

The design and analysis of steering interactions between automated vehicle and human driver using hybrid control framework

Mani Kaustubh

The design and analysis of steering interactions between automated vehicle and human driver using hybrid control framework

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft
University of Technology

Mani Kaustubh

August 28, 2014

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of
Technology



The research work in this thesis was performed at TNO Automotive, Helmond. The author hereby gratefully acknowledges their support and cooperation.



Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.



DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

THE DESIGN AND ANALYSIS OF STEERING INTERACTIONS BETWEEN AUTOMATED VEHICLE AND HUMAN DRIVER USING HYBRID CONTROL FRAMEWORK
by

MANI KAUSTUBH

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: August 28, 2014

Supervisor(s): _____
dr.ir. Manuel Mazo jr.

ir. D.M.C.(Dehlia) Willemse

Reader(s): _____
prof.dr.ir. J. Hellendoorn

dr.ir. Riender Happee

Abstract

Advanced Driver Assistance Systems (ADAS) are primarily radar and camera-based technologies that capture the vehicle's surrounding environment and assist the driver by keeping him informed about current vehicle state, and if necessary intervene to prevent an impending danger, while the driver is in control of the vehicle at all times. These technologies pose two types of requirements, one on the *Human-Machine interface* which consists of graphics-based platform or interaction devices like knobs, handles and acoustics etc. for interaction between electromechanical system and the end user, and other, on the *Transition of control* from manual to automated driving and vice-versa. Whilst the former is a topic of ongoing research in the industry, the latter clearly demands more attention (from a control engineering perspective) than it already receives.

The motivation behind this thesis lies in the unavailability of a rigorous mathematical framework, to design and assess the rich dynamic phenomena underlying the steering interactions that take place during a transfer of control authority between human driver and automated vehicle. The current approaches in the academia are based on a moncausal treatment (either from the purview of human factors or from systems engineering) and hence, are too conservative for a sound analysis of combined human-automation interaction. The approach outlined in this thesis addresses these challenges by using a switched system framework to solve the problem of “effecting a smooth switching of control authority between human driver and automated vehicle, and investigating the underlying parameters to address the issues of driver comfort and safety”.

The Human driver has been modeled as preview controller with a neuromuscular dynamics component, whereas the automated vehicle has been developed using PID control strategy for speed control and PD control strategy for steering control. This research uses a 4DOF(degree-of-freedom) ‘two track’ vehicle model for control design and after subsequent linearization, the 2DOF vehicle model for formal verification. Using the concepts of hybrid automata, both the systems were modeled to obtain a two-mode switching automaton. A scheme was then setup using the concept of average dwell time to evaluate stability and temporal logic to incorporate verification of different parameters that affect the switching. For guaranteeing ‘safe’ switching on the combined driver-vehicle system, the dwell time of automated mode was *bounded* to $\tau_s^{q_1} = 1.5 \text{ s}$. Then, respecting the conditions of *average dwell time* switching, the ‘dwell time’ of Manual driving mode would satisfy $\tau_s^{q_2} \geq 2 \times (\hat{\tau}_D^*) - 1.5 \text{ s}$, where, $(\hat{\tau}_D^*)$ is the average dwell time, which was determined to 5.13 s for the switched system.

Furthermore, the BREACH Matlab toolbox was used to perform the parametric verification of the three parameters under investigation, namely, human preview distance, automation preview distance, and driver gain, which were varied for different longitudinal velocities , maximum allowed lateral deviation and time of switching (or the time during the lane change when the switch takes place). In conclusion, the experimental results obtained, validate the correctness and usability of the framework for future developments.

Keywords: Advanced Driver Assistance Systems, Human Driver modeling, Hybrid Automata, Average Dwell Time Switching, Parametric Verification.

Table of Contents

Acknowledgements	viii
1 Introduction	1
1-1 Scope of the project	2
1-2 Motivation	4
1-3 Thesis problem statement	5
1-4 Thesis outline	6
2 Preliminaries	7
2-1 Hybrid Dynamical Systems	7
2-1-1 Overview of the Hybrid Phenomena	8
2-1-2 Modeling of Hybrid Systems	9
2-2 Linear Matrix Inequalities (LMIs)	11
2-3 Formal Verification of Hybrid Systems	14
2-3-1 Tools for formal verification	15
2-3-2 Breach MATLAB Toolbox	15
2-4 Temporal Logic (TL) and Metric Interval Temporal Logic (MITL)	17
2-5 Experimental Scenario	19
2-5-1 Vehicle model	19
2-5-2 Trajectory Generation for verification	24
3 Designing the Cruise Control and Steering Control	25
3-1 Modeling the Automated Controller	25
3-1-1 Deriving the non-autonomous state-space model	26
3-1-2 Transformation to autonomous state-space model	29
3-2 Modeling the Human Controller	30
3-2-1 Transformation to Linear autonomous system	33
3-3 Formulation of equidimensional models	34
3-4 Designing the Speed Control	36
3-5 Designing the Steering Control	39

4 Hybrid Control Design and Verification	45
4-1 Modeling Semantics : Hybrid Automata	45
4-1-1 Brief note on Existence and Uniqueness	47
4-2 Quantifying the interactions: Defining parameters	48
4-3 Stability in the sense of Lyapunov	50
4-4 Switching based on Average Dwell Time	52
4-5 Reducing Conservativeness: Parametric Verification	57
5 Results	59
5-1 Experimentation with Average Dwell time	59
5-2 Verification of safety constraints	64
5-2-1 Iterative procedure	65
5-2-2 Fixed position of switching	66
5-2-3 Variable position of switching	69
6 Conclusions and Recommendations	74
6-1 Discussions on the project	74
6-2 Future Recommendations	76
A Miscellaneous	78
A-1 Mathematical Derivations	78
A-2 List of Acronyms	81
B Parameter Values	82

List of Figures

1-1	Referenced from IRTAD Database: Normalization of road data (on1965) depicts decrease in total number of car accidents and fatalities, with the advent of active vehicle safety systems, even as total distance travelled increases . Source: [3] .	2
1-2	Diagrammatic representation of levels of automation defined for <i>imobility forum</i> , source:University of Twente	3
2-1	Diagram showing discrete time and continuous time interaction	8
2-2	Bouncing ball example, (a.) The discrete and continuous dynamics, (b.) vertical trajectory of the 'falling' ball. Source: [3]	9
2-3	Water tank system and the corresponding hybrid system. Source: [29]	11
2-4	Temporal Operators (a.) next, (b.) sometimes (c.) always, source: [48]	17
2-5	Two Track (4 DOF) vehicle model representing, longitudinal velocity (u), lateral velocity, (v), roll angle (ϕ), and yaw rate ($\dot{\psi}$), source: [52]	19
2-6	(a.) The 2DOF Bicycle model, (b.) Lateral Force F_z of the the TNO Delft tyre model plotted against side-slip angles α	20
2-7	(a.) The side-slip angles, (b.) The yaw-rates for both non-linear and linear vehicle models coincide in the linear tire region	21
2-8	Effect of varying longitudinal vehicle velocities V_x on yaw angle. In all cases the $0 < \psi < 2$ degrees, thereby validating small angle assumption	22
2-9	Stateflow diagram describing the actions and states that have been used in thesis for Time based Switching	23
2-10	Variation of Single Lane Change maneuvers with different ω_n values.	24
3-1	Block Diagram of Automated Vehicle	26
3-2	Block Diagram of Human Controlled Vehicle	30
3-3	ACC equipped vehicle, Source: [63]	36
3-4	Block Diagram of Cruise Controller design using PID control logic	37
3-5	Realized velocity profile of the Cruise Control equipped vehicle	38

3-6	Driver Preview Tracking model, source: [64]	39
3-7	The Force Feedback Driver Model, depicting different functional blocks, source: [66]	40
3-8	The Human Driver Model used in this thesis	42
3-9	Trajectory following behaviour for driver gain $k_p = 1$	42
3-10	Validating the Human Driver Model for different driver gains $k_p = 1, k_p = 3$: (a.) Output Driver torques, (b.) Observed Lateral Accelerations	43
4-1	Graphical representation of hybrid automata describing two states (q_1) and (q_2), with their invariants, reset maps, guards and the initial state	45
4-2	Selecting the nominal values of parametric intervals for human preview distance H_{th}	49
4-3	Selecting the nominal values of parametric intervals for automated preview distance A_{th}	49
4-4	Graphical representation of Hybrid Automata describing two states (q_1) and (q_2), with their invariants, reset maps, invariant and the initial state	56
5-1	Driver and controller Torques for $\hat{\tau}_D^* = 1s$	60
5-2	Driver and controller Torques for $\hat{\tau}_D^* = 10s$	61
5-3	Lateral Accelerations a_y for $\hat{\tau}_D^* = 1s$ and for $\hat{\tau}_D^* = 10s$	62
5-4	Yaw rates $\dot{\psi}$ for $\hat{\tau}_D^* = 1s$ and for $\hat{\tau}_D^* = 10s$	63
5-5	Steps of the iterative procedure	65
5-6	Verification of driver gain intervals k_p as observed with variation of the longitudinal velocity V_x . The dotted black and blue lines shows the constraints on maximum allowed lateral deviation $0.3 \leq y - y_{ref} \leq 0.3$	67
5-7	vehicle lateral response for $H_{th} = 17 m$	68
5-8	Yaw rate $\dot{\psi}$ and steering wheel rate $\dot{\delta}$ response for $H_{th} = 17 m$	68
5-9	vehicle lateral response for $k_p = 1.1$ and $A_{th} = 56$. The lateral deviation remains bounded: $-6cm \leq y_{lat} \leq 6cm$	70
5-10	Yaw rate $\dot{\psi}$ and steering wheel rate $\dot{\delta}$ response for $k_p = 1.1$ and $A_{th} = 56$. The magnified portion shows the responses at the instant of switching	70
5-11	Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 km/h$, $k_p = 2$, $H_{th} = 13 m$	71
5-12	Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 km/h$, $k_p = 2$, $H_{th} = 18 m$	71
5-13	Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 km/h$, $k_p = 1.2$, $H_{th} = 18 m$	72
5-14	Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 km/h$, $k_p = 1.2$, $H_{th} = 13 m$	72
5-15	Verification of yaw-rate based on variation of Human preview distances with different Time of Switching. The 'crossed' red line shows the constraints on bound on yaw rate values $-0.07 \leq \dot{\psi} \leq 0.07$. The colored plots illustrate: (a.) The Blue line :ToS= 20s, $H_{th} = 16 m$, (a.) The Orange line :ToS= 15s, $H_{th} = 15 m$, (c.) The Cyan line :ToS= 10s, $H_{th} = 14 m$, (d.) The Pink line :ToS= 5s, $H_{th} = 13 m$	73
5-16	Heavy oscillations in steering wheel rate $\dot{\delta}$ and yaw rate $\dot{\psi}$ for $A_{th} = 64$	73

List of Tables

3-1	Manual Driving mode, Closed-loop test I: Determining safe value ranges for $\dot{\psi}$ and $\dot{\delta}$	44
5-1	Parameter settings for Cases I and II	60
5-2	Determining optimal parameter intervals through Iterative Procedure	65
5-3	Effect of varying maximum Lateral deviation on parametric intervals	66
5-4	Effect of varying longitudinal velocity on parametric intervals	66
5-5	Effect of varying time of switching on parametric intervals	69
B-1	Parameter settings used in this thesis	82

Acknowledgements

This exploratory study, just like any worthy scientific work could never have been possible in isolation. I would like to thank so many people both at the Delft Centre for Systems and Control (DCSC) and TNO, helmond, as I worked in close collaboration with individuals from both institutions. It would be apt to say that if I learnt a lot about *hybrid control theory* from Manuel, then Dehlia was the go-to person for *vehicle dynamics* and *human factors*. It was due to their encouragement and support, that I could eventually come up with a solution to the research problem at hand.

Furthermore, I would like to whole-heartedly thank dr.ir. Riender Happee for initiating the project talks with TNO and thus, making this MSc. thesis a reality. Also, my sincere thanks to prof.dr.ir.J. Hellendoorn for accepting to be on my MSc. defence committee.

Last but not the least, I extend my gratitude to all my colleagues at IVS department at TNO, for their inspiration and the TNO management for providing requisite financial support and a dynamic work environment.

Delft, University of Technology
August 28, 2014

Mani Kaustubh

“Our *imagination* is stretched to the utmost, not, as in *fiction*, to imagine things which are not really there, but just to comprehend those things which are there.”

— *Richard Feynman, The Character of Physical Law (1965)*

Chapter 1

Introduction

Despite the qualitative improvements in our lives that an ever-evolving technology has bought, accident statistics throughout the world present a rather desolate picture. One of the largest road assessment programmes iRAP[1] states that burden of road crashes costs 1 – 3% of the world’s GDP. Also, their recent report states that annual number of road deaths worldwide without intervention, is projected to increase to an approximate 2.4 million by 2030. In addition, a survey by NHTSA [2] pointed out that driver error is by far (95%) the most common factor implicated in vehicle accidents (followed by road/weather condition 2.5%, mechanical failure 2.5%). Thus, to address the issues traffic accidents pose to the society, there has been a paradigm shift in attitude of industrial, academic and military research institutes towards vehicle safety. Initial efforts addressed driver and passenger safety by utilizing the ‘passive’ safety systems such as shock absorbing chassis, safety belts, airbags, etc. but this has now paved way for ‘active’ safety systems, that combine computational intelligence and real-time environmental data to assist the human driver. The increasing presence of on-board proprioceptive sensors and gradual reliance on exteroceptive sensors, like video cameras, radars, etc. rightly describes the rationale behind transformation in the auto industry.

ERTICO¹, one of the biggest European conglomerates on Intelligent Transport Systems and services, defines these active-safety systems or Advanced Driver Assistance Systems (ADAS), as a collection of in-vehicle technologies designed to improve vehicle safety by aiding the driver. For instance, Cruise Control (CC), regulates the speed of the vehicle; Adaptive Cruise Control (ACC) combines the regulation of speed with a distance control between the host and preceding vehicle, and the Lane Keeping Assistance Systems (LKAS) supports lateral guidance of the vehicle. Each of these systems implies a ‘transition of control’ (albeit of a different nature) from the driver to the automated-assist/ drive system and vice-versa.

The continual developments (figure 2-1) in the field of ADAS show a clear trend towards increased automation. These technologies are receiving increased acceptance, have been found to relieve driver related stress and result in successful accident mitigation². However, an

¹<http://www.ertico.com/objectives-approach/>

²<http://safety.trw.com/from-adas-to-automated-driving/1104/>

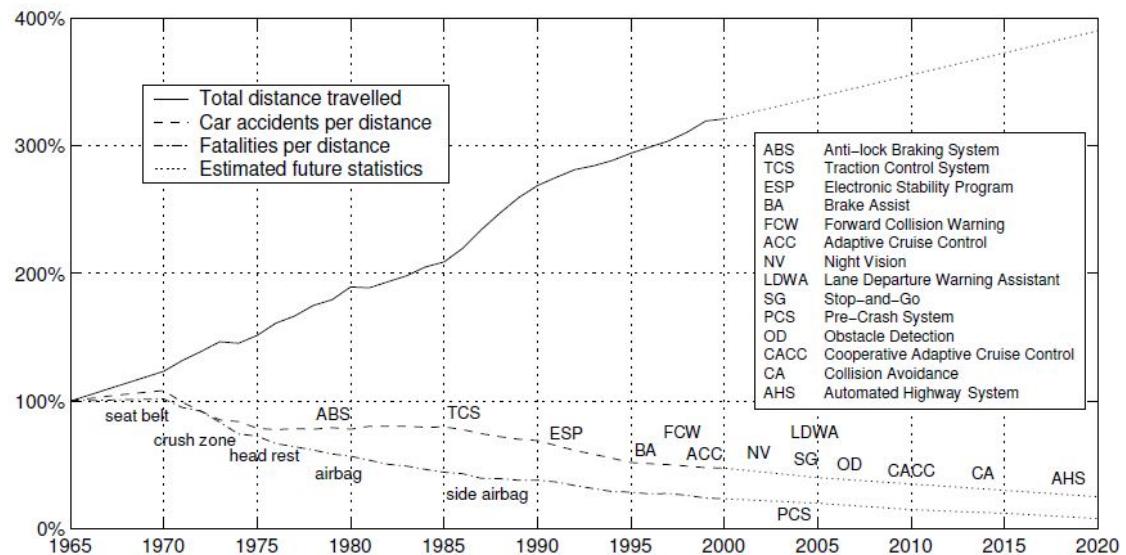


Figure 1-1: Referenced from IRTAD Database: Normalization of road data (on 1965) depicts decrease in total number of car accidents and fatalities, with the advent of active vehicle safety systems, even as total distance travelled increases . Source: [3]

increasing dependence on automation poses two types of functional requirements on the human driver, resulting from an increased gap in functional safety. On one hand, it is the *Human-Machine interface* and other, on the *Transition of control* between manual and automated driving. This is in general non-trivial, because predicting the effects of introducing driving-assist systems into real-time operations requires the knowledge of those factors which cause the human operator to select a ‘particular’ strategy while interacting with the on-board automation.

1-1 Scope of the project

The Working Group Automation in Road Transport created under the *imobility* forum, in its recent report³ (presented to public on June 2013) defines the scope and context under which vehicles can be termed as ‘automated’. The following presents an overview of taxonomy for levels of automation,

³<http://www.imobilitysupport.eu/library/imobility-forum/working-groups/active/automation/reports-3/2185-auto-wg-automation-roadmap-final-report-june-2013/file>

- i. **Driver Only:** Human driver executes manual driving task.
- ii. **Driver Assistance:** The driver permanently controls either longitudinal or lateral control. The other task can be automated to a certain extent by the assistance system.
- iii. **Semi automation:** The system takes over longitudinal and lateral control, the driver shall permanently monitor the system and shall be prepared to take over control at any time.
- iv. **High Automation:** The system takes over longitudinal and lateral control; the driver must no longer permanently monitor the system. In case of a take-over request, the driver must take-over control with a certain time buffer.
- v. **Full Automation:** The system takes over longitudinal and lateral control completely and permanently. In case of a take-over request that is not carried out, the system will return to the minimal risk condition by itself.

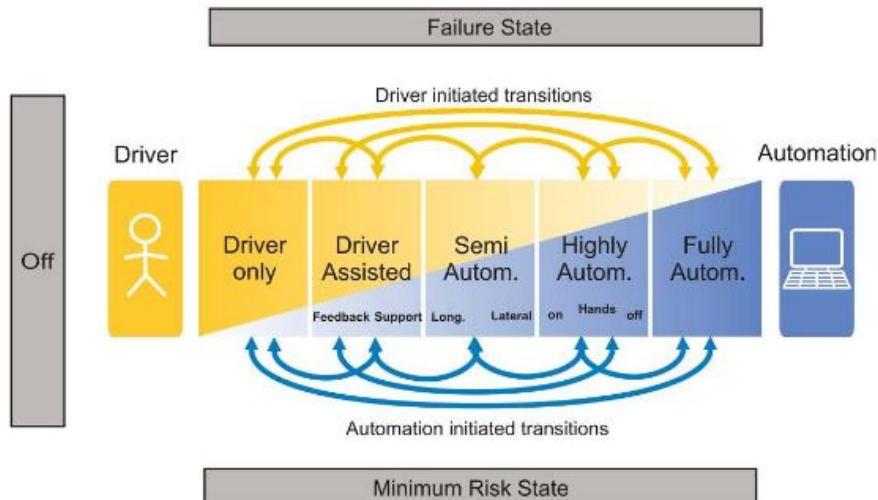


Figure 1-2: Diagrammatic representation of levels of automation defined for *imobility forum*,
source:University of Twente

The present research concerns entirely with definition of '*Level 3- Semi-automation*'. Effective 'sharing' of control requires that the partially automated vehicle system has the intelligence regarding the safe and ideal task performance by human operator. Ideally the automation perceives and plans its tasks and those required of the human with whom it will be trading control. The automation proceeds autonomously until 'critical' scenarios are imminent and human intervention is required. In this regard, researchers [5] propose that, automation should also have 'improvisational' situation assessment capabilities, i.e it could query for human assistance even if it was not originally necessary. However, such re-planning behavior doesn't lie within the scope of this research and the architecture presented in forthcoming sections only incorporates the perception and planning stage of previously mentioned automation behavior.

1-2 Motivation

The Digital Revolution that began in late 60s bought along among many other things, a sustained drive towards automation of tasks at workplace. Myriad benefits of automating tasks found their way into the industrial domains of aviation, automobiles, manufacturing, and medicine. Since its conception, the emphasis of automation design has always been on whether the automation is technologically superior (works more efficiently, is more reliable and guarantees greater accuracy) than human, at lesser costs of operation. However such an economy-centric design has proven catastrophic in a large number of cases (Airbus A320 crash over Strasbourg, 1993, Baltimore Railroad accident 1987), which leads us to exploring a much responsible, reliant yet cost-effective solution, called *adaptive automation*. Thus, before any important design decision related to automating the lateral control task of human driver could be taken, a literature survey was undertaken to gather knowledge and critically appraise the related issues.

In the entire study, two main ideas were omnipresent: *First*, depending on the type of indicator used (for instance, global statistics, accident detailed analysis, fatality etc.) and the country where experiments were performed, *road departure* accidents make up a significant proportion of all road accidents (from 35% to 70 % [4]), thereby stressing on the need for investigating issues related to safe *Lateral control* of the vehicle both by humans and automation. *Second*, lack of quantitative research based on human-automation transfer and reclaiming of control in lateral driving maneuvers necessitates the need for a more human-centered control design. [6] concluded that the problem of automation is not over-automation, but rather inappropriate feedback and inadequate interaction. Furthermore, [7] attributes the issue of driver distraction to an ill-coordinated activity, thus stressing the need for better human-machine coordination.

Lessons from aerospace domain

Studies concerning review of future technologies for the automobile sector, have long been dabbling with the prospect of inheriting intelligent human-flight design concepts from the aviation sector into the automotive domain. Case in point being the publications [8], [9] that describe safety implications for automating driving tasks based on the assumption that automation in aviation represents the basic model for driver automation. [9] explores the key benefits of automation in aviation and discusses their portability in the automotive sector based on which drivers' tasks are likely to yield benefits. They conclude that automation and success are not synonymous and the costs and benefits of automating human-centric tasks can't be identified effectively by underplaying the impact of human factors. [10] presents the case for adaptive automation by proposing that many of the benefits of automation can be maximized and the costs minimized if automation is implemented in an adaptive manner rather than in an all-or-none fashion. [11] present the interactions between human and flight by discussing Pilot Induced Oscillations (PIOs) (occurs when aircraft suffers from uncontrolled oscillations as pilot inadvertently tries to apply over-corrective actions). Another research [12] cites the industrial example of Airbus A320, 340 and 777 aircrafts for presenting a case on how the design philosophy driven by different levels of automation affects the safety and performance. They examine the differences between 'hard' and 'soft' automation employed in the afore-mentioned aircrafts. They conclude by saying that instead 'one-off' approach of either soft or hard automation for driving, something sort of 'blending' of control throughout

the driving subtasks may prove most efficient. However, they also point out that with soft automation causing problems of mental workload as described by [13], applying the concept verbatim is tougher than expected.

1-3 Thesis problem statement

This research work forms a part of a larger, more comprehensive project at TNO, Helmond, titled '*Transition of Control*'. This project investigates guidelines for a 'good transition of control' between the automated driving system and manual driving. One special case is the voluntary switching between automated driving and manual driving, which forms the basis of this MSc. thesis. The experimental scenario would be as follows: The automated vehicle will be equipped with a CC, to maintain a constant velocity, and a steering control, to maintain a reference trajectory. The coupled driver-automated vehicle system will have to navigate a single lane change (SLC) of width 3m. The vehicle would be initially driving in *automated mode* and just before the lane change the driver will initiate a take-over of control and would perform the lane change maneuver in *manual mode*. Finally, after stabilizing the vehicle the driver will switch back to automated mode, transferring the control to the automated vehicle. Now, regarding the experimental scenario it is important to mention two important points. Firstly, the use of CC to maintain a constant longitudinal velocity in the envisaged maneuver proves sufficient enough for developing a primary evaluation scheme. Secondly, although automated lane changes look promising barring a few studies [16],[17] this idea is still mostly under investigation and conclusive real-time implementations have yet to be developed, thereby stressing the need for a manual lane change.

To design, implement and analyze the above scenario, a natural question comes to mind: *How does one 'effect' a smooth transfer of control authority from manual driving mode to automated mode and which parameters affect the safety of systems involved?*

In their pivotal work on Human factors, [14] present the case for *use, misuse and abuse* of automation. The authors, through various experimental observations suggest that better operator knowledge about automation, active operator involvement, ease of transfer of control authority, go a long way in avoiding hazardous or destructive incidents. Also, in another related work[15], the authors suggest that a possible solution for flexible and responsive function allocation, is to allocate a task briefly to automation before returning it to human-operator. Thus, taking a cue from these suggestions, this MSc. thesis describes an approach based on 'steering interactions' to provide an answer to the above problem, and thus tries to solve the two research challenges posed: *First*, to investigate guidelines for 'good transition of control' between the automated driving system and manual driving system. *Second*, to design a temporary driver take-over mode for lateral control, e.g. when the driver wants to change lane or avoid a small obstacle on the road.

The sub-questions that need to be answered in order to have a comprehensive evaluation of the problem statement are discussed below:

- i. How can one model the manually driven vehicle and the automated vehicle?
- ii. How should automation take over?
- iii. How can driver take back control and how to involve him ?
- iv. How can a smooth switching/transition be effected?
- v. What are critical scenarios ? Define the situations for implementation?

1-4 Thesis outline

This MSc. thesis is divided into six chapters. This chapter (Chapter 1) provided a basic introduction, presented the motivation behind this thesis and also, presented the problem statement as well as the scope of this MSc. thesis. Chapter 2, prepares the background for forthcoming sections by defining the preliminary concepts in hybrid control systems, LMI theory and hybrid systems verification. Chapter 3, describes the modeling of both the human and automated systems, and ends with the speed and steering control design and validation. Chapter 4, revolves around another integral aspect of this research work, verification of hybrid systems. After an initial discussion of hybrid automata and Lyapunov theory, time-based switching and parametric verification is discussed. Results for both time-based switching and parametric verification are discussed in Chapter 5, whereas, Chapter 6 puts forth recommendations for possible future work scenarios.

Familiarity with basic concepts in Systems and Control theory, Temporal Logic, and Hybrid systems and control is expected from the reader.

Chapter 2

Preliminaries

2-1 Hybrid Dynamical Systems

Hybrid systems are composed of coexisting continuous and discrete dynamical components, that interact and evolve based on either response to continuous dynamics or occurrence of discrete events or both. A ‘discrete’ variable can be defined as function of a set of integers and can only take on a finite set of values, on the other hand, ‘continuous’ variables are functions of set of real numbers and within the allowed variable limits, can take any value. Throughout this thesis, discrete states have been represented by symbol $(q_i) \forall i \in [1, \dots, n]$ and continuous states have been represented by $(x_i) \forall i \in [1, \dots, n]$ or by other variables in Euclidean space viz. $v, y, u \in \mathbb{R}^n$. The complex and rich dynamical interaction of continuous-time and discrete-time components in a hybrid system, makes the analysis and design of these systems using the traditional mathematical tools in control systems design very challenging.

Furthermore, hybrid systems in essence is a multi-disciplinary (control engineering, computer science, signals and systems) concept and hence various scientific communities have their own ‘view’ regarding system analysis and design. Typically, the *computer science* community focuses on the theory of logic and the discrete event dynamics, *formal verification* lies at the core of such pursuits. Alternatively, *control engineering* community emphasizes more on continuous dynamics, thereby validating functionality and performance of the model through analysis and simulation (examples include Switching control, Supervisory control etc.). These approaches largely ignore issues related to software verification of control laws.

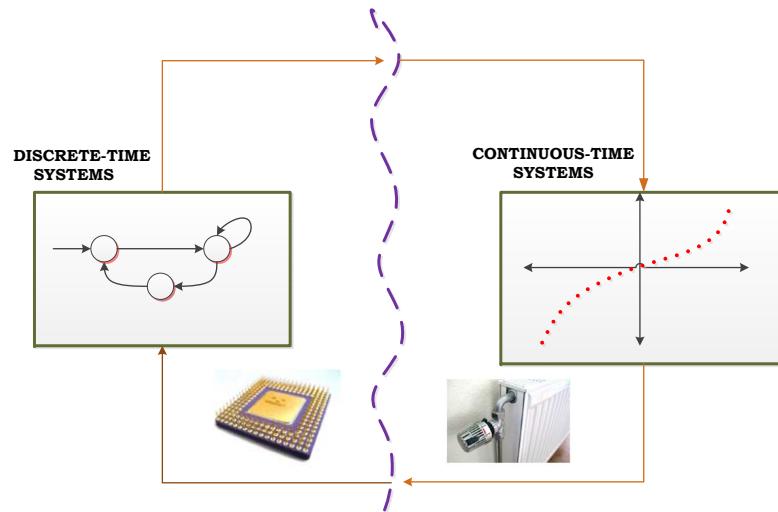


Figure 2-1: Diagram showing discrete time and continuous time interaction

2-1-1 Overview of the Hybrid Phenomena

The heterogeneous interaction between continuous and discrete behavior of the dynamics can be modeled as a hybrid system with the continuous ‘flow’ (or evolution) described by differential equations and the discrete ‘jumps’ or mode changes being described by difference equations. The discrete ‘modes’ have constraints describing the regions of space where continuous dynamics are allowed and their transitions describe the conditions where discrete evolutions occur.

Bouncing Ball

A benchmark example of simple Hybrid system is the *bouncing ball*. The Figure 2-2 describes the model with the discrete state q_0 , and continuous states x_1, x_2 describing the vertical position of the ball and velocity respectively. The inequality $x_1 \geq 0$ is the *invariant* condition or the constraint that prohibits the discrete state from changing its value. Only the logical constraints $x_1 = 0 \wedge x_2 \leq 0$, denoted as *Guard*, when proven ‘True’ allow for a ‘jump’ in value of the discrete state. This essentially means that during this transition the continuous state is constant and after the transition is effected the continuous states start to evolve(governed by Newton’s laws of motion) based on the differential equations as illustrated in the figure. The assignment equation $x_2 := -cx_2$ where, $c \in [0, 1]$ describes the *Reset map* (explained in further sections), which basically translates to loss of ball’s speed and change of direction after each contact of the ball with ground.

