d,a}$, $b_{d,a}$, $k_{d,a}$ represent the activated inertia, damping and stiffness properties of the impedance model of the driver's arms as seen at the steering wheel axis. These time-varying parameters describe the co-contracted state (i.e., increased damping and stiffness) impedance properties of the arms.

Remark 11. The active inertia, damping and stiffness of the driver's arms, $J_{d,a}(t)$, $b_{d,a}(t)$ and $k_{d,a}(t)$, are zero either when there is no guidance from the assistance system or when driver does not oppose the torque intervention.

Therefore combining the equation (7.2) with (7.3), we derive the coupled dynamics of the driver and the MDPS system.

$$J_{eq}(t)\ddot{\theta} = -b_{eq}(t)\dot{\theta} - k_{eq}(t)\theta + T_c, \tag{7.4}$$

where the equivalent inertia, damping and the stiffness of the coupled steering dynamics model is given by $J_{eq}(t) = \big(J_h + J_{d,p} + J_{a,p}(t)\big)$, $b_{eq}(t) = \big(b_h + b_{d,p} + b_{d,a}(t) + b_r(t)\big)$ and $k_{eq}(t) = \big(k_{d,p} + k_{d,a}(t) + k_r(t)\big)$, respectively.

The state-space representation of the coupled steering dynamics model is given as :

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \omega\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{k_{eq}(t)}{J_{eq}(t)} & -\frac{b_{eq}(t)}{J_{eq}(t)}\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{J_{eq}(t)}\end{bmatrix} T_c, \tag{7.5}$$

where we define the hand wheel rate as $\omega = \dot{\theta}$.

We approximate the continuous-time model of the process given in (7.5) by using a forward difference method with a sampling period of Δt_c. The discrete-time model is obtained as :

$$\begin{bmatrix}\theta(k+1)\\ \omega(k+1)\end{bmatrix} = \begin{bmatrix}1 & \Delta t_c\\ \varphi_1(k) & \varphi_2(k)\end{bmatrix}\begin{bmatrix}\theta(k)\\ \omega(k)\end{bmatrix} + \begin{bmatrix}0\\ \varphi_3(k)\end{bmatrix} u(k), \tag{7.6}$$

where k denotes the current discrete-time step such that $t = k\Delta t_c$. The true parameters are defined as $\varphi_1(k) = -\frac{\Delta t_c k_{eq}(t)}{J_{eq}(t)}$, $\varphi_2(k) = 1 - \frac{\Delta t_c b_{eq}(t)}{J_{eq}(t)}$ and $\varphi_3(k) = \frac{\Delta t_c}{J_{eq}(t)}$. The control input is denoted as $u(k) = T_c$.

We discretize the set of differential equations for the vehicle model given in (3.16) using a forward difference method and augment it with the data-driven steering model given in (7.6). The resulting prediction model, which is employed in the controller, can be compactly written as:

$$\boldsymbol{\xi}(k+1) = f_d\big(\boldsymbol{\xi}(k), u(k), \boldsymbol{v}(k)\big), \tag{7.7}$$

where the state vector is $\boldsymbol{\xi} = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s, \theta, \omega]^\intercal$, the control input is $u = T_c$ and the exogenous input vector is $\boldsymbol{v} = [\kappa, \mu, a_x, \varphi]^\intercal$. The unknown parameter vector $\boldsymbol{\varphi}$ is identified online by a recursive least squares algorithm while the system is running.

7.3.2 Parameter identification

We have already introduced the parameter identification algorithms and the regression models in Chapter 6. In order to identify the unknown parameters of the coupled dynamics given in (7.6), we use the following linear model to be used for recursive least squares parameter identification algorithms.

$$\hat{\omega}(k) = \hat{\varphi}_{0,k-1} + \hat{\varphi}_{1,k-1}\,\omega(k-1) + \hat{\varphi}_{2,k-1}\,\theta(k-1) + \hat{\varphi}_{3,k-1}\,u(k-1), \tag{7.8}$$

where we define the measurement $y_k = \omega(k)$ and a bias term $\hat{\varphi}_0$ is integrated in order to account for unmodeled input disturbances. The proposed regression model

is compactly written as $\hat{y}_k = \boldsymbol{x}_k^{\mathsf{T}} \hat{\boldsymbol{\varphi}}_{k-1}$ where the regression vector is given as $\boldsymbol{x}_k = [1,\ \omega(k-1),\ \theta(k-1),\ u(k-1)]$ and the identified parameter vector is given as $\hat{\boldsymbol{\varphi}}_k = [\hat{\varphi}_{0,k}, \cdots,\ \hat{\varphi}_{3,k}]$. Under sufficient excitation of the steering dynamics, the identified parameters should converge to the true values of the unknown parameters.

We use the EFRA method, given in (A.12), in order to identify the unknown parameters of the driver-steering system. Apart from the system identification experiments in Chapter 6, in this chapter the system is excited by the driver and the feedback forces calculated by the proposed control structure. Therefore we call this type of system identification as the closed-loop identification since the excitation input is calculated by a feedback controller.

7.3.3 Driver intention module

The driver intention module provides the high-level driving mode of the driver to the proposed control framework. These modes could be specified as lane keeping (LK), a lane change to the right (LCR) and a lane change to the left (LCL). When the mode of the driver is estimated, we can generate an estimated lateral trajectory of the driver in order to provide it to the controller. This allows the controller to have a knowledge on the desired trajectory of the driver.

We use the driver model proposed in [84]. The readers can refer to the reference for detailed information on the design of the driver model.

7.3.4 Hands on/off detection module

We use a powerful machine learning approach named as Support Vector Machine (SVM) for classification of the presence of the driver's hands on the steering wheel. The proposed SVM structure uses hand wheel angle position and velocity $\theta, \dot{\theta}$ along with steering column torque T_s and commanded control input T_c as inputs and SVM ouputs a binary classification (0/1) which corresponds to a hands on situation (the class 0) and a hands off situation (class 1). The SVM structure is illustrated in Figure 7.2.

First of all, the SVM should be trained with known data points. This is called as supervised learning since we provide the known output to the algorithm. The training

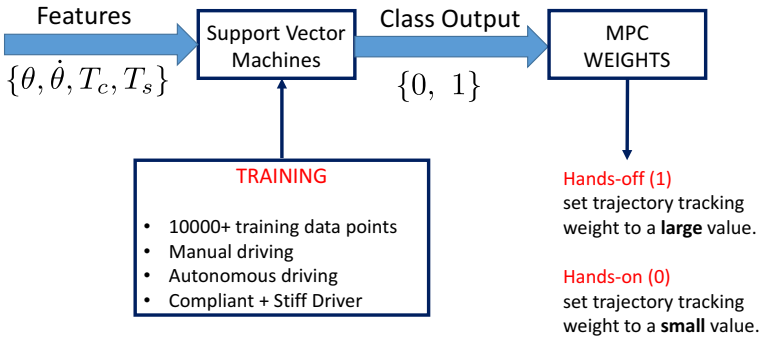


Figure 7.2 : The proposed SVM classifier to detect the presence of the driver in the hand wheel.

data consists of more than 10000 data points which are experimentally logged in various automated driving scenarios. After the training, we can use the SVM in the control structure. The output of the SVM is binary and it switches between zero and one. Since the tracking weight in the objective function of the MPC depends on the class output of SVM, we use a low-pass filter to provide a smooth transition between the weight values. Instead of having discontinuous weight changes in the objective function, a simple low-pass filter increases or decreases the weight value in a continuous manner. The filter parameters decide the settling time and the characteristics of the output.

7.3.5 Constraints

We formulate the safety constraints as the lateral position constraints and the slip angle constraints. The lateral position constraints are calculated as a function of road boundaries and the target vehicles around. The slip angle constraints are calculated as a function of road friction coefficient. The safety constraints are compactly written as:

$$h^{\xi}(\boldsymbol{\xi}_{i+1}, u_i) \leq \mathbf{0} \quad i = 0 \ldots N-1, \tag{7.9}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

We enforce the magnitude and the rate constraints on the overlay torque (i.e., actuator input) T_c in order to have a smooth and adequate level of feedback forces considering the comfort of the driver. The control input constraints could be compactly written as:

$$h^{u}(u_i, \Delta u_i) \leq \mathbf{0}, \quad i = 0 \ldots N-1, \tag{7.10}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

7.3.6 Optimal control problem

The constrained finite time optimal control (CFTOC) problem in (7.11) is solved at each discrete time step k to compute an optimal control input sequence $\boldsymbol{u}_k$. Only the first control input is applied to the vehicle, and this optimization routine is repeated at the next time step using new state measurements as initial conditions.

$$\min_{\boldsymbol{u}_k,\varepsilon} \sum_{i=0}^{N-1} \left(\|u_{i|k}\|_Q^2 + \|\Delta u_{i|k}\|_R^2 \right) \tag{7.11a}$$

$$+ \sum_{i=1}^{N} \|\boldsymbol{\xi}_{i|k} - \boldsymbol{\xi}_{i|k}^{ref}\|_S^2 + W\boldsymbol{\epsilon}$$

$$\text{s.t. } \boldsymbol{\xi}_{i+1|k} = f_d(\boldsymbol{\xi}_{i|k}, u_{i|k}, \boldsymbol{v}_k), \tag{7.11b}$$

$$h^{\xi}(\boldsymbol{\xi}_{i+1|k}, u_{i|k}) \leq \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \geq \mathbf{0} \tag{7.11c}$$

$$h^{u}(u_{i|k}, \Delta u_{i|k}) \leq \mathbf{0}, \tag{7.11d}$$

$$i = 0 \ldots N-1$$

$$\boldsymbol{\xi}_{0|k} = \boldsymbol{\xi}(k), \qquad u_{-1|k} = u(k-1), \tag{7.11e}$$

where $\boldsymbol{\xi}_{i|k}$ denotes the i-th state prediction at time step k, obtained by applying the optimal control input sequence $\boldsymbol{u}_k = \{u_{0|k}, u_{1|k}, \ldots, u_{N-1|k}\}$ to the discrete-time dynamics (7.11b) starting from the measured state at the current time step (7.11e). The cost function in (7.11a) reflects control objectives such as utilizing smooth and minimal control action and tracking the desired lateral trajectory. We penalize the magnitude and rate of control input as well as the tracking error, using corresponding weight matrices Q, R and S, respectively. The input constraints are given in (7.11d). We introduce slack variable vector ε for the safety constraints (7.11c) in order to keep the optimization problem feasible in case of constraint violations due to model mismatch. A large matrix W strongly penalizes constraint violation.

7.4 Experimental Results : Road Departure Prevention

The experimental setup is introduced in Section 5.4.1. Besides the general setup, the parameters given in Table B.1 and Table 7.1 are used in the control framework. The weight of the lateral tracking cost term w_{svm} is associated with the hands on/off detection algorithm where $w_{svm} = 1$ for hands on driving and $w_{svm} = 100$ for full autonomous driving.

Table 7.1 : The design parameters of the controller for the experiments.

Parameter	Value	Parameter	Value
(Q,R,M)	(10, 10, 10^4)	S	**diag**(0,0,0,0,w_{svm},0,0,0)
$(T_{c,\min}, T_{c,\max})$	(-5, 5) Nm	$(\Delta T_{c,\min}, \Delta T_{c,\max})$	(-5, 5) Nm/s
$(\alpha_{\min}, \alpha_{\max})$	(-4, 4) deg.	Δt_c	0.1 s
N	12		

The experiments are performed at a constant longitudinal velocity of 20 m/s by setting the cruise control feature. We perform the experiments in Hyundai-Kia California Proving Grounds (CPG) in CA, USA. We use the straight stability track where there is a two-lane closed traffic road. The nonlinear optimization problem presented in Section 7.3.6 is solved at a sampling interval of $\Delta t_c = 0.1$ s in real time. Also the recursive parameter identification algorithm runs at the same sampling interval with the controller. The prediction horizon is selected as $N = 12$ because of real-time computational reasons. The longer horizon selection caused problems in real-time operation.

7.4.1 Compliant driver behavior

In this experiment, we ask the driver to steer towards the road boundary and not to resist the haptic forces that are applied by the proposed assistance system. The vehicle start initially at $e_y = 0.1$ m. The driver starts steering to the right hand side of the road around $t = 3$ s. The system intervenes the driver around $t = 4$ s with a magnitude of 2.5 Nm. The driver accepts the guidance from the system and steers back to the centerline. Around $t = 10$ s, the driver performs one more time and the results show a similar action from the assistance system. The results of the experiment is illustrated in Figure 7.3.

The lateral position plots show that the safety constraints are not violated due to the collaborative behavior of the driver and the system. The system exerts enough guidance torque as the driver accepts the intervention. The Figure 7.4 shows the hand wheel angle measurement and the resulting open-loop predictions that are calculated by the predictive controller. The adaptive behavior of the model shows accuracy in predictions which enhances the control performance and therefore the human-machine

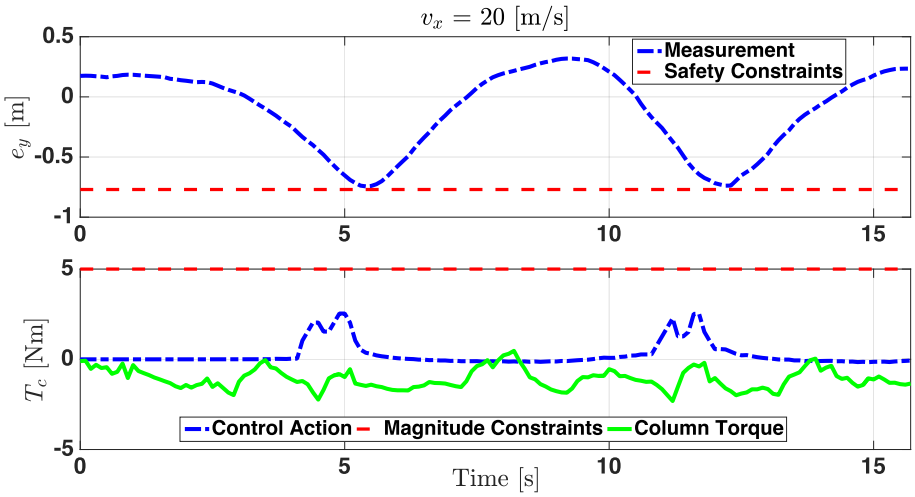


Figure 7.3 : The lateral position, commanded control input and the measured steering column torque for the experiment case *Compliant Driver*.

interactions. Finally, we illustrate the identified parameters of the data-driven model in Figure 7.5.

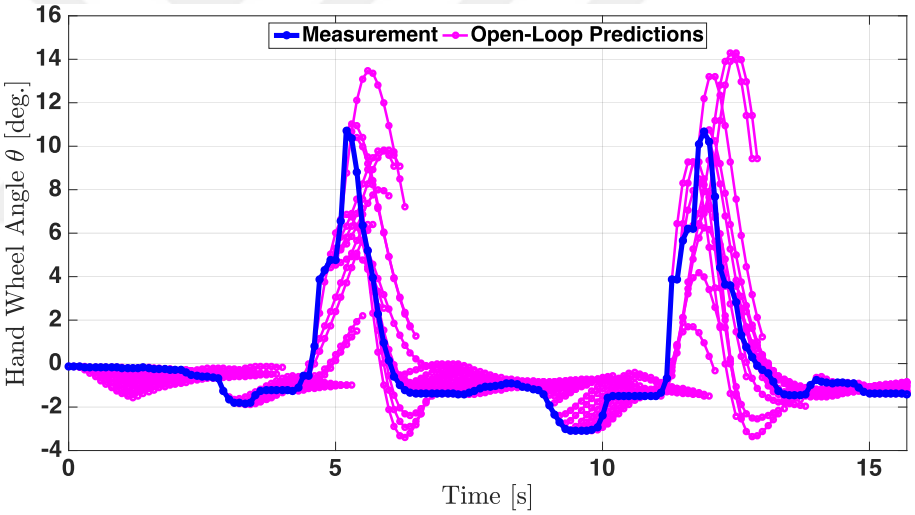


Figure 7.4 : The hand wheel measurements and the open-loop predictions for the experiment case *Compliant Driver*.