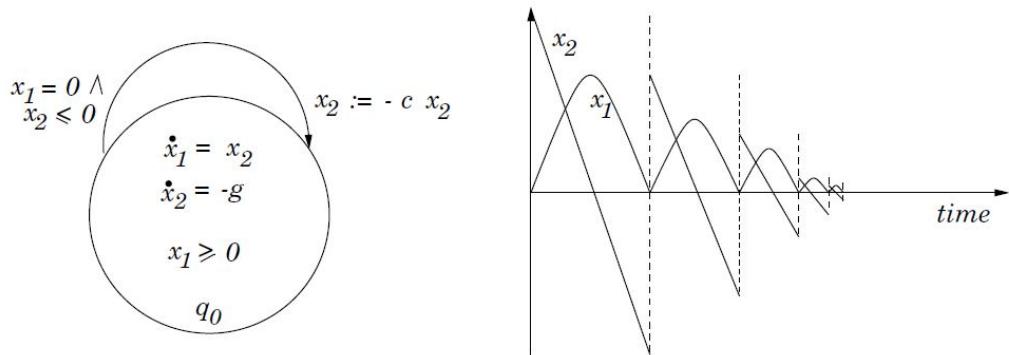


Figure 2-2: Bouncing ball example, (a.) The discrete and continuous dynamics, (b.) vertical trajectory of the 'falling' ball. Source: [3]

2-1-2 Modeling of Hybrid Systems

Mathematical modeling of Hybrid systems has been a topic of great interest to researchers in various communities. The challenging notions of *description* (ability to describe the continuous-discrete interaction in a multitude of ways), *abstraction* (ability to redefine system design depending upon the needs of the problem) and *composition* (ability to aggregate smaller building blocks to obtain a large-scale models) as presented in [18], negates the presence of unilateral approach to modeling and poses interesting questions on the compromise between model generality and expressibility. To elucidate, in a survey on modeling, analysis, and control of hybrid systems [20], the authors suggest that choice of a suitable modeling framework is a trade-off between two conflicting criteria: the *modeling power* and the *decisive power*. The modeling power indicates the size of the class of systems allowing a reformulation in terms of the chosen model description. The decisive power is the ability to prove quantitative and qualitative properties of individual systems in the framework.

This thesis considers the semantics of modeling language referred to as *Hybrid Automata*. The *motivation* behind the use of hybrid automata is its good *descriptive* power or *modeling* power. Also, due to a broad model structure, analysis and verification can be done in a relatively easy manner. Although it should be emphasized that as proven in [21], many classes of hybrid automata suffer from *undecidable reachability problem*, i.e. they are an NP hard analysis problem [22]. Some other notable works in modeling of hybrid systems are Tavernini's Differential Automata [23], Controlled Hybrid Dynamical System (CHDS) by Branicky [24], Stiver and Antsaklis model of continuous plant and discrete regulator [25], modeling with Petri Nets as described by Peleties and DeCarlo [26], and Digital Sequential Control Automata technique by Nerode and Kohn [27].

The Hybrid Automata

Hybrid automata [21], is a formal model that forms an extension of discrete control graphs, referred to as *finite state automata*, by incorporating continuous variables. It describes the evolution in time of set of instantaneous discrete mode changes or jumps and gradual continuous transitions. As the case with any modeling/programming language, the definition of *Syntax* (concerns with Symbolic representation/structure of language) and definition of *Semantics* (concerns with relation between signifier, such as words, phrases, signs and symbols or the study of meaning) is prerequisite before any investigative analysis can be performed.

Syntax

The discrete states are denoted by $q_i \in Q$, and are modeled by vertices of graph (also called as locations/modes). The discrete changes of the automaton are modeled by the *transitions* (also called as guard/control switches). Furthermore, the continuous states are modeled by $x_i \in X$, wherein the continuous dynamics of the particular mode/location are modeled by *invariant* conditions defined by differential equations. Finally, the change in value of continuous state after each transition is defined by *Reset map*. The definition of hybrid automata as described by [18] is explained in the next section.

Definition 1.1 (Hybrid Automaton)

A Hybrid automaton \mathbb{H} , is a tuple $\mathbb{H} = (Q, X, I, f, Inv, E, G, R)$

- $Q = \{q_1, q_2, \dots\} \subseteq \mathbb{Q}$ is a set of discrete states;
- $X = \{x_1, x_2, \dots\} \subseteq \mathbb{R}^n$ is a set of continuous variables;
- $I = (q_0, x_0) \subseteq Q \times X$, is a set of initial states;
- $f(\cdot, \cdot) : Q \times X \rightarrow \mathbb{R}^n$, is a vector field;
- $Inv : Q \rightarrow P(X) := 2^X$, is a set of invariant conditions;
- $E = (q_i, q_{i+1}) \subset Q \times Q$, is a collection of discrete transitions;
- $G = E \rightarrow P(X)$, assigns to each transition a collection a guard condition;
- $R = (q_i, q_{i+1}, x) : E \times X \rightarrow P(X)$ assigns to each transition a Reset Relation;

Here, $P(X)$ denotes the set of all subsets of X . The *state* of \mathbb{H} is described by $(q, x) \in Q \times X$. Continuous evolution can go on as long as x remains in domain of discrete state q . Consider for example the water tank system of Figure 2-3. The objective is to maintain the water level in both tanks above pre-fixed values of r_1 and r_2 . The inflow rate is w and the outflow rates are v_1, v_2 . Let the system start in discrete state q_1 . The evolution of continuous states x_1, x_2 in q_1 are continuous until the trajectory hits the guard condition $G(q_1, q_2) = x_2 \leq r_2$ belonging to the upper edge $(q_1, q_2) \in E$, the discrete state then changes the value to q_2 . In addition, had there been a reset map $R(q_1, q_2, x) \subseteq \mathbb{R}^n$ it would have caused the continuous variable to reset its value. Finally, after the discrete transition is made, continuous evolution in the new state q_2 resumes and the process is repeated again.

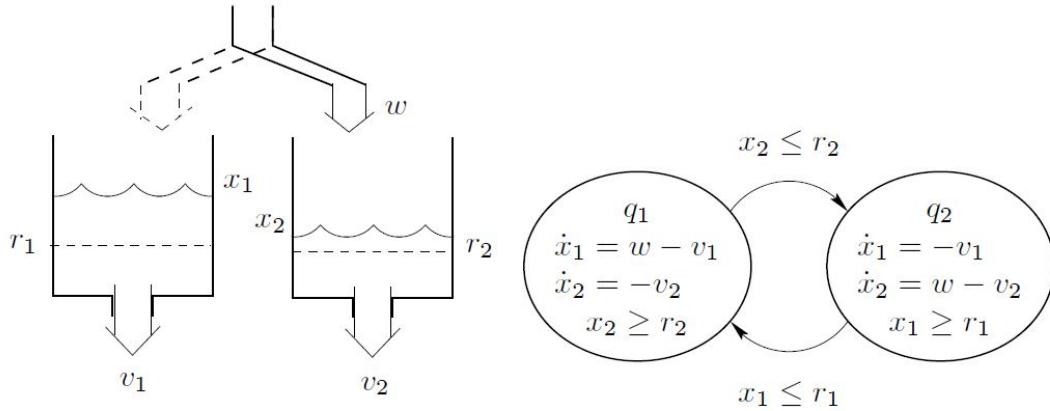


Figure 2-3: Water tank system and the corresponding hybrid system. Source: [29]

2-2 Linear Matrix Inequalities (LMIs)

Linear matrix inequality (LMI) denotes a constraint of the form $F_0 + \sum_{i=1}^n F_i x_i > 0$, where F_i are fixed symmetric matrices and $x \in \mathbb{R}^n$ is the decision variable. The inequality symbol ‘ $>$ ’ represents a *positive-definite* matrix, i.e. $y^T F(x)y > 0 \forall$ non-zero $y \in \mathbb{R}^n$, which means a matrix with all eigenvalues positive. Although the equation described above is based on *strict* inequality, LMIs can also be formulated with *non-strict* inequalities, i.e $F(x) \geq 0$. Linear inequalities, quadratic inequalities, convex quadratic matrix inequalities, as well as, Lyapunov constraints can all be cast as an LMI problem. For a more detailed and exhaustive discussion, the reader is referred to one of pivotal works in the LMIs [33]. The authors describe that the basic approaches in LMI formulation is to solve the following questions:

- Is the LMI problem *feasible*?
- Can the cost function be minimized to satisfy a certain constraint, *convex optimization problem*?

The authors also describe the concept of *tractability*, which basically involves ‘solving’ the problem in polynomial-time and in a practical and efficient manner. As to the question of what solving essentially is, the following statement outlines their message: By “solve the problem” we mean: *Determine whether or not the problem is feasible, and if it is, compute a feasible point with an objective value that exceeds the global minimum by less than some prespecified accuracy.*

Furthermore, another interesting work done by Scherer, and Weiland [34] describes the concepts of stability and performance characterizations with LMIs, optimal performance synthesis and polytopic uncertainties using motivating examples.

Now, the LMI concepts that have been used in this MSc. thesis for *average dwell-time switching* (explained in Section 4-4) involve *Lyapunov inequality*. An LMI based treatment for this approach is defined in the following example:

Lyapunov Inequality

One of the most elementary problems in control and systems theory is stability analysis using Lyapunov theory. The linear system described by $\dot{x} = Ax$ is *asymptotically stable* if and only if the real part of all eigenvalues of A are negative, or equivalently, there exist a symmetric matrix $P = P^T$ to the following Lyapunov inequality.

$$A^T P + PA < 0 \quad (2-1)$$

$$P > 0 \quad (2-2)$$

If there exists such a P , we say the Linear differential inclusion $\dot{x} = Ax$ is asymptotically stable and we call $V(x) = x^T Px$ a quadratic Lyapunov function. Thus, checking quadratic stability for the LTI system essentially falls under the category of LMI *feasibility* problem, in the variable P .

LMIs in System and Control Theory:

Research in robust control theory has seen a remarkable uprise in the previous decade, due to the emergence of numerical interior point algorithms [30], that have lead to fast, efficient and reliable LMI solvers seeing the light of the day. Although research in linear matrix inequalities is nothing new, with the first steps being taken in the 1940s with the application of Lyapunov's stability theory to practical problems in control engineering. Starting from the work of Lur'e and Postnikov [31] to the celebrated work of Kalman, Yakubovich, Popov [32], and further to largely known Popov criterion and Circle criterion, the LMIs were initially formulated manually, then solved via graphical methods, then solved by computer via convex programming, and most recently solved using interior-point algorithms.

The YALMIP MATLAB toolbox:

YALMIP(stands for Yet Another LMI Parser) is a modelling language for advanced modeling and solution of convex and nonconvex optimization problems. It was developed by Johan Löfberg, at the Automatic Control Laboratory, at ETHZ. The flexibility and ease in usage can be understood by the fact that the author [36] suggests that just about 3 YALMIP specific commands will be enough for most users to model and solve their optimization problem. The most recent release, YALMIP 3, interfaces around 20 solvers and supports linear programming (LP), quadratic programming (QP), second order cone programming (SOCP), semidefinite programming, determinant maximization, mixed integer programming, semidefinite programs with bilinear matrix inequalities (BMI), and multiparametric linear and quadratic programming.

Decision variables are represented in YALMIP by `sdpvar` objects. YALMIP supports strict inequality operators ($>$ and $<$) to describe both semidefinite constraints and standard element-wise constraints, as well as non-strict inequalities (\leq and \geq). Also, the most commonly used constraints in YALMIP are element-wise, semidefinite and equality constraints. The command to define these is called `set`. The command `solvesdp` is used for solving all optimization problems and typically takes two arguments, a set of constraints and the objective function.

The following commands represent the script for a sample problem of finding a common Lyapunov function for two different systems with state matrices A_1 and A_2 :

```
P = sdpvar(n,n);
F = set(P > Q);
F = F + set(A1' * P + P * A1 < 0);
F = F + set(A2' * P + P * A2 < 0);
solvesdp(F);
```

2-3 Formal Verification of Hybrid Systems

Hybrid Control Systems concerns with the mutual interactions taking place between continuous time dynamics and discrete event systems and as such are more complex to analyze, as compared to the classical control systems. The analysis of such systems is centered upon the formal verification of *safety* properties. Formal verification lies at the core of theoretical computer science and is defined as the use of computer simulation to analyze whether a system under test satisfies a set of desired specifications or not. The properties to be specified are defined in terms of Temporal Logic. Consider an example specification in terms of temporal logic: *Variable A is not set to zero until variable B eventually decreases to zero.* Here the terms ‘until’ and ‘eventually’ are referred to as *temporal operators* and are discussed in detail in the next section. An important remark regarding formal verification is the seemingly questionable result of the *correctness* of discrete system under study. This is because problems can occur either in creation of the formal model of a RT(real time) system or in its implementation.

Before addressing the methods used in formal verification, it is pertinent at this point to provide a lexical definition of what the verification problem actually is. Fortunately, a pivotal work at the Lund University by [28] formulates the problem as:

Given a collection of automata defining the system and a set of formulas of temporal logic defining the requirements, derive conditions under which the system meets the requirements.

Another interesting study of computational techniques for verification [37] cites verification of safety properties as essentially a *reachability* question. According to them it basically amounts to obtaining a rigorous mathematical solution to the following question:

Is a potentially unsafe configuration, or state, reachable from an initial configuration?

Broadly speaking, the formal verification methods can be classified as - *model checking*, which basically explores the entire state space of a system and automatically checks the formal model for the correctness with respect to a temporal logic specification. [38] define the two ways this can be done - (1) Enumeratively (considers each individual state) (2) Symbolically (computes constraints that represent state sets). Although it suffers from large simulation times which increases exponentially with increase in the number of variables, model checking is a really useful technique for system design, as it aids in debugging by providing diagnostic information regarding the satisfaction of correctness requirement.

The second method for formal verification is, *theorem proving*, which bases upon a representation of the system to be analyzed and attempts to arrive at a proof of correctness by a set of inference rules. However, many supposedly automated theorem proving methods require substantial time and human attention (variable ordering etc.) for completing the requisite task. These methods also suffer from lack of insight into what went wrong during verification and mostly provide a ‘Yes’ or ‘No’ solution. Popular theorem provers include (Coq - based on type theory, EVES - based on set theory, HOL - based on type theory).

2-3-1 Tools for formal verification

Consider the problem of controlling a four-stroke gasoline engine. It is usually modeled using four discrete modes corresponding to the position of the pistons, while the continuous behavior results from combustion and power train dynamics. To analyze the overall behavior of the controlled system, the interactions between discrete and continuous domains necessitate the combination of different mathematical representations. This aspect makes verification and synthesis impossible, unless a careful analysis of the interaction semantics is carried out. Problems like this and many more, provided the much needed motivation in the industry and academia for the development of computational tools.

The computational tools presently available for formal verification can be grouped as: (1) tools developed for commercial usage, primarily useful for *simulation* and (2) tools developed within localized academic communities, primarily dedicated to *verification*. Popular tools like *Matlab's Simulink, Stateflow, DYMOLA* fall under the first category. Their excellent abstraction power, ease of use, and intuitive modeling capability makes them a boon for designers. However, they lack the notion of rigorous mathematical semantics and as such are not suitable for formal verification purposes. On the other hand, there exist academic tools for verification like *HyTECH, HSOLVER, PHAVer* and *BREACH* (which is used for verification purposes in this thesis). These tools build upon the basic notion of intelligent exhaustive search on the state-space and avoid exploration on uninteresting parts.

HyTECH [39] is a symbolic model checker for linear hybrid automata, a subclass of hybrid automata that can be analyzed automatically by computing with polyhedral state sets. *PHAVer* [40] uses exact arithmetic whose robustness is guaranteed by the use of the Parma Polyhedral Library. *HSolver* [41] uses the general idea of reducing the infinite state space of a hybrid system to a finite one by partitioning the continuous space into boxes. *BREACH* [42] relies on an efficient numerical solver of ordinary differential equations that can also provide information about sensitivity with respect to parameter variation.

The selection of one type of tool or the other falls in the hand of designer based on the requirements of the research. This is aptly summarized by a comprehensive study [44] on computational tools where the authors base their experiments on two case studies, a system of three point masses and a full wave rectifier :

The essence is to balance the gains in analysis and synthesis power versus the loss of expressive power.

2-3-2 Breach MATLAB Toolbox

BREACH MATLAB toolbox [42] was developed by Alexandre Donzé at the Verimag Laboratory in France. It is based on the original idea of *verification using simulation* [45], where the authors equip transition systems with bisimulation metrics (defined as topologies on the set of trajectories of a system), thereby enabling the development of a simulation-based verification algorithm. They compute an over-approximation of the reachable set of a metric transition system in only finite (possibly large) number of simulations. This provides a remarkably new approach as owing to the almost infinite possible behaviors of hybrid systems, their formal verification by exhaustive simulation is impractical. *BREACH* has undergone improvements with time and now prides itself with an intuitive GUI, makes parametric verification possible and allows for writing Metric interval temporal logic (MITL) {explained in the forthcoming subsection} formulas and monitor their quantitative satisfaction.

Hybrid System Model and simulation

BREACH considers hybrid dynamical systems with piecewise-continuous dynamics, described mathematically as:

$$\dot{x} = f(q, x, p), x(0) = x_0 \quad (2-3)$$

$$q^+ = e(q^-, \lambda), q(0) = q_0 \quad (2-4)$$

$$\lambda = \text{sgn}(g(x)) \quad (2-5)$$

where, $x \in \mathbb{R}^n$ = is the continuous state

$p \in \mathcal{P} \subset \mathbb{R}^{n_p}$ = is the parameter vector

$q \in \mathcal{Q}^n$ = is the discrete state

g, e = is the guard function and event function respectively

BREACH defines a trajectory ξ_p as a function that maps from a time set $\mathcal{T} = \mathbb{R}^+$ to \mathbb{R}^n and satisfies the equations 2-6, 2-7 and 2-8 for all $t \in \mathcal{T}$. Represented mathematically:

$$\dot{\xi}_p(t) = f(q(t), \xi_p(t), p), q(t^+) = e(q^-(t), \lambda(t)) \text{ and } \lambda(t) = \text{sgn}(g(\xi_p(t))) \quad (2-6)$$

Another point worth observing is that BREACH doesn't allow either *sliding*(trajectory remains on the transition surface) or *grazing*(trajectory hits the transition surface tangentially). This is guaranteed in BREACH because, the dynamics of the systems leans strictly toward the guard before the switch and strictly away from it after the transition. BREACH builds on the concept of reachability using *sensitivity analysis*. Parametric Verification is concerned with the question of the influence of a parameter change δ_p on a trajectory ξ_p . Taylor expansion of trajectory ξ_p in terms of p , results in :

$$\xi_{p+\delta_p}(t) = \xi_p(t) + \frac{\partial \xi_p}{\partial p}(t)\delta_p + \psi(t, \delta_p), \text{ where, } \psi(t, \delta_p) = \mathcal{O}\|\delta_p\|^2 \quad (2-7)$$

For efficiency reasons the toolbox interfaces MATLAB with CVODES [43], a numerical solver designed to solve ODEs and sensitivity equations of type 2-10, efficiently and accurately. In BREACH a *new system* needs to be created with its dynamics defined in C++, before starting any verification process. The BREACH toolbox consists of classes and functions dedicated to this. For instance, a MATLAB script `CreateSystem` defines a structure with information about the system along-with options for the solver and default values. Also, C++ source files are provided that help in assigning dynamics of each subsystem, including information about 'Guards', 'Invariants', etc. and also conditions on switching. These files namely, `dynamics.cpp` and `dynamics.h`, need to be 'compiled' before starting the simulation. BREACH also provides ease in defining and modifying parameter sets, which are defined using in-built functions `CreateParamSet` and `CreateSampling`. For refining the parametric intervals into smaller grids of separation given by δ , the functions `Refine` allows manual refining, while `HaltonRefine` is a method to sample uniformly using quasi-random numbers.

2-4 Temporal Logic (TL) and Metric Interval Temporal Logic (MITL)

Temporal Logic (TL):

Temporal Logic is a mathematical framework that allows for formal (mathematical) description of specifications that the designed system should adhere to (or satisfy). A seminal work [46] in the 1970s applied the concept of “tense logic” in philosophy, to propound the technique of temporal logic which was used in the specification and verification of computers. Temporal logics have been generally considered useful for specifying and verifying programs consisting of digital sequences of states and events. However, with the recent advances in research, it has been extended to the specification of properties of real-valued signals defined over dense time and applied to domains as diverse as analog circuits or biochemical reactions.

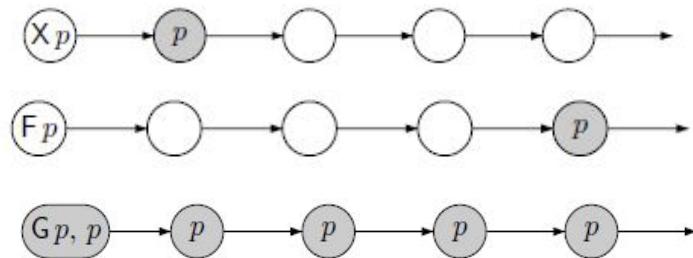


Figure 2-4: Temporal Operators (a.) next, (b.) sometimes (c.) always, source: [48]

Metric Interval Temporal Logic (MITL):

The notion of extending the ubiquitous LTL(Linear Temporal Logic) to incorporate more complex system behaviors, such as those observed in hybrid systems has been in place for about two decades now. These include the addition of a notion of ‘metric’ or ‘distance/depth’ to the existent discrete time concepts. Metric Temporal Logic (MTL) and Metric Interval Temporal Logic (MITL) are extensions designed to handle dense time logics. One of the earlier works [48] defines MITL as a linear temporal logic that is characterized by timed state sequences. Furthermore, [49] describes MITL as logic characterizing timed behaviors. In addition, every language needs a syntax and semantics, the MITL language is not exception. MITL grammar is given by [48]:

$$\varphi := p \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U}_I \varphi_2 \quad (2-8)$$

where, $\varphi, \varphi_1, \varphi_2$ are MITL formulae, $p \in P$ is the proposition and I is nonsingular interval with integer end-points. Also, the symbol \neg is the negation operator, \wedge is the logical AND operator, and \mathcal{U} is the temporal operator *until*. To understand the until operator \mathcal{U} , consider the example formula $\psi \mathcal{U} \phi$, which basically reads as : ψ has to be true at least *Until* ϕ which is currently true or will be true at a future position.

To elucidate the principle of time state sequences, the authors [48] consider a random formula $\varphi_1 \mathcal{U}_I \varphi_2$ that holds at a certain instance t_k , now this supposition can only be proven iff φ_2 holds at some time instance t_k' , where $t_k' \in I$ and φ_1 holds throughout the interval (t_k, t_k') . [47] provides a newer perspective on using MITL for verification and also forms the basis of BREACH Matlab Toolbox (discussed in the next section).

Monitoring or Runtime verification is the commonly used method for verifying whether a system with complex dynamics (like Hybrid Systems) or systems with mixed-signal circuits, satisfies a desired specification or not. This is because verification of Hybrid systems is generally undecidable, except only for a very-small class of systems (Timed automata [21]). Thus, research on the problems resulting from exhaustive-verification has led to utilization of property specifications for verification in discrete-time systems, to the technique of monitoring for simulation in hybrid systems. As with many systems in real-world, the *quantitative* information about property satisfaction by a system is of vital importance, specially in situations where the *degree of robustness* can play a vital role. Simply, stated the YES/NO type answers are of relatively less importance as compared to How much? or how less? Based on their previous work [49], where the measure of robustness is provided in space and time, the authors develop a robust monitoring algorithm [50] for specification of properties in Signal Temporal Logic, STL (which is a variation of MITL) that forms the basis of property based verification in the BREACH Matlab toolbox.

Although Run-time verification is vital tool to extend the techniques of formal verification for discrete time systems to continuous time systems, it suffers from the problem of *incomplete* exploration of the state-space. To provide a more systematic analysis of hybrid systems, the technique of *simulation-based verification* developed by [45] guides the trajectory generation by appropriate sampling of initial sets, to satisfy or violate a desired property for all the system trajectories based only on a finite number of simulations. The BREACH matlab toolbox used in this thesis utilizes a sensitivity-based parameter-space exploration technique to guide the generation of traces in parameter space and combines it with an efficient monitoring algorithm to compute the robustness measure based on space and time. In essence, it couples both the techniques of *run-time verification* and *verification based on simulation* to implement systematic parametric-verification in Matlab.

2-5 Experimental Scenario

This section serves as a guide for Matlab implementation of the ideas discussed in thesis. The forthcoming subsections define the vehicle model that has been used, present the concept of small-angle approximations needed for linearisation in the next chapter, discuss the Matlab STATEFLOW environment and finally explain the generation of reference trajectory that has been used for formal verification.

2-5-1 Vehicle model

In order to develop controllers for a system, it is important to have a good understanding of the states and parameters that affect the system dynamics. Vehicle dynamics modeling provides a methodical approach to analyze the dynamic behavior of vehicle when it reacts to forces imposed by tires, gravity and aerodynamics. Most of the vehicular control strategies in use today utilize different dynamical models to focus upon suitable variables of interest, be it acceleration, braking, ride or turning.

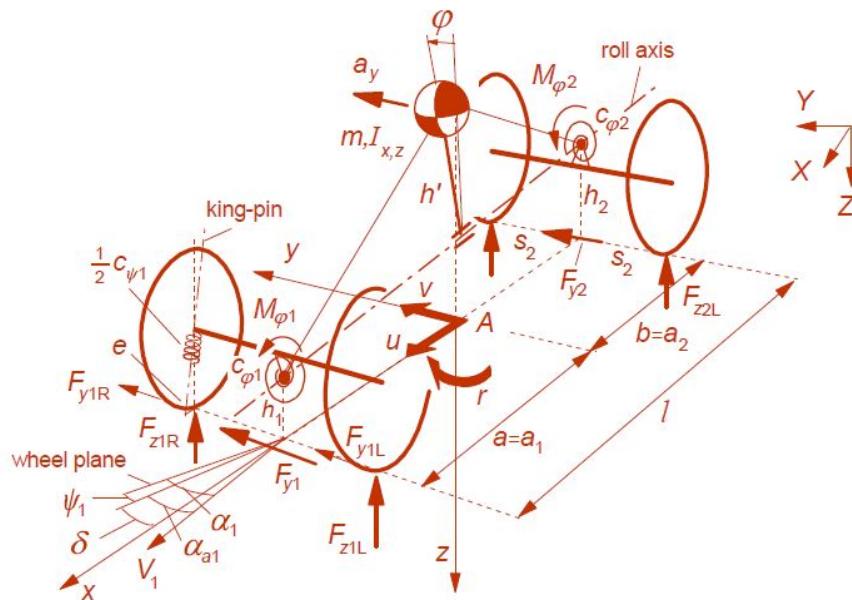


Figure 2-5: Two Track (4 DOF) vehicle model representing, longitudinal velocity (u), lateral velocity, (v), roll angle (ϕ), and yaw rate ($\dot{\psi}$), source: [52]

The vehicle model used in this thesis is a 4 Degree-of-Freedom (DOF) model as illustrated in the Figure 2-5. This model is used because studying a single lane change requires 3 degrees of freedom: Longitudinal, lateral and yaw rate. A fourth degree of freedom, roll rate is required to make the model more adaptive for future experiments at high velocities, or steep turns

or bank angles. Furthermore, this model includes steering compliance on the front axle (\approx steering system flexibility) and rear axle. A brief description is as follows: Consider the reference point A, the model defines longitudinal velocity u , and lateral velocity v , the yaw velocity r and the roll angle ψ . Point A is located in the ground plane. When the roll angle ϕ is equal to zero, this point is below the center of gravity (COG). Furthermore, a motion of vehicle along a curved path causes the body to roll about the roll axis. As illustrated in the Figure 2-5, the virtual roll axis joins the front roll center c_{ϕ_1} and the rear roll center c_{ϕ_2} . The heights of the roll centers are given by h_1 and h_2 . The roll stiffness and damping, which results from suspension springs and anti-roll bars, are modeled with torsional springs and dampers with roll stiffnesses k_{ϕ_i} and damping coefficients c_{ϕ_i} in the roll centers. The distance from center of gravity (COG) to the roll axis is given by h' . Also, included are the tyre dynamics for each tyre based on Pacejka's *Magic Formula*.

Need for a 2 Degree-of-Freedom (DOF) model Although a 4 Degree-of-Freedom (4DOF) vehicle model provides much qualitative information about the vehicle dynamics behavior, it is desirable to linearize the vehicle model to obtain a simpler yet accurate representation of the vehicle behavior. Also, in order to perform 'formal verification' of the hybrid automata (which is affected by the underlying vehicle dynamics), a structured approach of first using a simpler vehicle model for investigations purposes and then progress to more advanced vehicle models, achieves simplicity and allows to focus more on the steering interactions.

The non-linear vehicle model presented previously, can be linearized by assuming the small angle approximation (discussed in next section) and neglecting the higher order terms. However, to make such transitions from non-linear model to linear models, careful assumptions need to be made. Firstly, *constant longitudinal velocity* is required to construct a linear vehicle model. Secondly, it is also assumed that the tire behaves linearly in the vehicle model, which amounts to the use of the linear equations which in turn requires maintaining *tire slip angle* below a certain value. Since, this slip angle value is determined by the vertical tire load, road friction coefficient, and longitudinal slip of the tire, it can be argued that the linear model is valid under constant longitudinal speed, constant normal load on tires, constant coefficient of friction of the road, and constant longitudinal slip of the tires.

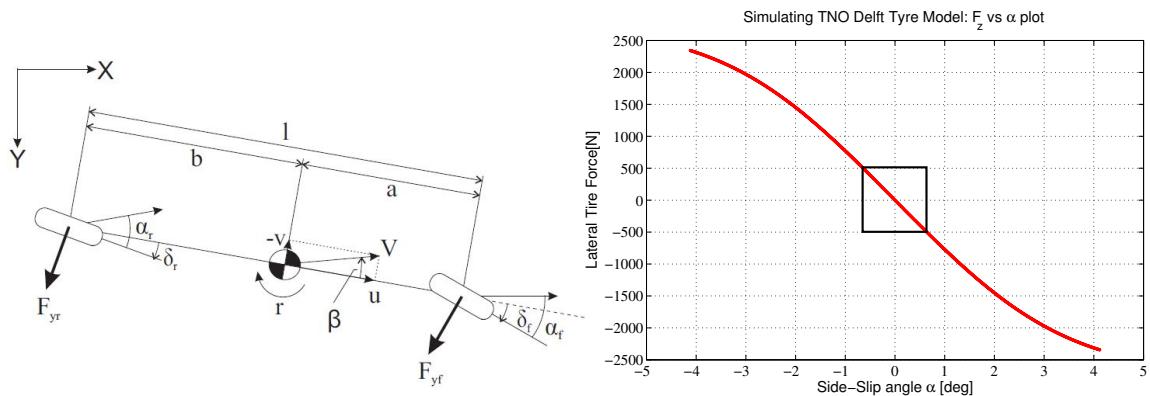


Figure 2-6: (a.) The 2DOF Bicycle model, (b.) Lateral Force F_z of the TNO Delft tyre model plotted against side-slip angles α .

First, we plot the Lateral force against the tire slip angle to obtain the limits under which both the non-linear and linear vehicle models show almost similar behavior. The black rectangle in the Figure 2-6 denotes the *Linear* region of operation of the tire model. The values of obtained side-slip angles for linear region of operation are. $-0.5^\circ \leq \alpha \leq 0.5^\circ$.