7.4.2 Stiff driver behavior

In this experiment, we ask the driver to steer towards the road boundary and resist the haptic forces that are applied by the proposed assistance system. The results of the experiment is illustrated in Figure 7.6. The vehicle start initially at $e_y = 0.5$ m. The driver starts steering to the right hand side of the road around $t = 1$ s. At $t = 5$ s, the vehicle is around 0.5 m away from the lane centerline and the system intervenes with a torque of 2.5 Nm. As the vehicle travels towards the road boundary (i.e., the

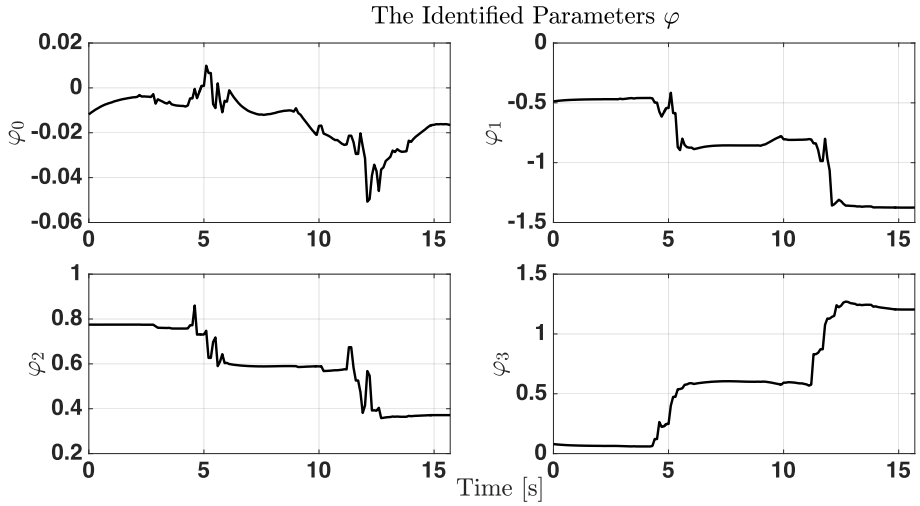


Figure 7.5 : The identified parameters of the model given in (7.5) for the experiment case *Compliant Driver*.

lateral safety constraint), the assistance system increases the amount of intervention to the maximum level of 5 Nm. The system and the driver struggles power between $t \in [5,6]$ s and the driver puts out a huge amount of effort to overcome the system's activity but since it is a safety measure the system does not let driver to overcome. After $t = 6$ s, the driver accepts the guidance from the system and steers back to the centerline. It is seen that the safety constraints are violated slightly around 5 cm. The driver repeats the same maneuver at $t = 10$ s and the system reacts as the same way since the driver behavior is the same.

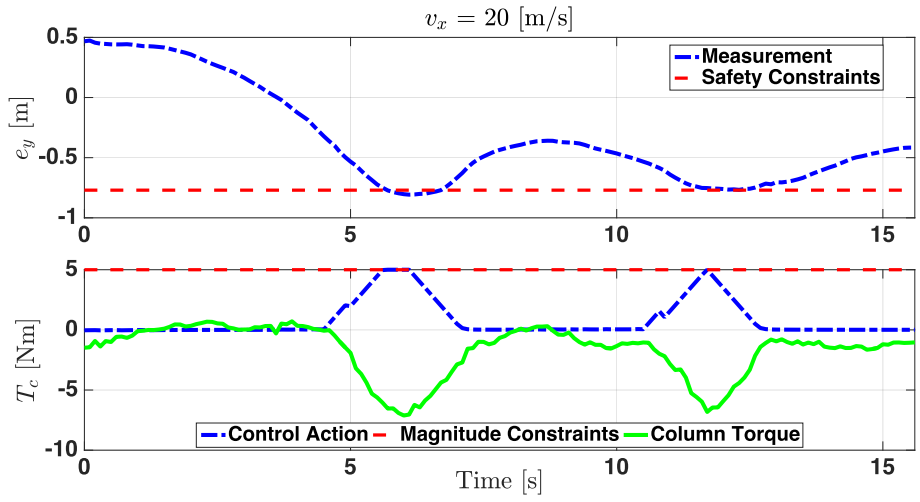


Figure 7.6 : The lateral position, commanded control input and the measured steering column torque for the experiment case *Stiff Driver*.

The Figure 7.7 shows the hand wheel angle measurement and the resulting open-loop predictions that are calculated by the predictive controller. The adaptive behavior of

the model shows accuracy in predictions which enhances the control performance and therefore the human-machine interactions.

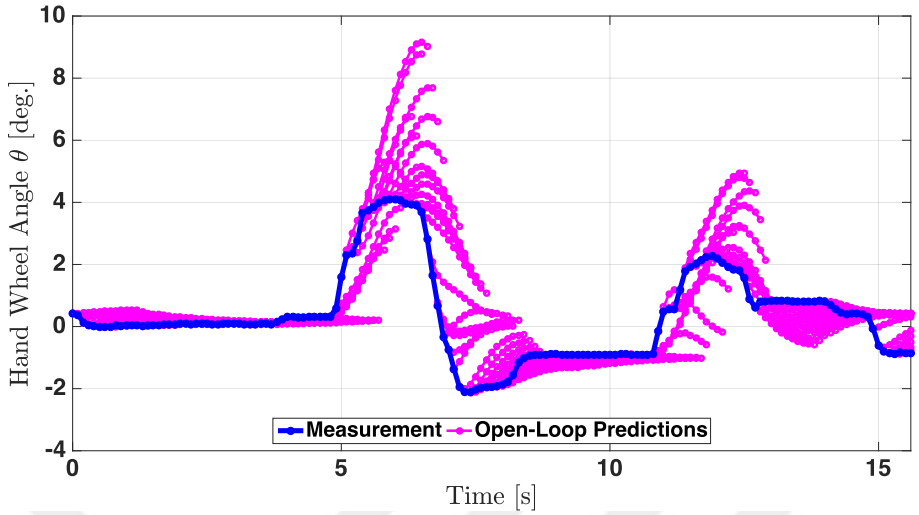


Figure 7.7 : The hand wheel measurements and the open-loop predictions for the experiment case *Stiff Driver*.

Finally, we illustrate the identified parameters of the data-driven model in Figure 7.8.

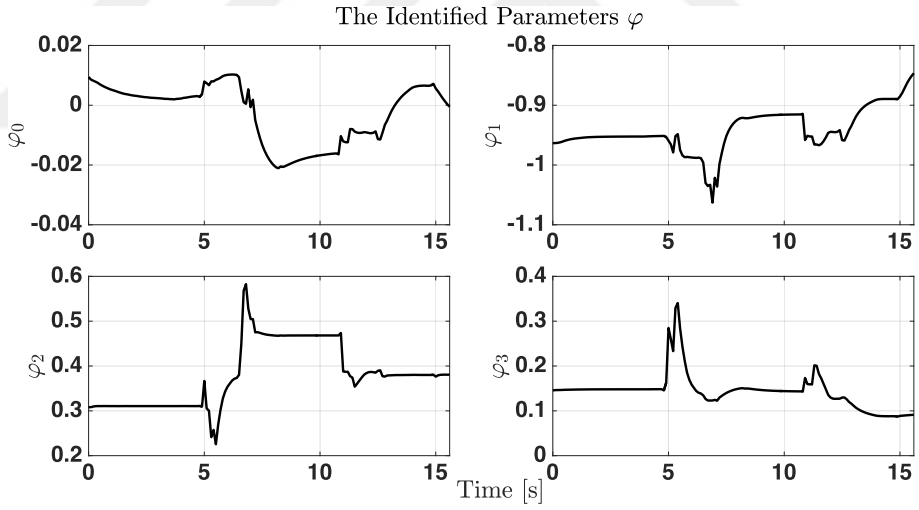


Figure 7.8 : The identified parameters of the model given in (7.5) for the experiment case *Stiff Driver*.

7.4.3 Conventional approach vs. adaptive approach

The main differences between the control architecture in Section 5.2 (conventional approach) and Section 7.3.6 (adaptive approach) is the existence of a recursive parameter identification method and the lateral position reference tracking term in the optimal control problem formulation.

From the experimental results presented both in Sections 5.4 and 7.4, we could conclude that updating the parameters of the steering model enhances the performance of the assistance system. The results show that by updating the prediction model, the controller intervenes earlier with a comfortable amount of torque.

7.5 Human-Machine Interactions in Lane Keeping Guidance

In this section, we present the results of our experimental study where the assistance system guides the driver to the lane center line from an initial lateral position while the driver shows different levels of interaction ranging from following the guidance torque T_c (i.e., *compliant* driver) to actively resisting it (i.e., *stiff* driver).

One of the objectives of this study is to show that the proposed approach can identify the corresponding differences in the drivers' mechanical impedance properties and suitably adapt the guidance torque by updating the model employed in the controller with respect to the interaction level.

7.5.1 Experimental procedure

We performed an experimental study with five participants aged between 25 to 30 years at the Berkeley Global Campus. The vehicle was initially at rest on a straight road with two lanes in each direction and was positioned around 1.5 m to the left of the rightmost lane center line. We ask each participant to accelerate and maintain a constant speed of 11 m/s and to check the speed display of the vehicle frequently to simulate a visual distraction. The participants can activate the assistance system by pushing a marked button on the hand wheel after passing a visual marking on the road.

The objective of the experimental study is to identify the neuromuscular differences in various participants either when the assistance system is activated or not activated. Therefore we define four test cases where in *driver only* case the participant performs the lane keeping maneuver when the assistance system is deactivated. In *hands off* case, the participant activates the assistance system and takes its hands off the hand wheel and the system performs the lane keeping maneuver alone

The interaction between the haptic forces and the participants are experimented in *compliant* and *stiff* driver cases. When the assistance system applies guidance torque

to bring the vehicle to the lane center line, we ask the participants to show different neuromuscular response to the haptic forces without actively steering the vehicle. This corresponds to an impedance like behavior where the *compliant* driver ideally follows the haptic forces acting on the hand wheel and *stiff* driver opposes these forces by holding the hand wheel firmly for a definite amount of time and releases the hand wheel afterwards.

The recursive identification algorithm was also started when the participant activated the controller. The parameters were initialized as $\hat{\Phi}_0 = [0,\ 0,\ 0,\ 0.1]^\intercal$ to introduce a modeling uncertainty to the controller (i.e., the driver-in-the-loop steering model is initially unknown).

7.5.2 Results

We present the results of the test cases that are performed by all participants in Figure 7.9. In this figure, the lateral deviation from lane center line is illustrated in the left columns. The plots in the middle column illustrate the hand wheel angle θ measurements. The right top column plot illustrates the steering effort of the participants and the rest of the plots illustrate the guidance torque applied by the controller.

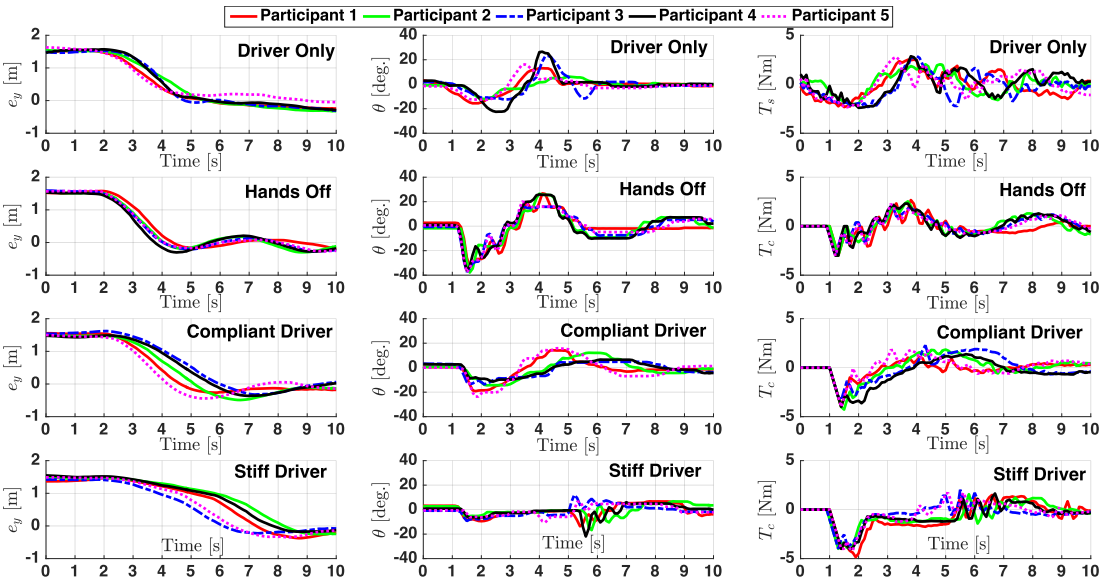


Figure 7.9 : The comparison of the test cases among five participants.

In the *driver only* case, it is observed that participants' lane centering trajectories without the controller are similar in terms of small overshoot but a steady-state error

around 0.2 m when they complete the maneuver. Each participant also performed the *hands off* case without holding the hand wheel. There are some differences in the associated plots which is caused by the different longitudinal speed behavior of the participants. Nevertheless, the controller performance was consistent. We observed an overshoot around 0.25 m and a slow settling time of around 8 s during the trajectory tracking maneuver. This is attributed to improper tuning of the controller since the focus of the paper is not autonomous driving.

If we consider the *compliant* driver case, we see that the response of the participants to the guidance torque is quite different. The spread of the lateral trajectories is significantly larger in the *compliant* driver case than in the *driver only* and *hands off* cases. This result reveals that participants have different levels of interaction with the guidance torque with respect to their individual preferences.

Among the five participants, *Participant* 3 and *Participant* 5 exhibited the most extreme responses. *Participant* 3 opposed to the guidance torque more than *Participant* 5 and therefore increased the impedance of the arms because the guidance torque might have felt too aggressive for the participant's preferences. On the other hand, *Participant* 5 followed the guidance torque without introducing a noticeable impedance. The steering system related measurements are also illustrated in Fig. 7.10. The convex hull of the related measurements show the steering regions of the different test cases for *Participant* 3 and 5. It is seen that the participants operate in different regions of the measurement space while interacting with the guidance torque of the controller.

It is interesting to note that for *Participant* 5, there is a large overlap between the regions corresponding to the *hands off* and *compliant* driver cases. On the other hand, for *Participant* 3, the regions corresponding to the *compliant* driver and *stiff* driver cases exhibit a relatively large overlap.

In the *stiff* driver case, the participants opposed to the guidance torque by co-contracting their muscles for about four seconds. At $t = 5$s, the participants released their hands from the steering wheel to introduce an abrupt change in the steering dynamics. The RLS algorithm updated the parameters of the model to account for the changes in the dynamics while the controller applied T_c to track the trajectory. We

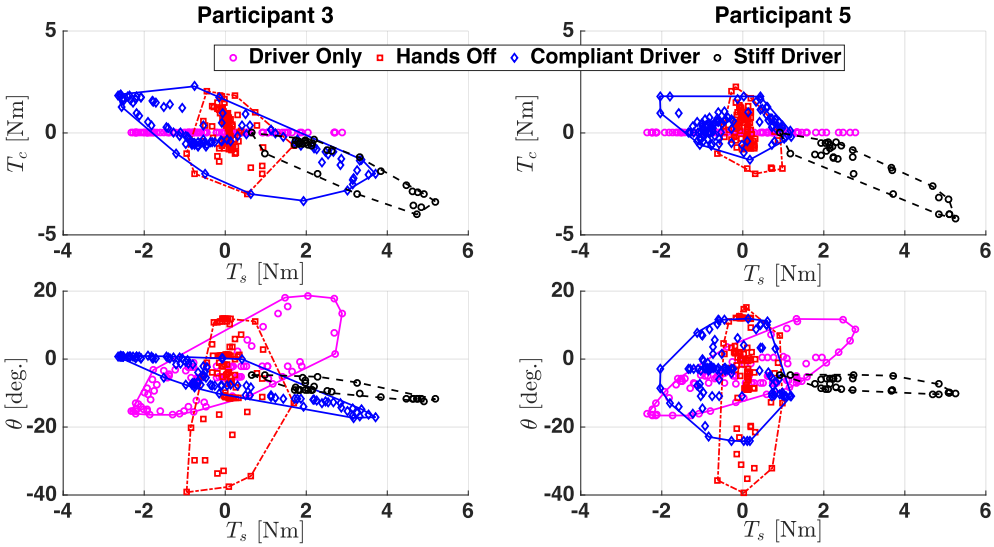


Figure 7.10 : The plots of the guidance torque (T_c) vs. the steering column torque (T_s) and the hand wheel angle (θ) vs. T_s for *Participant* 3 and 5.

observed some oscillations in T_c during the transient period of identification before the parameters converged to their steady-state values.

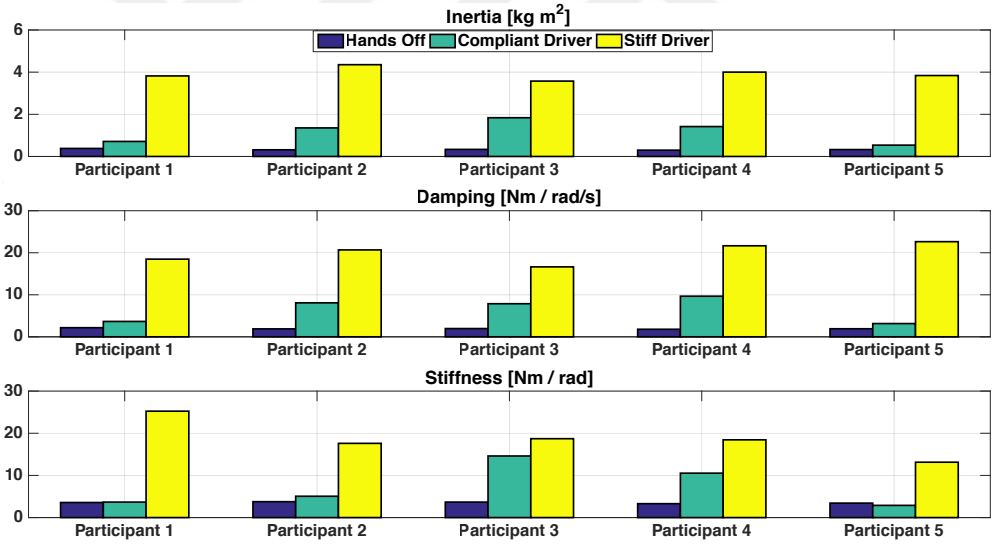


Figure 7.11 : The mean values of the estimated mechanical impedance properties of the driver-in-the-loop steering model for all participants.

The mean values of mechanical impedance properties of all the participants are illustrated in Fig 7.11. The estimated parameters of the *hands off* case for all participants are similar and close to the values that are presented in Section 6.5. This result validates our assumption that the excitation of the steering system is adequate for the closed-loop identification in the *hands off* case. The average impedance of each participant in the *compliant* driver case varies with respect to how participants responded to the guidance torque. As mentioned earlier, it is observed

that *Participants* 2, 3 and 4 introduced more impedance to the steering dynamics than *Participants* 1 and 5. We observed a remarkable increase in the equivalent impedance especially in terms of damping and stiffness during the *stiff* driver case as expected.

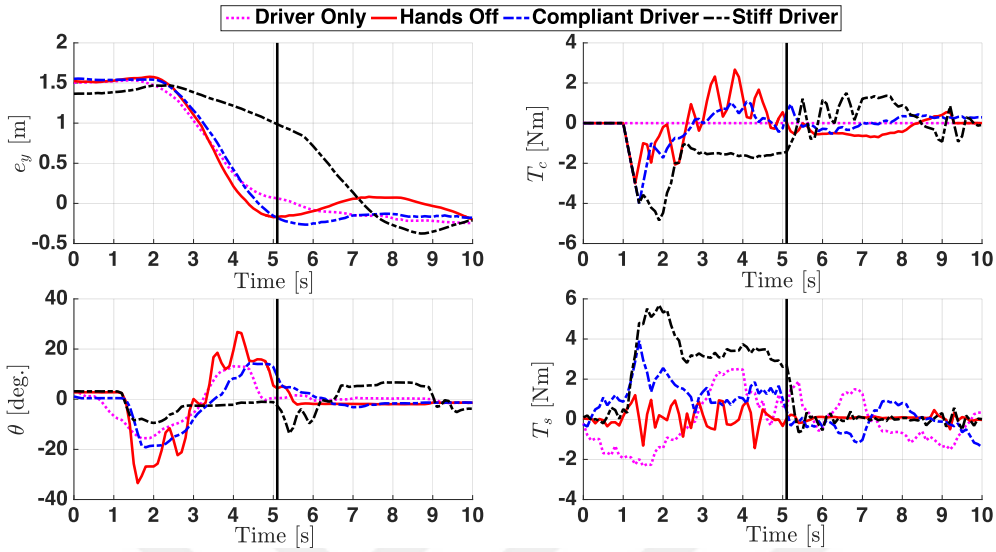


Figure 7.12 : The individual measurements of test cases that are performed by *Participant* 1. The vertical line is associated with *stiff* driver case and indicates the time when the participant released hands from the steering wheel.

We investigate the test cases that are performed by *Participant* 1 separately. This participant is chosen randomly without any bias. The related measurements of the test cases and the estimated impedance properties of the driver-in-the-loop steering model are illustrated in Fig. 7.12 and Fig. 7.13, respectively. The controller is activated around $t = 1$ s in all test cases except the *driver only* case. If the *stiff* driver case is considered, it is seen that the controller applied a guidance torque around -4 Nm after activation. As the participant opposed to the guidance, the RLS algorithm increased the impedance parameters of the steering model employed in the controller. Around $t = 2$ s, the controller started to decrease the guidance torque. This behavior is explained as follows.

The predictive controller calculates the optimal guidance torque that minimizes the objective function. In a lane keeping guidance scenario when there is a conflict between the actions of the driver and the controller, the impedance of the steering model increases. This causes an increase in the input cost since the controller needs a higher amount of corrective torque T_c to steer the vehicle. We would like to stress that when the controller predicts an unsafe situation (e.g., lane/road departure), which

corresponds to the violation of constraints, the controller delivers an adequate level of guidance to keep the vehicle in the safe portion of the road even if the driver opposes the intervention. This is caused by the dominant constraint violation penalty in the cost function.

To introduce an abrupt change in the steering dynamics, the participant released its hands from the steering wheel at $t = 5$ s. However, we observe that the parameters do not converge to the values in the *hands off* case after the participant releases the hands. This is due to the fact that the level of excitation decreases as the vehicle approaches to the lane center line. The controller should introduce a sufficient level of excitation to the steering dynamics in order for the parameters to converge to the true values.

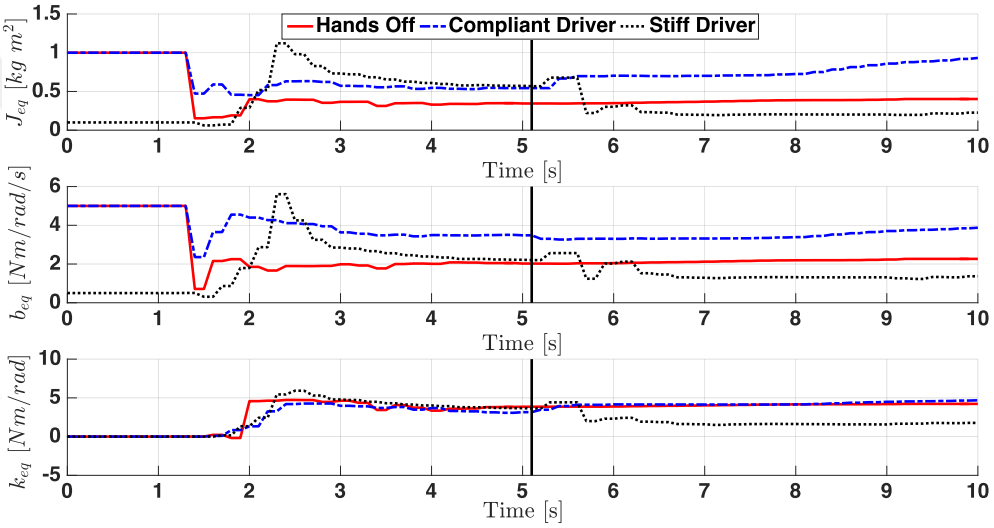


Figure 7.13 : The estimated mechanical impedance properties of driver-in-the-loop steering model for the three test cases. The values of *stiff* driver case is scaled down by a factor of 10. The vertical line is associated with *stiff* driver case and indicates the time when the participant released hands from the steering wheel.

7.5.3 Model comparison results

In this section, we conduct experiments with two control structures in a conflict scenario between the driver and the system during a lane change guidance. In this scenario, the system guides the driver to change the lane but the driver does not agree and opposes the guidance torque to override the system's actions.

The first control structure *Controller 1* is the proposed controller where a recursive parameter identification is employed in order to update the parameters of the driver-in-the-loop steering model. On the other hand, the second control

structure *Controller 2* uses constant parameters in the driver-in-the-loop steering model at all times. These are selected as the average values of *compliant* driver case where $J_{eq} = 0.84$ kgm^2, $b_{eq} = 2.52$ Nm/rad/s and $k_{eq} = 9.40$ Nm/rad.

The aim of this experiments is to show and validate the effectiveness of updating the parameters of the prediction model during interactions between the driver and the assistance system. We present the experimental results in Figure 7.14. The vehicle was initially at rest and positioned at the lane centerline of the rightmost lane (i.e., $e_y(t = 0) = 0$ m). The driver activated the assistance system around $t = 2$ s and the controller initiated the lane change maneuver by applying guidance torque in order to steer the vehicle to the target lane (i.e., $e_y^d = 3.4$ m).

When the system was activated, the guidance torque suddenly jumped to the value of 2 Nm because of the lateral tracking term in the cost function. As a result the vehicle steered towards to the target lane while the driver was griping the hand wheel firmly. Therefore the vehicle traveled around 0.6 m to the left of the original lane. Since the objective of the driver was just holding the hand wheel firmly without actively steering the vehicle back to the lane centerline, the experiment was ended with a lateral offset.

The most important result of this experiment was the guidance behavior of the *proposed controller*. It is seen from the bottom-right subplot of Figure 7.14 that the *proposed controller* decreased the amount of torque guidance as the driver opposed the system's activity. This is the result of updating the steering model parameters of the *proposed controller*. When the driver opposes the torque intervention, the mechanical impedance properties of the coupled system dynamics (i.e., driver arms and steering system) increases as well. As a result of the increasing impedance, more torque is needed to steer the vehicle which will increase the input cost. The optimization routine calculates the optimal value of torque intervention which minimizes the cost function that includes input cost term and lateral reference tracking cost term. This adaptive behavior of the *proposed controller* is important for the comfort of the driver during the interaction since the workload of the driver decreases in the conflict scenarios. The results show the effectiveness of the proposed assistance system to resolve the conflicts in shared steering tasks when the assistance system and the driver have different objectives.

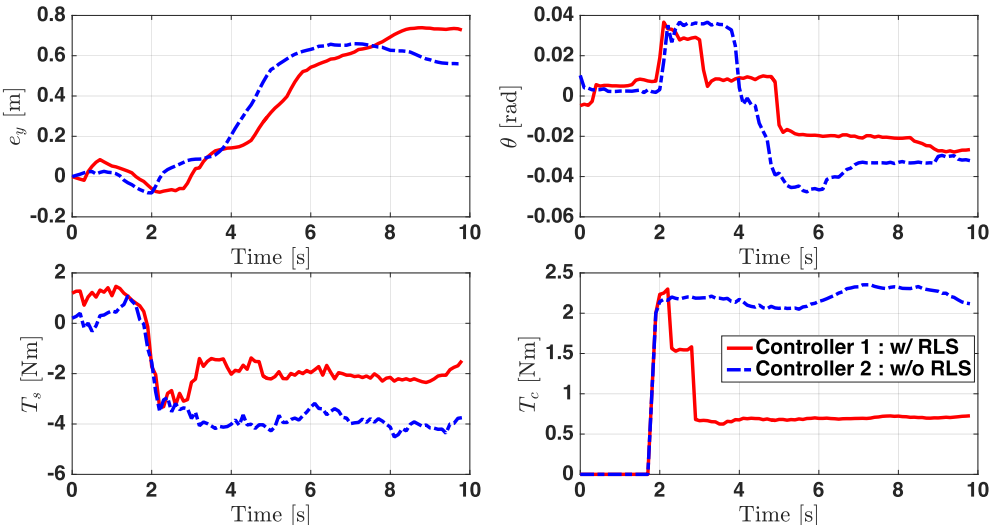


Figure 7.14 : The results of the experiment with two different controllers.

7.6 Human-Centered Lateral Automated Driving

In this experiment, we test the capabilities of the proposed unified control framework while driving at a longitudinal speed of 25m/s. The human driver can activate the proposed system by pushing a button on the hand wheel. The proposed system detects the presence of the driver's hands on the wheel by using the SVM classifier. By using the driver intent module, the system can comprehend the desired lateral behavior of the driver.

The lateral position results of the experiment is illustrated in Figure 7.15.

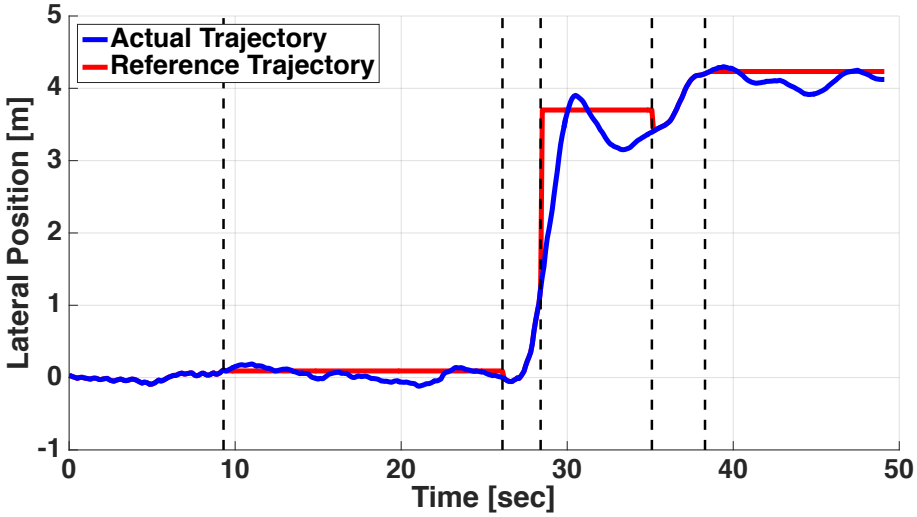


Figure 7.15 : The lateral position results.

Around $t = 9.0$ s, the driver activates the proposed system and takes the hands off from the steering wheel. The proposed system operates in full autonomous mode in

lane keeping driving task. Around $t = 26$ s, the driver takes the control authority and steers the vehicle to the left in order to perform a left lane change. The driver takes the hands off while the lane change maneuver is not finished at $t = 28$ s and the controller completes the lane change maneuver.

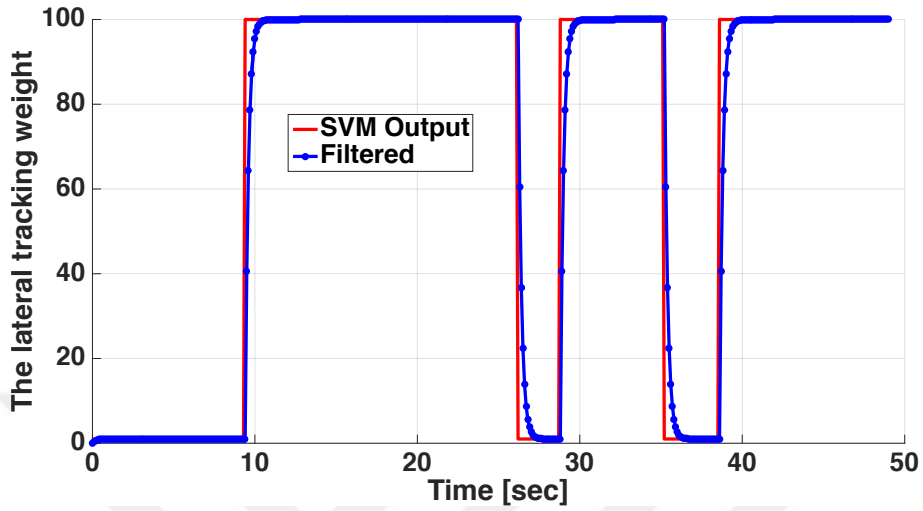


Figure 7.16 : The value of the lateral trajectory tracking weight.