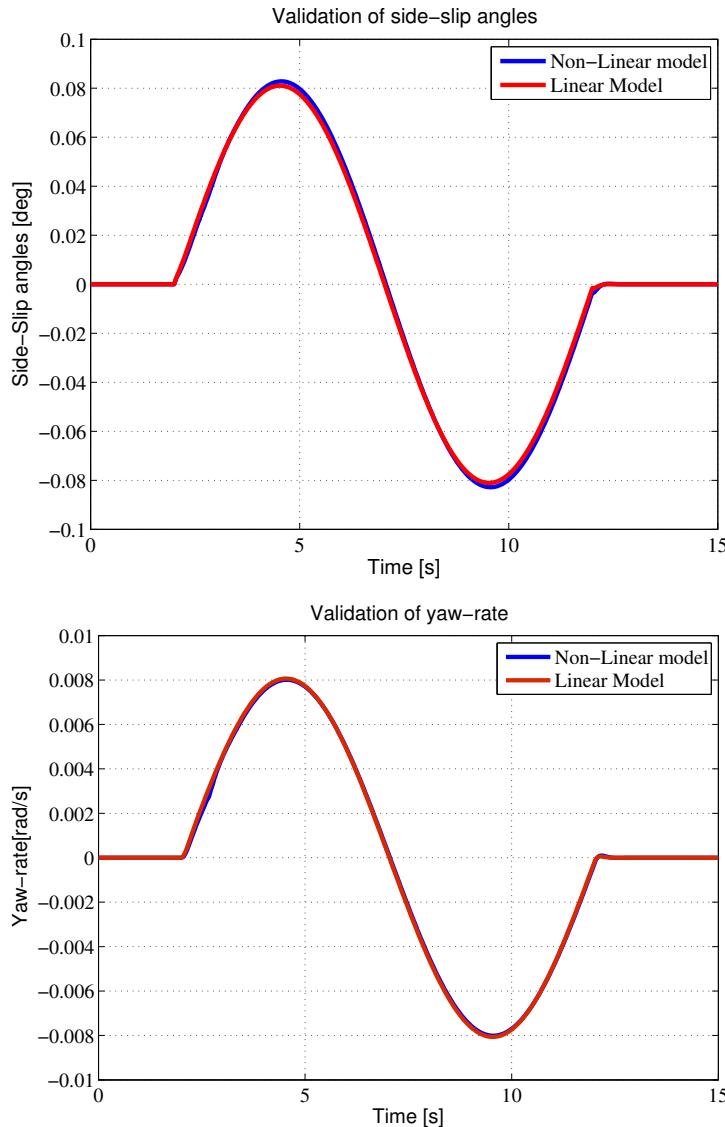


Figure 2-7: (a.) The side-slip angles, (b.) The yaw-rates for both non-linear and linear vehicle models coincide in the linear tire region

The Figure 2-7 illustrates that for a constant longitudinal velocity $V_x = 100$ km/h and a sinusoidal steering angle input for realizing a single lane change of width 3m, indeed the tire *side-slip* angles stay within the bounds as observed previously and also the yaw-rate behaviour is indeed the same for the two models. Thus, from the above experiments it can be argued that indeed the Linear 2DOF model and the Non-Linear 4 DOF model have same responses in the given experimental conditions.

Small Angle approximations

The small-angle approximation is a useful simplification of the basic trigonometric functions which is approximately true in the limit where the angle approaches zero. They are truncations of the Taylor series for the basic trigonometric functions to a second-order approximation. In the domain of vehicle dynamics, the standard linearizing small angle assumptions are made for the tire-slip angles and the front steering angles. Generally if the value of angle lies in the range $-6^\circ < \theta < 6^\circ$, then it can be sufficiently assumed that $\sin(\theta) \approx \tan(\theta) \approx \theta$, $\cos(\theta) \approx 1 - \frac{\theta^2}{2}$, where θ is measured in radians

Validation of small-angle approximations In the forthcoming sections (3-1-1) and (3-2-1), we make use of the small-angle approximations on the yaw angles (ψ) for use in the state-space modeling of the automated controller and human driver model.

To necessitate such a step, the Figure 2-8 illustrates the simulations performed using different longitudinal velocities ($V_x = 80, 90, 100, 110$ [km/h]), for a single lane change of width 3 m as the reference input. It can be observed that the yaw angle responses of the 2 DOF ‘Bicycle model’ respect the small-angle constraints and hence, the following relations can be successfully assumed ($\sin(\psi) \approx \tan(\psi) \approx \psi$, and $\cos(\psi) \approx 1 - \frac{\psi^2}{2}$).

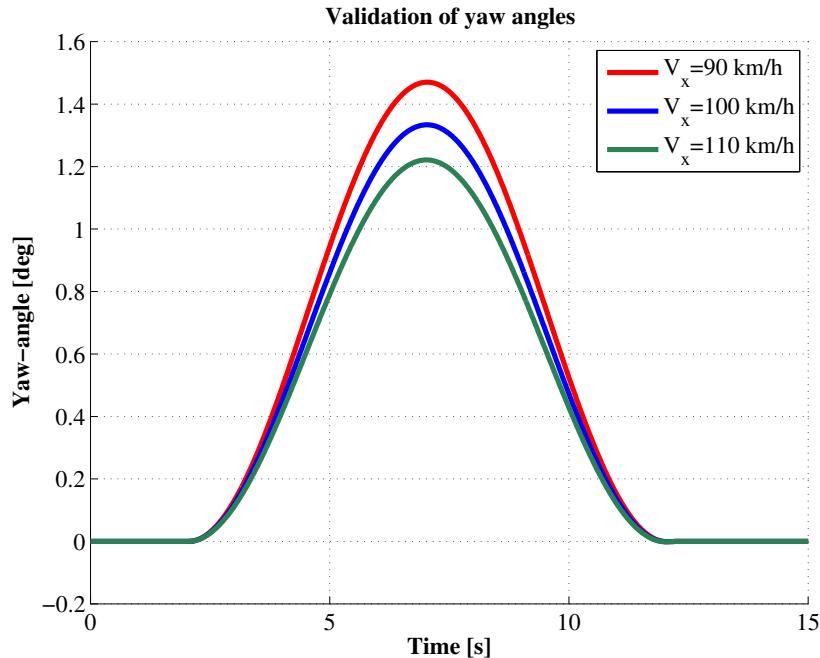


Figure 2-8: Effect of varying longitudinal vehicle velocities V_x on yaw angle. In all cases the $0 < \psi < 2$ degrees, thereby validating small angle assumption

MATLAB Stateflow Environment

STATEFLOW is an interactive Matlab environment for modeling and simulation of complex control and supervisory logic problems, used in conjunction with Simulink. For the design of hybrid systems STATEFLOW is generally used to describe discrete-event systems whereas, Simulink is used to specify the continuous behavior of plant and controllers involved. A Hybrid system consisting of Simulink and STATEFLOW models is *simulated* by transferring the control of the execution in an alternate manner, between their respective embedded simulation engines, a technique referred to in literature as *co-simulation*.

STATEFLOW uses a variation of finite state machines (FSM) coupled together with flow diagram notations, and state-transition diagrams for behavioral modeling of dynamical systems. More accurately, it owes its existence to a mathematical formalism developed in the late 80's [51] referred to as *Statecharts*.

SYNTAX:

- States are represented by a rounded rectangle and connected via arcs/transition. Each transition has a label that provides the qualitative information about the transition. It is defined in as: `event[condition]{condition action}/transition action`.
- A state's syntax consists of the following identifiers in the exact order as they appear here. The name of the state is defined first with the `name` identifier; the `entry` action is executed upon entering the state; the `during` action is executed whenever the model is evaluated and the state cannot be left; the `exit` action is executed when the state is left; finally, the `event_name` action is executed each time the specified event is enabled.
- A STATEFLOW model incorporates `data` and `event` I/O ports, which when set to `external` allow communication with Simulink through ports.

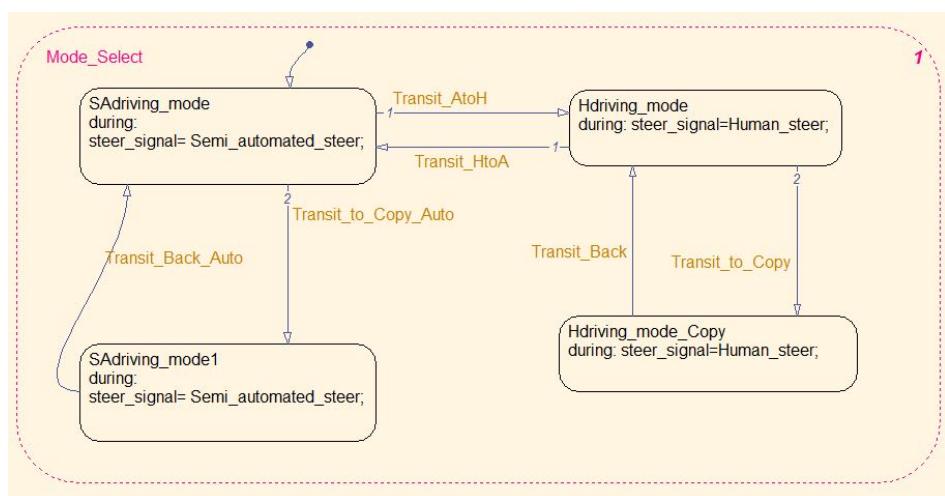


Figure 2-9: Stateflow diagram describing the actions and states that have been used in thesis for Time based Switching

2-5-2 Trajectory Generation for verification

This section explains the formulation of single lane change trajectory of width 3m, that will be used in the forthcoming sections on modeling the ‘automated’ vehicle and the ‘human driver’. The notion that a reference input can be represented as a dynamical system is indeed a helpful one albeit not generally used. A single lane change basically consists of three sections: A *straight* section which is generally the centerline of the current lane, a *curved* section for traversing into the next lane, and a final *straight* section that corresponds to the center line of the next lane. Simulating such a path requires the presence of ‘second order’ dynamics, mathematically represented with differential equations of the form $(s^2 + 2\xi\omega_n s + \omega_n^2)u_{ref} = 0$, where u_{ref} is the reference variable. The Figure 2-10 illustrates the trajectories generated for different lane changes ($\omega_n = 0.4$ represents ‘slow’ lane change, whereas, $\omega_n = 1$ represents a ‘fast’ lane change).

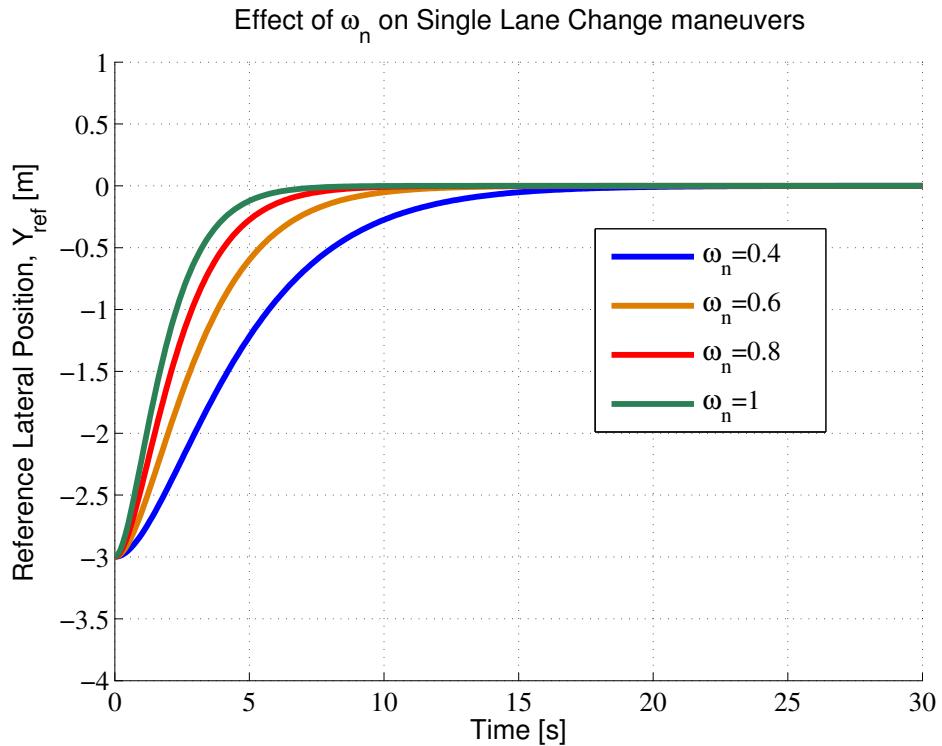


Figure 2-10: Variation of Single Lane Change maneuvers with different ω_n values.

The initial conditions for simulation have been selected as $[\dot{u}_{ref}, u_{ref}] = [0, -3]$. Now, a question may arise regarding the resemblance of such a trajectory to (e.g. ISO 3888 double-lane change) actual lane changes prescribed by ISO, for which it suffices to state that for the purpose of verification via BREACH toolbox the first *straight* section doesn’t correspond to a significant change in the dynamical behavior. In fact the vehicle just drives straight in the absence of any obstacle. The real challenge lies in simulating the behavior when the human takes the control of the vehicle before entering the lane change(so the *second* section of lane change) and when the automated reclaims control (the *third* section).

Chapter 3

Designing the Cruise Control and Steering Control

Studies in man-machine control have been a rich and ever-expanding. Owing to the enormity of research, understanding and modeling human driver behaviour, or its effects on automation thereof, have become too broad in scope to be exhaustively considered in this research work. To elucidate, studies in human driver modeling have a two-pronged classification structure: First, based on Control theoretic models, identification theory, fuzzy and neural control etc., and second, based on Human factors aspects, which focuses on evaluating driver behaviour and performance, wherein the former is concerned with choices people make regarding driving goals, performance trade-offs, and acceptable safety margins and latter is associated with cognitive limits of attention and perception. Due to the nature of this research, the focus will be on the control-theoretic aspect of the human driver and its subsequent computer-based modeling, so the steering and speed control attributes of human driver as a longitudinal and lateral controller of the vehicle are discussed. For a comprehensive and detailed review of human driver interactions with vehicle, the reader is referred to seminal work by [55].

3-1 Modeling the Automated Controller

Consider, the closed-loop block diagram described in the Figure 3-1. The three blocks represent the PD steering controller, the steering dynamics and the vehicle dynamics block. The methodology described in the next sub-sections is, to first obtain the *state-space* form of the entire plant-controller system that represents the equivalent ‘non-autonomous’ system and then by incorporating the reference dynamics (2nd order differential equation of trajectory, refer § 2-5-2), transform it into the corresponding autonomous system.

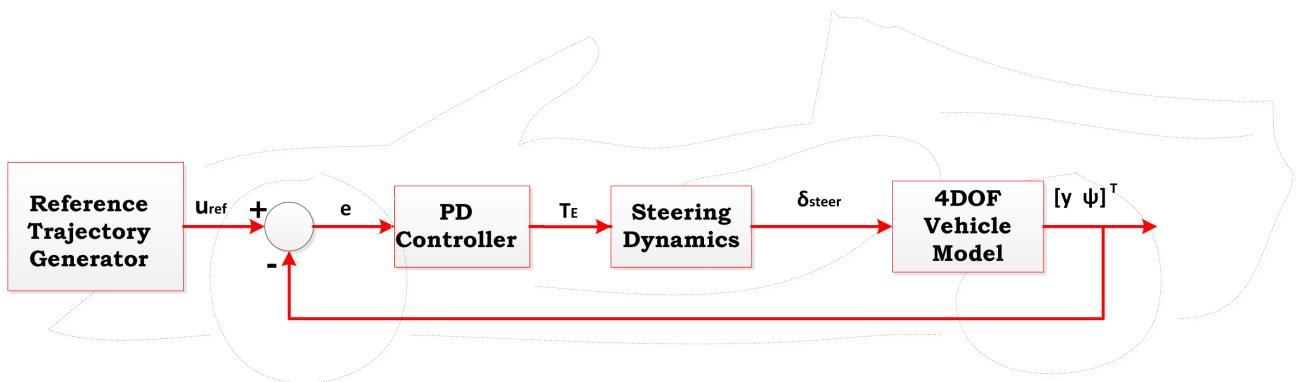


Figure 3-1: Block Diagram of Automated Vehicle

3-1-1 Deriving the non-autonomous state-space model

The idea is to obtain a model of the form :

$$\dot{X}_c = A_c X_c + B_c u \quad (3-1)$$

$$Y_c = C_c X_c + D_c u \quad (3-2)$$

where,

X_c , are the system states, u , Y_c are the input, and output to the system respectively and A_c, B_c, C_c, D_c represent the state-space matrices.

Let's start with the 2 DOF (Degree-of-freedom) vehicle dynamics model, which can be represented in state-space form as

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{(C_{\alpha_1}+C_{\alpha_2})}{mV_x} & -(V_x + \frac{aC_{\alpha_1}-bC_{\alpha_2}}{mV_x}) \\ -\frac{(aC_{\alpha_1}-bC_{\alpha_2})}{V_x I_{zz}} & -\frac{(a^2C_{\alpha_1}+b^2C_{\alpha_2})}{V_x I_{zz}} \end{bmatrix} \begin{bmatrix} V_y \\ r \end{bmatrix} + \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} \delta_{st} \quad (3-3)$$

V_x = Longitudinal velocity

V_y = Lateral velocity

m = Vehicle mass

a, b = distance from vehicle COG to front and back axle respectively

I_{zz} = Moment of Inertia in z-direction

$C_{\alpha_1}, C_{\alpha_2}$ = Steering Coefficients

r = yaw rate

δ_{st} = steering angle input

$$B_{11} = \frac{C_{\alpha_1}}{m}$$

$$B_{21} = \frac{aC_{\alpha_1}}{I_{zz}}$$

Now, we *augment* the steering dynamics and vehicle dynamics block to form a extended state-space model. The equations are then described as follows-

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \\ \ddot{\delta}_{st} \\ \dot{\delta}_{st} \end{bmatrix} = \begin{bmatrix} -\frac{(C_{\alpha_1}+C_{\alpha_2})}{mV_x} & -(V_x + \frac{aC_{\alpha_1}-bC_{\alpha_2}}{mV_x}) & 0 & B_{11} \\ -\frac{(aC_{\alpha_1}-bC_{\alpha_2})}{V_x I_{zz}} & -\frac{(a^2C_{\alpha_1}+b^2C_{\alpha_2})}{V_x I_{zz}} & 0 & B_{12} \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \dot{\delta}_{st} \\ \delta_{st} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ J_w \\ 0 \end{bmatrix} [T_E] \quad (3-4)$$

where,

- B_w = Damping
- K_w = Steering wheel Stiffness
- J_w = Moment of inertia of steering wheel
- T_E = Steering torque applied by automated controller

As a final step in the formulation of state-space of entire closed loop system (A_c, B_c, C_c, D_c), we try to augment the extended state-space model developed in the equation Eq. (3-4) with the PD controller (refer, Figure 3-1). A point worth noting in this context is that, a pure *derivative action (D)* can't be implemented in the state-space form as the system becomes *improper*. We then try to use a **softened derivative action** by considering the equivalent of a *low pass filter*.

Defining the Input-output dynamics of PD controller mathematically, gives the following relation :

$$T_E = (C_1 + C_2 \frac{s}{\tau s + 1})e \quad (3-5)$$

where, as illustrated in the Figure 1,

- T_E = Engaging torque applied by controller
- e = Error input to the controller
- C_1 = Proportional Gain
- C_2 = Derivative Gain
- τ = $\frac{C_2}{10}$ (as a rule of thumb) [59]

State space derivation of PD controller

It is worth deriving the state-space model of the PD controller at this stage, as it is required for the modeling of a combined plant-controller system as indicated at the outset of this section, refer Eq. (3-1) and Eq. (3-2). Also, explicitly modeling the PD controller would allow us to investigate the effects of controller gains in the later sections involving hybrid dynamical systems. Furthermore, exploiting the flexibility allowed in modeling the state-space model of any system, we introduce a dummy state state z , which represents an algebraic relation between Torque T_E and error input to controller e .

$$\text{From Eq. (3-5): } T_E(\tau s + 1) = [C_1(\tau s + 1) + C_2 s]e \quad (3-6)$$

$$\text{Inverse Laplace transform: } \tau \dot{T}_E + T_E = K\dot{e} + C_1 e \quad (3-7)$$

$$\text{Define dummy state 'z': } z = \tau(T_E) - K(e) \quad (3-8)$$

$$\text{State Equation: } \dot{z} = -\frac{z}{\tau} + (C_1 - \frac{K}{\tau})e \quad (3-9)$$

$$\text{Output Equation: } T_E = \frac{z}{\tau} + \frac{K}{\tau}e \quad (3-10)$$

where for simplification we consider, $K = (C_1\tau + C_2)$

Now, with the PD controller mathematically defined in the previous section, we *augment* the vehicle-steering dynamics Eq. (3-4) with the Eq. (3-9) and Eq. (3-10). The resulting state-space model then completely defines the plant-controller dynamics, and relates the input (*the error signal, e*) and the output (*Engaging torque, T_E*).

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \\ \ddot{\delta}_{st} \\ \dot{\delta}_{st} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\frac{(C_{\alpha_1}+C_{\alpha_2})}{mV_x} & -(V_x + \frac{aC_{\alpha_1}-bC_{\alpha_2}}{mV_x}) & 0 & B_{11} & 0 \\ -\frac{(aC_{\alpha_1}-bC_{\alpha_2})}{V_x I_{zz}} & -\frac{(a^2C_{\alpha_1}+b^2C_{\alpha_2})}{V_x I_{zz}} & 0 & B_{12} & 0 \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} & \frac{J_w}{\tau} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \dot{\delta}_{st} \\ \delta_{st} \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{J_w K}{\tau} \\ 0 \\ -\frac{C_2}{\tau} \end{bmatrix} [e] \quad (3-11)$$

Now, for the obtaining the closed-loop state-space model of the system, we consider the following equations:

$$\text{Error signal: } e = (u_{ref} - y - d_c \psi) \quad (3-12)$$

$$\text{Where, } d_c = \text{Human preview distance} \quad (3-13)$$

$$\text{Output feedback: } \dot{y} = V_x \sin(\psi) + V_y \cos(\psi) \quad (3-14)$$

$$\text{Small angle approx. (Ref. Section 2.3): } \dot{y} = V_x(\psi) + V_y \quad (3-15)$$

$$\text{yaw rate feedback: } \dot{\psi} = r \quad (3-16)$$

The final system model in terms of reference trajectory input $u = u_{ref}$, and the output (*Engaging torque, T_E*) is given by following state-space formulation:

$$\dot{X}_c = [A_c] X_c + [B_c] u_{ref} \quad (3-17)$$

$$T_E = [C_c] X_c + [D_c] u_{ref} \quad (3-18)$$

where,

$X_c = [V_y, r, \dot{\delta}_{st}, \delta_{st}, z, y, \psi]^T$, are the system states

$Y_c = T_E$, Engaging torque applied by controller.

$$A_c = \begin{bmatrix} -\frac{(C_{\alpha_1}+C_{\alpha_2})}{mV_x} & -(V_x + \frac{aC_{\alpha_1}-bC_{\alpha_2}}{mV_x}) & 0 & B_{11} & 0 & 0 & 0 \\ -\frac{(aC_{\alpha_1}-bC_{\alpha_2})}{V_x I_{zz}} & -\frac{(a^2C_{\alpha_1}+b^2C_{\alpha_2})}{V_x I_{zz}} & 0 & B_{12} & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} & \frac{J_w}{\tau} & -\frac{J_w K}{\tau} & -\frac{J_w K d_c}{\tau} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} & \frac{C_2}{\tau} & \frac{C_2 d_c}{\tau} \\ 1 & 0 & 0 & 0 & 0 & 0 & V_x \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0 & 0 & \frac{J_w K}{\tau} & 0 & -\frac{C_2}{\tau} & 0 & 0 \end{bmatrix}^T,$$

$$C_c = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\tau} & -\frac{K}{\tau} & -\frac{K d_c}{\tau} \end{bmatrix}, \quad D_c = \begin{bmatrix} \frac{K}{\tau} \end{bmatrix}$$

3-1-2 Transformation to autonomous state-space model

The stability analysis of hybrid systems that follows is done based on the fact that, the states of hybrid automata and their related dynamics are self-contained, i.e. there are no ‘exogenous inputs’ to the system that affect its dynamics in any form. Hence, the non-autonomous state-space obtained previously in Eq. (3-17) and Eq. (3-18) must be converted into a completely autonomous form, represented mathematically as:

$$\dot{X}_{auto} = A'_1 X_{auto} \quad (3-19)$$

The section 2.3 explains in detail how the modeling of reference trajectory can be done suitably by 2nd order differential equations of the form $(s^2 + 2\xi\omega_n s + \omega_n^2)u_{ref} = 0$, with the initial conditions as $[\dot{u}_{ref}, u_{ref}] = [0, -3]$.

The following equations represent the state-space realization of the reference trajectory:

$$\begin{bmatrix} \ddot{u}_{ref} \\ \dot{u}_{ref} \end{bmatrix} = \begin{bmatrix} -2\xi\omega_n & -\omega_n^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{u}_{ref} \\ u_{ref} \end{bmatrix} \quad (3-20)$$

Augmenting the system of equations Eq. (3-17), Eq. (3-18) with Eq. (3-20), we obtain a 9th order *autonomous* model, that describes the plant-controller dynamics of automated vehicle completely.

$$\begin{bmatrix} \dot{X}_c \\ \ddot{u}_{ref} \\ \dot{u}_{ref} \end{bmatrix} = \begin{bmatrix} A_c & 0 & B_c \\ 0 & -2\xi\omega_n & -\omega_n^2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ \dot{u}_{ref} \\ u_{ref} \end{bmatrix} \quad (3-21)$$

Now, comparing to the equation Eq. (3-19), we observe the following state-space model with:

$\dot{X}_{auto} = [X_c, \dot{u}_{ref}, u_{ref}]^T$, are the states of the system

$A'_1 = \begin{bmatrix} A_c & 0 & B_c \\ 0 & -2\xi\omega_n & -\omega_n^2 \\ 0 & 1 & 0 \end{bmatrix}$ is the state matrix

3-2 Modeling the Human Controller

The Human controller is described in Figure 3-2 as a closed-loop system with different blocks, Controller, Human dynamics, Steering dynamics, and Vehicle dynamics. Here again, in a manner similar to that used in the last section, the idea is to derive the state space model of the form :

$$\dot{X}_h = A_h X_h + B_h u \quad (3-22)$$

$$Y_h = C_h X_h + D_h u \quad (3-23)$$

where,

X_h, Y_h, u = States, output, and input respectively of the combined human-vehicle system.

A_h, B_h, C_h, D_h = state-space matrices.

We progress with a block-by-block approach to arrive at the state-space formulation. We start with augmenting the blocks, *steering dynamics* and *vehicle dynamics* and focus on deriving a state-space representation of the system dynamics.

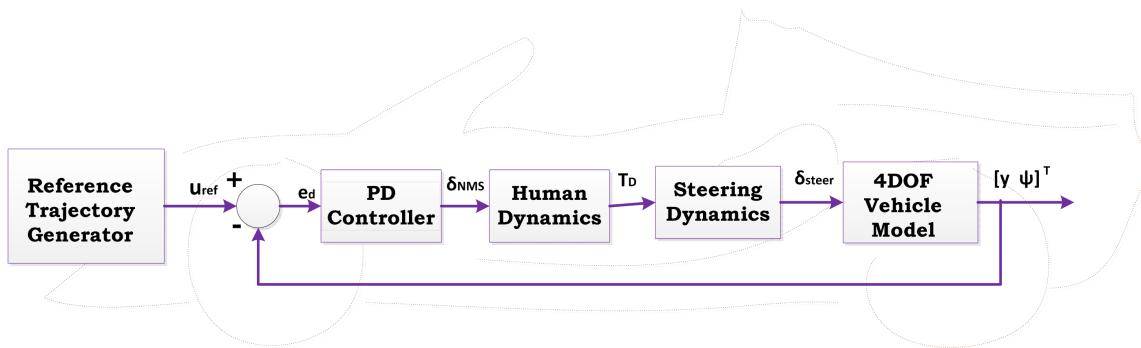


Figure 3-2: Block Diagram of Human Controlled Vehicle

$$\begin{bmatrix} V_y \\ \dot{r} \\ \ddot{\delta}_{st} \\ \dot{\delta}_{st} \end{bmatrix} = \begin{bmatrix} -\frac{(C_{\alpha_1} + C_{\alpha_2})}{mV_x} & -(V_x + \frac{aC_{\alpha_1} - bC_{\alpha_2}}{mV_x}) & 0 & B_{11} \\ -\frac{(aC_{\alpha_1} - bC_{\alpha_2})}{V_x I_{zz}} & -\frac{(a^2 C_{\alpha_1} + b^2 C_{\alpha_2})}{V_x I_{zz}} & 0 & B_{12} \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \dot{\delta}_{st} \\ \delta_{st} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ J_w \\ 0 \end{bmatrix} [T_D] \quad (3-24)$$

where,

$$B_{11} = \frac{C_{\alpha_1}}{m}$$

$$B_{12} = \frac{aC_{\alpha_1}}{I_{zz}}$$

B_w = Damping

K_w = Steering wheel Stiffness

J_w = Moment of inertia

T_D = Steering torque applied by Human driver

Next, we augment the other two blocks *PD controller* and *human dynamics* into a single state-space representation. *Appendix A* describes in detail the mathematical derivation of the State-space of the combined block. Here, we provide the final result for completeness:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{m}_1 \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & \tau_d \\ 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ m_1 \end{bmatrix} + \begin{bmatrix} \frac{k_d}{\tau_d} \\ 0 \\ -\frac{C_{2D}}{\tau_d} \end{bmatrix} [e_d] \quad (3-25)$$

$$T_D = \begin{bmatrix} \xi_1 & \xi_2 & \frac{K_0}{\tau_d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ m_1 \end{bmatrix} + \begin{bmatrix} \frac{K_0 k_d}{\tau_d} \end{bmatrix} [e_d] \quad (3-26)$$

where, if k_p is Human Driver Gain, τ_L and τ_I are the Lead and Lag Constants and τ_d is the neuromuscular delay constant.

$$\begin{aligned} K_0 &= -\frac{\tau_L \cdot k_p}{\tau_I} \\ l_1 &= \frac{\tau_L + \frac{(\tau_d + \tau_N)}{2}}{\tau_I \frac{(\tau_d + \tau_N)}{2}}, \quad l_2 = \frac{1}{\tau_I \frac{(\tau_d + \tau_N)}{2}} \\ \xi_1 &= \frac{\tau_L - \frac{(\tau_d + \tau_N)}{2}}{\tau_I \frac{(\tau_d + \tau_N)}{2}}, \quad \xi_2 = \frac{k_p}{\tau_I \frac{(\tau_d + \tau_N)}{2}} \end{aligned}$$