Without using a driver intention module, the expected behavior of the system would be to steer back to the original lane. So this will issue a conflict between the driver's trajectory and the system's trajectory. When a driver intention module is integrated, the controller is aware of driver's intentions and could complete the driver's initiated maneuver. The associated weights with the lateral trajectory tracking is illustrated in Figure 7.16. Using a low-pass filter enables a smooth transition between hands on and

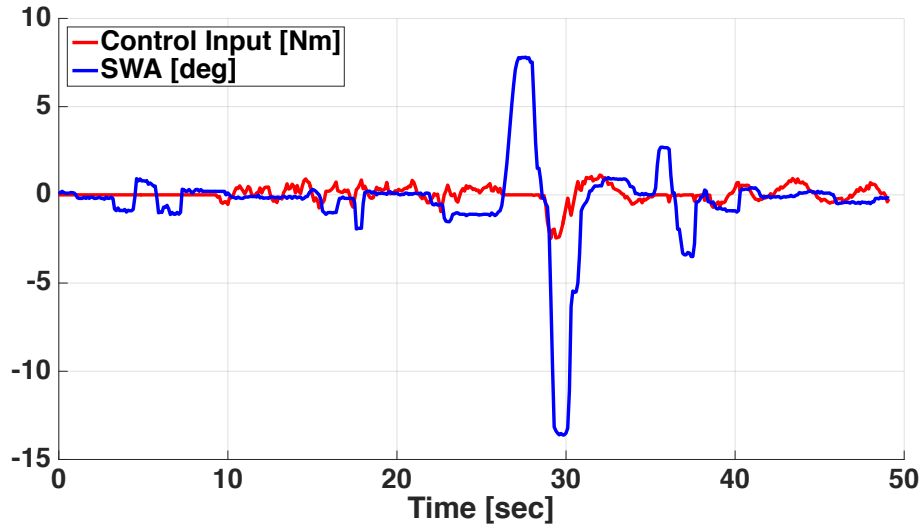


Figure 7.17 : The steering wheel angle and the control input.

hands off states of the SVM classifier. Finally we present the plots of the hand wheel

position and the feedback forces that is calculated by the controller in Figure 7.17. The system exerts more torque input when the driver is out of the loop in order to steer the vehicle in lane change maneuvers. However, when the driver is in the loop and the steering actions of driver does not cause any safety issues, the system utilizes the minimum control input for the comfort and convenience of the driver.

7.7 Discussions and Conclusions

The focus on this chapter is to formulate a unified control structure based on model predictive control framework which is designed for human-centered and safe automated driving applications. The proposed control structure integrates several concepts such as modeling the physical interactions taking place at the neuromuscular level of the driver, an online parameter identification scheme to update the prediction model of MPC, a binary classification algorithm to detect the presence of driver's hands on the wheel and a HMM based driver intention module to estimate the lateral trajectory of the driver based on the history of driving behavior.

We first test the controller in a distracted driving scenario where the driver unintentionally steers to the road boundaries. The proposed system intervenes by applying feedback forces to the steering wheel in order to prevent road departure. We perform experiments in two distinct driver behavior, compliant and stiff, and observe the adaptive behavior of the controller under different levels of interaction. The results validate the effectiveness of updating the prediction model of the MPC for better control performance.

Next, we perform an experimental study with five participants to validate the performance. The results show that the assistance system adapts the guidance torque systematically with respect to the interaction level. This approach could be a practical method to resolve conflicts between the driver and the controller while performing a steering task cooperatively. However, it is important to point out that current sample size is a limitation to justify the statistical reliability of the results.

Finally, we show the results on human-centered automated driving application where the driver is free to take its hands off the steering wheel for full autonomous driving. Since we use a machine learning approach for classification of the driver's presence,

an extra capacitive hardware is not needed. We emphasis the importance of using a driver intention module for minimizing the conflicts between the driver and the system. The experimental results justify the effectiveness of the proposed unified control architecture in shared automated driving scenarios.

8. LATERAL CONTROL OF AN AUTONOMOUS VEHICLE IN VARIOUS ROAD CONDITIONS

8.1 Introduction

The technological development of motor driven power steering (MDPS) systems enables the implementation of automated driving features, such as lane keeping/changing assistance to automated collision avoidance, for most of the vehicles in market [87]. One of the key element in implementation of such automated driving features is the steering control framework which should provide robustness in all road conditions while allowing the driver to intervene during autonomous functioning. The authors in [35] investigated the control performance of lane tracking systems that are based on steering angle/torque control. It is concluded that both methods provide decent tracking characteristics but the *torque based control* has advantages such as the self-aligning torque and the driver torque are considered in the controller. Also the driver can easily intervene/override the system's activity.

Model predictive control (MPC) framework has been extensively used in automated driving applications. The previous works in general use the steering angle as the control input and therefore neglecting the steering actuator dynamics in the prediction model [45, 46]. The authors in [88] propose to use an online system identification to estimate the tire cornering stiffness values of the bicycle model in order to improve the prediction accuracy. On the other hand, an offline system identification is used to identify the closed-loop dynamics of the steering actuator (i.e., from desired steering angle to actual steering angle) in [76]. The identified model is augmented with the bicycle model and results showed an improved tracking performance. When the actual input of the steering actuator is torque, one should use a low-level controller to calculate the required torque to track the desired steering angle reference. When the actual control input and optimized control input does not match, it might affect the control performance because of the additional dynamics and nonlinearities introduced by the low-level controller.

This chapter is based on the work of the author which is published in the proceedings of IEEE International Conference on Vehicular Electronics and Safety (ICVES) in 2017 [89]. We focus on the steering control of an autonomous vehicle by using MPC framework which optimizes the steering torque of the MDPS system. Since the optimized control input is the actual actuator input, there is no need for any extra low-level controllers and saturation functions after the control input is calculated by MPC. MPC-based controllers form an optimization problem and solve it to find an optimal open-loop control sequence which minimizes the objective function while satisfying all the state and input constraints. The control performance depends on the prediction accuracy of future state trajectories which are obtained by employing a model of the plant, (i.e., prediction model) in the controller. In order to increase the prediction accuracy, we propose to use a regression model based on previous input-output values whose parameters are identified by a recursive least squares (RLS) algorithm. The resulting model is simple and adapts to the varying road conditions.

8.2 The Predictive Controller Architecture

The block diagram of the proposed approach is illustrated in Figure 8.1. A *Trajectory Generator* calculates a global/local state trajectories for the autonomous vehicle which might be infeasible for the vehicle to track. The controller, *MPC*, optimizes the control input in order to track the state trajectories as close as possible while considering the vehicle and actuator dynamics. The steering torque T_c is generated by the steering actuator and applied to the vehicle. A set of sensors are assumed to be available to provide all the necessary measurements $\boldsymbol{x}$.

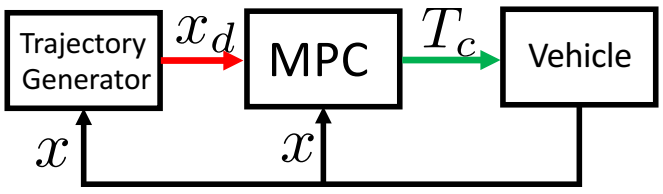


Figure 8.1 : The block diagram for the proposed control method.

8.2.1 Prediction models

Employing a high fidelity model enhances the prediction accuracy, the optimization problem could become too complex to solve and it may not guarantee real-time control

performance. Therefore, simple but accurate prediction model is desired in order to decrease the computational cost while improving the control performance.

We present the following prediction models, which are employed in MPC framework, in order to perform comparison. In both models, we assume that the acceleration command a_x is provided from a low-level speed controller since the main focus of the work is the control of lateral dynamics.

8.2.1.1 Conventional prediction model

The conventional prediction model is derived here in order to compare the performance with the proposed approach. We assume that the vehicle operates in linear region of the tire forces (i.e., small slip angle assumption) and therefore a linear tire model, introduced in Section 3.4.3, is used.

The conventional prediction model consists of the nonlinear bicycle model introduced in Section 3.2.2 where the road-aligned reference frame, given in (3.14) and the steering model in (3.33) is discretized with a fixed sampling period Δt_c and compactly written as:

$$\xi_c(k+1) = f_c(\xi_c(k), u(k), d(k)), \tag{8.1}$$

where the state vector is $\boldsymbol{\xi}_c = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s, \theta, \dot{\theta}]^\intercal$, the control input is $u = T_c$ and the exogenous input vector is $\boldsymbol{d} = [\kappa, a_x]^\intercal$. The driver torque T_d is assumed to be zero since the vehicle operates autonomously.

8.2.1.2 Proposed prediction model

Instead of using the simplified steering dynamics model in (3.33), we propose to use a data driven model for the steering dynamics which adapts itself with respect to current driving conditions. Since the steering dynamics is represented by a second order system, we propose to use a regression model, which is introduced in (A.1) with $m = n = 2$, whose parameters will be identified recursively by using *EFRA* method introduced in (A.12). The proposed control approach with data-driven prediction model is illustrated in Figure 8.2.

$$\theta_{k+1} = \hat{b}_k + \hat{\beta}_{1,k}\theta_k + \hat{\beta}_{2,k}\theta_{k-1} + \hat{\gamma}_{1,k}u_k + \hat{\gamma}_{2,k}u_{k-1}. \tag{8.2}$$

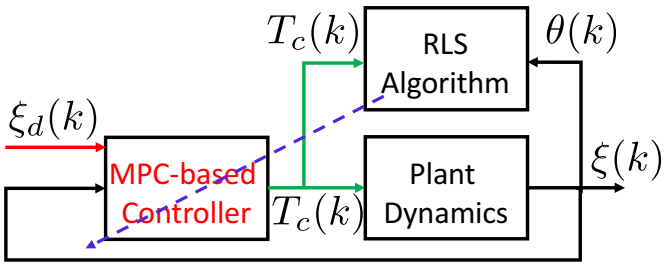


Figure 8.2 : The block diagram for the proposed data-driven controller.

The vehicle model in (3.10)-(3.14) is discretized with a fixed sampling period Δt_c and integrated with (8.2) to yield the prediction model of the proposed controller given as:

$$\boldsymbol{\xi}_p(k+1) = f_p(\boldsymbol{\xi}_p(k), u(k), \boldsymbol{d}(k)), \tag{8.3}$$

where the state vector is $\boldsymbol{\xi}_p = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s, \theta]^\intercal$, the control input is $u = T_c$ and the exogenous input vector is $\boldsymbol{d} = [\kappa, \hat{\varphi}, a_x]^\intercal$.

8.2.2 Constraints

The main objective of the controller is to track the desired lateral trajectory while keeping the vehicle in road and improving the lateral stability for all road conditions. We define the following safety constraints imposed on the vehicle states in Section 8.2.2.1 and the actuator constraints in Section 8.2.2.2.

8.2.2.1 Safety constraints

Keeping the vehicle within the road is achieved by imposing upper and lower bounds on the lateral position e_y as:

$$e_{y,\min} \leq e_{y,i} \leq e_{y,\max}, \qquad i = 1 \ldots N, \tag{8.4}$$

where $e_{y,\min}$ and $e_{y,\max}$ denote the limits of the drivable part of the road. $e_{y,i}$ denotes the lateral position prediction of the vehicle at prediction index i. The prediction horizon is denoted by N.

We also impose slip angle constraints on the vehicle dynamics to enforce the slip angles to stay in the linear region of tire forces.

$$\alpha_{\min}(\mu) \leq \alpha_{j,i} \leq \alpha_{\max}(\mu), \quad j \in \{f, r\}, \quad i = 1 \ldots N, \tag{8.5}$$

where $\alpha_{\min}(\mu)$ and $\alpha_{\max}(\mu)$ denote the upper and lower bounds on slip angle depending on the road friction coefficient. $\alpha_{j,i}$ denotes the slip angle prediction of tire j at prediction index i.

The safety constraints in (8.4) - (8.5) are compactly written as:

$$h^{\xi}(\xi_{i+1}, u_i) \leq \mathbf{0} \quad i = 0 \ldots N-1, \tag{8.6}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

8.2.2.2 Actuator constraints

We enforce the following magnitude and the rate constraints on the actuator input $u = T_c$ in order to utilize a smooth and adequate level of control.

$$T_{c,\min} \leq T_{c,i} \leq T_{c,\max}, \tag{8.7a}$$

$$\Delta T_{c,\min} \leq \Delta T_{c,i} \leq \Delta T_{c,\max}, \quad i = 0 \ldots N-1, \tag{8.7b}$$

where $T_{c,j}$ and $\Delta T_{c,j}$, $j \in \{\min, \max\}$ denote the boundaries on the magnitude and rate of control input, respectively.

The control input constraints (8.7) could be compactly written as:

$$h^{u}(u_i, \Delta u_i) \leq \mathbf{0}, \quad i = 0 \ldots N-1, \tag{8.8}$$

where $\mathbf{0}$ is a vector of zeros with appropriate dimension.

8.2.3 Optimal control problem

The constrained finite time optimal control (CFTOC) problem in (8.9) is solved at each discrete time step k to compute an optimal control input sequence $\boldsymbol{u}_k$. Only the first control input is applied to the vehicle, and this optimization routine is repeated at the next time step using new state measurements as initial conditions.

$$\min_{\boldsymbol{u}_k, \varepsilon} \sum_{i=0}^{N-1} \left(\|u_{i|k}\|_Q^2 + \|\Delta u_{i|k}\|_R^2 \right) \tag{8.9a}$$

$$+ \sum_{i=1}^{N} \|\xi_{i|k} - \xi_{i|k}^{ref}\|_S^2 + W\varepsilon$$

$$\text{s.t. } \xi_{i+1|k} = f(\xi_{i|k}, u_{i|k}, d_k), \tag{8.9b}$$

$$h^{\xi}(\xi_{i+1|k}, u_{i|k}) \leq \varepsilon, \quad \varepsilon \geq \mathbf{0} \tag{8.9c}$$

$$h^{u}(u_{i|k}, \Delta u_{i|k}) \leq \mathbf{0}, \tag{8.9d}$$

$$i = 0 \ldots N-1$$

$$\xi_{0|k} = \xi(k), \quad u_{-1|k} = u(k-1), \tag{8.9e}$$

where $\xi_{i|k}$ denotes the ith state prediction at time step k, obtained by applying the optimal control input sequence $\boldsymbol{u}_k = \{u_{0|k}, u_{1|k}, \ldots, u_{N-1|k}\}$ to the discrete-time dynamics (8.9b) starting from the measured state at the current time step (8.9e). The cost function in (8.9a) reflects control objectives of utilizing smooth and minimal control action while tracking the desired lateral trajectory. We penalize the magnitude and rate of control input as well as the tracking error, using corresponding weight matrices Q, R and S, respectively. The input constraints are given in (8.9d). We introduce slack variable vector ε for the safety constraints (8.9c) in order to keep the optimization problem feasible in case of constraint violations due to model mismatch. A large matrix W strongly penalizes constraint violation.

8.3 Simulation Environment and Results

We validate the performance of the proposed control framework through simulations in Matlab/Simulink. The simulation environment properties are summarized in Table 8.1.

Table 8.1 : Simulation environment.

Dynamics	***Actual Dynamics (plant model)***	***Conventional prediction model***	***Proposed prediction model***
Vehicle dynamics	Bicycle model (3.10)-(3.14)	Bicycle model (3.10)-(3.14)	Bicycle model (3.10)-(3.14)
Steering dynamics	Simplified steering model (3.33)	Simplified steering model (3.33)	Data driven model (8.2)
Tire dynamics	Fiala (Brush) tire model (3.20)	Linear tire model (3.26)	Linear tire model (3.26)
Sampling Period	10^{-4} s	10^{-1} s	10^{-1} s
Prediction horizon	n.a.	2 s	2 s

We compare two MPC-based controllers where *conventional controller* employs the prediction model in (8.1) and *proposed controller* uses the prediction model in (8.3). The parameters of the predictive controllers are listed in Table 8.2. The vehicle model parameters are given in Appendix. The optimal control problem in (8.9) is solved using the general purpose nonlinear solver NPSOL [80].

The lateral tire forces and the alignment moments are illustrated in Figure 8.3 and they are calculated by using the Fiala tire model in (3.20) with the parameters given

Table 8.2 : MPC design parameters.

Parameter	**Value**	**Parameter**	**Value**
Δt_c	0.1 s	Δt_{sim}	$10^{-4}s$
$T_{c,\min}$	-8 Nm	$\Delta T_{c,\min}$	-16 Nm/s
$T_{c,\max}$	8 Nm	$\Delta T_{c,\max}$	16 Nm/s
$e_{y,\min}$	-0.67 m	$e_{y,\max}$	4.07 m
w_{lane}	3.4 m	Q,R	$(1,10)$
S	**diag**$(0,0,0,$ $0,5,0,0)$	W	**diag**$(10^5,10^5)$
		N	20

in Table B.1. The figure shows different characteristics of lateral tire forces and alignment moments with respect to various road conditions which are characterized by the road friction coefficient μ. We have four different road conditions where $\mu = \{1, 0.6, 0.4, 0.2\}$ characterizes dry, wet, snowy and icy road conditions, respectively.

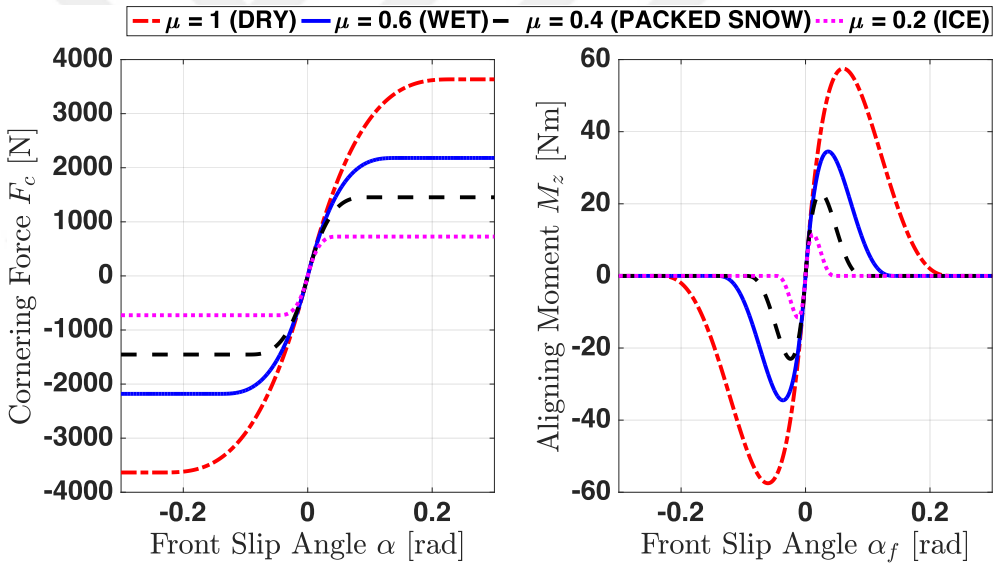


Figure 8.3 : The resulting lateral tire forces F_c and the alignment moments M_z as a function of road friction coefficient μ.

Since we assume that the vehicle operates in linear region of the tire forces, it is important to determine the linear region in terms of maximum and minimum slip angle values as a function of μ. It is observed that the linear region of the lateral tire force shrinks as the road becomes more slippery. Depending on the road condition, the MPC-based controller updates the slip angle constraints given in (8.5). The max. and min. limits of the slip angles are given in Table 8.3.

Table 8.3 : Slip angle limits as a function of road friction coefficient.

Road Condition	μ	$\alpha_{\max,\min}$ **[rad]**
Dry Road	1	± 0.05 rad.
Wet Road	0.6	± 0.035 rad.
Snowy Road	0.4	± 0.02 rad.
Icy Road	0.2	± 0.01 rad.

In the beginning of each simulation, we initialize the parameters of the model given in (8.2) as $\varphi_0 = [0,0,0,0.01,0]^\intercal$ in order to introduce a high modeling uncertainty. We employ the *EFRA* method for recursive identification of the unknown parameters of the model which runs at a sampling period of Δt_c.

8.3.1 Lane change maneuver

We compare the controllers under different road conditions such as dry road ($\mu = 1$), wet road ($\mu = 0.6$), packed snow road ($\mu = 0.4$) and icy road ($\mu = 0.2$) in a automated lane change scenario where the road is assumed to be straight ($\kappa = 0$). Throughout the simulation, we assume a constant longitudinal velocity of 25 m/s and therefore the longitudinal tire force components are zero. The lane changes are triggered by an external signal which is provided to the controllers.

Here we present the results of the lane change maneuver simulation. The lane change reference is triggered with a period of 10 s. The future values of the lateral reference is assumed to be constant with the current value of reference (i.e., $e_y^d(i|k) = e_y^d(k), \quad i = 0 \ldots N-1$). Therefore the control algorithm lacks of anticipation in reference tracking because of the unknown future reference values.

We illustrate the lateral position plots of both controllers in Figure 8.4 for various road conditions. It is seen that *proposed controller* achieves a better tracking performance in terms of less overshoot and a faster settling time than the *conventional controller*. It is important to notice that the tracking performance of the proposed approach improves as the parameters converge to their correct values.

We also illustrate the comparison of lateral tracking performance of each controller with respect to different road conditions in Figure 8.5. Since the longitudinal velocity is fixed to 25 m/s (i.e., no braking/acceleration), both controllers behave differently

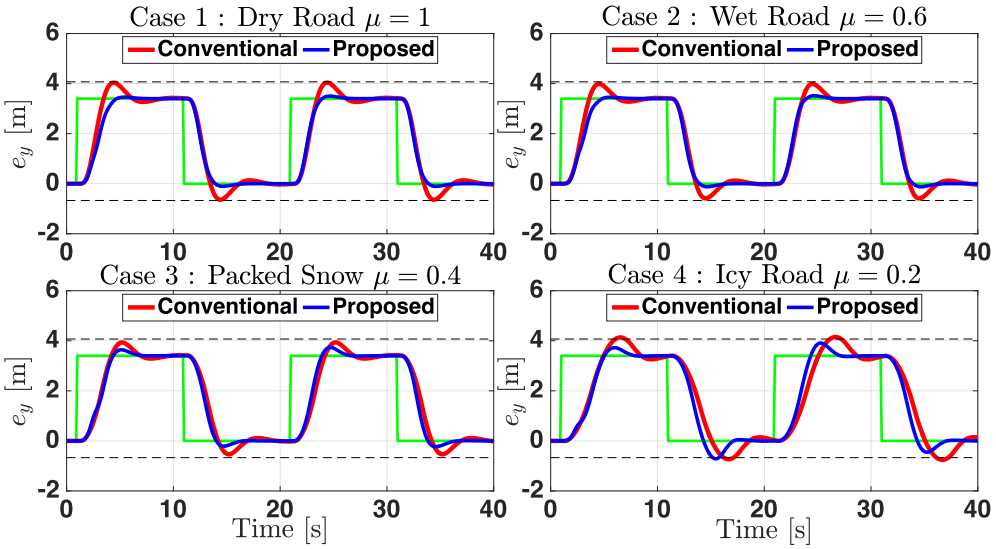


Figure 8.4 : The comparison of lateral tracking performance of both controllers in various road conditions.

depending on the road conditions. It is observed that as the road becomes more slippery the settling time and overshoot increases.

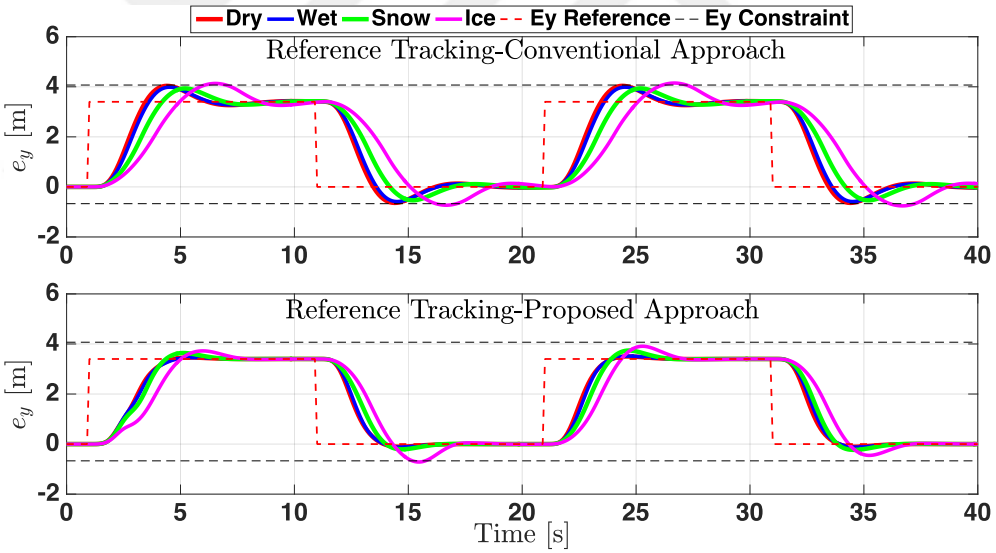


Figure 8.5 : The intra-comparison of lateral tracking performance of each controller in various road conditions.

The *proposed controller* achieves less control input utilization with respect to the 1-norm of the control signal. This is shown in Table 8.4.

Finally, we illustrate the front slip angle plots in Figure 8.6 where we compare the resulting slip angles in various road conditions. It is observed that *conventional controller* satisfies the slip angle constraints for all conditions. However *proposed controller* fails to satisfy the slip angle constraints especially when the road type is

Table 8.4 : Control input utilization of both controllers.

Norm	Dry road	Wet road	Snowy road	Icy road
Conventional $\|\|T_c\|\|_1$	319.1 Nm	280.9 Nm	260.1 Nm	271.53 Nm
Proposed $\|\|T_c\|\|_1$	251.2 Nm	226.6 Nm	227.2 Nm	199.5 Nm

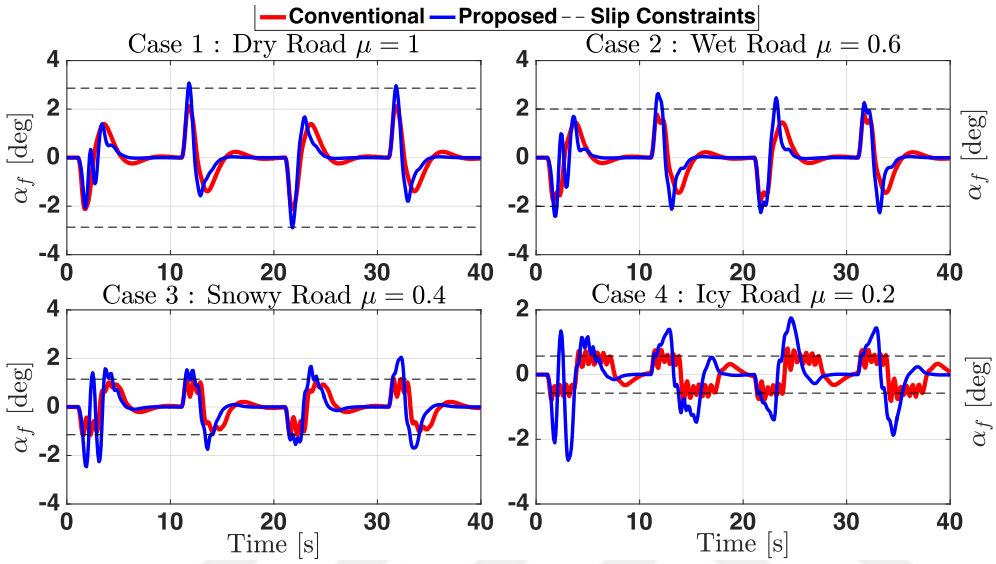


Figure 8.6 : The results of the front slip angles of both controllers in various road conditions.

icy. This could be attributed to the modeling approach that is used for the *proposed controller*. The alignment torque term is linearly dependent to front slip angle and this relation is preserved in *Conventional controller*. But nevertheless, the lateral stability of the vehicle is maintained even if the linear region of the tire forces are violated for the *proposed controller*. In order to have a better tracking performance, the proposed controller sacrifices the slip angle constraints.

8.3.2 Sinusoidal reference tracking

We also simulated a sinusoidal reference tracking scenario in various road conditions in order to compare the performance of the *Conventional controller* and *Proposed controller*. The lateral position reference is selected as $e_y^d(t) = K\sin(2\pi f_d t)\,[m]$ where $K = 1.7$ and $f_d = 0.1$ Hz. Throughout the simulation, we assume a constant longitudinal velocity of 25 m/s and therefore the longitudinal tire force components are zero. The vehicle initially starts at a lateral position of $e_y(0) = 2$ m.

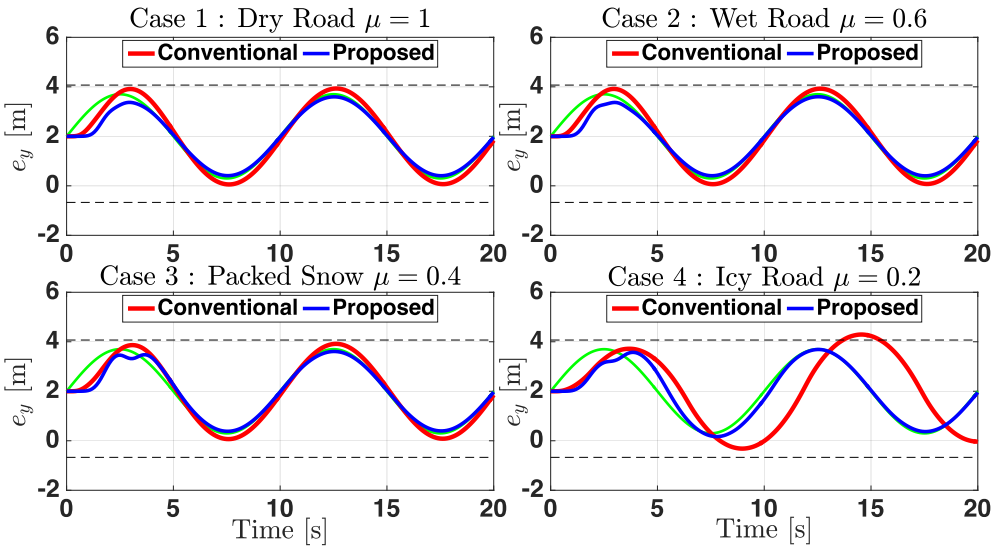


Figure 8.7 : The comparison of lateral tracking performance of both controllers in various road conditions.

We illustrate the sinusoidal reference tracking results in Figure 8.7. Both controllers have a close tracking performance while satisfying the safety constraints in dry, wet and snowy road conditions. However, the *conventional controller* has a degraded tracking performance and the constraints are violated in the icy road condition. On the other hand, the *proposed controller* improves the tracking performance in each case as the parameters converge to their correct values. We compare the tracking performance of each controller by calculating the root-mean square (RMS) error values where $e_{RMS} \triangleq \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(e_y^d(k) - e_y(k)\right)^2}$. The RMS of lateral tracking error is given in Table 8.5.

Table 8.5 : The RMS of lateral tracking error of both controllers in various road conditions.

e_{RMS}	**Dry road**	**Wet road**	**Snowy road**	**Icy road**
Conventional	0.198 m	0.190 m	0.184 m	1.542 m
Proposed	0.082 m	0.078 m	0.086 m	0.210 m

8.4 Discussions and Conclusions

We present an adaptive and predictive control framework where we focus on improving the lateral stability and the tracking performance of an autonomous vehicle operating in various road conditions. While most of the approaches in literature ignore the steering

dynamics and use the steering angle as the control input, we formulate the predictive controller to optimize the actual control input (i.e., torque input to steering actuator). We propose to use a data-driven model based on the previous input-output values of the steering system where the parameters are identified online by a recursive least squares algorithm. First, we validate the system identification method with our test vehicle using a sine sweep test signal while traveling at different speeds. Then, we test and compare the performance of the proposed model with a conventional model through simulations. We focus on the performance of both formulations in various road conditions and the results show that the proposed approach enhances the lateral stability and tracking while utilizing less control action.