Now, with the individual blocks defined, we *augment* the vehicle-steering dynamics Eq. (3-24) with the Human-controller dynamics Eq. (A-9) and Eq. (A-10). The resulting state-space model then completely defines the plant-controller dynamics, and relates the input (*the error signal, e*) and the output (*Disengaging torque, T_D*)

$$\dot{X}'_h = A'_h X'_h + B'_h [e_d] \quad (3-27)$$

$$T_D = C'_h X'_h + \left[\frac{K_0 k_d}{\tau_d} \right] [e_d] \quad (3-28)$$

where,

$$\begin{aligned} X'_h &= \begin{bmatrix} V_y & r & \dot{\delta}_{st} & \delta_{st} & x_1 & x_2 & m_1 \end{bmatrix}^T \\ A'_h &= \begin{bmatrix} -\frac{(C_{\alpha_1}+C_{\alpha_2})}{mV_x} & -(V_x + \frac{aC_{\alpha_1}-bC_{\alpha_2}}{mV_x}) & 0 & B_{11} & 0 & 0 & 0 \\ -\frac{(aC_{\alpha_1}-bC_{\alpha_2})}{V_x I_{zz}} & -\frac{(a^2 C_{\alpha_1} + b^2 C_{\alpha_2})}{V_x I_{zz}} & 0 & B_{12} & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} & J_w \xi_1 & J_w \xi_2 & \frac{J_w K_0}{\tau_d} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -l_1 & -l_2 & \frac{1}{\tau_d} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_d} \end{bmatrix} \\ B'_h &= \begin{bmatrix} 0 & 0 & \frac{J_w k_d K_0}{\tau_d} & 0 & \frac{k_d}{\tau_d} & 0 & -\frac{C_2}{\tau_d} \end{bmatrix}^T \\ C'_h &= \begin{bmatrix} 0 & 0 & 0 & 0 & \xi_1 & \xi_2 & \frac{K_0}{\tau_d} \end{bmatrix} \end{aligned}$$

Now, for obtaining the closed-loop state-space model of the system, we consider the following equations. We also introduce a variable '*d_h*' which is the *preview distance* of the human controller. It is defined as the distance up-to which the controller observes the road trajectory ahead and applies corrective action.

$$\text{Error signal: } e_d = (u_{ref} - y - d_h \psi) \quad (3-29)$$

$$\text{Where, } d_h = \text{Human preview distance} \quad (3-30)$$

$$\text{Output feedback: } \dot{y} = V_x \sin(\psi) + V_y \cos(\psi) \quad (3-31)$$

$$\text{Small angle approx. (Ref. Section 2.3): } \dot{y} = V_x(\psi) + V_y \quad (3-32)$$

$$\text{yaw rate feedback: } \dot{\psi} = r \quad (3-33)$$

The final system model in terms of reference trajectory input $u = u_{ref}$, and the output (*Disengaging torque*, T_D) is given by following state-space formulation:

$$\dot{X}_h = [A_h] X + [B_h] u_{ref} \quad (3-34)$$

$$T_D = [C_h] X + [D_h] u_{ref} \quad (3-35)$$

where,

$X_h = [V_y, r, \dot{\delta}_{st}, \delta_{st}, x_1, x_2, m_1, y, \psi]^T$, are the system states

$Y_h = T_D$, Disengaging torque applied by human.

$$A_h = \begin{bmatrix} A_{11} & A_{12} & 0 & B_{11} & 0 & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & B_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} & J_w\xi_1 & J_w\xi_2 & \frac{J_w K_0}{\tau_d} & -\frac{J_w k_d K_0}{\tau_d} & -\frac{J_w k_d K_0 d_h}{\tau_d} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -l_1 & -l_2 & \frac{1}{\tau_d} & -\frac{k_d}{\tau_d} & -\frac{k_d d_h}{\tau_d} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_d} & \frac{C_{2D}}{\tau_d} & \frac{C_{2D} d_h}{\tau_d} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & V_x \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_{11} = -\frac{(C_{\alpha_1} + C_{\alpha_2})}{mV_x}, \quad A_{12} = -(V_x + \frac{aC_{\alpha_1} - bC_{\alpha_2}}{mV_x})$$

$$A_{21} = -\frac{(aC_{\alpha_1} - bC_{\alpha_2})}{V_x I_{zz}}, \quad A_{22} = -\frac{(a^2 C_{\alpha_1} + b^2 C_{\alpha_2})}{V_x I_{zz}}$$

$$B_h = \begin{bmatrix} 0 & 0 & \frac{J_w k_d K_0}{\tau_d} & 0 & \frac{k_d}{\tau_d} & 0 & -\frac{C_{2D}}{\tau_d} & 0 & 0 \end{bmatrix}^T,$$

$$C_h = \begin{bmatrix} 0 & 0 & 0 & 0 & \xi_1 & \xi_2 & \frac{K_0}{\tau_d} & -\frac{K_0 k_d}{\tau_d} & \frac{K_0 k_d d_h}{\tau_d} \end{bmatrix}, \quad D_h = \begin{bmatrix} \frac{K_0 k_d}{\tau_d} \end{bmatrix}$$

3-2-1 Transformation to Linear autonomous system

In a manner similar to that described in § 3-1-2, the non-autonomous state-space of Eq. (3-34) and Eq. (3-35) is converted into a completely autonomous form, represented mathematically as:

$$\dot{X}_{hum} = A'_2 X_{hum} \quad (3-36)$$

Augmenting the system of equations Eq. (3-34), Eq. (3-35) and Eq. (3-20), we obtain a 11th order *autonomous* model, that describes the plant-controller dynamics of manually controlled vehicle completely.

$$\begin{bmatrix} \dot{X}_h \\ \ddot{u}_{ref} \\ \dot{u}_{ref} \end{bmatrix} = \begin{bmatrix} A_h & 0 & B_h \\ 0 & -2\xi\omega_n & -\omega_n^2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_h \\ \dot{u}_{ref} \\ u_{ref} \end{bmatrix} \quad (3-37)$$

Comparing above with the equation Eq. (3-36), we obtain the following relations:

$$\dot{X}_{hum} = [X_h, \dot{u}_{ref}, u_{ref}]^T, \text{ are the states of the system}$$

$$A'_2 = \begin{bmatrix} A_h & 0 & B_h \\ 0 & -2\xi\omega_n & -\omega_n^2 \\ 0 & 1 & 0 \end{bmatrix}, \text{ is the state matrix}$$

3-3 Formulation of equidimensional models

A closer look at the equations Eq. (3-21) and Eq. (3-37) reveals the dimensions of the plant-controller dynamics of automated vehicle (**9th-order**) and manually controlled vehicle (**12th-order**). Since, for analyzing the stability of hybrid automata, it is necessary that we combine the models to obtain equidimensional state space models for representing ‘continuous’ dynamics in both modes of driving, we proceed mathematically as follows:

Consider, two systems described by first-order differential equations $\dot{Y}_1 = A_{y1} \times Y_1$ and $\dot{Y}_2 = A_{y2} \times Y_2$, where, $A_{y1} \in \mathbb{R}^{n \times n}$ and $A_{y2} \in \mathbb{R}^{m \times m}$, $\forall n < m$. Now, let's assume that the state vector Y_1 or Y_2 is composed of states common to both, as well as states exclusive to individual subsystems. Represented mathematically, this would amount to $Y_1 = [Y, K_1, K_2, \dots, K_{n-q}]^T$ and similarly for second state, $Y_2 = [Y, M_1, M_2, \dots, M_{m-q}]^T$, if, $Y \in \mathbb{R}^{q \times 1}$. We then, create a common state vector $Y_{com} \in \mathbb{R}^{n+m-q}$, which basically augments all the states into one vector, taking into account the redundancy of common states that could occur. So, the resulting vector would look something like: $Y_{com} = [Y, K_1, K_2, \dots, K_{n-q}, M_1, M_2, \dots, M_{m-q}]$. Then it remains to show that resulting matrices A_{y1}^{cm} and A_{y2}^{cm} have equidimensional spatial representation, i.e. they both belong to $\mathbb{R}^{n+m-q \times n+m-q}$.

In accordance with logic put forth in the last paragraph, considering equations Eq. (3-19) and Eq. (3-36) we define a common state vector X_{com} , where

$$X_{com} = [V_y, r, \dot{\delta}_{st}, \delta_{st}, y, \psi, \dot{u}_{ref}, u_{ref}, z, x_1, x_2, m_1]^T, \text{ where, } X_{com} \in \mathbb{R}^{12 \times 1}.$$

For system (q₁) this corresponds to:

$$\dot{X}_{com} = [A_1] \times X_{com} \quad (3-38)$$

where, $\mathbf{A}_1[12 \times 12] =$

$$\begin{bmatrix} A_{11} & A_{12} & 0 & B_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & B_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} & -\frac{J_w K}{\tau} & -\frac{J_w K d_c}{\tau} & 0 & \frac{J_w K}{\tau} & \frac{J_w}{\tau} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & V_x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2\xi\omega_n & -\omega_n^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_2}{\tau} & \frac{C_2 d_c}{\tau} & 0 & -\frac{C_2}{\tau} & -\frac{1}{\tau} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For system (q₂) this corresponds to:

$$\dot{X}_{com} = [A_2] \times X_{com} \quad (3-39)$$

where, $\mathbf{A}_2[12 \times 12] =$

$$\begin{bmatrix} A_{11} & A_{12} & 0 & B_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & B_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_w}{J_w} & -\frac{K_w}{J_w} & -\frac{J_w K_0 k_d}{\tau_d} & -\frac{J_w K_0 k_d d_h}{\tau_d} & 0 & \frac{J_w K_0 k_d}{\tau} & 0 & J_w \xi_1 & J_w \xi_2 & \frac{J_w K_0}{\tau} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & V_x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2\xi\omega_n & -\omega_n^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{k_d}{\tau_d} & -\frac{k_d d_h}{\tau_d} & 0 & \frac{k_d}{\tau_d} & 0 & -l_1 & -l_2 & \frac{1}{\tau_d} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{C_{2D}}{\tau_d} & -\frac{C_{2D} d_h}{\tau_d} & 0 & -\frac{C_{2D}}{\tau_d} & 0 & 0 & 0 & -\frac{1}{\tau_d} \end{bmatrix}$$

3-4 Designing the Speed Control

Vehicles equipped with longitudinal controller relegate the human driver from the act of constantly maintaining a ‘certain’ velocity (referred to as Cruise Control, CC) or both a constant velocity and pre-fixed distance (referred to as Adaptive Cruise Control, ACC), between the host vehicle and preceding vehicle. Studies [56, 60, 61] describe the longitudinal control model of an ACC vehicle as usually composed of two separate controllers,(1) the *upper level controller* and (2) the *lower level controller*. The upper level controller uses the range (spacing difference) and range-rate (velocity difference) between the ACC vehicle and its preceding vehicle and the ACC vehicle’s velocity and acceleration to determine the acceleration commands to perform the maneuvers required in the spacing-control law of the ACC vehicle. The lower-level controller uses this acceleration signal to generate the throttle and braking commands to track the spacing-control law computed by the upper-level controller.

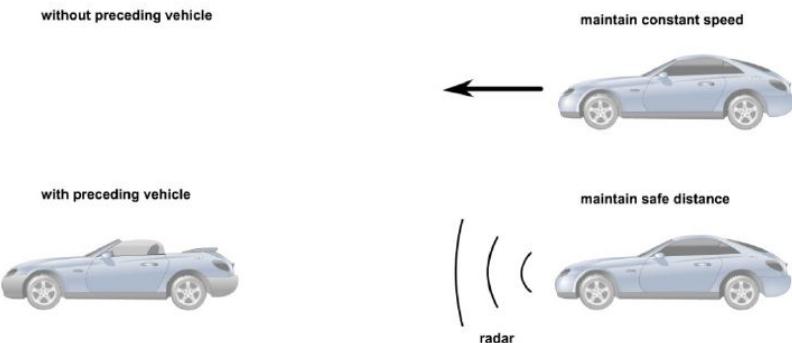


Figure 3-3: ACC equipped vehicle, Source : [63]

Scope of the present research

- It should be noted that the scope of this research lies only till the development of cruise control (longitudinal control). This is because the speed control is merely required to maintain a constant vehicle speed during the lane change maneuver and is not really a part of the system analysis. So the concepts of advanced driver assistance systems like Adaptive Cruise Control (ACC) and Cooperative adaptive cruise control (CACC) will not be discussed.
- This research work is limited to design and analysis of upper level controllers and a very brief introduction will be provided on lower-level controllers (which require actuation of throttle and brakes to achieve ‘desired’ acceleration output of upper level controller). For further reading in this domain [62] provides an interesting perspective.

Design Objectives and Motivation

One of the most influential works in the area of vehicle dynamics control [63] states that, in performance specifications for the design of the longitudinal controller, it is necessary to specify that the steady state tracking error of the controller should be zero. In other words, the speed of the vehicle should converge to the desired speed set by the driver. Taking a cue from the observations put forth by [63], the longitudinal controller to be designed for this thesis should satisfy the following objectives.

1. Steady-state error= 0.
2. Rise time < 12 s.
3. Zero overshoot.

The *motivation* behind using a PID control strategy for the longitudinal controller is attributed to the fact that the test scenario for experimentation (refer Section 2.3) is limited to:

- A. Standard highway driving scenario. Sharp turns, wet asphalt, off-road conditions will never be accounted for in this research. So, using advanced control techniques would make the problem more cumbersome to tackle, than it already is.
- B. Since it is desirable that the controller should be able to maintain constant speed during stationary conditions it is natural to choose a controller with integral action.
- C. Since, we are not interested in experimenting with the vehicle behavior at its *handling limits*, we observe very small steering angle deviation as aggressive maneuvers don't take place. Hence, a PID controller suffices for reference tracking with very small deviations.

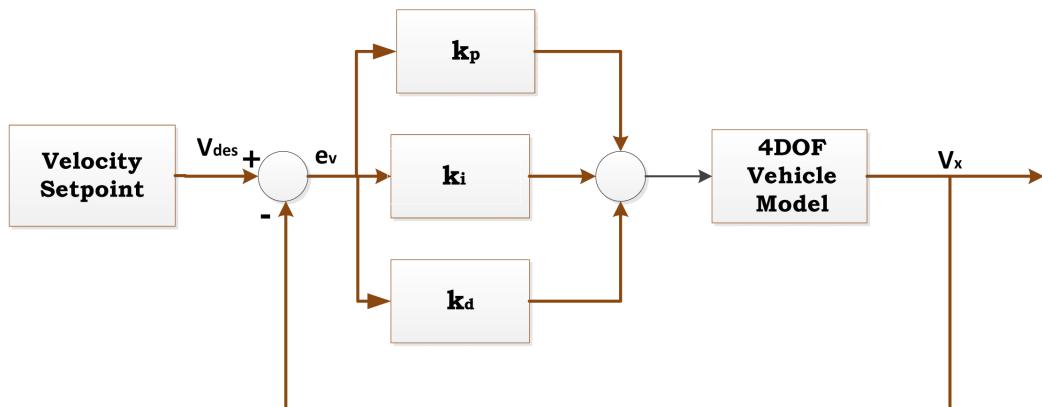


Figure 3-4: Block Diagram of Cruise Controller design using PID control logic

Mathematical Formulation

The purpose of a cruise control system is to accurately maintain the driver's desired set speed, without intervention from the driver, by actuating the throttle-accelerator pedal linkage. A modern automotive cruise control is a feedback control loop that operates on the error ($e = V_{des} - V_x$) between desired velocity and the actual velocity, and based on a PID (proportional-integral-derivative) action drives the steady state error to zero. The control algorithm:

$$u_c = k_p(V_{des} - V_x) + k_i \int_0^t (V_{des} - V_x(\tau)) d\tau + k_d \frac{d(V_{des} - V_x)}{dt} \quad (3-40)$$

where,

- V_x = actual vehicle longitudinal velocity
- V_{des} = Desired speed set by the user
- k_p, k_i, k_d = Proportional, Integral and Derivative gain constants

Validation of Cruise control

Figure 3-5 illustrates the desired velocity profile for the Cruise Control (CC) equipped vehicle. It starts from zero speed, ramps up, and finally settles to a desired user velocity of 100 km/h [27.778 m/s]. As can be observed the PID controller is able to meet the desired specifications (zero steady state error, zero overshoot and the rise time is = 7.5 s) thereby validating its efficiency.

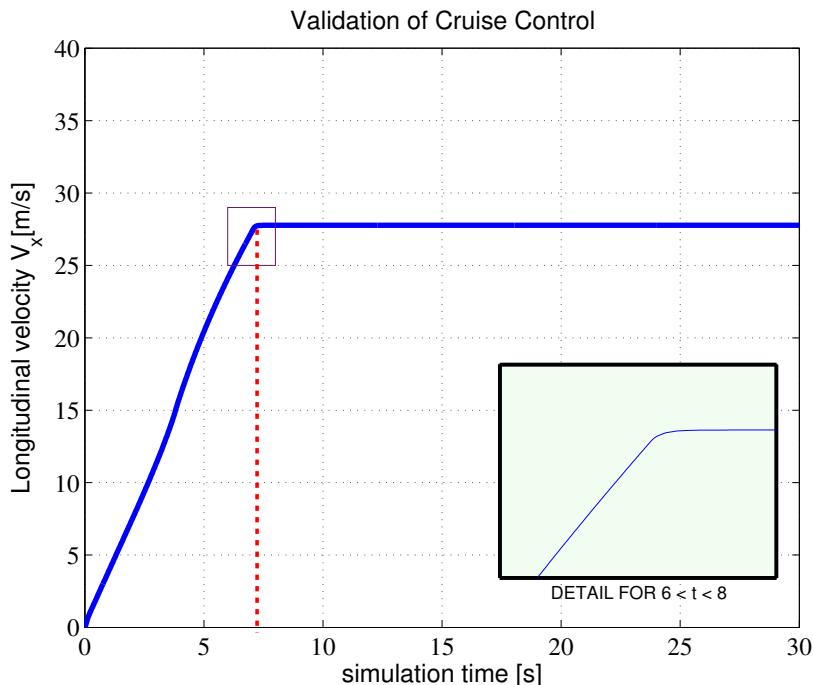


Figure 3-5: Realized velocity profile of the Cruise Control equipped vehicle

3-5 Designing the Steering Control

In all generality, the task of driving a vehicle is generally two-fold. It is controlling the longitudinal behaviour (throttle and brake control) and the lateral behaviour (steering/lane keeping control) while trying to navigate a set trajectory. The former aspect (cruise control) has been designed in the previous section. In this section, the lateral control that occurs in human driving is focused upon, based on which the steering controller for the automated vehicle is also developed.

Human driver steering control

Human driving is basically adaptive in the sense that a driver tries to adjust his motor actions in order to minimize the error between current path and desired path. The first preview model was proposed by [64] wherein the author describes a linear predictive model. The steering action in this case is decided upon the lateral error between the desired road and the predicted vehicle position a certain distance ahead of the vehicle.

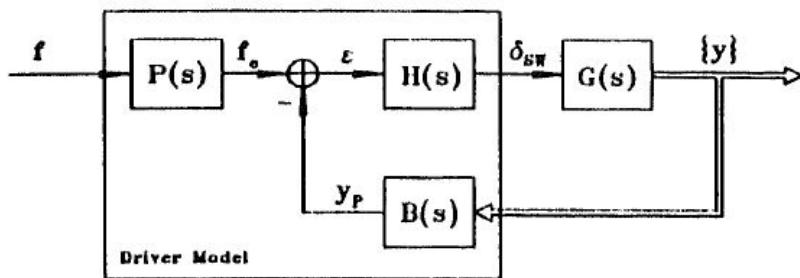


Figure 3-6: Driver Preview Tracking model, source: [64]

The Preview Tracking principle has been used in this thesis to design the steering controller part of Human driver model. Its working can be briefly summarized as follows : Consider, that while driving the human driver perceives a lateral error (e_p) between *actual* and *target* preview point at a distance (d_p) ahead, also called as the *preview distance*. The distance d_p is obtained by multiplying the velocity V_x along the X-axis with the parameter ‘preview time’ (t_p). Furthermore, the actual preview point is defined as (x_p, y_p) , and is obtained by extrapolating the current vehicle X- and Y-position (x_{car}, y_{car}) along with the current yaw angle (ψ_{car}) at the distance d_p . The target preview point(x_t and y_t) is positioned at the predefined target path at the same X-position ($x_t = x_p$) as the actual preview point. Then, depending upon the controller gain K (k_p, k_d as we use a PD controller!), the driver tries to minimize the error e_p , resulting in a feedback loop that essentially responds to heading angle ψ and lateral position y . Mathematically, represented:

$$\text{for the Feedback vector: } \quad y(t) = [y(t) \ \psi(t)]^T \quad (3-41)$$

$$\text{actual preview point: } y_p(t + t_p) = y(t) + t_p \cdot V \cdot \psi(t) \quad (3-42)$$

$$y_p(t + t_p) = y(t) + d_p \cdot \psi(t) \quad (3-43)$$

$$\text{preview error: } \quad e_p = y_t - y_p \quad (3-44)$$

Neuromuscular Driver Model

The goal of this subsection is to describe in brief the Neuromuscular Driver Model (NMS) [65], [67] and the Force-Feedback driver model (FFDM) [66] based on the NMS model, that were developed at the TU Delft. The motivation behind the use of such a model is the faster response as spinal reflexes contribute to control action, better driver acceptance and regular driver-system communication. The following argument provided by the authors explains the benefits of such a model :

“Designing a support system with the design approach of continuous haptic feedback may partly or entirely resolve the discussed issues with BWS (binary warning systems) and automation. The driver can remain in the loop, but also be supported in the assessment phase as well as the control phase”.

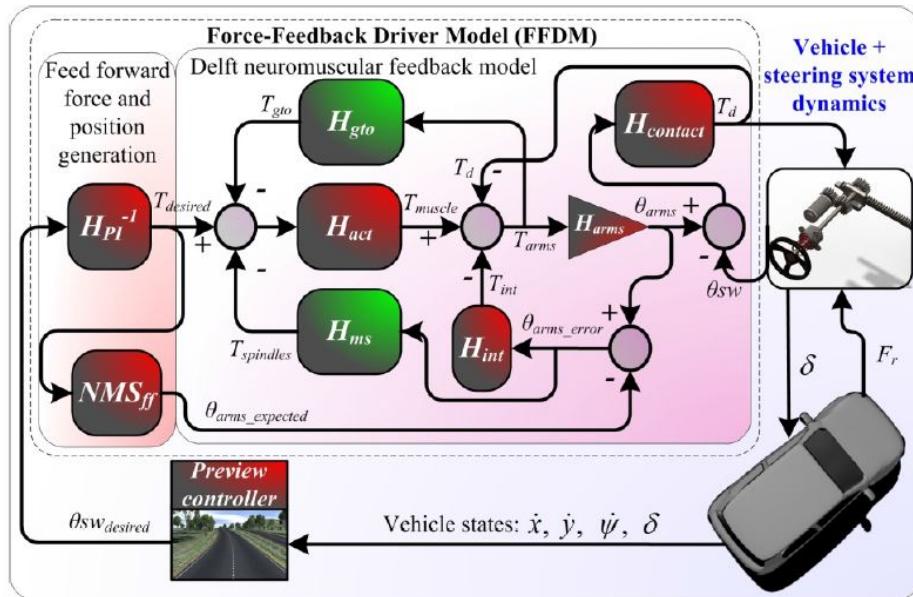


Figure 3-7: The Force Feedback Driver Model, depicting different functional blocks, source: [66]

The Figure 3-4, illustrates the Force Feedback Driver Model (FFDM) that takes in the desired steering wheel angle $\theta_{SWdesired}$ (output of the preview controller) as an input and yields the driver steering torque T_d , that acts on the steering dynamics associated with the vehicle. The vehicle states that are fed-back are longitudinal and lateral velocities, the vehicle yaw-rate and the steering angle ($\dot{x}, \dot{y}, \dot{\psi}, \delta$). Building on the work of [68], the Neuromusculoskeletal (NMS) model also adds an *internal model* H_{PI}^{-1} of the driver, which in real life scenarios comes into play when humans apply a desired torque to achieve a $\theta_{SWdesired}$. [66] provides detailed motion control mechanisms, NMS feedback parameters and analytical expressions of the blocks involved.

The Simulink model of FFDM developed by the Delft BioMechanics group¹, was re-structured to develop the human driver simulink model that we use in this project. This human driver simulink model consists of :

1. Visual Stimuli transformation block

Here global lane position and heading, and global vehicle states are input. An index is computed to spot the vehicle position along the two lanes and the road to follow (which computes the future 50 points) is calculated.

2. Perception block

Visual controller: This module predicts the future trajectory based on a 50-pts road category. It acts upon vehicle parameters (V_y , $\dot{\psi}$, δ) and global path parameters to generate future vehicle trajectory. Here, it is believed that the human operator possesses certain knowledge about his environment and system under consideration, in order to be able to change this environment according to his plans [57]. The brain decides arm displacement and velocity to perform a ‘set’ steering task (travel upon the optimal path determined in the above section). These optimal outputs are used in the intrinsic feedback under the NMS dynamics block.

3. NMS (Neuromusculoskeletal) dynamics (Action block)

This section concerns with the control of muscle force and limb movements to exert contact torque (in coordination with dynamics involving skin contact) for steering wheel actuation. One of the classic examples is McRuer’s Lumped model based on aviation studies [58]. It considers the NMS as a lumped 1st/2nd order system. However, when higher-order dynamics are encountered, mere ‘lumping’ the parameters doesn’t accurately represent the adaptive nature of NMS and the need for more elaborate models arise. Thus, with regard to aforementioned requirements on the human controller, a related but different neuromuscular model than above is proposed. The Golgi Tendon Organs (GTO) are responsible for muscular force generation, whereas the muscle spindles directly impact the position and velocity.

3. Combined steering wheel and contact dynamics block:

The contact dynamics represent the interplay of forces between hand (acting on response to stimuli) and the steering wheel. The position and angular velocity of hand and steering wheel after subsequent multiplication of human gain (hum_on) produce the contact torque. This contact torque results in displacement of steering wheel, which forms an input of the forthcoming vehicle dynamics block. Mathematically, the steering wheel dynamics can be expressed as:

$$\ddot{\theta} = \frac{-T_b - B_w\dot{\theta} - k_w\theta}{Jw} \quad (3-45)$$

where,

θ = Angular position of hand.

T_b = Total contact torque (N/m)

B_w, K_w = Gains of the steering wheel system

Jw = Moment of Inertia of the steering wheel.

¹For detailed description refer to: David ABBINK, WB2306 *The Human Controller*, ON: Delft University of Technology, December 2013. Lecture Notes.

- **Vehicle dynamics block**

The vehicle model that was used in conjunction with the human-driver model is primarily a 4DOF, ‘two-track’ model. It describes the longitudinal, lateral, yaw and roll dynamics of the vehicle. Further discussions on motivation, design and mathematical treatment of this model have been presented earlier in the section 2-5-1.

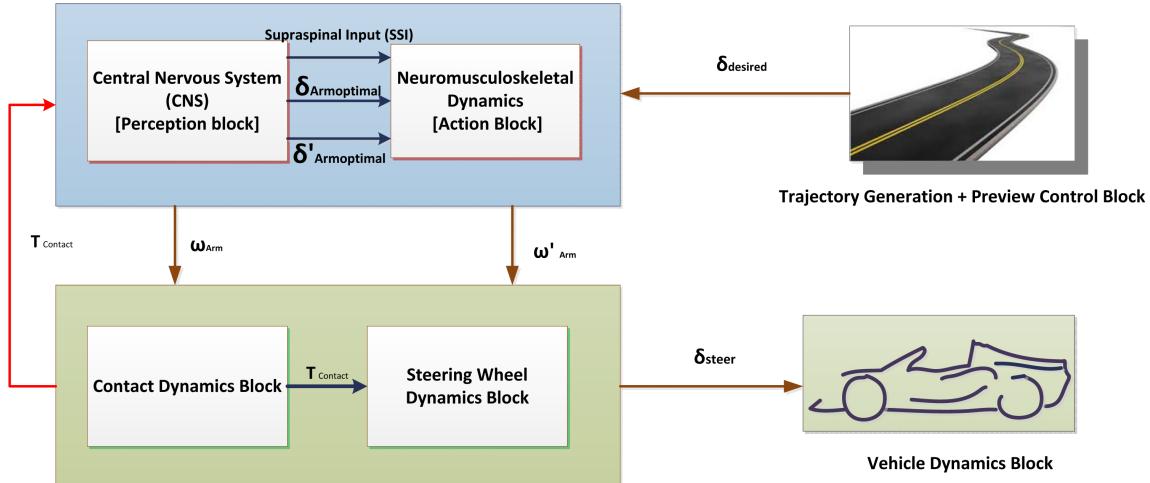


Figure 3-8: The Human Driver Model used in this thesis

Validation of Human Driver Model

The Figure 3-9 represents the trajectory following behavior of ‘relaxed’ driver ($k_p = 1$), whereas the Figure 3-10 illustrates driver torque T_d responses and lateral accelerations responses a_y for different driver gains $k_p = 1, k_p = 3$. As can be observed, application of higher driver gains result in more oscillatory torque responses. This can be attributed to a ‘stiff’ driving behavior at higher gains.

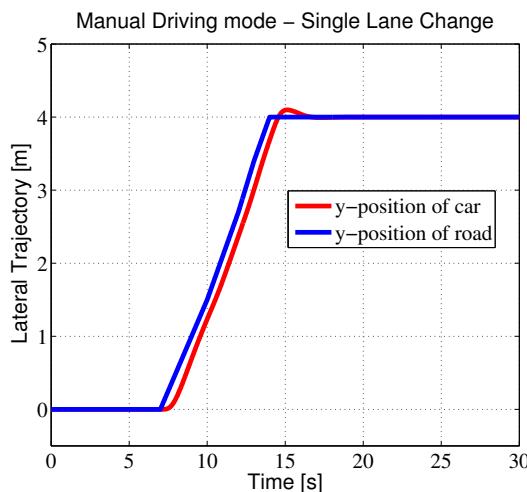


Figure 3-9: Trajectory following behaviour for driver gain $k_p = 1$

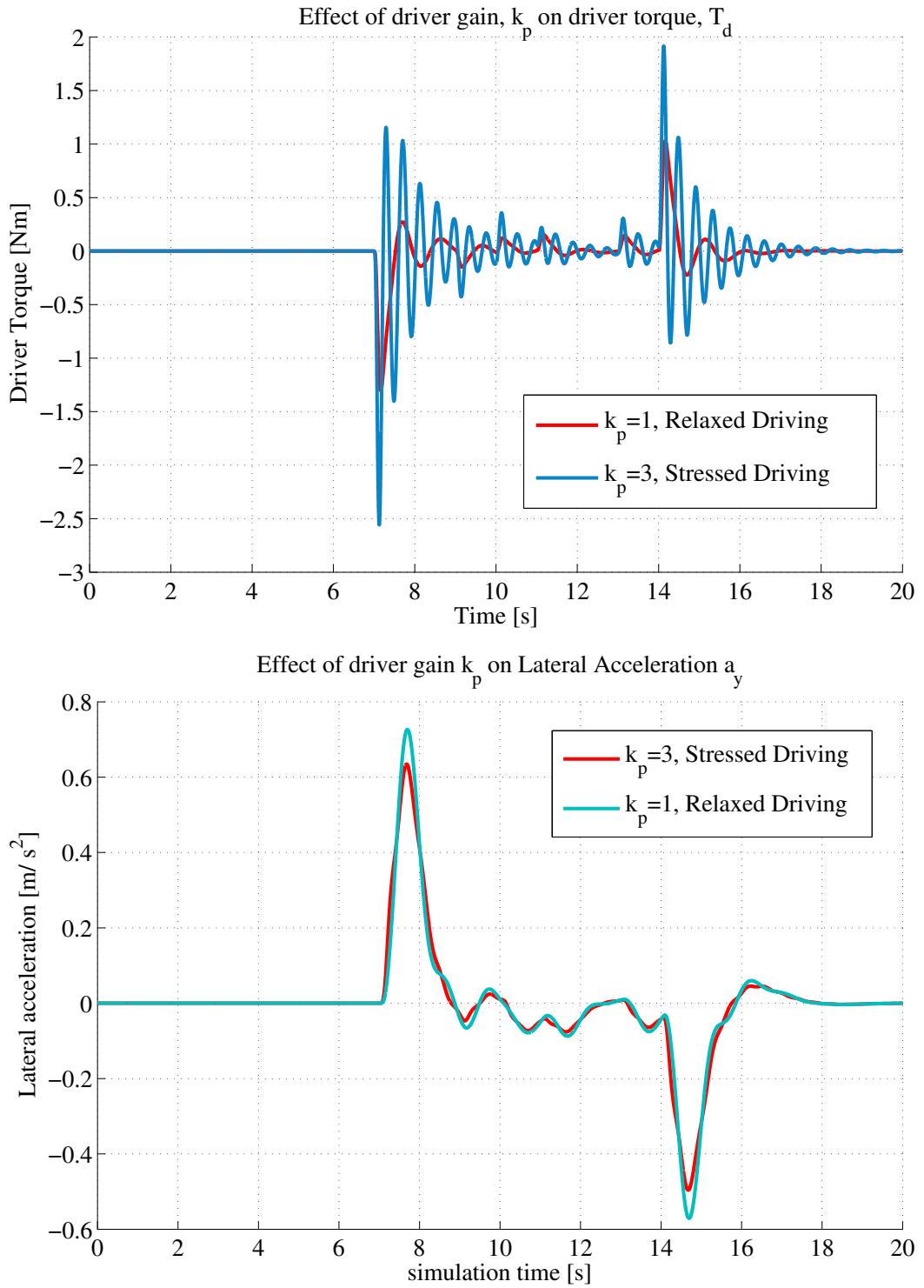


Figure 3-10: Validating the Human Driver Model for different driver gains $k_p = 1, k_p = 3$: (a.) Output Driver torques, (b.) Observed Lateral Accelerations

Single Mode closed-loop tests: Closed-loop testing refers to the subjecting the process/system under test, with inputs that are based on outputs from previous experiments. It provides important quantitative information about the combined human- vehicle dynamics. To perform closed-loop testing, the human driver navigates a single lane change of width 3 m at different velocities. The measured values of yaw rate $\dot{\psi}$ and steering wheel rate $\dot{\delta}$ during such maneuvers serve as nominal values for applying safety constraints during parametric verification using BREACH matlab toolbox.