9. CONCLUDING REMARKS

The research presented in this thesis focuses on designing a control algorithm for the human-machine system in automated, shared driving applications. More specifically, the thesis discusses the current design methodologies of the advanced driver assistance systems (ADAS) which supports the steering actions of the driver through feedback forces generated by an electric motor in the steering wheel.

One of the most important problem is to design the feedback forces with a high driver acceptance such that the conflicts are minimized and the driver's mental and physical load is reduced. We consider the haptic shared control methodology as the guidelines for designing such a human-machine system. The previous studies show that a simple control structure struggles to generalize the feedback forces and therefore they are designed individually. This thesis aims to investigate the control design problems of the driver steering assistance system (DSAS).

In the context of control approach, we design the controller of the DSAS by using model predictive control (MPC) strategy. The proposed controller optimizes the feedback forces depending on the interaction level of the driver and the system, both in cognitive and neuromuscular levels. In MPC strategy,the control law is optimized by minimizing a user-defined objective function in a finite-time prediction horizon while satisfying all the constraints. This approach has several advantages. The state trajectory predictions are functions of a finite-time control input sequence (to be optimized) and a mathematical model of the dynamics. Therefore it is a model-based control approach with optimization procedure at each time interval. On the other hand, the state and input constraints are considered in the optimization problem before the control law is calculated.

This work has contributions on the modeling of the driver-steering dynamics as well as integrating an online parameter identification scheme in order to track the time-varying interaction level between the driver and the feedback forces. The prediction model

of the MPC is updated based on the identified parameters. This enables to have an adaptive feedback forces based on the driver's preferences at the muscular level.

We also propose the control design of a human-centered steering assistance system which switches its control strategy from assisting to full autonomous driving mode with respect to driver's presence on the steering task. By estimating the driver's desired trajectory, the proposed controller can complete the maneuvers which are initiated by the driver. These properties open up new perspectives in designing a human-centered assistance systems where driver can manipulate the control objectives of the system through haptics.

In the context of implementation, the proposed system is evaluated in a real-world driving scenarios with a production-level vehicle which is modified for research purposes. We first conduct experiments of distracted driving scenarios such as unintentional road/lane departure and high-risk lane change maneuvers under various interaction levels. Then we perform an experimental study to evaluate the performance of the proposed system with multiple human participants. The system is able to capture the participants' intentions individually and optimize the feedback forces accordingly. Finally, we implement the unified control structure which includes a hands on/off classifier based on SVM and a high-level driver model based on HMM. The unified approach is validated through experiments and the results show the effectiveness of the system.

We conclude this dissertation with a summary of the suggestions of the author for the possible future directions.

9.1 Final Discussions and Future Directions

Although the proposed system is implemented and validated in real-time applications, there is a lack of proofs on stability of the closed loop system. It is a known fact that a stability proof for a nonlinear MPC is very limited to specific conditions under several assumptions. However, one can extend the proposed framework by integrating a terminal state constraint and a terminal state cost term in order to guarantee a convergence and stability for the finite-horizon.

Nevertheless, the stability of the closed loop system is obtained by a proper tuning of the cost weight terms and selecting an appropriate prediction horizon. Although this approach does not guarantee stability in every case, the simulation and the experimental results show that the closed-loop system operates in the stable region for the automated driving scenarios.

I would like to point out the fact that the tuning process of the weights in the objective function depends on heuristic methods where the highest priority of the controller is assigned to a relatively larger value than the other cost terms. But one could discuss on implementing a systematical way to tune these weights. Moreover, these weights could be an optimization variable in the optimal control problem and they might be optimized with respect to a given objective.

On the other hand, the selection of the prediction horizon is more straightforward. If the prediction model is accurate enough, then choosing a long prediction horizon gives better results. But the numeric value of the prediction horizon depends on the computational power and the size of the constrained optimization problem. When the prediction horizon increases, the size of the optimization problem increases which yields high computational times. For real-time performance of dynamic systems, one should consider having small sampling intervals (i.e., $T_s \leq 0.1$ s). Therefore it is important to be aware of this trade-off between a larger horizon and a small sampling frequency. Because of our problem size and computational power, we conduct the experiments with a prediction horizon of two seconds.

As the next step of our research we would like to conduct an experimental study with a larger sample set in order to assess the generalizability and the reliability of the proposed DSAS. Also we would like to investigate the system's adaptation to driver's preferences in a long-term driving scenario and show how the driver's experience might affect the assistance system.

More applications could be done by integrating the longitudinal dynamics or a high level motion planner for full autonomous navigation which includes lane changing and merging behaviors.

REFERENCES

[1] **Sivaraman, S. and Trivedi, M.M.** (2013). Towards cooperative, predictive driver assistance, *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp.1719–1724.

[2] **Abbink, D.A. and Mulder, M.** (2010). Neuromuscular analysis as a guideline in designing shared control, *Advances in Haptics*, 499–517.

[3] **Camacho, E.F. and Alba, C.B.** (2007). *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing, Springer-Verlag London, 2 edition.

[4] **Pick, A. and Cole, D.J.** (2007). Dynamic properties of a driver's arms holding a steering wheel, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *221*(12), 1475–1486.

[5] **Gray, A.**, (2013), *An Integrated Framework for Planning and Control of Semi-Autonomous Vehicles* (Ph.D. thesis), university of California, Berkeley.

[6] **Toroyan, T. and et al.** (2013). Global status report on road safety 2013: supporting a decade of action, **Technical Report**, World Health Organization.

[7] **Blosseville, J.M.** (2005). Driving assistance systems and road safety: State-of-the-art and outlook, *Annales Des Télécommunications*, *60*(3), 281–298.

[8] **Navarro, J.**, **Mars, F. and Young, M.S.** (2011). Lateral control assistance in car driving: classification, review and future prospects, *IET Intelligent Transport Systems*, *5*(3), 207–220.

[9] **Brandt, T.**, **Sattel, T. and Bohm, M.** (2007). Combining haptic human-machine interaction with predictive path planning for lane-keeping and collision avoidance systems, *2007 IEEE Intelligent Vehicles Symposium*, pp.582–587.

[10] **Inagaki, T.** (2003). Adaptive Automation for Comfort and Safety, *International Journal of ITS Research*, *1*(1).

[11] **Itoh, M. and Inagaki, T.** (2014). Design and evaluation of steering protection for avoiding collisions during a lane change, *Ergonomics*, *57*(3), 361–373.

[12] **Abbink, D.A.**, **Mulder, M. and Boer, E.R.** (2012). Haptic shared control: smoothly shifting control authority?, *Cognition, Technology & Work*, *14*(1), 19–28.

[13] **Alirezaei, M.**, **Corno, M.**, **Katzourakis, D.**, **Ghaffari, A. and Kazemi, R.** (2012). A Robust Steering Assistance System for Road Departure Avoidance, *IEEE Transactions on Vehicular Technology*, *61*(5), 1953–1960.

[14] **Katzourakis, D.**, **Alirezaei, M.**, **Winter, J.C.F.D.**, **Corno, M.**, **Happee, R.**, **Ghaffari, A. and Kazemi, R.** (2011). Shared control for road departure prevention, *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1037–1043.

[15] **Gray, A.**, **Ali, M.**, **Gao, Y.**, **Hedrick, J.K. and Borrelli, F.** (2012). Integrated threat assessment and control design for roadway departure avoidance, *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp.1714–1719.

[16] **Gray, A.**, **Ali, M.**, **Gao, Y.**, **Hedrick, J.K. and Borrelli, F.** (2013). A Unified Approach to Threat Assessment and Control for Automotive Active Safety, *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1490–1499.

[17] **Katzourakis, D.**, **Alirezaei, M.**, **Winter, J.C.F.D.**, **Corno, M.**, **Happee, R.**, **Ghaffari, A. and Kazemi, R.** (2011). Shared control for road departure prevention, *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1037–1043.

[18] **Steele, M. and Gillespie, R.B.** (2001). Shared Control between Human and Machine: Using a Haptic Steering Wheel to Aid in Land Vehicle Guidance, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *45*(23), 1671–1675.

[19] **Panat Leelavansuk, H.Y. and Masao** (2003). Cooperative Steering Characteristics of Driver and Lane-Keeping Assistance System, *International Journal of ITS Research*, *1*(1).

[20] **Griffiths, P. and Gillespie, R.B.** (2004). Shared control between human and machine: haptic display of automation during manual control of vehicle heading, *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings. 12th International Symposium on*, pp.358–366.

[21] **Mulder, M.**, **Abbink, D.A. and Boer, E.R.** (2008). The effect of haptic guidance on curve negotiation behavior of young, experienced drivers, *2008 IEEE International Conference on Systems, Man and Cybernetics*, pp.804–809.

[22] **Enache, N.M.**, **Netto, M.**, **Mammar, S. and Lusetti, B.** (2009). Driver steering assistance for lane departure avoidance, *Control Engineering Practice*, *17*(6), 642 – 651.

[23] **Tsoi, K.K.**, **Mulder, M. and Abbink, D.A.** (2010). Balancing safety and support: Changing lanes with a haptic lane-keeping support system, *2010 IEEE International Conference on Systems, Man and Cybernetics*, pp.1236–1243.

[24] **Saleh, L.**, **Chevrel, P.**, **Claveau, F.**, **Lafay, J.F. and Mars, F.** (2013). Shared Steering Control Between a Driver and an Automation: Stability in the Presence of Driver Behavior Uncertainty, *IEEE Transactions on Intelligent Transportation Systems*, *14*(2), 974–983.

[25] **Sentouh, C.**, **Soualmi, B.**, **Popieul, J.C. and Debernard, S.** (2013). Cooperative Steering Assist Control System, *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp.941–946.

[26] **Iwano, K.**, **Raksincharoensak, P. and Nagai, M.** (2014). A Study on Shared Control between the Driver and an Active Steering Control System in Emergency Obstacle Avoidance Situations, *IFAC Proceedings Volumes*, *47*(3), 6338–6343.

[27] **Nishimura, R.**, **Wada, T. and Sugiyama, S.** (2015). Haptic Shared Control in Steering Operation Based on Cooperative Status Between a Driver and a Driver Assistance System, *arXiv preprint arXiv:1504.05770*.

[28] **Zafeiropoulos, S. and Tsiotras, P.** (2014). Design of a lane-tracking driver steering assist system and its interaction with a two-point visual driver model, *2014 American Control Conference*, pp.3911–3917.

[29] **Penna, M.D.**, **van Paassen, M.M.**, **Abbink, D.A.**, **Mulder, M. and Mulder, M.** (2010). Reducing steering wheel stiffness is beneficial in supporting evasive maneuvers, *2010 IEEE International Conference on Systems, Man and Cybernetics*, pp.1628–1635.

[30] **Mulder, M. and Abbink, D.A.** (2011). Correct and faulty driver support from shared haptic control during evasive maneuvers, *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1057–1062.

[31] **Abbink, D.A.**, **Cleij, D.**, **Mulder, M. and v. Paassen, M.M.** (2012). The importance of including knowledge of neuromuscular behaviour in haptic shared control, *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.3350–3355.

[32] **Boink, R.**, **van Paassen, M.M.**, **Mulder, M. and Abbink, D.A.** (2014). Understanding and reducing conflicts between driver and haptic shared control, *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.1510–1515.

[33] **Forsyth, B.A.C. and Maclean, K.E.** (2006). Predictive haptic guidance: intelligent user assistance for the control of dynamic tasks, *IEEE Transactions on Visualization and Computer Graphics*, *12*(1), 103–113.

[34] **Winter, J.C.F.D. and Dodou, D.** (2011). Preparing drivers for dangerous situations: A critical reflection on continuous shared control, *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1050–1056.

[35] **Nagai, M.**, **Mouri, H. and Raksincharoensak, P.** (2002). Vehicle Lane-Tracking Control with Steering Torque Input, *Vehicle System Dynamics*, *37*(sup1), 267–278.

[36] **Soualmi, B.**, **Sentouh, C.**, **Popieul, J. and Debernard, S.** (2013). A shared control driving assistance system : interest of using a driver model in both lane keeping and obstacle avoidance situations, *IFAC Proceedings Volumes*, *46*(15), 173 – 178.

[37] **Oufroukh, N.A. and Mammar, S.** (2014). Integrated driver co-pilote approach for vehicle lateral control, *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp.1163–1168.

[38] **Nguyen, A.T.**, **Sentouh, C.**, **Popieul, J.C. and Soualmi, B.** (2015). Shared lateral control with on-line adaptation of the automation degree for driver steering assist system: A weighting design approach, *2015 54th IEEE Conference on Decision and Control (CDC)*, pp.857–862.

[39] **Smisek, J.**, **Sunil, E.**, **van Paassen, M.M.**, **Abbink, D.A. and Mulder, M.** (2017). Neuromuscular-System-Based Tuning of a Haptic Shared Control Interface for UAV Teleoperation, *IEEE Transactions on Human-Machine Systems*, *47*(4), 449–461.

[40] **Na, X. and Cole, D.J.** (2013). Linear quadratic game and non-cooperative predictive methods for potential application to modeling driver AFS interactive steering control, *Vehicle System Dynamics*, *51*(2), 165–198.

[41] **Na, X. and Cole, D.J.** (2015). Game-Theoretic Modeling of the Steering Interaction Between a Human Driver and a Vehicle Collision Avoidance Controller, *IEEE Transactions on Human-Machine Systems*, *45*(1), 25–38.

[42] **Na, X. and Cole, D.J.** (2017). Application of Open-Loop Stackelberg Equilibrium to Modeling a Driver's Interaction with Vehicle Active Steering Control in Obstacle Avoidance, *IEEE Transactions on Human-Machine Systems*, *PP*(99), 1–13.

[43] **Flad, M.**, **Otten, J.**, **Schwab, S. and Hohmann, S.** (2014). Steering driver assistance system: A systematic cooperative shared control design approach, *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.3585–3592.

[44] **Flad, M.**, **Fröhlich, L. and Hohmann, S.** (2017). Cooperative Shared Control Driver Assistance Systems Based on Motion Primitives and Differential Games, *IEEE Transactions on Human-Machine Systems*, *PP*(99), 1–12.

[45] **Borrelli, F.**, **Falcone, P.**, **Keviczky, T.**, **Asgari, J. and Hrovat, D.** (2005). MPC-based approach to active steering for autonomous vehicle systems, *International Journal of Vehicle Autonomous Systems*, *3*(2-4), 265–291.

[46] **Anderson, S.J.**, **Peters, S.C.**, **Pilutti, T.E. and Iagnemma, K.** (2010). An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios, *International Journal of Vehicle Autonomous Systems*, *8*(2-4), pp. 190–216.

[47] **Balachandran, A.**, **Brown, M.**, **Erlien, S.M. and Gerdes, J.C.** (2016). Predictive Haptic Feedback for Obstacle Avoidance Based on Model Predictive Control, *IEEE Transactions on Automation Science and Engineering*, *13*(1), 26–31.

[48] **Zafeiropoulos, S. and Di Cairano, S.** (2013). Vehicle yaw dynamics control by torque-based assist systems enforcing driver's steering feel constraints, *American Control Conference (ACC), 2013*, pp.6746–6751.

[49] **Erlien, S.M.**, **Fujita, S. and Gerdes, J.C.** (2016). Shared Steering Control Using Safe Envelopes for Obstacle Avoidance and Vehicle Stability, *IEEE Transactions on Intelligent Transportation Systems*, *17*(2), 441–451.

[50] **Mayne, D.**, **Rawlings, J.**, **Rao, C. and Scokaert, P.** (2000). Constrained model predictive control: Stability and optimality, *Automatica*, *36*(6), 789 – 814.

[51] **Pacejka, H.B.** (2012). *Tire and Vehicle Dynamics (Third Edition)*, Butterworth-Heinemann.