The values in Table [2-1] describe the closed-loop tests performed under variable speeds (70 km/h-100km/h) and two different conditions based on lateral accelerations observed during simulation.

The first one describes a *slow* lane change maneuver with lateral accelerations $< 2 \text{ m/s}^2$. The second conditions describe a *fast* lane change maneuver with lateral accelerations $\geq 4 \text{ m/s}^2$

Longitudinal Velocity (m/s)	Lateral accelerations (m/s ²)	Measured Yaw rate ($\dot{\psi}$)	Max. abs. Measured Steering Wheel rate ($\dot{\delta}$)
SLOW			
80	0.67	0.018	0.375
90	0.71	0.033	0.375
100	0.78	0.0345	0.375
110	0.83	0.035	0.375
FAST			
80	1.12	0.058	0.75
90	1.25	0.059	0.741
100	1.35	0.061	0.734
110	1.5	0.061	0.72

Table 3-1: Manual Driving mode, Closed-loop test I: Determining safe value ranges for $\dot{\psi}$ and $\dot{\delta}$

The Table 3-1 describes the values of yaw rate $\dot{\psi}$ and steering wheel rate $\dot{\delta}$ obtained for manual driving mode. The maximum values of $\dot{\delta} = 0.75$ and $\dot{\psi} = 0.061$ have been selected. Under no circumstances should the switching between modes cause a behavior that allows the yaw rate and steering wheel rate to exceed these values.

Chapter 4

Hybrid Control Design and Verification

4-1 Modeling Semantics : Hybrid Automata

Hybrid Automata provide a general modeling formalism to model dynamical systems consisting of both discrete and analog components. An initial discussion of the switched mode stability problem and an overview of Hybrid systems modeling (as described earlier in section 2-1-2) leads us to casting the problem of switching between human driver and automated vehicle into the formalism of hybrid automata. The Figure 4-1 illustrates the Hybrid Automata that results from such an approach.

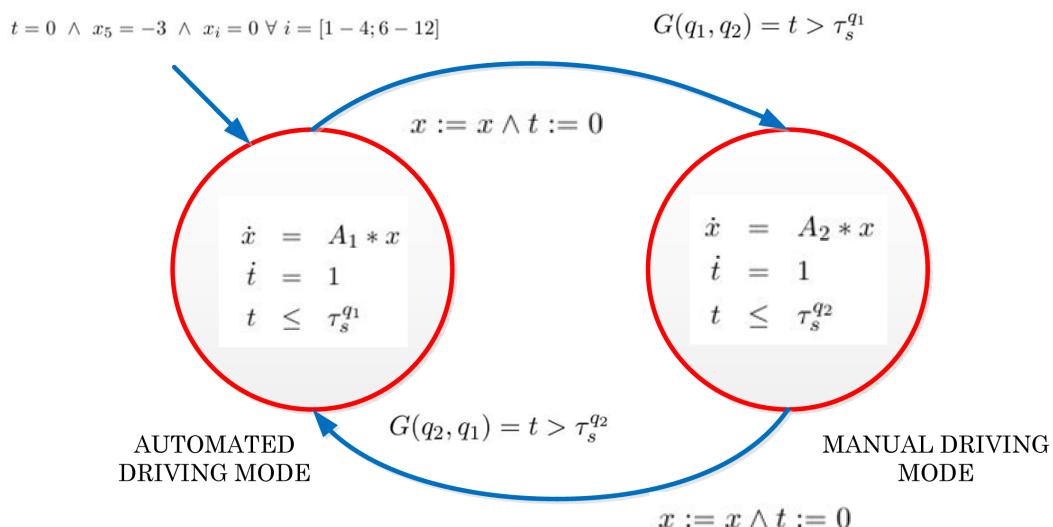


Figure 4-1: Graphical representation of hybrid automata describing two states (q_1) and (q_2), with their invariants, reset maps, guards and the initial state

Definition 4.1 (Mode Switching automaton, \mathcal{HA})

- $Q = \{q_1, q_2\}$, (Human Driving mode, Automated Driving mode), are the discrete States.
- $X = [V_y, r, \dot{\delta}_{st}, \ddot{\delta}_{st}, y, \psi, \dot{u}_{ref}, u_{ref}, z, x_1, x_2, m_1, t]^T \forall x \in \mathbb{R}$ and $t \in \mathbb{R}^+$
- $I = (0, 0, 0, 0, -3, 0, 0, -3, 0, 0, 0, 0)$, are the Initial conditions.
- $f(q_1, x) = A_{1[12 \times 12]}$ and $f(q_2, x) = A_{2[12 \times 12]}$, are the flow conditions (Eqns. 3-38, 3-39)
- $Inv = \{q_1, \{t \leq \tau_s^{q_1}, t \in \mathbb{R}^+\}\}, \{q_2, \{t \leq \tau_s^{q_2}, t \in \mathbb{R}^+\}\}$ are the set of invariant conditions, where $\tau_s^{q_1}$ and $\tau_s^{q_2}$ are switching times for modes q_1 and q_2 respectively, the relation between which is defined in § 4-4.
- $G(q_1, q_2) = \{t \in \mathbb{R}^+ > \tau_s^{q_1}\}$ and $G(q_2, q_1) = \{t \in \mathbb{R}^+ > \tau_s^{q_2}\}$ where, the guard conditions $G(q_1, q_2)$ denotes the switch from $q_1 \rightarrow q_2$ and $G(q_2, q_1)$ denotes the switch from $q_2 \rightarrow q_1$.
- $R(q_1, q_2, X) = R(q_2, q_1, X) = \{x\}$; which denotes Identity Reset for all states in X except, $X_{13} = t := 0$ which is state of the timer and hence is reset to zero.

With the definition of hybrid automata explained, the challenge now lies in proving the stability of this mode switching automata \mathcal{HA} , which forms the topic of discussion in the forthcoming sections. But before progressing to stability analysis, it is pertinent to understand the notion of time ‘evolution’ of a hybrid system.

Definition 4.2 (Hybrid Time Sets, T_i)

A *hybrid time set* or time trajectory of hybrid automata [18], represents a finite(or possibly infinite) sequence of intervals $\tau = \{T_j\}_0^M$, such that $\forall j < M$, $T_j = [\tau_j, \tau'_j]$. Also, if $M < \infty$ then either $T_M = [\tau_M, \tau'_M]$ or $T_M = [\tau_M, \tau'_M)$ and $\tau_j \leq \tau'_j = \tau_{j+1}$ for all j .

Definition 4.3 (Execution)

An execution [18] of a hybrid automaton H is a hybrid trajectory, (τ, q, x) , which satisfies the following conditions

- Initial conditions are contained in set I , $(q_o(0), x_o(0)) \in I$.
- Discrete evolution is defined by: $(q_i(\tau'_1), q_{i+1}(\tau_{i+1})) \in E$, $x_i(\tau'_i) \in G(q_i(\tau'_i))$ and $x_{i+1}(\tau_{i+1}) \in R(q_i(\tau'_1), q_{i+1}(\tau_{i+1}), x_i(\tau'_i)) \forall i \in \mathbb{Z}^+$.
- Continuous evolution is defined as:
 - $q_i(\cdot) : I_i \rightarrow Q$ is constant over $t \in I_i$, i.e. $q_i(t) = q_i(\tau_i) \forall t \in I_i$
 - $x_i(\cdot) : I_i \rightarrow X$ is the solution to the differential equation $\frac{dx_i}{dt} = f(q_i(t), x_i(t))$

4-1-1 Brief note on Existence and Uniqueness

It is fundamental to understand before investigating the steering interactions, if there are any executions (trajectories) that satisfy the equations of the system; Is the hybrid system \mathcal{HA} , *well-posed* or not? In general this is a non-trivial question, owing to the fact that a mathematical problem needs to be solved to find the system evolution. Thus, we try to explain briefly the concepts required to show that, for the automaton \mathcal{HA} infinite executions *exist and they are unique*, based on the following definitions from the work of Lygeros and Sastry [18]:

First, let us define REACH, as the reachable set of the hybrid automata. A set $(q_r, x_r) \in Q \times X$ is called a reachable set if there exists a finite execution (τ, q, x) that ends in (q_r, x_r) , i.e. for a Hybrid Time Set defined by $\tau = \{[\tau_i, \tau'_i]\}_0^M$, $M < \infty$ and $(q_M(\tau'_M), x_M(\tau'_M)) = (q_r, x_r)$. Also, the set SWITCH is the set of states for which continuous evolution along the differential equation forces the system to exit the domain instantaneously. The characterization of SWITCH is quite involved and lies outside the scope of this thesis.

A Hybrid Automaton is *non-blocking* if, $\forall (q, x) \in \text{REACH} \cap \text{SWITCH}, \exists q^+ \in Q$, such that $(q, q^+) \in E$ and $x \in G(q, q^+)$, where, q^+ is the successive discrete state of q . These conditions basically translate to : If for all states belonging to the *reachable set*, the impossibility of continuous evolution would lead to a discrete transition to the next discrete state, then the automata is called as non-blocking.

Also, a Hybrid Automaton is *deterministic* iff, $\forall (q, q^+) \in \text{REACH} :$

- (a.) if $x \in G(q, q^+)$ for some transition $(q, q^+) \in E$, then $(q, x) \in \text{SWITCH}$,
- (b.) if $(q, q^+) \in E$ and $(q, q^{++}) \in E$, then for, $q^+ \neq q^{++}$, $x \notin G(q, q^+) \cap G(q, q^{++})$
- (c.) if $(q, q^+) \in E$ and $x \in G(q, q^+)$, reset map contains unique element, i.e. $R(q, q^+, x) = x_k$, where q^{++} is another discrete state to which transitions can occur from previous state q .

More intuitively this means, a hybrid automata is deterministic if the impossibility of continuous evolution for a particular state guarantees the discrete transition to next state, provided the discrete transitions have unique destinations (multiple discrete transitions are not allowed). Then to justify that the mode switching automata \mathcal{HA} is indeed well-posed we would like to assume the following proposition:

Proposition 1

The Mode Switching Automaton, \mathcal{HA} is non blocking and deterministic.

This proposition guarantees that for the hybrid system \mathcal{HA} infinite executions exist and they are unique. Also, while defining the mode switching automata care has been taken not to allow blocking and non-determinism. The proof of these propositions is quite complex and require rigorous mathematical derivations for obtaining conditions for local existence and uniqueness of executions of hybrid automata. Such investigations lies outside the scope of the topics considered in this thesis and hence, will not be discussed. For a review of these concepts the interested reader is referred to [19].

4-2 Quantifying the interactions: Defining parameters

An essential prerogative of a structured analysis of the steering interaction, is the defining of the underlying ‘parameters’. This section defines the main parameters involved in the studying the steering interaction between human driver and automated vehicle. Knowledge of these parameters helps in identification of optimal switching conditions, that allow a ‘smooth’ transition of control between the different driving modes (human and automated). Furthermore, potentially dangerous/risky situations can also be accurately accounted for, if the domain in which these parameters lie is provided. The experimental scenario as discussed in section 2-5 is a single lane change.

The human driver is best represented by a behaviorally-complex, resilient, and optimal controller. Driver-focused studies[53] have suggested that the human operator uses different physical and mental processes in response to visual, motion and acoustic clues and regularly adapts his/her behavior to suit the vehicle dynamics and task variables. The following classification presents the four parameters/metrics that have been used in the present research to characterize human driver behavior. These are important as they form the basis for the forthcoming analysis and discussion phases.

1. Driver Competence:

The parameters that describe *driver competence* are the Gain bandwidth (k_p), the look-ahead distance of human driver (H_{th}) and the preview distance (in automated mode) of the steering controller (A_{th}). A comparative study of driving abilities between experienced and inexperienced drivers show that the former have probably smaller gains at higher speeds and smaller preview times, as compared to the inexperienced driver, who suffer from large driver gains for similar conditions.

2. Situation Awareness:

The ability of a human driver to perceive the changes taking place in surrounding environment, as measured in a interval of space and time is defined as *situation awareness*. This research work defines a metric (**TTS**) (or Time to Switch) to quantify driver’s reactive capabilities. Awareness refers to the impromptu cognitive reaction of humans in response to a change in stimuli. So, an aware driver has a smaller TTS than a distracted one. Thus, **TTS** provides important quantitative information on the nimbleness of driver.

Defining Nominal Values of parametric intervals

The Figure 4-2 and Figure 4-3 plot the real \mathbb{R} and imaginary \mathbb{I} part of the poles of both the closed loop human-vehicle and automation-vehicle systems. The plots represent 6 poles for each mode that lie close to origin, that were chosen to analyze the closed loop system response for varying H_{th} and A_{th} distances. As can be observed the nominal values for intervals that have been selected are: $A_{th} = [45 \ 55]\text{m}$, $H_{th} = [13 \ 18]\text{m}$ and $k_p = [0.98 \ 1.02]$.

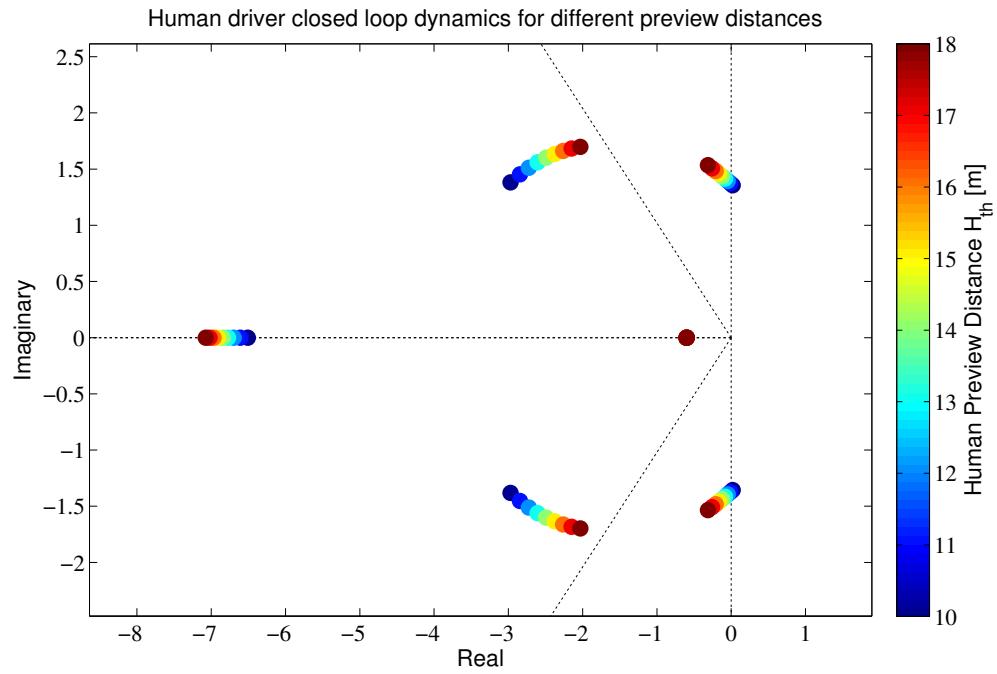


Figure 4-2: Selecting the nominal values of parametric intervals for human preview distance H_{th} .

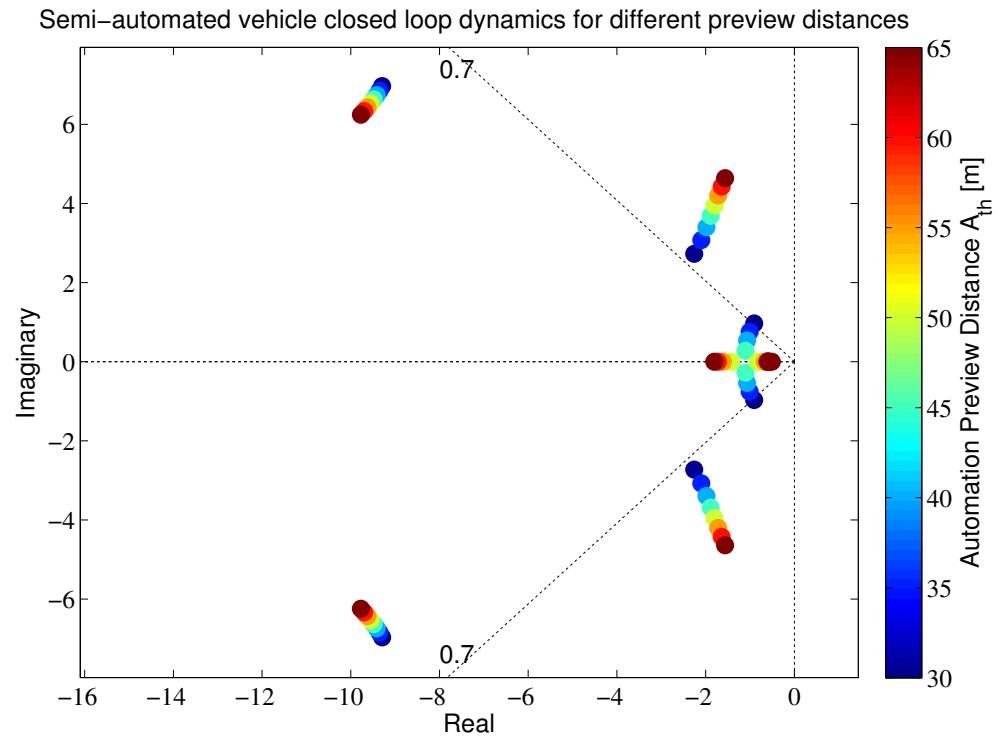


Figure 4-3: Selecting the nominal values of parametric intervals for automated preview distance A_{th} .

4-3 Stability in the sense of Lyapunov

In this chapter, so far we have discussed the modeling of hybrid automata and quantified the steering interactions using four different parameters. This section explains the Lyapunov function and its extension to concepts of Common Quadratic Lyapunov Function (CQLF) and Multiple Lyapunov Functions (MLF). Lyapunov stability criterion is vital mathematical to analyze the stability of nonlinear systems and forms the basis of all the stability discussions that follow in this thesis. The concept of Lyapunov direct method for stability analysis has been presented as follows:

Consider a autonomous, time-invariant nonlinear system of the form :

$$\dot{x}(t) = f(x(t)) \quad (4-1)$$

where, $x(t) \in \mathbb{R}^n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Let us consider an equilibrium point $x_e = 0$, of the system Eq. (4-3) such that $\dot{x} = f(x_e) = 0$, then for the Lyapunov stability, the following relations hold:

Stability theorem For the equilibrium point x_e of the system described by Eq. (4-3). Let $D \subset \mathbb{R}^n$ be a set containing x_e . If $V : D \rightarrow \mathbb{R}$ be a continuously differentiable function such that

- $V(0) = 0$
- $V(x) > 0 , V \in D \setminus \{0\}$
- $\dot{V}(x) \leq 0 , V \in D \setminus \{0\}$

then, $x_e = 0$ is stable in the sense of Lyapunov.

Asymptotic stability theorem Under the stability conditions described previously, if V satisfies the following conditions:

- $V(0) = 0$
- $V(x) > 0 , V \in D \setminus \{0\}$
- $\dot{V}(x) < 0 , V \in D \setminus \{0\}$

then, $x_e = 0$ is asymptotically stable in the sense of Lyapunov.

Common Quadratic Lyapunov Functions (CQLF)

The Lyapunov stability theory plays a key role in deriving stability results for switched systems. It is a widely known fact in the systems and control community that, the necessary and sufficient conditions for asymptotic stability of LTI systems is the *existence* of Lyapunov functions of the form discussed above. Consider then, switched systems of the form -

$$\Sigma_S : \dot{x}(t) = A_\sigma(t)x(t) \quad (4-2)$$

where, $A \in \mathcal{A} = \{A_1, A_2, \dots, A_n\}$ is a set of state matrices defining a family of LTI systems, that switch based on value of switching signal $\sigma(t)$, where, $\sigma(\cdot) \in \mathcal{S}$ a set of admissible piecewise switching signals.

Then for a collection of LTI systems Eq. (4-3) that switch based on certain conditions, proving asymptotic stability based on Lyapunov stability calls for the notion of Common Quadratic Lyapunov Functions (CQLF).

Consider a quadratic function of the form :

$$V(x) = x^T P x, \text{ with } P = P^T \quad (4-3)$$

now if, V satisfies the asymptotic stability conditions mentioned previously then it is indeed a Lyapunov function. However, if the quadratic function V satisfies simultaneously the following conditions for a set of systems Σ_S :

$$A_i^T P + P A_i = -Q_i \quad (4-4)$$

$$\text{where, } Q_i = Q_i^T > 0 \quad (4-5)$$

where, $A_i \in \mathcal{A} \forall i \in \{1, 2, \dots, n\}$, then such quadratic function V is called as a CQLF. However, an important point of observation is that existence of such a common quadratic lyapunov function is a *sufficient* condition for asymptotic stability for a set of systems Σ_S . For further reading on CQLF [70] provides a good overview.

Multiple Lyapunov Functions (MLF)

The existence of CQLF is a convenient technique to analyze the stability of switched systems, however quite often when dealing with physical systems in real-life, it is plausible that such a function does not exist. It then remains to utilize another advanced concept based on Lyapunov theory, called the *Multiple Lyapunov functions*. Essentially, the idea is to utilize a number a Lyapunov functions for each mode/subsystem and switching to a certain mode q_1 is only allowed, if the associated Lyapunov Function V_{q_1} takes up a value that is *lesser* than the value it had when it was in the mode q_1 last time. The idea of MLFs is attributed to [71]. Another excellent approach for nonlinear systems can be found at [73].

4-4 Switching based on Average Dwell Time

This chapter so far has focussed on modeling of the research problem using hybrid systems framework. Also, the use of Lyapunov functions as a vital analysis tool have been discussed. However, analysis of hybrid systems is in general a non-trivial task. This is described in [74] where the authors prove analytically that switching between stable systems can even lead instability, or switching between unstable subsystems can also lead to an overall stable system, making the task of stability analysis really complex. This MSc. thesis utilizes the concepts of *time domain* restrictions, specifically the notion of *average-dwell time* switching to develop the first *metric/parameter*, called the **Time To Switch** (TTS), that helps in realizing a smooth switching between automated vehicle and human driver.

Based on benchmark studies on the issues related to stability of switched systems [74], it can be observed that rapid switching between individual subsystems causes instability due to failure in containing the increase in system energy. Thus, selecting stability analysis methods based on constrained switching methodology seems an obvious solution. The following idea was first propounded in seminal work [72] on switched systems. It forms the basis of switching based on constraining the rate at which switching takes place:

If all of the matrices in the switching set $\{A_1, \dots, A_n\}$ are Hurwitz, then it is possible to ensure the stability of the associated switched system by switching sufficiently slowly between the asymptotically stable constituent LTI systems.

Definition 4.4 - Concept of Dwell Time [72] Given a positive constant τ_D , let $S[\tau_D]$ denote the set of all switching signals with interval between consecutive discontinuities being no smaller than τ_D . The constant τ_D is called the (fixed) “dwell time”. It was shown in [77] that one can pick τ_D sufficiently large so that the system $\dot{x} = A_\sigma x$ is exponentially stable for every $\sigma \in S(\tau_D)$. In fact, there exist positive constants c, λ such that $\|A_\sigma(t, \tau)\| \leq ce^{-\lambda(t-\tau)}$ for all, $t \geq \tau \geq 0, \sigma \in S([\tau_D])$

Based on the fundamental concept of dwell time, the authors [72] extended the concept to contain the signals $\sigma \in S_{ave}[\tau_D, N_0]$ that have consecutive discontinuities (defined by N_0) lesser than dwell time (τ_D) but for these signals on an average the switching time between the discontinuities is no less than $\hat{\tau}_D$ (average dwell time). Mathematically stated, for a switching signal $\sigma \in S_{ave}[\tau_D, N_0]$ and a time interval (t, τ) , let us denote the discontinuities that system undergoes in this time interval by $N_\sigma(t, \tau)$. Then for $\hat{\tau}_D > 0$, and a constant $N_0 > 0$ called as chatter bound, the following relation holds for system satisfying average dwell time: $N_\sigma(t, \tau) \leq N_0 + \frac{t_2 - t_1}{\hat{\tau}_D}$.

Theorem 4.1 - Average Dwell Time [72] Consider that for a family of switched systems defined by Eq. (4-3) the matrices $\{A_1, \dots, A_n\}$ are hurwitz and there exists a constant $\lambda_0 > 0$, such that $A_i + \lambda_0 I$ is Hurwitz $\forall i = [1, \dots, n]$. Then , for any $\lambda \in [0, \lambda_0)$ \exists a finite constant $\hat{\tau}_D^*$ such that the system (4-2) is uniformly exponentially stable over $S_{ave}[\tau_D, N_0]$ with stability margin λ , for any average dwell time $\hat{\tau}_D \geq \hat{\tau}_D^*$ and any chatter bound $N_0 > 0$.

Checking the Hurwitz stability condition

An important problem in Switching control design is to find conditions that guarantee that the switched systems is stable for any switching signal. [78] state that this situation is of great importance when a given plant is being controlled by switching among a family of stabilizing controllers, each of which is designed for a specific task. Therefore, as a first step in analyzing stability, the Hybrid Automaton \mathcal{HA} defined in the section 4-1 can be recast in the switched systems formalism as:

$$f(x) = \begin{cases} A_1x, & \text{if } t \leq T_1, \forall t \in \mathbb{R}^+ \\ A_2x, & \text{if } t \leq T_2, \forall t \in \mathbb{R}^+ \end{cases} \quad (4-6)$$

The above system of equations fall under the category of those mentioned in 4-2. Now, before applying the concept of average dwell time to this switched system framework it is pertinent to check whether the matrices are indeed *Hurwitz* i.e. matrices for which all its eigenvalues have strictly negative real part ($Re(\lambda) < 0$). A MATLAB script evaluates the state matrices $A_1_{[12 \times 12]}$ and $A_2_{[12 \times 12]}$, which then turn out to be *Non-Hurwitz*. It should be remarked that, although individual state matrices of mode q_1 and q_2 in their non-equidimensional form as described by equations 3-21 and 3-38 are *Hurwitz*, the process of reformulating them into a equidimensional system (refer Chapter 3, § 3-3) leads to loss of the rank of each of these matrices, hence failing the Hurwitz stability condition. However, to utilize the Average Dwell Time approach it is important to have matrices that are Hurwitz. So, we add really ‘fast’ dynamics to the states of Human controller (x_1, x_2, m_1) for the system of mode q_1 and the state of automated controller (z) for the system of mode q_2 . An argument that follows such a step is that, adding ‘fast’ dynamics just forces the controller states in either mode to go quickly to zero, thereby not disturbing the system stability but guaranteeing that state matrices A_1, A_2 are now Hurwitz.

Checking the existence of CQLF (common quadratic lyapunov functions)

It is pertinent at this stage to verify whether a Common quadratic Lyapunov function exists for both systems or not. This verification can be done by solving the *Dual Problem*[75]. Consider the systems used in this thesis, as defined by equation 4-6, if there exists matrices $R_i > 0 \forall i \in \mathcal{I} = [1, 2]$, that satisfy the following equations:

$$\sum_i A_i^T R_i + R_i A_i > 0 \quad (4-7)$$

Then the Lyapunov inequalities defined for exponentially stability $A_i^T P + PA_i < 0, i \in \mathcal{I}$ do not admit a solution to $P = P^T$. The YALMIP matlab toolbox (refer section) is used to solve the system of LMIs in equation 4-7. The LMI problem was found to *feasible* suggesting the fact that a common quadratic lyapunov *doesn't exist*.

In this thesis, the two state matrices A_1 (belonging to state q_1 , the automated mode) and A_2 (belonging to state q_2 , the human driving mode) both have non-strict eigenvalues $\text{Re}(\lambda \leq 0)$. So we define a pair of positive scalars λ_1, λ_2 such that the matrices $A_i + \lambda_i I$, where, $i \in \mathcal{I} = [1, 2]$, are Hurwitz stable. To ensure the exponential stability we utilize a Multiple Lyapunov Function approach. Then based on the work [79] we state the time-based switching theorem:

Theorem 4.2 : Time-based Switching theorem [72] For the system matrices A_1 and A_2 belonging to the family of switched systems defined by 4-2, select a positive constant λ_s , such that such that the matrices $A_i + \lambda_s I$, (where, $i \in \mathcal{I} = [1, 2]$) are asymptotically stable. Then for any selected $\lambda_p \in [0, \lambda_s]$ \exists a finite constant $\hat{\tau}_D^*$ such that the system defined by 4-6 is uniformly exponentially stable over $S_{\text{ave}}[\hat{\tau}_D, N_0]$ with stability margin λ_p , for any average dwell time $\hat{\tau}_D \geq \hat{\tau}_D^*$ and any chatter bound $N_0 > 0$.

Proof: Since, no CQLF exists, the concept of Multiple Lyapunov functions is used. Consider, two symmetric, positive definite matrices P_1, P_2 such that system 4-6 admits:

$$\begin{cases} (A_1 + \lambda_s I)^T P_1 + P_1 (A_1 + \lambda_s I) < 0 \\ (A_2 + \lambda_s I)^T P_2 + P_2 (A_2 + \lambda_s I) < 0 \end{cases} \quad (4-8)$$

Consider a Multiple quadratic lyapunov function of the form, $V_i = x^T P_i x$, where, P_i takes values based on the solutions of the equation (4-8). Then the POLF V_{PQ} satisfies the following properties:

- Each $V_i = x^T P_i x$ is continuous and its derivative along the solutions of the each of subsystems of 4-6 satisfies:

$$\frac{\partial V_P}{\partial x} A_i x \leq -2\lambda_s V_i \quad (4-9)$$

- There exists functions α_1, α_2 of class ¹ \mathcal{K}_∞ such that:

$$\alpha_1(\|x\|^2) \leq V_i(x) \leq \alpha_2(\|x\|^2), \forall x \in \mathbb{R}^n, \forall i \in \mathcal{I} \quad (4-10)$$

- there exists a constant scalar $\mu \geq 1$ such that

$$V_i(x) \leq \mu V_j(x) \quad \forall x \in \mathbb{R}^n, \forall i, j \in \mathcal{I} \quad (4-11)$$

Based on this theorem, a computation scheme can be setup to check the stability of the mode switching automata $\mathcal{H}\mathcal{A}$. The **Algorithm 1** described in the next page has been used to generate the average dwell time $\hat{\tau}_D^*$. The functions, `sdpvar`, `checkset`, `ineqs` are defined in the YALMIP Matlab toolbox used for solving LMIs.

¹class \mathcal{K}_∞ functions are

Algorithm used for generating $\hat{\tau}_D^*$ **Algorithm 1** Calculating the value of $\hat{\tau}_D^*$

```

1: Start
2:   Choose the value of(  $\lambda_s$ );
3:   Next;
4:   Start
5:     Construct the matrix( $A_1 + \lambda_s I$ );
6:
7:     Construct the matrix( $A_2 + \lambda_s I$ );
8:
9:   Start
10:    Choose sdpvar variables ( $P_1, P_2$ );           ▷ Start the LMI formulation
11:    Next;
12:    Construct : ineq1=(set( $P_1 > 0$ )+ set( $A_1^T * P_1 + P_1 * A_1 < 0$ ));
13:
14:    Construct : ineq2=(set( $P_2 > 0$ )+ set( $A_2^T * P_2 + P_2 * A_2 < 0$ ));
15:
16:    Verify sdpvar(ineq1);           ▷ Check the feasibility of LMI formulation
17:    Verify Checkset (ineq1);       ▷ Check the value of constraint residuals
18:    Verify sdpvar(ineq2);
19:    Verify Checkset (ineq2);
20:    Compute  $\lambda_c = [eig(P_1) \ eig(P_2)]$       ▷ matrix containing eigenvalues of  $P_1, P_2$ 
21:    Save  $\lambda_c$ 
22:   Start
23:     Compute  $\alpha_1 = (\min(\min(\lambda_c)))$           ▷ Compute Infimum
24:     Save  $\alpha_1$ 
25:     Compute  $\alpha_2 = (\max(\max(\lambda_c)))$           ▷ Compute Supremum
26:     Save  $\alpha_2$ 
27:   End
28: End
29:   Compute the expr.( $\mu = \frac{\alpha_2}{\alpha_1}$ );
30:   Save  $\mu$ 
31: End
32:   Compute the expr.( $\hat{\tau}_D^* = \frac{\log(\mu)}{2(\lambda_s - \lambda_p)}$ );      ▷ Compute the value for ADT,  $\hat{\tau}_D \geq \hat{\tau}_D^*$  [72]
33:   Save  $\hat{\tau}_D^*$ 
34: End

```

The resulting value from Algorithm 1 was $\hat{\tau}_D^* = 5.13$ s, then according to theorem 4.2 the average dwell time for the system $\mathcal{H}\mathcal{A}$ to remain stable under switching is given by $\hat{\tau}_D \geq 5.13$ s. Represented mathematically:

$$\hat{\tau}_{\mathbf{D}}^{\mathcal{H}\mathcal{A}} = \frac{\tau_s^{q_1} + \tau_s^{q_2}}{2} \geq \hat{\tau}_D^*$$

Simulating the switched systems using Stateflow

The Figure 4-4 explains the STATEFLOW implementation of the time-based switching methodology that was used to develop the Time to switch (TTS) metric for evaluating the switching between two different modes (human driving and automated driving). The state 1 (*Mode_Select*) defines the two modes and switching between them based on events *Transit_AtoH* and *Transit_HtoA*, whereas the state 2 (*Time_Switching_Logic*) defines the time-based switching logic based on which the events in the first state are called. Both these states have *parallel* (AND) decomposition, describing a simultaneous execution with time.

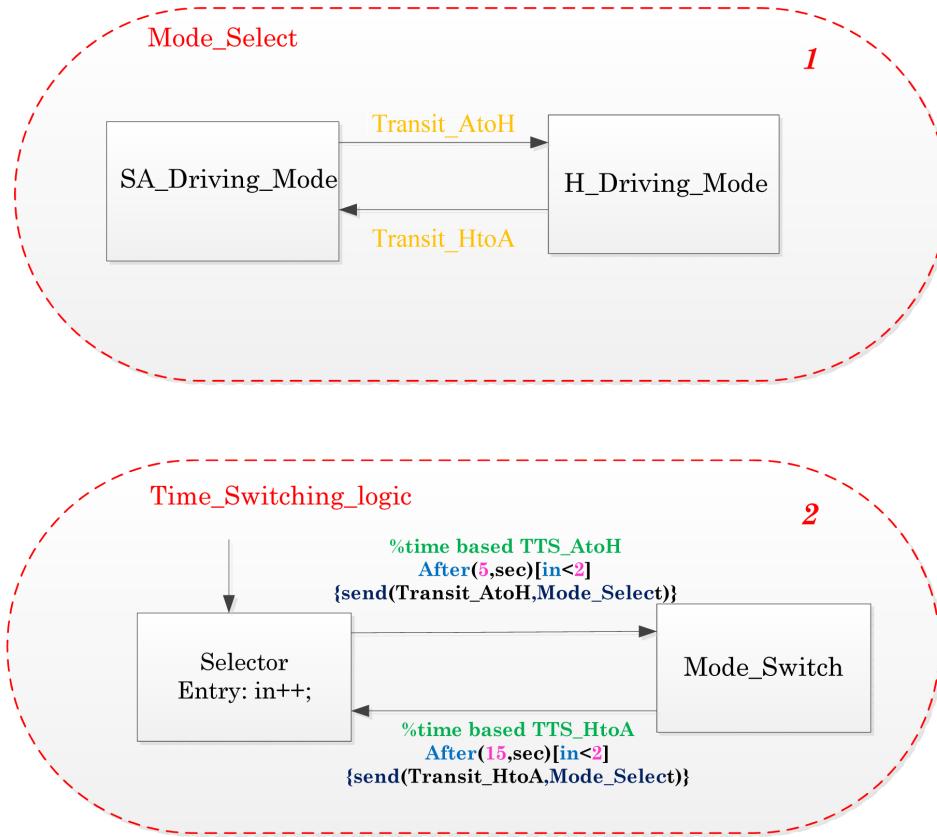


Figure 4-4: Graphical representation of Hybrid Automata describing two states (q_1) and (q_2), with their invariants, reset maps, invariant and the initial state

The state 2 in the Figure 4-4 illustrates the time-based switching implementation based on the value of $\hat{\tau}_D^*$ obtained from Algorithm 1. The switch from automated driving to manual driving occurs at the call of event $\{send(Transit_AtoH), Mode_Select\}$ after 5 secs. Then, referring to Figure 4-1 the TTS for mode q_1 becomes $\tau_s^{q_1} = 5$ s. Similarly, the switch from manual driving to automated driving occurs at the call of event $send(Transit_HtoA), Mode_Select$ after 15 secs, then the TTS for mode q_2 becomes $\tau_s^{q_2} = 15$ s. Thus, the system stays in each mode for an average dwell time of $\tau_{D}^{sys} = \frac{\tau_s^{q_1} + \tau_s^{q_2}}{2} = \frac{5+15}{2} = 10$ s, where the condition $\tau_{D}^{sys} > \hat{\tau}_D^*$ is respected and hence the switched-stable remains stable.

Remark §4.1 This section explained the time-based switching methodology that was used to develop the Time to switch (TTS) metric for evaluating the switching between two different modes (human driving and automated driving). This metric provides the much needed design freedom when transferring the control between the modes. Consider this, had we used merely the dwell time concept this would have put unnecessary constraint on switching. To elucidate, using the τ_D metric would impose that each of the modes *had* to dwell for a time duration of τ_D secs. But what if the driver panics or wants an early reclaiming of control ? He would have never got the control back from the automated vehicle untill the automation drove the vehicle till τ_D secs. Now, let us assume instead that the Time To Switch between the modes is $\hat{\tau}_D^*$. Since, this is essentially an *average* value, the driver could take back control from the automated vehicle anytime he so feels like, provided he still is able to maintain an average dwell time of greater than/equal to $\hat{\tau}_D^*$.

Remark §4.2 However, such an interesting approach could possibly lead to instability if the driver is ill-trained and switches too frequently. Thus, for guaranteeing ‘safe’ switching on the combined driver-vehicle system, we *bound* the TTS (time to switch) for mode q_1 (automated mode) to $\tau_s^{q_1} = 1.5$ s, which is based on the minimum time of reaction of human driver [76]. Then, the time to switch of mode q_2 (human driving) can be determined by

$$\tau_s^{q_2} \geq 2 \times (\hat{\tau}_D^*) - 1.5 \text{ s}$$

4-5 Reducing Conservativeness: Parametric Verification

An experimental approach where parameters attain fixed values, proves conservative for performance analysis on the account that it does not allow one to investigate the complete effects on performance of the system for an exhaustive range of parameters. The calculations in previous section were entirely based on the following parameter values: The look-ahead distance of human driver (\mathbf{H}_{th}) = 15m, the preview distance (in automated mode) (\mathbf{A}_{th}) = 50m, the Driver bandwidth ($\mathbf{k_p}$) = 1. Thus, for an insightful analysis on the effects of these parameters we proceed to the parametric verification of the Mode Switching Automata (Def. 1.2).

Parametric Verification using BREACH Matlab Toolbox

A ‘parameter synthesis’ problem consists of identifying a set of parametric valuations that guarantee a certain acceptable behavior of system under consideration. The BREACH Matlab toolbox explained in section § 2-3-2, will be used to perform the parametric verification of the remaining three parameters (\mathbf{H}_{th}), (\mathbf{A}_{th}),($\mathbf{k_p}$) that describe the interaction. The authors in [42] describe the following algorithm for obtaining a parameter synthesis.

Definition 4.5 (Parameter Synthesis Problem) A parameter synthesis problem is a 4-tuple $(\mathcal{H}, \mathcal{P}, \mathcal{B}, T)$ where \mathcal{H} is an hybrid system defined by Eqns. [2.6-2.8], \mathcal{P} a compact set of parameters, \mathcal{B} is a set of “Bad” sets and T a non-negative real number;

- *A solution of the parameter synthesis problem $(\mathcal{H}, \mathcal{P}, \mathcal{B}, T)$ is a partition of \mathcal{P} into three sets, \mathcal{P}_{saf} (safe sets), \mathcal{P}_{unc} (uncertain sets), \mathcal{P}_{bad} (bad sets) such that: (I.) for all $p \in \mathcal{P}_{bad}$, $\xi_p \in \mathcal{B}$ for some $0 \leq t \leq T$; and (II.) for all $p \in \mathcal{P}_{saf}$, $\xi_p \notin \mathcal{B}$ for all $0 \leq t \leq T$ and (III.) $\mathcal{P}_{unc} = \mathcal{P} - \mathcal{P}_{saf} \cup \mathcal{P}_{bad}$.*

Essentially what the ‘parameter synthesis problem’ defines is the partitioning the set of parameters $p \in \mathcal{P}$ into three sets *safe*, *uncertain* and *bad* based on sensitivity analysis to generate approximations to reachable sets very efficiently. As soon as a simulated trajectory crosses the bad set \mathcal{B} it returns ‘unsafe’ and the *falsifying* trajectory. While no such falsifying trajectory is found, it partitions the sampling space into ‘safe’ trajectories and ‘uncertain’ trajectories. The uncertain trajectories are those for which the neighborhood induced by the expansion function has a non-empty intersection with the bad set, indicating the possibility of a falsifying trajectory in this neighborhood. The algorithm then refines ² the uncertain trajectories, and when the local refining of uncertain trajectories becomes less than given parameter δ , the algorithm stops, it returns the uncertain set. Safe trajectories are returned when an uncertain set is empty. The Bad set \mathcal{B} are described by the help of Signal Temporal Logic (STL) formulas, which can be implemented in BREACH using the class QMITL_Formula.

Motivation for Safety Logic

The logic that describes the safety condition ‘ ϕ ’ for the experimental verification, is defined in terms of two important safety variables, the yaw rate ($\dot{\psi}$) and the steering angle rate ($\dot{\delta}$). The reason for selecting these parameters lies in the fact that: (1) Yaw-rate provides an apt description of the lateral state of a vehicle along with the slip-angle. (2) Steering Wheel Rate (SWR) is defined as the rate of change of steering wheel angle with time. In general, steering behavior is an important driver activity measure due to the fact that a large number of driving characteristics like the driver traits (e.g. driving experience), and driver states (e.g. distraction or fatigue) can be described in terms of experiments steering wheel angles, steering wheel velocity. Also if the yaw-rate and steering wheel rate can be maintained low, then in presence of low friction, the side-slip angle of the vehicle will be also be small.

Formulating the Bad Set, \mathcal{B}

We first formulate the safety logic conditions based on the yaw rate ($\dot{\psi}$), the steering angle rate ($\dot{\delta}$), and the lateral displacement ($y - y_{ref}$), which has been described in terms of an QMITL_formula as:

$$\boxed{\phi = (\text{alw } (\dot{\psi} < \alpha_1) \wedge \text{alw } (\dot{\delta} < \alpha_2)) \wedge \text{alw } (y - y_{ref} = \alpha_3)}$$

where, **alw** is the temporal operator *always* and ϕ is the safety condition imposed on the dynamics of the switched system. Keep in mind that these values are really extreme values and would not necessarily be overridden in acceptable ranges of A_{th} & H_{th} . However, it provides a starting point for the experimentation.

The, Bad set then refers to all those values that negate this safety condition. For instance, the values ($\dot{\psi} \geq \alpha_1$), ($\dot{\delta} \geq \alpha_2$) and ($y - y_{ref} > \alpha_3$) violate the requirements of driver safety and driving comfort and hence, are considered as ‘bad’ values. Logically represented, this means:

$$\boxed{\text{Bad Set, } \mathcal{B} \dashv \phi}$$

²for refining the parameter sets BREACH requires a bound on the dimensions of the refining grid denoted by constant δ . Refer [42] for further information on refining of sets.

Chapter 5

Results

The problem statement defined at the outset of this thesis was: *How does one ‘effect’ a smooth transfer of control authority from manual driving mode to automated mode and which parameters affect the safety of systems involved?*. This report has so far been able to explain what the problem was, define and discuss core ideas and mathematical tools and develop a switched systems model that was able to cast the problem in the hybrid control framework. That said, mere description of theories without experimental validation belies the notion of performing a sound research analysis. Thus, this chapter describes the experiments performed, their results obtained and alongside provides conclusions regarding the nature and outcome of experiment.

5-1 Experimentation with Average Dwell time

The first parameter that quantifies the transfer of control from one mode to another is the Time To Switch (TTS). A detailed explanation regarding how the concept of Average Dwell time ($\hat{\tau}_D \geq \hat{\tau}_D^*$) relates to this and how can we find a lower bound on this value was considered in Chapter 4, § 4-4. In this section we proceed to application of these theories to the problem at hand.

Two cases are investigated in this section that are believed to be comprehensive enough to explain the dynamical behavior of the switched system. Also, these cases provide a quantitative explanation for probable unsafe behaviors arising out of the application of theorem 1 § 4-4.

- **Case I** (When $\hat{\tau}_D < 5.13$ s): The first scenario we investigate is a lane change of width 3 m and the driver is supposed to be ill-trained and thus, switches *too frequently*. Such an experiment demonstrates the so called *worst-case* scenario. The TTS for each mode will be 1 second i.e. $\tau_s^{q_1} = 1$ s for mode q_1 and $\tau_s^{q_2} = 1$ s for mode q_2 . Although, it has to be pointed out that such a constraint seldom applies to normal highway driving scenarios, but serves as proof of concept for not respecting the Dwell time condition.

- **Case II** (When $\hat{\tau}_D > 5.13$ s): The second scenario is more practical in the sense that we expect to observe such a behavior when the system is eventually implemented for real-time testing purposes. Here, we first stay in the q_1 (automated driving mode) for a duration of $\tau_s^{q_1} = 5s$, then at the instant the vehicle approaches the *curved* of the lane change maneuver, the mode q_2 (manual driving) is activated for a duration of $\tau_s^{q_2} = 15s$ and finally for the last section, the control is transferred back to the automated vehicle which then steers the vehicle till the end of lane change i.e. the activation time $\tau_{sk}^{q_2} = 10s$. So, for a simulation time of $t_{sim} = 30$, the average dwell time can be calculated by $\hat{\tau}_D = \frac{\tau_s^{q_1} + \tau_s^{q_2} + \tau_{sk}^{q_1}}{3} = \frac{5+15+10}{3} = 10s$, which is greater than $\hat{\tau}_D^* = 5.13s$ and thus system remains stable.

Parameters	Values
Longitudinal velocity, V_x [m/s]	27.778
Automation preview distance, A_{th} [m]	55
Human preview distance, H_{th} [m]	15
Driver gain, k_p	1

Table 5-1: Parameter settings for Cases I and II

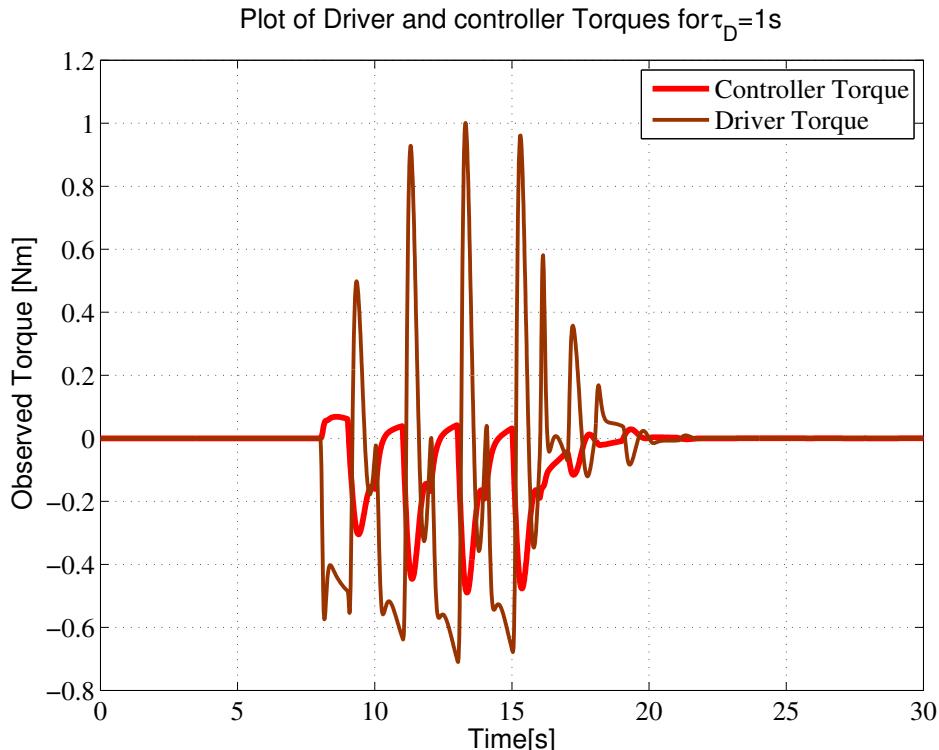


Figure 5-1: Driver and controller Torques for $\hat{\tau}_D^*=1s$

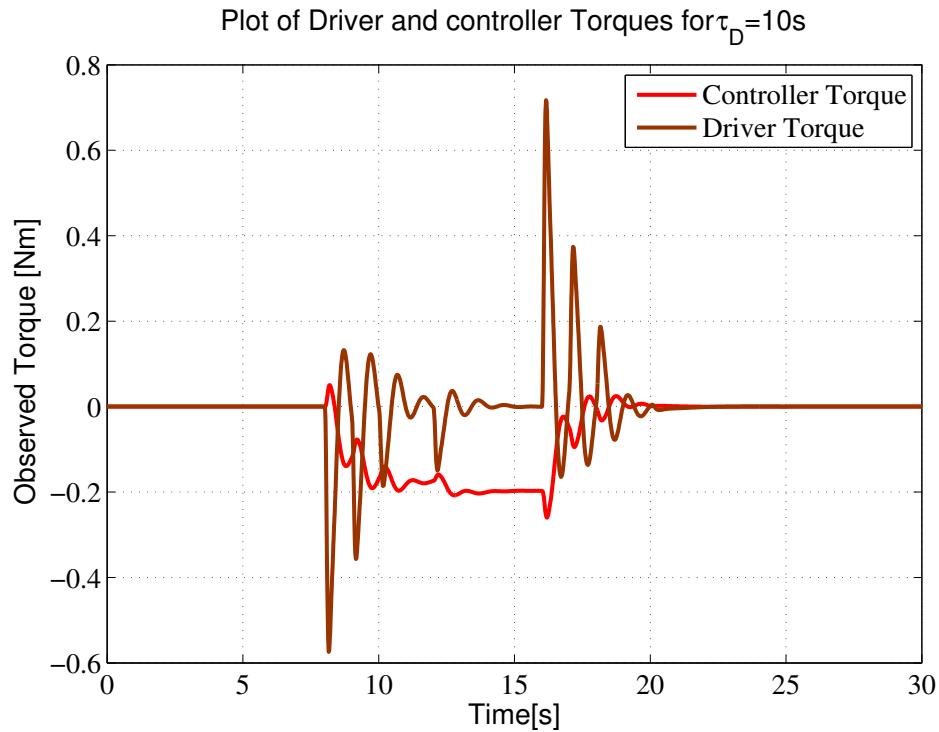


Figure 5-2: Driver and controller Torques for $\hat{\tau}_D^* = 10s$

Remark § 5.1 The Figure 5-1 illustrates the observed steering torque responses of both the human driver and automated vehicle (referred to as controller torque) for Case I. The Driver's aggressive steering behavior results from his/her inability to keep up with frequent switches that take place. Although, the controller torque also shows oscillatory behavior, its values are bounded to between $-0.6 \leq T_D \leq 0$ Nm. The oscillatory torque responses can be explained by the fact that the increase in system energy when a mode switch takes place, is not allowed to dissipate quickly due to inadequate 'dwell time' for each mode. This clearly depicts the loss of one of our important driver-related factors, referred to as *Driver Comfort* and also threatens *Driver Safety*, if the maneuver is performed at higher velocities.

Remark § 5.2 The Figure 5-2 illustrates the observed steering torque responses for Case II. The Average dwell time $\hat{\tau}_D = 10s$ is long enough to allow the driver to take control in and negotiate the lane change. The Driver torque values stay bounded between $-0.6 \leq T_D \leq 0.8$ Nm. Furthermore, the driver torque response successfully decays after perturbations (at the entry and exit of cornering maneuver) thereby confirming the decrease of system energy when 'dwell time' for each mode is sufficiently large. Meanwhile, since the automated controller is *switched-off* during the time of lane change, it's torque characteristic shows an initial slow decay followed by a rise in trajectory, when the driver hands over the control and finally settles down to zero.

Remark § 5.3 The Figure 5-3 and Figure 5-4 on the next page, illustrate the observed Lateral accelerations a_y and yaw rates ψ and compares them for both case I and II. For case I, the application of aggressive steering inputs to stabilize the vehicle and follow the reference trajectory leads to generation of large lateral tire forces, thereby causing oscillatory lateral acceleration responses. Such accelerations can cause *car-rolling* behavior due to the effect of dynamic weight transfer, which can lead to loss of *driver comfort*. Also, as one increases the longitudinal velocity for navigating the same lane change, the lateral acceleration values are observed to be much higher. The peaks in values of a_y for case II, at simulation time, $t_{sim} \approx 7$ secs and 17 secs, respectively at entry and exit of the *curved* section, result from generation of centripetal forces (tire forces) when cornering on road, however, these decay quickly to zero when the driver is successful in stabilizing the vehicle on the centerline of road.

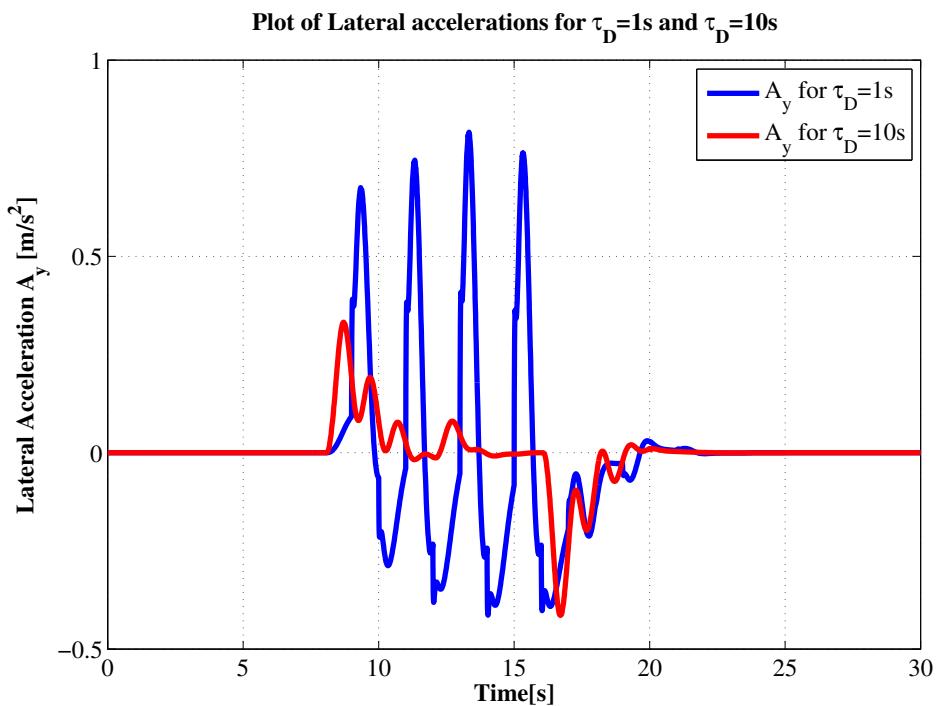


Figure 5-3: Lateral Accelerations a_y for $\hat{\tau}_D^* = 1s$ and for $\hat{\tau}_D^* = 10s$

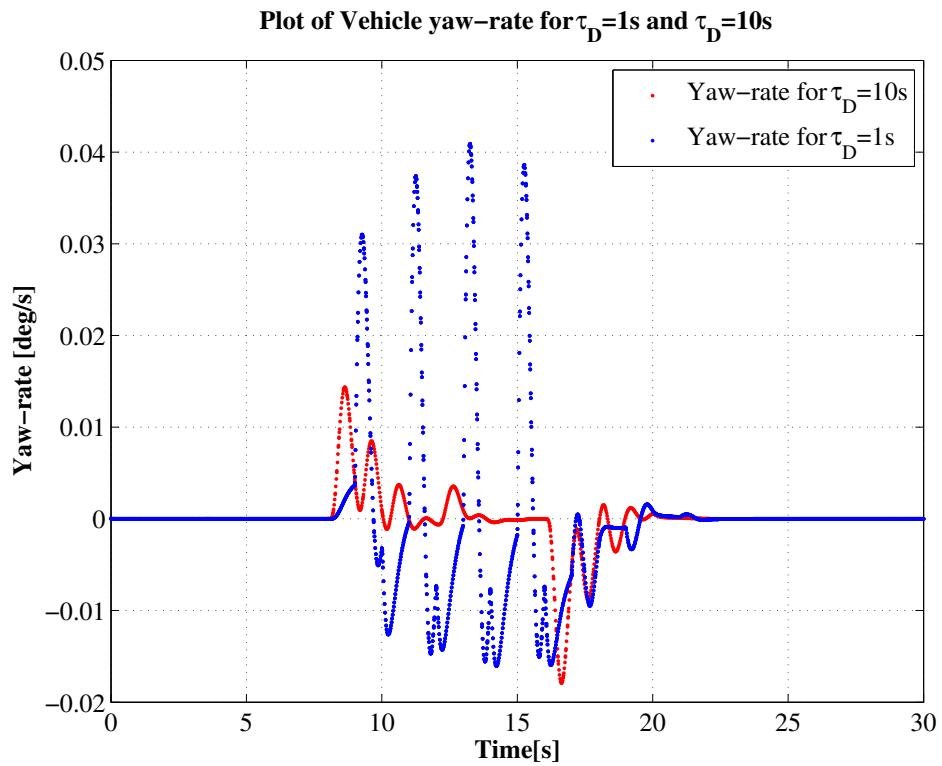


Figure 5-4: Yaw rates $\dot{\psi}$ for $\hat{\tau}_D^*=1s$ and for $\hat{\tau}_D^*=10s$

5-2 Verification of safety constraints

This section describes the experimental results obtained when the constraints written as Metric Interval Logic formulas (MITL). The parameters under investigation are human preview distance H_{th} , automation preview distance A_{th} , and the driver gain k_p . These were varied for different longitudinal velocities V_x , maximum allowed lateral deviation $y_{lat} = y - y_{ref}$ and in the next subsection for different time of switching (ToS) or the position during the lane change when the switch takes place.

The conditions below describe the values of the *safety constraints* that have been imposed on the switched system to avoid any dangerous/unsafe lane change maneuvers. These values are the outcome of closed-loop tests (Chapter 2, § 2-5-2) that had been performed on single mode driving, where the human driver navigated a single lane change at variable velocities. To build up on discussion in (Chapter 4, § 4-5) the values of $\dot{\psi}$, $\dot{\delta}$, y_{lat} that describe the safety constraints imposed on the system are:

- The maximum value of yaw rate $\dot{\psi} = 0.061 \text{ deg/s}$.
- The maximum value of steering wheel rate $\dot{\delta} = 0.75 \text{ deg/s}$.
- The maximum lateral deviation $y(t) - y_{ref}(t) \leq 0.3 \text{ m}$

Then, the logic can be described in terms of an QMITL_formula as:

$$\phi = (\text{alw } (\dot{\psi} < \alpha_1) \wedge \text{alw } (\dot{\delta} < \alpha_2)) \wedge \text{alw } (y - y_{ref} = \alpha_3)$$

where, the constants are:

$$\begin{aligned}\alpha_1 &= 0.061 \\ \alpha_2 &= 0.75 \\ \alpha_3 &= 0.3\end{aligned}$$

Now, for performing the said experiments it is important to describe the algorithm that has been followed in this thesis, for falsification of constraints that in turn leads to obtaining the values of the acceptable values of parametric intervals, that forms the core of this thesis. The *Parameter Synthesis* algorithm described in [80] explains the underlying methodology. The authors base separation of sets into safe, unsafe or uncertain based on an approximation of the reachable set. Keeping the uncertain subset possibly very small, the algorithm basically refines iteratively the parameter set \mathcal{P} . The resulting subset falls under the category of ‘safe’ if it doesn’t intersect the *Bad set* \mathcal{B} and stays reliable within desired ‘toleration’ bounds. For termination, each refining iterations utilize subsets strictly smaller than the previous refined set.