[52] **Hindiyeh, R.Y. and Gerdes, J.C.** (2009). Equilibrium Analysis of Drifting Vehicles for Control Design, *ASME 2009 Dynamic Systems and Control Conference*, 48920, pp.181–188.

[53] **Rajamani, R.** (2006). *Vehicle Dynamics and Control*, Mechanical Engineering Series, Springer US, 1 edition.

[54] **Kim, W.**, **Son, Y.S. and Chung, C.C.** (2015). Torque Overlay-Based Robust Steering Wheel Angle Control of Electrical Power Steering for a Lane-Keeping System of Automated Vehicles, *IEEE Transactions on Vehicular Technology*, *PP*(99), 1–1.

[55] **Marouf, A.**, **Djemai, M.**, **Sentouh, C. and Pudlo, P.** (2012). A New Control Strategy of an Electric-Power-Assisted Steering System, *IEEE Transactions on Vehicular Technology*, *61*(8), 3574–3589.

[56] **Kim, J.H. and Song, J.B.** (2002). Control logic for an electric power steering system using assist motor, *Mechatronics*, *12*(3), 447 – 459.

[57] **Treiber, M.**, **Hennecke, A. and Helbing, D.** (2000). Congested traffic states in empirical observations and microscopic simulations, *Physical review E*, *62*(2), pp. 1805–1824.

[58] **Donges, D.E.** (1978). A Two-Level Model of Driver Steering Behavior, *Human Factors*, *20*(6), 691–707.

[59] **Salvucci, D.D. and Gray, R.** (2004). A Two-Point Visual Control Model of Steering, *Perception*, *33*(10), 1233–1248.

[60] **MacAdam, C.C.** (1981). Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving, *IEEE Transactions on Systems, Man, and Cybernetics*, *11*(6), 393–399.

[61] **Sharp, R.**, **Casanova, D. and Symonds, P.** (2000). A Mathematical Model for Driver Steering Control, with Design, Tuning and Performance Results, *Vehicle System Dynamics*, *33*(5), 289–326.

[62] **Cole, D.J.**, **Pick, A.J. and Odhams, A.M.C.** (2006). Predictive and linear quadratic methods for potential application to modeling driver steering control, *Vehicle System Dynamics*, *44*(3), 259–284.

[63] **Pick, A. and Cole, D.** (2003). Neuromuscular dynamics and the vehicle steering task, *The Dynamics of Vehicles on Roads and on Tracks*, *41*, 182–191.

[64] **Pick, A.J. and Cole, D.J.** (2008). A Mathematical Model of Driver Steering Control Including Neuromuscular Dynamics, *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, *130*.

[65] **Hoult, W. and Cole, D.J.** (2008). A neuromuscular model featuring co-activation for use in driver simulation, *Vehicle System Dynamics*, *46*(sup1), 175–189.

[66] **Cole, D.J.** (2012). A path-following driver–vehicle model with neuromuscular dynamics, including measured and simulated responses to a step in steering angle overlay, *Vehicle System Dynamics*, *50*(4), 573–596.

[67] **Mehrabi, N.**, **Sharif, M. and McPhee, J.** (2012). Study of human steering tasks using a neuromuscular driver model, *Advanced Vehicle and Control Conference (AVEC), Seoul, Korea*.

[68] **Saleh, L.**, **Chevrel, P.**, **Mars, F.**, **Lafay, J.F. and Claveau, F.** (2011). Human-like cybernetic driver model for lane keeping, *IFAC Proceedings Volumes*, *44*(1), 4368 – 4373, 18th IFAC World Congress.

[69] **Sentouh, C.**, **Chevrel, P.**, **Mars, F. and Claveau, F.** (2009). A sensorimotor driver model for steering control, *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp.2462–2467.

[70] **Keen, S.D. and Cole, D.J.** (2011). Application of time-variant predictive control to modeling driver steering skill, *Vehicle System Dynamics*, *49*(4), 527–559.

[71] **Qu, T.**, **Chen, H.**, **Ji, Y.**, **Guo, H. and Cao, D.** (2013). Modeling Driver Steering Control Based on Stochastic Model Predictive Control, *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp.3704–3709.

[72] **Qu, T.**, **Chen, H.**, **Cao, D.**, **Guo, H. and Gao, B.** (2015). Switching-Based Stochastic Model Predictive Control Approach for Modeling Driver Steering Skill, *IEEE Transactions on Intelligent Transportation Systems*, *16*(1), 365–375.

[73] **Marouf, A.**, **Pudlo, P.**, **Sentouh, C. and Djemai, M.** (2014). Identification of human arm viscoelastic properties during vehicle steering maneuver, *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.3342–3347.

[74] **Balachandran, A.**, **Brown, M.**, **Erlien, S.M. and Gerdes, J.C.** (2015). Creating predictive haptic feedback for obstacle avoidance using a model predictive control (MPC) framework, *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp.31–36.

[75] **Griffiths, P. and Gillespie, R.B.** (2005). Sharing Control Between Humans and Automation Using Haptic Interface: Primary and Secondary Task Performance Benefits, *Human Factors*, *47*(3), 574–590.

[76] **Kim, E.**, **Kim, J. and Sunwoo, M.** (2014). Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics, *International Journal of Automotive Technology*, *15*(7), 1155–1164.

[77] **Ercan, Z.**, **Carvalho, A.**, **Tseng, H.E.**, **Gökaşan, M. and Borrelli, F.** (2017). A predictive control framework for torque-based steering assistance to improve safety in highway driving, *Vehicle System Dynamics*, *0*(0), 1–22.

[78] **Ercan, Z.**, **Carvalho, A.**, **Lefèvre, S.**, **Tseng, H.E.**, **Gokasan, M. and Borrelli, F.**, (2016). Torque-based steering assistance for collision avoidance during lane changes, Crc Press, pp.43–48.

[79] **Lefevre, S.**, **Carvalho, A. and Borrelli, F.** (2015). Autonomous car following: A learning-based approach, *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp.920–926.

[80] **Gill, P.**, **Murray, W.**, **Saunders, M. and Wright, M.** (1986). User's Guide for NPSOL (Version 4.0), **Technical ReportSOL-86-2**, Systems Optimization Laboratory, Stanford University, Stanford, CA.

[81] **Katzourakis, D.I.**, **Abbink, D.A.**, **Velenis, E.**, **Holweg, E. and Happee, R.** (2014). Driver's Arms' Time-Variant Neuromuscular Admittance During Real Car Test-Track Driving, *IEEE Transactions on Instrumentation and Measurement*, *63*(1), 221–230.

[82] **Salgado, M.E.**, **Goodwin, G.C. and Middleton, R.H.** (1988). Modified least squares algorithm incorporating exponential resetting and forgetting, *International Journal of Control*, *47*(2), 477–491.

[83] **Lefèvre, S.**, **Carvalho, A. and Borrelli, F.** (2016). A Learning-Based Framework for Velocity Control in Autonomous Driving, *IEEE Transactions on Automation Science and Engineering*, *13*(1), 32–42.

[84] **Lefèvre, S.**, **Carvalho, A.**, **Gao, Y.**, **Tseng, H.E. and Borrelli, F.** (2015). Driver models for personalised driving assistance, *Vehicle System Dynamics*, *53*(12), 1705–1720.

[85] **Kumar, P.**, **Perrollaz, M.**, **Lefèvre, S. and Laugier, C.** (2013). Learning-based approach for online lane change intention prediction, *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp.797–802.

[86] **Ercan, Z.**, **Carvalho, A.**, **Gokasan, M. and Borrelli, F.** (2017). Modeling, Identification, and Predictive Control of a Driver Steering Assistance System, *IEEE Transactions on Human-Machine Systems*, *47*(5), 700–710.

[87] **Eidehall, A.**, **Pohl, J.**, **Gustafsson, F. and Ekmark, J.** (2007). Toward Autonomous Collision Avoidance by Steering, *IEEE Transactions on Intelligent Transportation Systems*, *8*(1), 84–94.

[88] **Chen, B.C.**, **Luan, B.C. and Lee, K.** (2014). Design of lane keeping system using adaptive model predictive control, *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp.922–926.

[89] **Ercan, Z.**, **Gokasan, M. and Borrelli, F.** (2017). An adaptive and predictive controller design for lateral control of an autonomous vehicle, *2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp.13–18.

[90] **Astrom, K.J. and Wittenmark, B.** (1994). *Adaptive Control*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.

APPENDICES

APPENDIX A.1

The least squares (LS) algorithm is used commonly in system identification and signal processing because it is a powerful method to estimate the unknown parameters of a given linear model of the process to be identified. The closed form solution of LS yields an optimal estimate of the unknown parameters such that the sum of the squared error between the actual measurement and the model output is minimized.

In this work, we utilize LS algorithms for linear model of the process which is based on the previous values of output and the input signal. If the output is a scalar variable (i.e., $y \in \mathbb{R}$), then the following linear model can be used.

$$\hat{y}_k = \hat{a}_0 + \sum_{i=1}^{n} \hat{a}_i y_{k-i} + \sum_{j=1}^{m} \hat{b}_j u_{k-j}, \tag{A.1}$$

where y_{k-i} is the i-th time step previous value of the output signal and u_{k-j} is the j-th time step previous value of the input signal. The model output at the current time step k is denoted by $\hat{y}_k$. The unknown coefficients of the output and input values are $\hat{\boldsymbol{a}} = [\hat{a}_0,\ \ldots,\ \hat{a}_n]$ and $\hat{\boldsymbol{b}} = [\hat{b}_1,\ \ldots,\ \hat{b}_m]$.

Remark 12. The unknown coefficient $\hat{a}_0$ corresponds for the constant bias term which is added to the model in order to account for input disturbances.

For simplicity consider the unknown parameter vector (or regression coefficients) $\hat{\boldsymbol{\varphi}} = [\hat{\boldsymbol{a}},\ \hat{\boldsymbol{b}}]$ and the regression vector $\boldsymbol{x}_k = [1,\ y_{k-1},\ \ldots,\ y_{k-n},\ u_{k-1},\ \ldots,\ u_{k-m}]$.

The compact form of the model output $\hat{y}_k$ could be written as:

$$\hat{y}_k = \boldsymbol{x}_k^{\mathsf{T}} \hat{\boldsymbol{\varphi}}, \tag{A.2}$$

where the unknown parameters are denoted by the column vector $\hat{\boldsymbol{\varphi}} = [\hat{\varphi}_0,\ \ldots,\ \hat{\varphi}_{n+m}] \in \mathbb{R}^{n+m+1}$ and the regression vector at time step k is denoted by $\boldsymbol{x}_k = [x_{0,k},\ \ldots,\ x_{n+m,k}] \in \mathbb{R}^{n+m+1}$.

The objective (cost) function to be minimized is defined as:

$$J(\hat{\boldsymbol{\varphi}}, N) = \frac{1}{2} \sum_{k=1}^{N} \left(y_k - \boldsymbol{x}_k^{\mathsf{T}} \hat{\boldsymbol{\varphi}} \right)^2 \tag{A.3}$$

where N is the size of the dataset which consists of previous time-step values of input and output variables from initial time-step to current time-step.

The LS could be written as unconstrained optimization problem in compact form.

$$\min_{\hat{\boldsymbol{\varphi}}} \quad J(\hat{\boldsymbol{\varphi}}, N) = \frac{1}{2} \left(\boldsymbol{y}_N - X_N^{\mathsf{T}} \hat{\boldsymbol{\varphi}} \right)^{\mathsf{T}} \left(\boldsymbol{y}_N - X_N^{\mathsf{T}} \hat{\boldsymbol{\varphi}} \right), \tag{A.4}$$

where $\boldsymbol{y}_N = [y_1,\ \ldots,\ y_N] \in \mathbb{R}^N$ is the vector of all output measurements up to time step N and $X_N = [\boldsymbol{x}_1^{\mathsf{T}}\ \ldots\ \boldsymbol{x}_N^{\mathsf{T}}] \in \mathbb{R}^{(m+n+1)\times N}$ is the matrix of all data vectors up to time step N. In order to solve the LS problem given in (A.4), N should be bigger than the size of the regression vector $\big($i.e., $N > (n+m+1)\big)$.

The closed form solution of the parameter vector $\hat{\boldsymbol{\varphi}}_N^*$, referred as least squares estimate (LSE) of $\hat{\boldsymbol{\varphi}}$ from data of size N, could be found by taking the derivative of the cost function respect to the parameter vector, equating it to zero and solving for the parameter vector.

$$\left.\frac{\partial J(.)}{\partial \hat{\boldsymbol{\varphi}}}\right|_{\hat{\boldsymbol{\varphi}}=\hat{\boldsymbol{\varphi}}_N^*} = 0 \rightarrow \hat{\boldsymbol{\varphi}}_N^* = \left(X_N X_N^\intercal\right)^{-1} X_N \boldsymbol{y}_N. \tag{A.5}$$

This solution is unique if the matrix $\left(X_N X_N^\intercal\right)^{-1}$ is nonsingular. The optimal parameters yield the minimum distance between the measurements and the model estimates.

When the system dynamics vary with respect to the time, one would consider using an adaptive approach to account for the changes in the system dynamics. Therefore forgetting factor λ, where $0 < \lambda < 1$, is introduced in order to weight the recent measurements more than the old measurements by simply manipulating the cost function as follows.

$$J(\hat{\boldsymbol{\varphi}},\, N,\, \lambda) = \frac{1}{2}\sum_{k=1}^{N} \lambda^{N-k}\left(y_k - \boldsymbol{x}_k^\intercal \hat{\boldsymbol{\varphi}}\right)^2 \tag{A.6}$$

The following unconstrained optimization problem given in (A.7) is called the weighted least squares (WLS) and it is written in compact form as:

$$\min_{\hat{\boldsymbol{\varphi}}} \quad J(\hat{\boldsymbol{\varphi}},\, N,\, \lambda) = \frac{1}{2}\left(\boldsymbol{y}_N - X_N^\intercal \hat{\boldsymbol{\varphi}}\right)^\intercal \Lambda \left(\boldsymbol{y}_N - X_N^\intercal \hat{\boldsymbol{\varphi}}\right), \tag{A.7}$$

where $\Lambda \in \mathbb{R}^{N\times N}$ is a diagonal matrix $\left(\text{i.e., } \Lambda = \textbf{diag}[\lambda^{N-1},\, \lambda^{N-2},\, \ldots,\, \lambda,\, 1]\right)$.

The closed form solution of the WLS is again calculated by taking the derivative of (A.7) with respect to $\hat{\boldsymbol{\varphi}}$ and equating it to zero and solving for the optimal parameter vector $\hat{\boldsymbol{\varphi}}_N^*$.

$$\left.\frac{\partial J(.)}{\partial \hat{\boldsymbol{\varphi}}}\right|_{\hat{\boldsymbol{\varphi}}=\hat{\boldsymbol{\varphi}}_N^*} = 0 \rightarrow \hat{\boldsymbol{\varphi}}_N^* = \left(X_N \Lambda X_N^\intercal\right)^{-1} X_N \Lambda \boldsymbol{y}_N. \tag{A.8}$$

There are couple of drawbacks in calculation of the optimal parameter vector $\hat{\boldsymbol{\varphi}}^*$.

1. When the size of the parameter vector $(m+n+1)$ increases, it is computationally intensive to perform matrix inversion such as $\left(X_N \Lambda X_N^\intercal\right)^{-1}$.

2. As the new measurements arrive at each time step, the size of the matrix X_N and the vector Y_N increases. Therefore more memory is needed to keep all the past measurement history.

A more efficient method for calculating the optimal parameters is presented in the next section. This method is based on recursive calculation of the optimal parameters at each time-step.

Recursive Least Squares Method

Recursive least squares (RLS) method provides a more efficient way to calculate the optimal parameter vector because it is formulated in recursive fashion which requires less computation and memory usage.

Assume that at time step $k = N$, we calculate the optimal parameter vector $\hat{\boldsymbol{\varphi}}_N^*$ by using all the N data. When a new measurement is available at $k = N+1$, $\hat{\boldsymbol{\varphi}}_{N+1}^*$ could be calculated by solving the LS problem using $N+1$ data. However in recursive formulation, $\hat{\boldsymbol{\varphi}}_{N+1}^*$ is calculated as an update of $\hat{\boldsymbol{\varphi}}_N^*$ without using all the data. The derivation of the RLS algorithm could be found in [90]. We would like to introduce some important matrices before writing the RLS algorithm. The information matrix at time step k is denoted by R_k and defined by $R_k = \sum_{i=0}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\intercal$. Recursively, it is calculated by $R_{k+1} = R_k + \boldsymbol{x}_k \boldsymbol{x}_k^\intercal$.