5-2-1 Iterative procedure

This subsection explains the *Iterative procedure* that will be used as a basis for further investigations on parametric verification related to different switching scenarios. Verification using BREACH is in essence an interval based verification methodology. We start by creating a set of parameters that have to be analyzed. These parameters are then assigned nominal interval values as required by BREACH's inbuilt verification algorithms. These values have been obtained (§ 4-2, Figures 4-2 and 4-3) from root locus analysis of closed-loop transfer functions of the manual and automated vehicle. These are $A_{th} = [45 \ 55]m$, $H_{th} = [13 \ 18]m$ and $k_p = [0.98 \ 1.02]$.

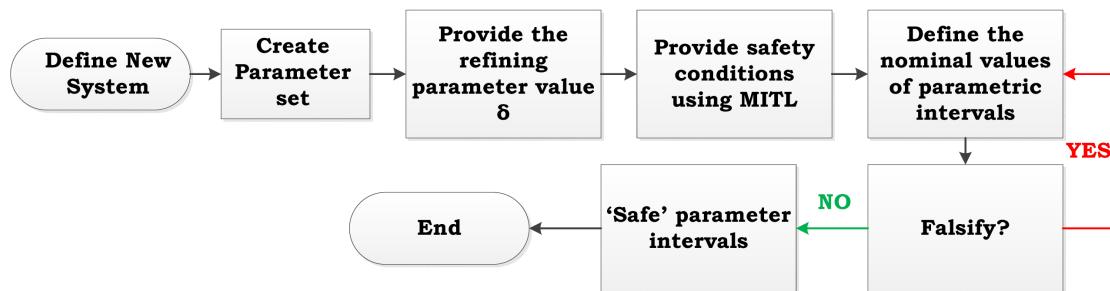


Figure 5-5: Steps of the iterative procedure

Then based on the value of constraints, we run a falsification algorithm which *stops* when a falsifying trajectory is encountered and returns the 'safe' intervals that respect the mentioned constraints. The values 'YES' in the Table 5-2 are the results for simulations that could not generate 'safe' trajectories. So, we keep varying the the parametric intervals and observe the 'reachable sets', as well as plot the trajectories of variables involved. After 4 iterations, we finally arrive at an interval that satisfies the constraints and values for each of the parameter sets are the final resulting values $A_{th} = [45 \ 55]m$, $H_{th} = [13 \ 18]m$ and $k_p = [0.98 \ 1.02]$. This procedure has been described in the Figure 5-5 and the table 5-2.

Desired Lateral Deviation y_{lat} , (m)	Automation Preview A_{th} [min max] (m)	Human Preview $H_{th}(m)$ [min max] (m)	Driver gains k_p [min max]	Falsification Result Yes/No?
0.5	[13.5 18]	[30 55]	[0.98 1.05]	YES
0.4	[13.5 18]	[30 55]	[0.98 1.05]	YES
0.3	[14.5, 51]	[14,18]	[0.98,1.0]	YES
0.2	[15, 51]	[14,16]	[0.98,1.0]	NO

Table 5-2: Determining optimal parameter intervals through Iterative Procedure

5-2-2 Fixed position of switching

This section describes the parametric verification done to observe the interactions that arise from *switching-on* the human driving mode as soon as the vehicle is about to navigate the lane change (vehicle enters the *curved* section). Thus switching is only allowed to take place *once*, that too at a fixed position (corresp. to simulation time, $t_{sim} = 20s$). We then use the *iterative procedure* to observe the inter-related effects of human preview distance, driver gains, and longitudinal velocity when the human driver is in control. To validate the system stability in such an experimental condition we utilize the results of time-constrained switching developed in the previous sections. The switch from mode $q_2 \rightarrow q_1$ occurs at $t_{sim} = 20$, implying switch at 20 seconds since start of simulation. Since, we still stay in the automated mode q_2 for 20s (considering total simulation time, $t_{sim} = 40$ s), the average dwell time condition is respected as $\hat{\tau}_D = 20s$ leads to $\hat{\tau}_D \geq \hat{\tau}_D^* = 5.13s$.

Effect of human preview distance on lateral deviation

Max. Lateral Deviation [m], $y - y_{ref}$	0.5	0.4	0.3	0.2	0.1	0.05
Human Preview Distance [m], (min-max)	13.5-18	14-18	14.5-18	15.5-18	16-18	Not feasible

Table 5-3: Effect of varying maximum Lateral deviation on parametric intervals

The table 5-3 represents the result of the first set of experiments. As can be observed, with decreasing maximum Lateral Deviation [m], $y - y_{ref}$, the minimum human preview distance H_{th} increases. Such results make sense, because for effective lane-keeping the Human driver should see far enough, so that sudden oscillations in steering can be avoided. Too close a preview distance leads to undamped steering maneuvers thereby resulting in a loss of accuracy for path tracking purposes.

Effect of velocity on driver gains and human preview length

Longitudinal velocity V_x (m/s)	Driver gains k_p (min-max)	Human Preview H_{th} (min-max) [m]
90	0.98-2	14.5-18
100	0.98-1.8	14.5-18
110	0.98-1.6	15-18
120	0.98-1.2	16-18

Table 5-4: Effect of varying longitudinal velocity on parametric intervals

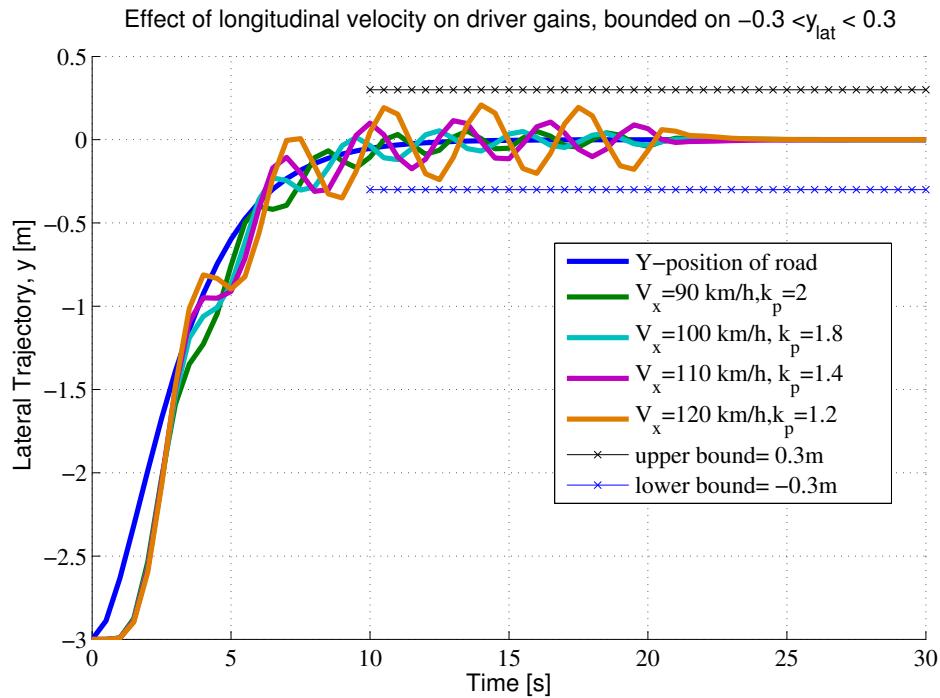


Figure 5-6: Verification of driver gain intervals k_p as observed with variation of the longitudinal velocity V_x . The dotted black and blue lines shows the constraints on maximum allowed lateral deviation $0.3 \leq y - y_{ref} \leq 0.3$

An important scenario to analyze is the effect of longitudinal velocities V_x on driver gains and human preview distances. The Table 5-3 describes the variation in longitudinal velocity, where $V_x \in [90 \text{ } 120]\text{km/h}$. As can be observed the gains decrease with an increase in velocity. This is because the task of lane-keeping while negotiating a curved trajectory at low velocities requires larger steering angles, on the account of low lateral vehicle forces and very low side-slip angles, hence resulting in a more ‘stiff/focused’ driving. But, at higher velocities the effect of ‘side-slip’ angles adds to the required ‘drift’ while making a cornering maneuver . However, such gains in driving effort come at the cost of larger preview distances.

The Figure 5-6 describes how the driver gains have to be *adjusted* with increasing longitudinal velocities, if the same safety constraints have to be followed. The vehicle starts negotiates a lane change maneuver of width 3m and the maximum allowed lateral deviation is 0.3 m.

Few remarks are in order after such results:

Remark § 5.4 In the first driving scenario $V_x = 90\text{km/h}$ and $k_p = 2$, although neither $-0.3 \not< y - y_{ref}$ nor $y - y_{ref} \not> 0.3$, it is observed that the oscillatory behavior of the yaw-rate $\dot{\psi}$ and steering-angle rate $\dot{\delta}$ increases as one increases the driver gain. This refers to the fact that a ‘stiff’ driver ($k_p = 2$) tries to steer aggressively to stabilize the vehicle specifically before transferring the control to automated vehicle, compared to a more relaxed driver.

Remark § 5.5 In the last driving scenario $V_x = 120\text{km/h}$ and $k_p = 1.3$ the ‘safe’ values for human preview distance returned by the algorithm is $H_{th} = [16 \text{--} 18]\text{m}$. However, the optimal value was found to be 17 m. When the human driver has $H_{th} = 17$ m, the observed lateral deviation is as low as 9cm and the oscillation for steering wheel rate response remain bounded between $-0.01 \leq \dot{\delta} \leq 0.01$. Figure 5-7 and Figure 5-8 illustrate the observed results.

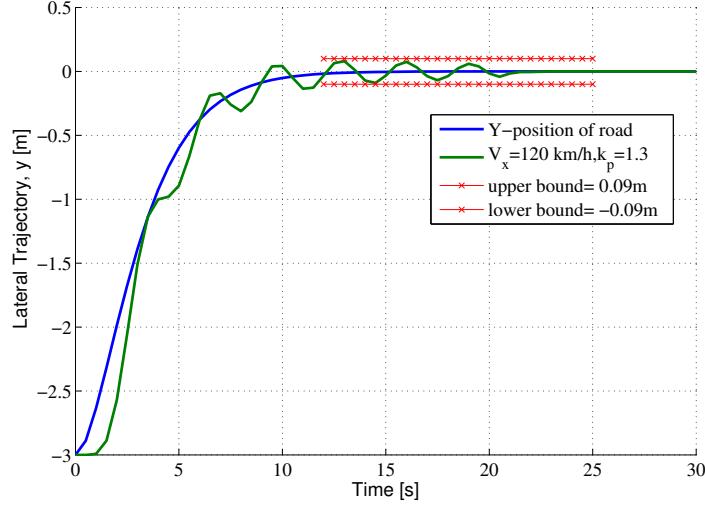


Figure 5-7: vehicle lateral response for $H_{th} = 17$ m

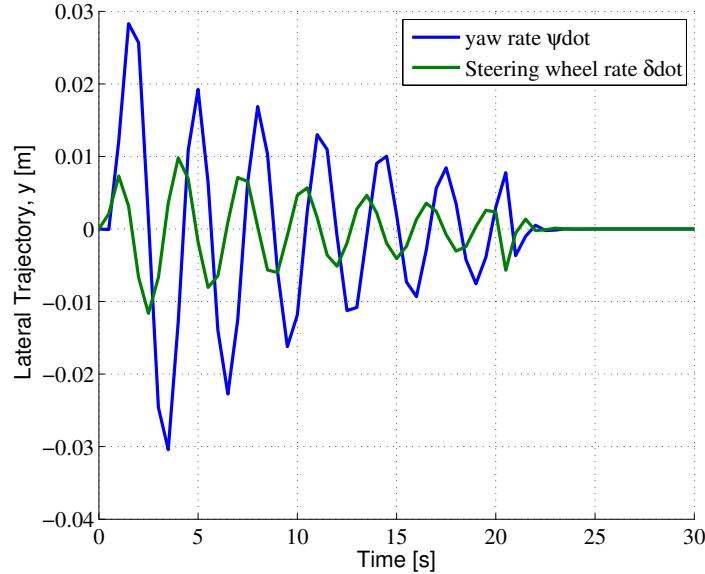


Figure 5-8: Yaw rate $\dot{\psi}$ and steering wheel rate $\dot{\delta}$ response for $H_{th} = 17$ m

5-2-3 Variable position of switching

Last subsection described the influence of parametric valuations for switching between modes at a fixed position. Although the results were insightful enough to understand the human driving behavior, a next logical step would be to answer the question: *What happens when instead of switching after the end of lane change maneuver, one decides to switch in between or switch at different locations?*. Then, it remains to investigate how would a variation of position of transfer of control effect the parameters. We introduce a new terminology called the Time of Switching (ToS) for this purpose. So, varying time of switching allows the switching to take place at different positions in a lane change. Although bear in mind that, irrespective of when/where one wants to switch in a lane change maneuver, the time one spends in each mode (or the TTS) has to respect the concepts of average dwell time, so $TTS = \frac{\tau_s^{q1} + \tau_s^{q2}}{2} \geq \hat{\tau}_D^*$.

Variation of Time of Switch (ToS) on driver gain, human preview and automation preview distance

Time of Switching ToS (s)	Max. Driver gains k_p	Controller Preview A_{th} (m)	Human Preview H_{th} (m)
20	1.8	40-65	16-18
15	1.8	40-65	15-18
10	1.6	45-55	14-18
5	1.4	55-65	13-18

Table 5-5: Effect of varying time of switching on parametric intervals

Remark § 5.6 For the case (I) where ToS=20s described in the Table 5-5 , so the transfer of control from human to automated vehicle takes place after the lane change, although the temporal logic constraint on $\dot{\psi}, \dot{\delta}, y_{lat}$ is still satisfied, the $\dot{\psi}, \dot{\delta}$ responses exhibit a dangerous trend as they remain very close to the bad set, \mathcal{B} . Furthermore, the driver gain $k_p = 1.8$ is the *maximum* driver gain that can be applied without reaching unsafe regions, and hence in further experiments it was observed that the values $k_p=1.1$ and $A_{th}=56$ show the best response in the entire interval as illustrated by Figure 5-9 and Figure 5-10. Good lane-keeping behavior is observed for $-6cm \leq y_{lat} \leq 6cm$ and a smooth transfer of control is realized as the observed values at the instant of switching for yaw-rate lie in $-0.005 \leq \dot{\psi} \leq 0.005 rad/s$ and steering angle lie in $\dot{\delta} \leq 10^{-4} rad/s$.

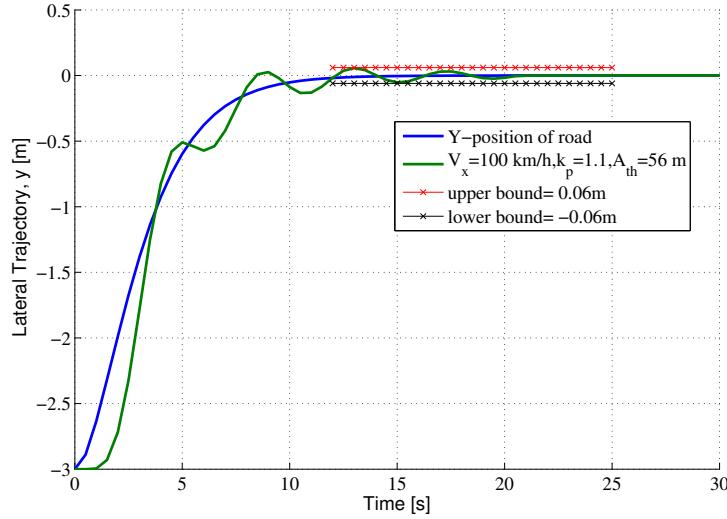


Figure 5-9: vehicle lateral response for $k_p=1.1$ and $A_{th}=56$. The lateral deviation remains bounded: $-6\text{cm} \leq y_{lat} \leq 6\text{cm}$

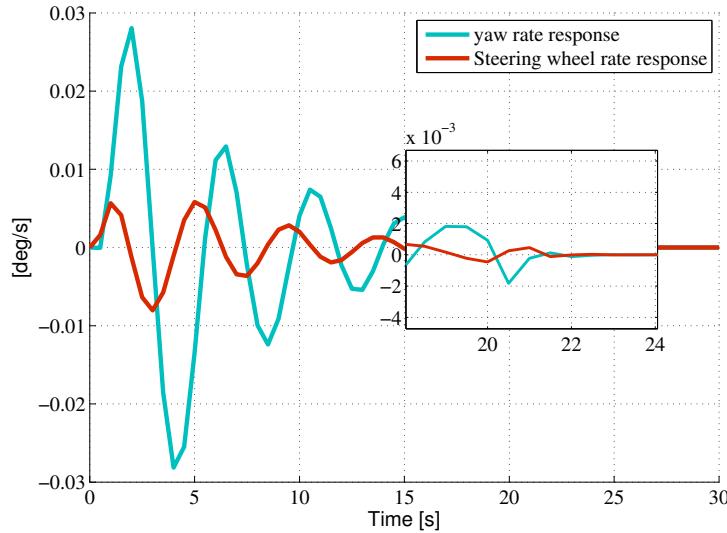


Figure 5-10: Yaw rate $\dot{\psi}$ and steering wheel rate $\dot{\delta}$ response for $k_p=1.1$ and $A_{th}=56$. The magnified portion shows the responses at the instant of switching

Now, to demonstrate the effect of time of switching (ToS) graphically, we consider the effects of *nature* of human driver by considering two gain values $k_p = 1.2$ representing a ‘relaxed driver’ and the value $k_p = 2$ representing a ‘stiff driver’. Here, the terms relaxed (and stiff) are just qualitative definitions allotted to human driving behavior. For these two types of drivers we then consider two different human preview distance values $H_{th} = 13\text{ m}$ and $H_{th} = 18\text{ m}$, described a driver with smaller and larger look-ahead distances respectively. Figures 5-11 and 5-12 illustrate the lane-keeping behavior of a stiff driver, and figures 5-13 and 5-14 illustrate the lane-keeping behavior of a relaxed driver.

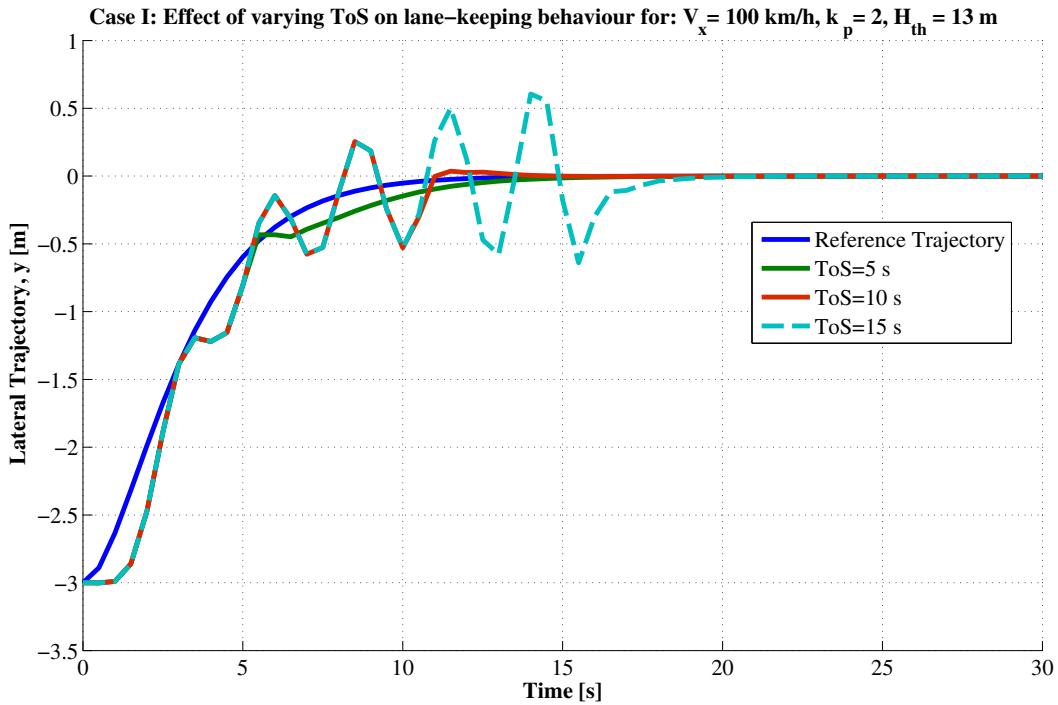


Figure 5-11: Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 \text{ km/h}$, $k_p = 2$, $H_{th} = 13 \text{ m}$

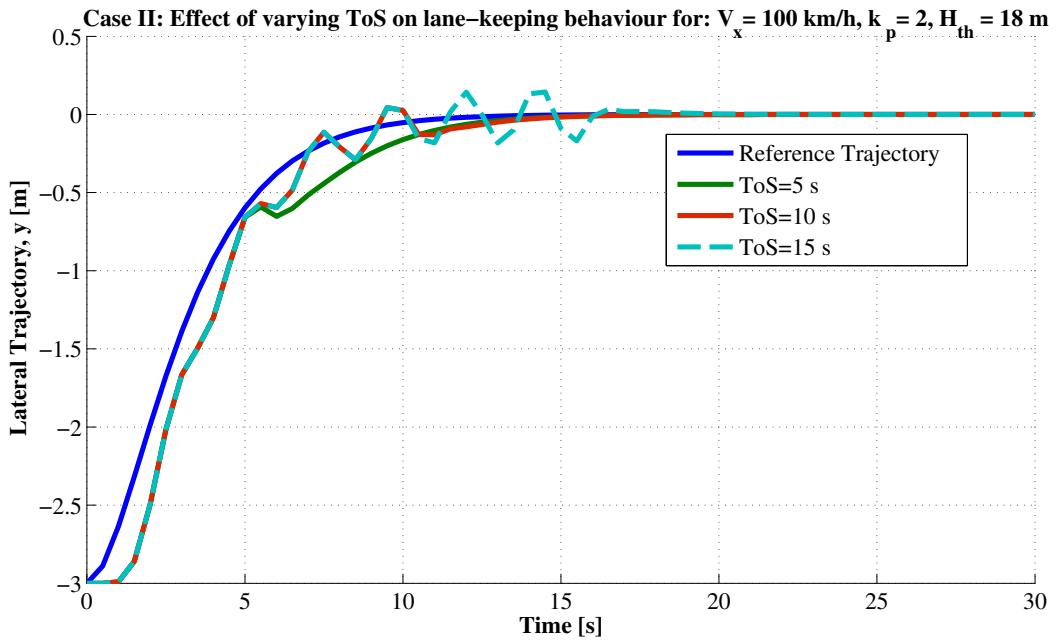


Figure 5-12: Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 \text{ km/h}$, $k_p = 2$, $H_{th} = 18 \text{ m}$

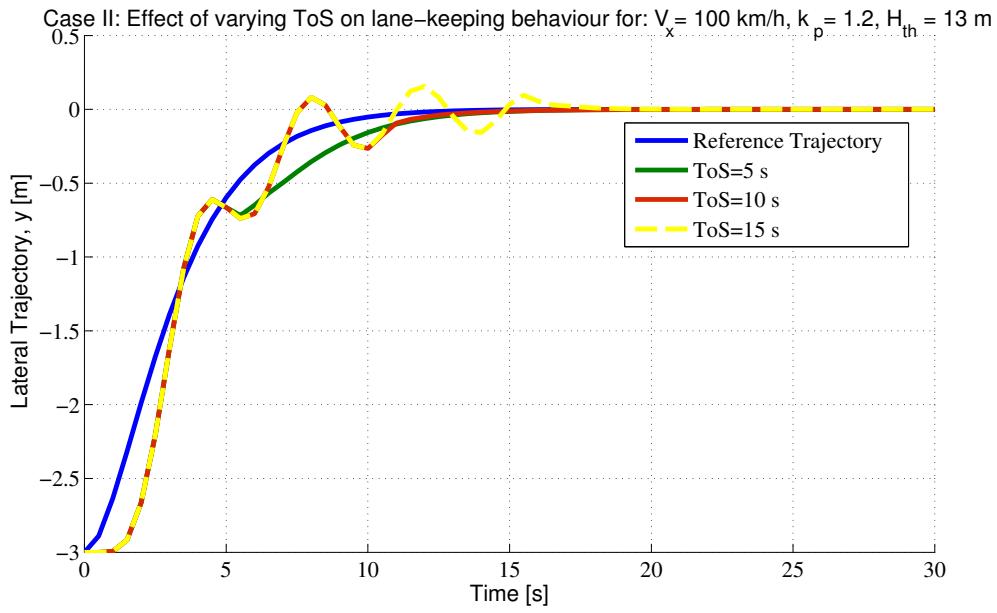


Figure 5-13: Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 \text{ km/h}$, $k_p = 1.2$, $H_{th} = 18 \text{ m}$

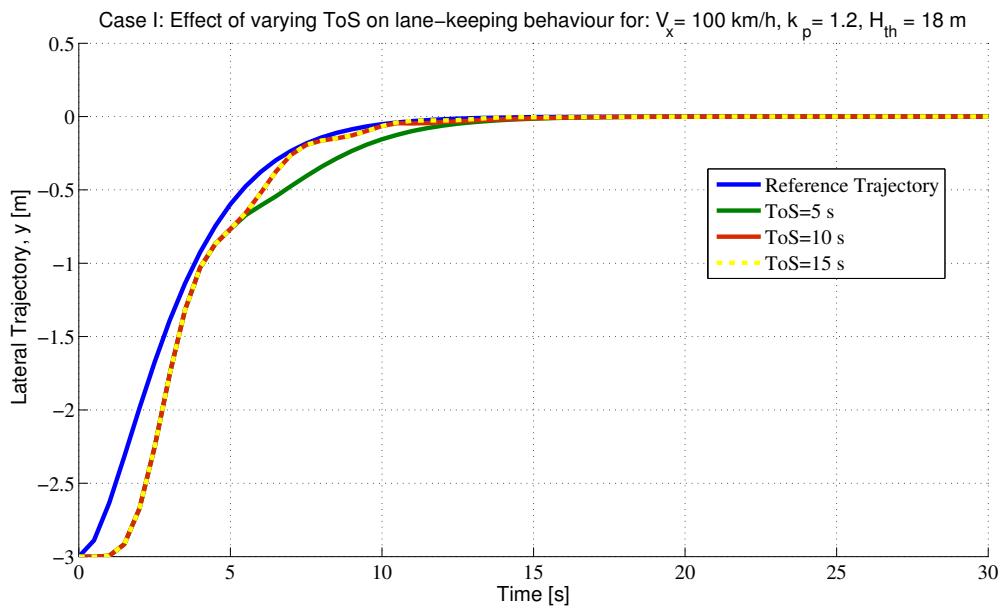


Figure 5-14: Effect of varying Time of Switching (ToS) on lane-keeping behaviour for: $V_x = 100 \text{ km/h}$, $k_p = 1.2$, $H_{th} = 13 \text{ m}$

Remark § 5.7 Referring to the table 5-5, we consider the case where ToS=15 s, so transferring control at the end of *curved* section of the lane change. As illustrated in the Figure 5-15 the yaw-rate values remain bounded well-within the constraints imposed. However, it is important to mention that these responses have been generated for *nominal* A_{th} values [45-55] m, however during experimentation with higher values of A_{th} (≈ 64 m), heavy oscillations in the steering wheel rate $\dot{\delta}$ and $\dot{\psi}$ were observed, Figure 5-16 illustrates the response.

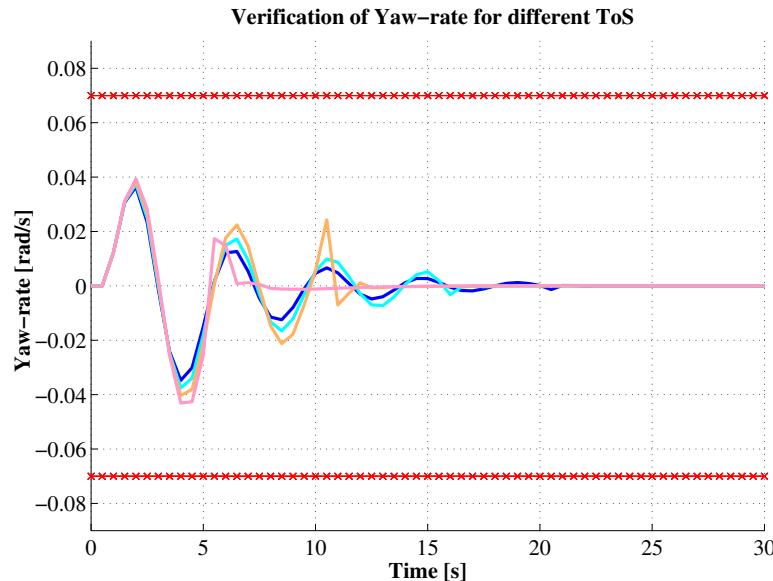


Figure 5-15: Verification of yaw-rate based on variation of Human preview distances with different Time of Switching. The ‘crossed’ red line shows the constraints on bound on yaw rate values $-0.07 \leq \dot{\psi} \leq 0.07$. The colored plots illustrate: (a.) The Blue line :ToS= 20s, $H_{th} = 16$ m, (a.) The Orange line :ToS= 15s, $H_{th} = 15$ m , (c.) The Cyan line :ToS= 10s, $H_{th} = 14$ m, (d.) The Pink line :ToS= 5s, $H_{th} = 13$ m].

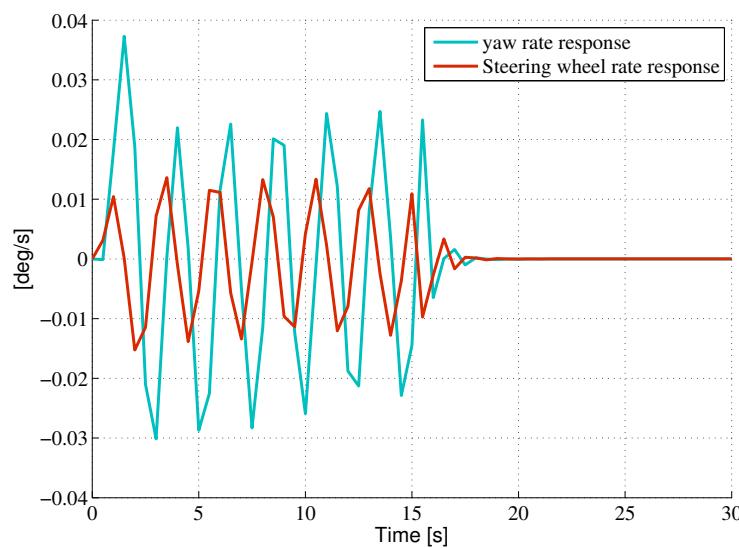


Figure 5-16: Heavy oscillations in steering wheel rate $\dot{\delta}$ and yaw rate $\dot{\psi}$ for $A_{th} = 64$.

Chapter 6

Conclusions and Recommendations

6-1 Discussions on the project

This thesis work developed a ‘primary evaluation scheme’ for analyzing the transition of control between automated and manual driving based on the hybrid systems framework. The results presented in sections § 5-1 and § 5-2 and the remarks therein, provided a quantitative explanation of the experimental observations made during application of the concepts of switching based on average dwell time and parametric verification of the manual-automated switched system and also correlates to the readers logical and intuitive experiences regarding everyday driving scenarios. However, it should be noted that these observations were made while working with certain assumptions, hence many improvements can be made. The next section provides a list of recommendations on selected issues that the author felt deserve more investigation, than it is accorded in this project.

To conclude, the main ideas, observations, tools, and insights developed in this project can be briefly explained by answering the sub-questions posed in Chapter 1,§ 1-2. This would allow the reader to connect the individual ‘units’ of this thesis together and hopefully obtain the final answer to the problem statement of this thesis.

Q1: How can one model the manually driven vehicle and the automated vehicle?

The Human driver has been modeled as preview controller with a neuromuscular dynamics component, whereas the automated vehicle has been developed using PID control strategy for speed control and PD control strategy for steering control. The automated and manually driven vehicle were modeled respectively as 9th ($q_1[9 \times 9]$) and 11th ($q_2[11 \times 11]$) order system using the *state-space* representation. Chapter 2 described how the dynamics of these individual modes were represented as a set of coupled first-order differential equations of the system variables. Furthermore, a classification of the four parameters, namely, Time To Switch (TTS), human preview distance, automation preview distance, and driver gain, has been provided to characterize human driver behavior under the notions of *Driver Competence* and *Situation Awareness*.

Q2: How can the driver take back control and how to involve him?

For analyzing the reclaiming of control by human driver, a time-based parameter, Time to Switch (TTS) has been developed. Each of the individual modes have their own TTS, described as $\tau_s^{q_1}$ for the automated driving mode(q_1) , and $\tau_s^{q_2}$ for the manual driving mode (q_2). The use of a time-dependent switching methodology based on average dwell time switching $\hat{\tau}_D$ allows the driver to take back control from the automated vehicle anytime he so feels like, provided he still is able to maintain an average dwell time greater than/equal to $\hat{\tau}_D^*$. Also, for guaranteeing ‘safe’ switching on the combined human-automation switched system, we *bound* the time to switch (TTS) of mode q_1 to $\tau_s^{q_1} = 1.5$ s, which is the minimum time that the switched system has to stay in the automated mode before a transfer of control occurs. Then, the time to switch of mode q_2 can be calculated by the mathematical relation: $\tau_s^{q_2} \geq 2 \times (\hat{\tau}_D^*) - 1.5$ s, where $\hat{\tau}_D^*$ was determined to be 5.13 s.

Q3: How can a smooth switching/transition be effected?

Smooth switching can be effected by allowing each of the individual modes to dwell for a time interval that when averaged over entire time range of simulation respects the mathematical conditions $\hat{\tau}_D \geq \hat{\tau}_D^*$ that result from the work of [72], where, $\hat{\tau}_D$ is the average dwell of the entire switched system. Furthermore, to obtain ‘acceptable’ values for other parameters like human preview distance, automation preview distance, and driver gain, the parametric verification of the switched system using the BREACH Matlab toolbox was performed. For this the safety conditions based on the yaw rate ($\dot{\psi}$), the steering angle rate ($\dot{\delta}$), and the lateral displacement ($y - y_{ref}$) were formulated and then a *Bad set* \mathcal{B} of values that negate this safety condition was searched for. Those parametric valuations that didn’t intersect with *Bad set* qualified as the acceptable values.

Q4: What are critical scenarios ? Define the situations for implementation?

The basic experimental scenario as discussed in section 2-5 is a single lane change of width 3 m that is navigated with a constant longitudinal velocity of 100 km/h. For investigating the metric Time To Switch (TTS), we extend the scenario to include two different cases: *Case I*, where the average dwell time of the system $\hat{\tau}_D < 5.13$ s and *Case II*, where $\hat{\tau}_D > 5.13$ s. Furthermore, for experimenting with values of human preview distance H_{th} , automation preview distance A_{th} and driver gain k_p , we again extend the basic scenario to define two different cases: the first case is *fixed position of switching*, where the switching takes places only once near the end of lane change. For this case, the effect of varying longitudinal velocity V_x and maximum lateral deviation on human driver gains was studied. The second case is *variable position of switching*, where the switch between manual driving and automated driving vehicle takes place at different positions along the single lane change. For this case, the effect of different time of switching was studied on the human preview distance, automation preview distance and driver gains.

6-2 Future Recommendations

1. Numerous driver modeling efforts in the literature focus on human steering models that are quite advanced. These are based on either *Control theoretic* models, identification theory etc., or based on *behavioral* models involving psycho-biological, cognitive approaches. For those interested in a classification based overview of driver models, [69] provides a comprehensive and insightful study. Each of these models have their own advantages and within acceptable limits, can very closely resemble the Human driving behaviour. However, the focus in this thesis is on the design and analysis of the steering interactions by applying the techniques of Hybrid control framework. Therefore, a relatively simple yet for the envisaged maneuver sufficiently accurate model of human steering behavior was set up. We use a *single point preview tracking* [64] model in conjugation with the *Human Driver Simulink* model based on the neuromuscular (NMS) driver model developed by the Delft BioMechanics group [65]. It is thus believed that using other advanced driver models provides great potential, and hence is best a subject of future investigations on this topic.
2. The preview controller used in this thesis to describe the lateral vehicle control by human driver is developed for a single preview point. In this control technique, depending upon the controller gain (k_p , k_d) the driver tries to minimize the lateral error e_p between actual and target preview point at a distance (d_p) ahead, resulting in a feedback loop that essentially responds to heading angle ψ and lateral position y . However, the challenge with such an approach lies in obtaining a ‘reasonable’ preview distance. Too far or too close a preview point can both lead to vehicle instability. Also, the notion of real-life human driving considering just a single-point preview is not accurate. With advances in research ‘multi-point preview’ driver models were developed to overcome these challenges. Starting from the work of [88], this concept was later extended by [83] to develop a time-invariant optimal (TI-optimal) control. These models compute the optimal road preview for good and effortless tracking. They also account for the stability degradation for restricted preview distances and present an attractive compromise between path-following accuracy and driver workload.
3. The Force-Feedback Driver Model (FFDM) developed by [65] was used to model the human driving behavior in this project. The motivation for using this model, as described by [66] is its sensitivity towards steering wheel systems with different dynamics and efficient prediction of both goal-directed steering wheel movements and neuromuscular feedback. However, the FFDM developed was only validated for two different scenarios: lane-keeping and lane-changing maneuver. The results for four different parametrizations of steering system and their effect on two different driver types ('Relaxed' and 'Stiff') that have been presented in [66] are insufficient to draw any larger conclusions. Thus, owing to the diverse behavior of human drivers the validation of FFDM using real-life driving data instead of only simulation based studies would be helpful in justifying its resemblance to actual human driving.

4. The use of Cruise Control (CC) in this thesis is justified by the need to maintain a constant longitudinal vehicle speed while navigating the lane change. However, many of the present day commercial vehicles have incorporated ACC (Adaptive Cruise Control) in their active safety architecture. Also, the use of ACC can account for a worst-case scenario when the human driver take over fails due to some reason. In such a case, the vehicle can easily decelerate if the preceding vehicle slows down/stops or keep just moving forward at a constant speed. Thus, it would be interesting to extend the longitudinal controller used in thesis to incorporate ACC and then analyze the resulting interactions between the modes.
5. This thesis utilizes the concepts of time-dependent switching methodology for analyzing the steering interactions between manual and automated vehicle. Extending this concept to switch between the modes based on applied steering torques will be of great value for future investigations. Such a scheme would envisage the human driver applying a certain threshold torque so as to not destabilize the vehicle but signal the automation for a take-over of control. Similarly, for transferring the control back to automated vehicle, if the automation observes that the applied driver torques have remained constant for a ‘certain’ duration of time, this would allow it to safely take-over the control of the vehicle. For such a case, the guards and invariants of the mode switching automata described in Chapter4, § 4-1 can then be re-designed to incorporate switching based on measured threshold values of human and controller torques.
6. The vehicle model that we have used for controller design is a 4DOF ‘two-track’ vehicle model. Within acceptable limits, this model is complex enough to resemble a real-life vehicle. However, for reasons of simplicity and the inherent requirements of the time-dependent switching methodology used, it was linearized to a 2 DOF model. That said, it still remains to observe the effect of parametric valuations on the hybrid automata consisting of a non-linear vehicle model. As a starting point the work done by [85] on application of average dwell time approach for switched systems with nonlinear dynamics can be referred.
7. The discussion on formal verification tools provided in section 2-3-1 is not exhaustive. This is quite obvious considering the fact that the domain of Hybrid Systems and Control has disparate requirements and design choices which merits the presence of a multitude of analytical tools. The tool selected for this thesis was BREACH . However, this was not the first tool used. Starting with many efficient tools like PHAVER, SPACEEx, we had narrowed down to ARIADNE, owing to its remarkable ability to perform a rigorous computable analysis for calculating over-approximations of reachable sets. Unfortunately, ARIADNE is an open, in-progress verification environment and so suffers from lack of working examples and compile time errors. However, it is advised to track the developments in ARIADNE, because a verification methodology that utilizes its full potential will go a long way in providing insightful details about the human-automation switching framework.

Appendix A

Miscellaneous

The Manually controlled vehicle was modeled in Chapter 2, § 3-2. The Equations 3-25 and 3-26 described the augmented state-space representation of PD controller and the Human dynamics block (Refer Fig.3-2). This section explains the step-by-step modeling procedure that led to the mathematical relations defined in the equations 3-25 and 3-26.

A-1 Mathematical Derivations

First, consider the equations 3-25 and 3-26 again. These equations explain the *combined* dynamics of the PD controller and the Human dynamical systems. Starting from modeling of individual units, we aim to arrive at these equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{m}_1 \end{bmatrix} = \begin{bmatrix} -l_1 & -l_2 & \tau_d \\ 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ m_1 \end{bmatrix} + \begin{bmatrix} \frac{k_d}{\tau_d} \\ 0 \\ -\frac{C_{2D}}{\tau_d} \end{bmatrix} [e_d]$$
$$T_D = \begin{bmatrix} \xi_1 & \xi_2 & \frac{K_0}{\tau_d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ m_1 \end{bmatrix} + \begin{bmatrix} \frac{K_0 k_d}{\tau_d} \end{bmatrix} [e_d]$$

The Human dynamics block in the Fig. 3-2 has neuromuscular and perception components, and can be described in terms of transfer function $G(s)$. Based on the McRuer's Crossover model [87]:

$$G(s) = \frac{T_D}{\delta_{NMS}} = k_p \cdot \frac{\tau_L s + 1}{\tau_I s + 1} \cdot e^{-(\tau_d + \tau_N)s} \quad (\text{A-1})$$

where,

k_p = Human Driver Gain

τ_L = Lead Constant

τ_I = Lag constant, together with k_p, τ_L , represents the nimbleness of driver

τ_d = neuromuscular delay constant

τ_N = action delay constant

Now, using pade's approximation :

$$G(s) = k_p \cdot \frac{\tau_L s + 1}{\tau_I s + 1} \cdot \frac{[1 - (\frac{\tau_d + \tau_N}{2})s]}{[1 + (\frac{\tau_d + \tau_N}{2})s]} \quad (\text{A-2})$$

$$= k_p \cdot \frac{-\tau_L(\frac{\tau_d + \tau_N}{2})s^2 + \tau_L s + 1 - (\frac{\tau_d + \tau_N}{2})s}{\tau_I(\frac{\tau_d + \tau_N}{2})s^2 + \tau_I s + 1 + (\frac{\tau_d + \tau_N}{2})s} \quad (\text{A-3})$$

$$= \frac{K_0 s^2 + \xi_1 s + \xi_2}{s^2 + l_1 s + l_2} \quad (\text{A-4})$$

where,

$$\begin{aligned} K_0 &= -\frac{\tau_L \cdot k_p}{\tau_I} \\ \xi_1 &= \frac{\tau_L - \frac{(\tau_d + \tau_N)}{2}}{\tau_I \frac{(\tau_d + \tau_N)}{2}} \\ \xi_2 &= \frac{k_p}{\tau_I \frac{(\tau_d + \tau_N)}{2}} \\ l_1 &= \frac{\tau_L + \frac{(\tau_d + \tau_N)}{2}}{\tau_I \frac{(\tau_d + \tau_N)}{2}} \\ l_2 &= \frac{1}{\tau_I \frac{(\tau_d + \tau_N)}{2}} \end{aligned}$$

Now, the transfer of Eqn. A-4 has to be converted to an equivalent state-space representation. Consider two dummy states x_1, x_2 for the above purposes, also T_D represents the applied Driver Torque on the steering wheel. Then the following state-space representation results:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -l_1 & -l_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \delta_{NMS} \quad (\text{A-5})$$

$$T_D = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} K_0 \end{bmatrix} \delta_{NMS} \quad (\text{A-6})$$

The first-order differential equations in Eqns. A-5 and A-6 represent the state and the output equations for the Human Neuromuscular System. Then, we move on to utilizing the PD controller state-space representation (Chapter3 § 3-1).

$$\dot{m}_1 = -\frac{m_1}{\tau_d} - \left(\frac{k_d}{\tau_d}\right) e_d \quad (\text{A-7})$$

$$\delta_{NMS} = \frac{m_1}{\tau_d} + \left(\frac{k_d}{\tau_d}\right) e_d \quad (\text{A-8})$$

where,

$$\begin{aligned} m_1 &= \text{state of PD controller} \\ k_d &= C_{1D}\tau_d + C_{2D} \\ C_{1D}, C_{2D} &= \text{The gains of the PD controller.} \\ \tau_d &= \frac{C_{2D}}{10} \text{ (as a rule of thumb) [59]} \end{aligned}$$

Then, as a final step we combine the equations, A-5, A-6, A-7 and A-8 :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{m}_1 \end{bmatrix} = \begin{bmatrix} -l_1 & -l_2 & \tau_d \\ 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ m_1 \end{bmatrix} + \begin{bmatrix} \frac{k_d}{\tau_d} \\ 0 \\ -\frac{C_{2D}}{\tau_d} \end{bmatrix} [e_d] \quad (\text{A-9})$$

$$T_D = \begin{bmatrix} \xi_1 & \xi_2 & \frac{K_0}{\tau_d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ m_1 \end{bmatrix} + \begin{bmatrix} \frac{K_0 k_d}{\tau_d} \end{bmatrix} [e_d] \quad (\text{A-10})$$

A-2 List of Acronyms

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
CACC	Cooperative Adaptive Cruise Control
CC	Cruise Control
COG	Centre of Gravity
CQLF	Common Quadratic Lyapunov Functions
DOF	Degree of Freedom
FFDM	Force Feedback Driver Model
NHTSA	National Highway Traffic Safety Administration
LKAS	Lane Keeping Assistance Systems
MLF	Mutliple Lyapunov Functions
MITL	Metric Interval Temporal Logic
PIO	Pilot Induced Oscillations
SLC	Single Lane Change

Appendix B

Parameter Values

Parameters	Values
Mass of the vehicle, m [kg]	1384
Distance from COG to front axle, a [m]	1.4
Distance from COG to back axle, b [m]	1.4
vehicle moment of Inertia, I_{zz} [kg-m ²]	2000
Front tire coefficient, $C_{\alpha 1}$	98389
Rear tire coefficient, $C_{\alpha 2}$	198142
Human driver Lead constant τ_L [s]	0.1
Human driver Lag constant τ_I [s]	0.04
Neuromuscular constant, τ_N [s]	0.16
Delay constant, τ_d [s]	0.15
Total moment of inertia of the steering system J_w [N-m ² /rad]	0.3
Viscous damping of the steering system B_w [N-m-s/rad]	2
Stiffness of the steering system K_w [N-m/rad]	4.2
Steering wheel gear ratio, g_{sw}	15

Table B-1: Parameter settings used in this thesis

Bibliography

- [1] International Road Assessment Programme, IRAP (2007) The true cost of road crashes: valuing life and the cost of a serious injury.
- [2] National Highway Traffic Safety Administration, NHTSA (2008) National motor vehicle crash causation survey: report to congress. DOT HS 811 05, July 2008
- [3] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, “Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations,” *Vehicle System Dynamics*, vol. 44, no. 7, pp. 569-590, July 2006.
- [4] Bar, Francois, and Yves Page. “An empirical classification of lane departure crashes for the identification of relevant counter-measures.” Publication of: ASSOCIATION FOR THE ADVANCEMENT OF AUTOMOTIVE MEDICINE (2002).
- [5] Kortenkamp, David, et al. “Traded control with autonomous robots as mixed initiative interaction.” *AAAI Symposium on Mixed Initiative Interaction*. 1997.
- [6] Norman, Donald A. Emotional design: Why we love (or hate) everyday things. Basic books, 2007.
- [7] Wiese, Emily E., and John D. Lee. “Attention grounding: a new approach to in-vehicle information system implementation.” *Theoretical Issues in Ergonomics Science* 8.3 (2007): 255-276
- [8] Stokes, A. F., C. D. Wickens, and K. Kite. “Display Technology: Human Factors Concepts. Warrendale, PA: Society of Automotive Engineers.” (1990).
- [9] Stanton, Neville A., and Philip Marsden. “From fly-by-wire to drive-by-wire: safety implications of automation in vehicles.” *Safety Science* 24.1 (1996): 35-49.
- [10] Parasuraman, Raja, et al. *Theory and design of adaptive automation in aviation systems*. CATHOLIC UNIV OF AMERICA WASHINGTON DC COGNITIVE SCIENCE LAB, 1992.

- [11] Bailey, Randall E., and Timothy J. Bidlack. *Unified Pilot-Induced Oscillation Theory. Volume 4. Time-Domain Neal-Smith Criterion.* No. CALSPAN-FR-8184-13-VOL-4. CALSPAN ADVANCED TECHNOLOGY CENTER BUFFALO NY, 1995.
- [12] Young, Mark S., Neville A. Stanton, and Don Harris. "Driving automation: learning from aviation about design philosophies." *International Journal of Vehicle Design* 45.3 (2007): 323-338.
- [13] Stanton, Neville A., and Mark S. Young. "Driver behaviour with adaptive cruise control." *Ergonomics* 48.10 (2005): 1294-1313.
- [14] Parasuraman, Raja, and Victor Riley. "Humans and automation: Use, misuse, disuse, abuse." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 39.2 (1997): 230-253.
- [15] Parasuraman, Raja, Mustapha Mouloua, and Robert Molloy. "Effects of adaptive task allocation on monitoring of automated systems." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 38.4 (1996): 665-679.
- [16] Hatipoglu, Cem, Umit Ozguner, and Keith A. Redmill. "Automated lane change controller design." *Intelligent Transportation Systems, IEEE Transactions on* 4.1 (2003): 13-22.
- [17] Cerone, Vito, Mario Milanese, and Diego Regruto. "Combined automatic lane-keeping and driver's steering through a 2-DOF control strategy." *Control Systems Technology, IEEE Transactions on* 17.1 (2009): 135-142.
- [18] Lygeros, John, Claire Tomlin, and Shankar Sastry. "Hybrid systems: modeling, analysis and control." *preprint* (1999).
- [19] Lygeros, John, et al. "Continuity and invariance in hybrid automata." *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on.* Vol. 1. IEEE, 2001.
- [20] B. De Schutter, W.P.M.H. Heemels, J. Lunze, and C. Prieur, "Survey of modeling, analysis, and control of hybrid systems," Chapter 2 in *Handbook of Hybrid Systems Control - Theory, Tools, Applications* (J. Lunze and F. Lamnabhi-Lagarrigue, eds.), Cambridge, UK: Cambridge University Press, ISBN 978-0-521-76505-3, pp. 31-55, 2009.
- [21] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. In *Proceedings of the 27th Annual Symposium on the Theory of Computing*, STOC'95, pages 373-382. ACM Press, 1995.
- [22] V.D. Blondel and J.N. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35(3):479-490, 1999.
- [23] Tavernini, L. (1987): "Differential automata and their discrete simulators." *Nonlinear Analysis, Theory, Methods and Applications*, 11:6, pp. 665-683.
- [24] Branicky, M. (1995): *Studies in Hybrid Systems: Modeling, Analysis, and Control.* PhD thesis, LIDS, Massachusetts Institute of Technology.

- [25] Stiver, J. and P. Antsaklis (1992): “Modeling and analysis of hybrid control systems.” *In Conference on Decision and Control*. Tucson, Arizona, USA.
- [26] Peleties, P. and R. DeCarlo (1989): “A modeling strategy with event structures for hybrid systems.” In Proc. 28th CDC, pp. 13081313. Tampa, FL.
- [27] Nerode, A. and W. Kohn (1992): “Models for hybrid systems: automata, topologies, stability” Technical Report. Mathematical Sciences Institute, Cornell University.
- [28] Malmborg, Jürgen. “Analysis and design of hybrid control systems.” *Department of Automatic Control Lund Institute of Technology, Sweden* (1998)
- [29] Johansson, Karl Henrik, John Lygeros, and Shankar Sastry. “Modeling of hybrid systems.” *Control Systems, Robotics and Automation, from Encyclopedia of Life Support Systems (EOLSS)* (2002).
- [30] Nesterov, Yurii, Arkadii Nemirovskii, and Yinyu Ye. *Interior-point polynomial algorithms in convex programming*. Vol. 13. Philadelphia: Society for Industrial and Applied Mathematics, 1994.
- [31] A. I. Luře. Some Nonlinear Problems in the Theory of Automatic Control. H.M. Stationery Off., London, 1957. In Russian, 1951.
- [32] V. A. Yakubovich. The method of matrix inequalities in the stability theory of nonlinear control systems, I, II, III. *Automation and Remote Control*, 25-26(4):905-917, 577-592, 753-763, April 1967.
- [33] El Ghaoui, Laurent, Eric Feron, and Venkataraman Balakrishnan. *Linear matrix inequalities in system and control theory*. Vol. 15. Philadelphia: Society for industrial and applied mathematics, 1994.
- [34] Scherer, Carsten, and Siep Weiland. “Linear matrix inequalities in control.” Lecture Notes, *Dutch Institute for Systems and Control, Delft, The Netherlands* (2000).
- [35] Pólik, Imre, and Tamás Terlaky. “A survey of the S-lemma.” *SIAM review* 49.3 (2007): 371-418.
- [36] Lofberg, Johan. “YALMIP: A toolbox for modeling and optimization in MATLAB.” *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004.
- [37] Tomlin, Claire J., et al. “Computational techniques for the verification of hybrid systems.” *Proceedings of the IEEE* 91.7 (2003): 986-1001.
- [38] Henzinger, Thomas A. *The theory of hybrid automata*. Springer Berlin Heidelberg, 2000.
- [39] Henzinger, Thomas A., Pei-Hsin Ho, and Howard Wong-Toi. “HyTech: A model checker for hybrid systems.” *Computer aided verification*. Springer Berlin Heidelberg, 1997.
- [40] Frehse, Goran. “PHAVer: Algorithmic verification of hybrid systems past HyTech.” *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2005. 258-273.

- [41] Ratschan, Stefan, and Zhikun She. "Safety verification of hybrid systems by constraint propagation based abstraction refinement." *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2005. 573-589.
- [42] Donzé, Alexandre. "Breach, a toolbox for verification and parameter synthesis of hybrid systems." *Computer Aided Verification*. Springer Berlin Heidelberg, 2010.
- [43] R. Serban and A. C. Hindmarsh. Cvodes: the sensitivity-enabled ode solver in sundials. In *Proceedings of IDETC/CIE 2005*, Long Beach, CA., Sept.2005.
- [44] Passerone, Roberto, and Alessandro Pinto. *Languages and tools for hybrid systems design*. Vol. 1. No. 1-2. now Publishers Inc, 2006.
- [45] Girard, Antoine, and George J. Pappas. "Verification using simulation." *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2006. 272-286.
- [46] Pnueli, Amir. "The temporal logic of programs." *Foundations of Computer Science, 1977., 18th Annual Symposium on. IEEE*, 1977.
- [47] Donzé, A., Maler, O., Bartocci, E., Nickovic, D., Grosu, R., & Smolka, S. (2012). On temporal logic and signal processing. In *Automated Technology for Verification and Analysis* (pp. 92-106). Springer Berlin Heidelberg.
- [48] Alur, Rajeev, Tomás Feder, and Thomas A. Henzinger. *The benefits of relaxing punctuality.* *Journal of the ACM (JACM)* 43.1 (1996): 116-146.
- [49] Donzé, Alexandre, and Oded Maler. *Robust satisfaction of temporal logic over real-valued signals*. Springer Berlin Heidelberg, 2010.
- [50] Donzé, Alexandre, Thomas Ferrere, and Oded Maler. "Efficient robust monitoring for STL." *Computer Aided Verification*. Springer Berlin Heidelberg, 2013.
- [51] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, p. 231:274, July 1987
- [52] Pacejka, Hans. *Tire and vehicle dynamics*. Elsevier, 2005.
- [53] Weir, David H., and Duane T. McRuer. "Dynamics of driver vehicle steering control" *Automatica* 6.1 (1970): 87-98.
- [54] UNECE- United Nations Economic Commission for Europe, Transport Programme, Electronic Stability Control Systems Regulation, Annex. 9, E/ECE/324, E/ECE/-TRANS/505, Regulation no.13-H, p. 3, November 2009
- [55] Lee, John D. "Fifty years of driving safety research." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 50.3 (2008): 521-528.
- [56] Hedrick, J. Karl, et al. "Longitudinal vehicle controller design for IVHS systems." *American Control Conference*, 1991. IEEE, 1991.
- [57] Veldhuyzen, Wim, and Henk G. Stassen. "The internal model concept: An application to modeling human control of large ships." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 19.4 (1977): 367-380.

- [58] McRuer, Duane. "Human dynamics in man-machine systems." *Automatica* 16.3 (1980): 237-253.
- [59] Kristiansson, Birgitta, and Bengt Lennartson. "Robust tuning of PI and PID controllers: using derivative action despite sensor noise." *Control Systems, IEEE* 26.1 (2006): 55-69.
- [60] Vahidi, Ardalan, and Azim Eskandarian. "Research advances in intelligent collision avoidance and adaptive cruise control." *Intelligent Transportation Systems, IEEE Transactions on* 4.3 (2003): 143-153. 2
- [61] Shladover, Steven E. "Review of the state of development of advanced vehicle control systems (AVCS)." *Vehicle System Dynamics* 24.6-7 (1995): 551-595.
- [62] Yanakiev, Diana, and Ioannis Kanellakopoulos. "A simplified framework for string stability analysis in AHS" *Proceedings of the 13th IFAC World Congress*. 1996
- [63] Rajamani, Rajesh. *Electronic Stability Control*. Springer US, 2012.
- [64] M. Kondo, A. Ajimine, Drivers' sight point and dynamics of the driver-vehicle-system related to it, SAE technical paper, 680104, 1968
- [65] D. A. Abbink, Neuromuscular Analysis of Haptic Gas Pedal Feedback during Car Following, Ph.D. Thesis, Faculty of Mechanical Engineering, Delft University of Technology, 2006.
- [66] Katzourakis, D. *Driver steering support interfaces near the vehicle's handling limits*. TU Delft, Delft University of Technology, 2012.
- [67] Schouten, Alfred C., Winfred Mugge, and Frans CT van der Helm. "NMClab, a model to assess the contributions of muscle visco-elasticity and afferent feedback to joint dynamics." *Journal of biomechanics* 41.8 (2008): 1659-1667.
- [68] Pick, Andrew J., and David J. Cole. "A mathematical model of driver steering control including neuromuscular dynamics." *Journal of dynamic systems, measurement, and control* 130.3 (2008): 031004.
- [69] Macadam, Charles C. "Understanding and modeling the human driver." *Vehicle System Dynamics* 40.1-3 (2003): 101-134.
- [70] Shorten, Robert, et al. "Stability criteria for switched and hybrid systems." *SIAM review* 49.4 (2007): 545-592.
- [71] Peleties, Philippos, and Raymond DeCarlo. "Asymptotic stability of m-switched systems using Lyapunov-like functions." *American Control Conference*, 1991. IEEE, 1991.
- [72] Hespanha, Joao P., and A. Stephen Morse. "Stability of switched systems with average dwell-time." *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*. Vol. 3. IEEE, 1999.
- [73] Branicky, Michael S. "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems." *Automatic Control, IEEE Transactions on* 43.4 (1998): 475-482.

- [74] Liberzon, Daniel, and A. Stephen Morse.“Basic problems in stability and design of switched systems.” *Control Systems, IEEE* 19.5 (1999): 59-70.
- [75] Rantzer, Anders. “A dual to Lyapunov’s stability theorem.” *Systems & Control Letters* 42.3 (2001): 161-168.
- [76] Green, Marc. “How Long Does It Take to Stop?” Methodological Analysis of Driver Perception-Brake Times.” *Transportation human factors* 2.3 (2000): 195-216
- [77] Morse, A. Stephen. “Supervisory control of families of linear set-point controllers Part I. Exact matching.” *Automatic Control, IEEE Transactions on* 41.10 (1996): 1413-1431.
- [78] Antsaklis, Panos, ed. *Hybrid systems II*. Vol. 2. Springer, 1995
- [79] Zhai, Guisheng, et al. “Stability analysis of switched systems with stable and unstable subsystems: an average dwell time approach.” *International Journal of Systems Science* 32.8 (2001): 1055-1061.
- [80] Donzé, Alexandre, Gilles Clermont, and Christopher J. Langmead. “Parameter synthesis in nonlinear dynamical systems: Application to systems biology.” *Journal of Computational Biology* 17.3 (2010): 325-336.
- [81] Swaroop, D., and J. K. Hedrick.“String stability of interconnected systems.” *Automatic Control, IEEE Transactions on* 41.3 (1996): 349-357.
- [82] Liang, Chi-Ying, and Huei Peng. “Optimal adaptive cruise control with guaranteed string stability.” *Vehicle System Dynamics* 32.4-5 (1999): 313-330.
- [83] Sharp, Robin S. “Driver steering control and a new perspective on car handling qualities.” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 219.10 (2005): 1041-1051.
- [84] Borrelli, Francesco, et al. “MPC-based approach to active steering for autonomous vehicle systems.” *International Journal of Vehicle Autonomous Systems* 3.2 (2005): 265-291.
- [85] Zhai, Guisheng, et al. “Piecewise Lyapunov functions for switched systems with average dwell time.” *Asian Journal of Control* 2.3 (2000): 192-197.
- [86] Michon, John A. *A critical view of driver behavior models: what do we know, what should we do?*. Springer US, 1986.
- [87] Jürgensohn, Thomas. “Control theory models of the driver.” *Modelling driver behaviour in automotive environments*. Springer London, 2007. 277-292.
- [88] MacAdam, Charles C. “Application of an optimal preview control for simulation of closed-loop automobile driving.”(1981).