The covariance matrix P_k is defined as the inverse of R_k (i.e., $P_k = R_k^{-1}$). Because of its simplicity in calculating the filter gain, the covariance matrix is used in the recursive algorithms.

The RLS algorithms require an initial value of the parameter vector $\hat{\boldsymbol{\varphi}}_0$ and initial value of the covariance matrix P_0. Therefore the only memory usage is just for P_k and $\hat{\boldsymbol{\varphi}}_k$. The recursive formulation of LS problem in (A.4) is given by:

$$\hat{y}_k = \boldsymbol{x}_k^\intercal \hat{\boldsymbol{\varphi}}_{k-1} \tag{A.9a}$$

$$K_k = P_k \boldsymbol{x}_k = \frac{P_{k-1}\boldsymbol{x}_k}{1 + \boldsymbol{x}_k^\intercal P_{k-1} \boldsymbol{x}_k} \tag{A.9b}$$

$$\hat{\boldsymbol{\varphi}}_k = \hat{\boldsymbol{\varphi}}_{k-1} + K_k\big(y_k - \hat{y}_k\big) \tag{A.9c}$$

$$P_k = (I - K_k \boldsymbol{x}_k^\intercal) P_{k-1}, \tag{A.9d}$$

where at each time step k, the model output $\hat{y}_k$ is calculated by using (A.9a). When the new measurement y_k is available, the covariance matrix is updated in (A.9b), the gain is calculated in (A.9b). The parameter vector and the covariance matrix is recursively updated in (A.9c) and (A.9d).

The RLS algorithm presented in (A.9) is suitable for identifying the parameters of a time invariant system. However, in most applications the actual parameters of the system could be time-varying and conventional RLS algorithm fails to track these time-varying parameters since the gain K_k converges to zero (i.e., the covariance matrix P_k tends to zero).

Exponential forgetting is usually a good practice when the actual parameters of the system are slowly changing with the time (i.e. data is non stationary) or when a nonlinear dynamics is approximated by a linear model. Exponential forgetting is achieved by the term λ^{t-k} in the cost function $\big($i.e., see (A.6)$\big)$ where λ is called the forgetting factor. The range of λ is $0 \leq \lambda \leq 1$ but in practice the typical value of λ is $0.95 < \lambda \leq 0.995$. Since λ is smaller than one, the most recent data has higher weight than the old data. As the time passes, the weight on the oldest data converges to zero therefore the algorithm is said to forgot these data. The memory horizon of the RLS algorithm is defined by:

$$T_\lambda = \frac{1}{1-\lambda}, \quad \lambda \neq 1, \tag{A.10}$$

which describes the number of the data that has weight more than one third of unit weight (i.e., when $T_\lambda = t - k \rightarrow \lambda^{T_\lambda} \approx e^{-1}$ where $e^{-1} \approx 0.3679$). For example ; $\lambda =$

$0.98 \rightarrow T_\lambda = 50$ samples and $\lambda = 0.95 \rightarrow T_\lambda = 20$ samples. The recursive formulation of WLS problem in (A.7) is given by:

$$\hat{y}_k = \boldsymbol{x}_k^\intercal \hat{\boldsymbol{\varphi}}_{k-1} \tag{A.11a}$$

$$K_k = P_k \boldsymbol{x}_k = \frac{P_{k-1}\boldsymbol{x}_k}{\lambda + \boldsymbol{x}_k^\intercal P_{k-1}\boldsymbol{x}_k} \tag{A.11b}$$

$$\hat{\boldsymbol{\varphi}}_k = \hat{\boldsymbol{\varphi}}_{k-1} + K_k\big(y_k - \hat{y}_k\big) \tag{A.11c}$$

$$P_k = \frac{1}{\lambda}(I - K_k \boldsymbol{x}_k^\intercal)P_{k-1}. \tag{A.11d}$$

The exponential forgetting introduces a time depended weighting of the data which allows tracking the changes in the dynamics of the system. However the incorporation of forgetting factor may lead to problems when the input excitation is poor since the covariance matrix could grow exponentially (i.e. estimator wind-up, covariance blow-up). This is equal to the situation where the information matrix R_k becomes singular. In order to prevent this singularity, it is a common procedure to add a positive-definite matrix to R_k. This procedure is known as regularization of R_k or in other words it is called covariance normalization (resetting). This approach is also useful when there is an abrupt change in the actual parameters of the system after a long time.

There are a lot of different approaches in the literature to increase the robustness and the performance of the RLS identification. One example could be given as the "Exponential Forgetting and Resetting Algorithm" (EFRA) which is proposed in [82]. EFRA bounds the trace of the covariance matrix from above and below in order to maintain the stability of the algorithm (e.g., preventing covariance blow-up) while preserving the tracking ability (i.e., preventing zero gain). The EFRA algorithm step are given as:

$$\hat{y}_k = \boldsymbol{x}_k^\intercal \hat{\boldsymbol{\varphi}}_{k-1} \tag{A.12a}$$

$$K_k = \frac{P_{k-1}\boldsymbol{x}_k}{\lambda + \boldsymbol{x}_k^\intercal P_{k-1}\boldsymbol{x}_k} \tag{A.12b}$$

$$\hat{\boldsymbol{\varphi}}_k = \hat{\boldsymbol{\varphi}}_{k-1} + \alpha K_k\big(y_k - \hat{y}_k\big), \tag{A.12c}$$

$$P_k = \frac{1}{\lambda}\big(I - K_k \boldsymbol{x}_k^\intercal\big)P_{k-1} + \beta I - \gamma P_{k-1}^2, \tag{A.12d}$$

where the recursive algorithm starts from an initial guess of P_0 and $\hat{\boldsymbol{\varphi}}_0$. At each time step, when a new measurement y_k, and therefore a new regression vector $\boldsymbol{x}_k$, is acquired, the filter gain K_k is calculated in (A.12b). P_k and $\hat{\boldsymbol{\varphi}}_k$ are updated in (A.12c)-(A.12d) where λ is the forgetting factor and the filter gain is denoted by α. The parameters β and γ are used for covariance resetting and normalization.

The details on the formulation and how to select the parameters could be found in [82]. A typical set of parameters could be $\alpha = 0.5$, $\lambda = 0.95$ and $\beta = \gamma = 0.005$ which corresponds to the maximum and minimum bounds on the eigenvalues of the covariance matrix where $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 10$.

Initialization of RLS and Practical Considerations

As it is aforementioned, the recursive formulations need an initial value (guess) of the covariance matrix and the parameter vector. If there is no a-priori knowledge on the values of the parameters, a common approach is setting them to zero (i.e. $\boldsymbol{\varphi}_0 =$

0). Since this corresponds to a huge uncertainty on the initial guess of the parameter vector, it is practical to set the information matrix to zero (i.e. $R_0 = 0$). However it is impossible to calculate P_0 when R_0 is singular. Therefore one can set the information matrix as $R_0 = \sigma I,\ 0 < \sigma \ll 1$ which corresponds to $P_0 = \sigma I,\ \sigma \gg 1$.

Remark 13. In fact, the true initialization of the information matrix require a-priori knowledge on the data history for $k < 0$.

Another important factor is choosing the forgetting factor λ. The value of λ corresponds to the memory of the RLS algorithm and it affects both the stability and the ability to track the time-varying nature of the actual system parameters. Therefore λ should be chosen carefully. A small forgetting factor corresponds to a short memory of the algorithm which enables a better tracking properties but a high sensitivity to noise in the regression vector where the stability is degraded. On the other hand, a large forgetting factor corresponds to a long memory which enables less sensitivity to the noise but the algorithm achieves a poor tracking performance.

APPENDIX A.2

The experimental vehicle used in this work is a prototype Hyundai Grandeur which is provided by the Hyundai Motor Company. This vehicle is equipped with sensors and electronically controlled actuators which are required for (semi-) autonomous driving. A diagram of the experimental vehicle is illustrated in Figure B.1 and also some pictures of the vehicle is given in Figure B.2.

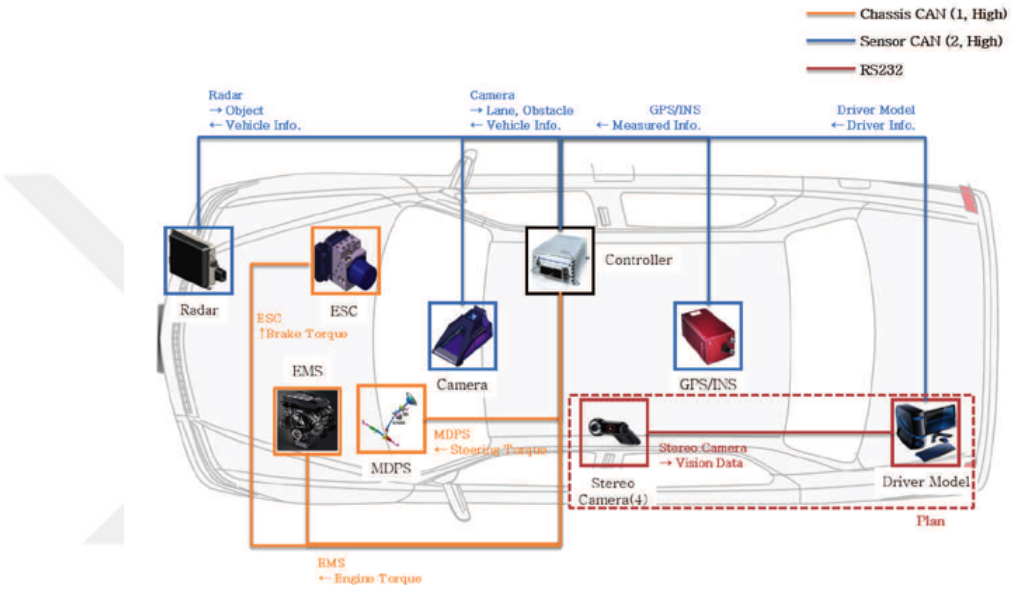


Figure B.1 : The diagram of the experimental vehicle [5].

Figure B.2 : The pictures of the experimental vehicle - a Hyundai Grandeur.

We used a dSPACE Rapid Prototyping Control Unit (MicroAutoBox II) which has 900 MHz IBM PowerPC processor. The control algorithm is implemented in Matlab/Simulink and embedded to the dSPACE hardware. The sensors, actuators and computing platform on the vehicle communicate via a CAN bus.

A production grade camera from Mobileye and a radar provide estimates of the relative positions and velocities of vehicles in front of the ego vehicle. In addition, the camera gives an estimate of the lane geometry in terms of a 3rd-order polynomial

fitted to the lane boundaries in the body fixed coordinate frame. An Oxford Technical Solutions (OxTS) RT 2002 sensing system, comprising of a global positioning system (GPS) and an inertial measurement unit (IMU), combined with a GPS base station are used for localizing the vehicle.

The vehicle is instrumented with a production level Motor Driven Power Steering (MDPS) system which provides a electric power assisted steering to the driver. The MDPS system could be controlled by the dSPACE embedded platform via the overlay torque function of the built-in MDPS control logic. The torque is additive to that applied by the driver to the steering column. Longitudinal speed control is achieved by means of a desired acceleration command to the built-in ACC system of the vehicle. The vehicle parameters are given in Table B.1.

Table B.1 : Ego vehicle model parameters.

Parameter	**Value**	**Parameter**	**Value**
m	1830 kg	I_z	3477 kgm^2
l_f	1.152 m	l_r	1.693 m
$C\alpha_f$	40703 N/rad	$C\alpha_r$	64495 N/rad
J_{eq}	0.172 kgm^2	b_{eq}	1.56 Nm/rad/s
N_s	14.5	k_{eq}	1.75 Nm/rad
ρ_f	0.15 m	ρ_r	0.18 m
c_p	10^6 N/m^2	N_p	0.1

CURRICULUM VITAE

Name Surname: Ziya ERCAN

Place and Date of Birth: KUŞADASI - 25/02/1986

E-Mail: zyercan@gmail.com

EDUCATION:

- **B.Sc. :** 2008, Anadolu University, Faculty of Engineering and Architecture, Department of Electrical and Electronics Engineering.
- **M.Sc. :** 2011, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Department of Control and Automation Engineering.

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2012-2017 Teaching and Research Assistant in Deparment of Mechanical Engineering of İstanbul Technical University.
- 2015-2016 Assistant Specialist at University of California, Berkeley.
- 2009-2012 Project Researcher at Mechatronics Eduacation and Reseach Center of Istanbul Techinical University.
- 2014 Awarded a one-year scholarship grant (2214-A) from The Scientific and Technological Research Council of Turkey (TUBITAK).

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Z. Ercan**, A. Carvalho, H. E. Tseng, M. Gokasan and F. Borrelli, 2017. A Predictive Control Framework for Torque-Based Steering Assistance to Improve Safety in Highway Driving, *Vehicle System Dynamics—Special Issue on Advanced Vehicle Control 2016* pp 1-22, Taylor&Francis Online.
- **Z. Ercan**, A. Carvalho, M. Gokasan and F. Borrelli, 2017. Modeling, Identification and Control of a Driver Steering Assistance System, *IEEE Transactions on Human-Machine Systems*, Volume 47 , Issue 5.
- **Z. Ercan**, M. Gokasan, F.Borrelli, 2017. An Adaptive and Predictive Controller Design for Lateral Control of an Autonomous Vehicle, *IEEE International Conference on Vehicular Electronics and Safety (ICVES)* pp 13 - 18, DOI: 10.1109/ICVES.2017.7991894, June 27–29, 2017, Vienna, Austria.
- **Z. Ercan**, A. Carvalho, S. Lefevre, H.E. Tseng, M. Gokasan, F. Borrelli, 2016. Torque-based steering assistance for collision avoidance during lane changes, *13th International Symposium on Advanced Vehicle Control (AVEC)*, September 13–16, 2016, Munich, Germany.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- M. Bujarbaruah, **Z. Ercan**, V. Ivanovic, H. Eric Tseng, F.Borrelli, 2017. Torque Based Lane Change Assistance with Active Front Steering, *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, October 16–19, 2017, Yokohama, Japan.

- C. Vallon, **Z. Ercan**, A. Carvalho, F.Borrelli, 2017. A machine learning approach for personalized autonomous lane change initiation and control, *IEEE Intelligent Vehicles Symposium (IV)* pp 1590 - 1595, June 11–14, 2017, Redondo Beach, CA, USA.

- V. Sezer, P. Boyraz, **Z. Ercan**, C. Dikilitas, H. Heceoglu, A. Oner, G. Oke and M. Gokasan,2014. Unmanned Ground Vehicle Otonobil: Design, Perception, and Decision Algorithms, *Smart Mobile In-Vehicle Systems*, pp 47-56, Springer New York, 2014.

- V. Sezer, **Z. Ercan**, H. Heceoglu, M. Gokasan and S. Bogosyan,2012. A new fuzzy speed planning method for safe navigation, *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, July 24–27, 2012, Istanbul, Turkey.

- V. Sezer, **Z. Ercan**, H. Heceoglu, M. Gokasan and S. Bogosyan, 2012. A new fuzzy speed control strategy considering lateral vehicle dynamics, *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, September 16–19, 2012, Anchorage, AK, USA.

- V. Sezer, C. Dikilitas, **Z. Ercan**, H. Heceoglu, A. Oner, M. Gokasan and A. Mugan,2011. Conversion of a conventional electric automobile into an unmanned ground vehicle (UGV), *IEEE International Conference on Mechatronics (ICM)*, April 13–15, 2011, Istanbul, Turkey.

- **Z. Ercan**, V. Sezer, H. Heceoglu, C. Dikilitas, M. Gokasan, A. Mugan and S. Bogosyan,2011. Multi-sensor data fusion of DCM based orientation estimation for land vehicles, *IEEE International Conference on Mechatronics (ICM)*, April 13–15, 2011, Istanbul, Turkey.