Machine learning control of dynamical systems in electric and autonomous vehicles

Marc-Antoine Beaudoin

Department of Electrical and Computer Engineering
McGill University, Montreal

A thesis submitted to McGill University
in partial fulfillment of the requirements of the degree of
Doctor of Philosophy

July 2022

To Gilles.[1]

[1]My grandfather Gilles Roy was a car mechanic for all his working years. As the older sibling, he had to start working at his father's garage when his brother went off to engineering school. The abrupt end to Gilles' agronomy studies displeased him—lead soldering cracked radiators in a thick blue haze was not exactly what he aspired to—but he had no choice, such was the custom at the time. Two generations later, as an older sibling myself, I am pursuing doctoral studies and hoping to advance the state-of-the-art in automotive engineering. I believe this contrast is due in no small part to the technological progress our society benefited from since then. May this story be a testimony to the importance of research work, and hopefully, give a glimpse of motivation to my fellow graduate students. May this also be a reminder of the chance bestowed on us.

Abstract

Machine learning algorithms can enhance the control of uncertain dynamical systems by learning dynamical models and tuning controller parameters. However, electric and autonomous vehicles present important challenges. First, the need to operate a physical system in order to collect data and iterate the control parameters leads to efficiency requirements for the learning algorithm. Also, vehicles must be kept safe at all time during operation, which requires to maintain criteria of robust stability and state constraint satisfaction during learning. This thesis presents learning methods that address these challenges. First, it is shown that with a proper choice of algorithm, a gearshift controller can be tuned from reinforcement learning with very few gearshift trials. The method could accelerate the development process for new multi-speed transmissions for electric drivetrains. Also, the thesis presents a new method to synthesize a linear controller from a learned model of arbitrary type, while preserving robust stability guarantees. The method is demonstrated by synthesizing a controller for autonomous vehicular maneuvers. Finally, the thesis presents the safe uncertainty learning principle. This principle introduces necessary conditions such that, while the dynamics of an uncertain system are being learned, state constraints are always enforced despite the modeling uncertainty. This research suggests that machine learning control can be employed to accelerate the development process of new electric drivetrains, and to enhance vehicular control while preserving safety guarantees.

Sommaire

Les algorithmes d'apprentissage automatique peuvent améliorer le contrôle de systèmes dynamiques incertains en apprenant des modèles dynamiques et en ajustant des paramètres de contrôle. Cependant, les véhicules électriques et autonomes présentent des défis importants. Tout d'abord, le besoin d'opérer un système physique afin de collecter des données et itérer les paramètres engendre un requis d'efficacité pour l'algorithme d'apprentissage. De plus, les véhicules doivent demeurer en sécurité tout au long de l'opération, ce qui requiert de maintenir des critères de stabilité robuste et de satisfaction de contraintes d'état pendant l'apprentissage. Cette thèse présente des méthodes d'apprentissage qui répondent à ces défis. En premier lieu, il est montré qu'avec un choix approprié d'algorithme, un contrôleur de changement de vitesse peut être ajusté avec l'apprentissage par renforcement, et ce, en nécessitant seulement quelques essais de changement de vitesse. Cette méthode peut accélérer le développement de nouvelles transmissions multi-vitesses pour véhicules électriques. Aussi, cette thèse présente une nouvelle méthode pour synthétiser un contrôleur linéaire à partir d'un modèle appris de la dynamique d'un système, tout en préservant des garanties de stabilité robuste. Cette méthode est démontrée en synthétisant un contrôleur pour des manoeuvres de véhicules autonomes. Finalement, la thèse présente le principe d'apprentissage sécuritaire de l'incertitude. Ce principe introduit des conditions nécessaires tel que, pendant que la dynamique d'un système incertain est apprise, les contraintes d'état sont respectées malgré l'incertitude du modèle. Cette recherche suggère que le contrôle par apprentissage automatique peut être utilisé afin d'accélérer le développement de véhicules électriques, ainsi que d'améliorer le contrôle de véhicule autonomes tout en préservant des garanties de sécurité.

Acknowledgments

I would like to thank my supervisor, Prof. Benoit Boulet, for his invaluable support in my studies. Thank you for all your comments, suggestions, and contagious enthusiasm toward this research. I learned a lot from you and am grateful for it.

This research was financially supported by Quebec's Fonds de recherche Nature et technologies, as well as a Mitacs Accelerate Fellowship. This fellowship started in partnership with Nordresa—an electric vehicle startup company whom I practiced engineering for prior to starting my doctoral studies. It was then continued with Dana Inc. following their acquisition of Nordresa. Thank you, Sylvain Castonguay, for making this fellowship possible.

As this thesis culminates my engineering education, I would like to thank the friends I made during my studies in Sherbrooke, Toronto, and Montreal. Thank you for the numerous learning opportunities and all the fun.

I would like to thank my parents, Sylvie and Jacques, for their support throughout my degrees. Lastly, I would like to thank my wife, Krysta, for everything.

Contents

Introduction **1**
Thesis structure . . . 3
Claims of originality . . . 3
Contributions of authors . . . 4
Notation . . . 5

I Development of electric vehicle drivetrains 6

1 Multi-speed transmissions **7**
1.1 Review of powertrain architectures . . . 7
1.2 Justification for multi-speed transmissions . . . 9
1.3 Vehicle and driveline models . . . 10
1.4 Transmission models . . . 12
1.4.1 Parallel shaft architecture with two frictional clutches . . . 13
1.4.2 Parallel shaft architecture with a one-way clutch . . . 13
1.4.3 Planetary gearset architectures . . . 14
1.5 New multi-speed transmission design . . . 16

2 Uninterrupted gearshifts **19**
2.1 Limits to uninterrupted gearshifts . . . 19
2.2 Example gearshift trajectories . . . 22
2.3 Theorems for fundamental limitations . . . 27
2.3.1 Theorems for a dual-clutch architecture . . . 27
2.3.2 Adaptations for planetary gearset architectures . . . 33
2.3.3 Other vehicle models and kinematic conditions . . . 35

3 Learning gearshift controllers **37**
3.1 Motivations for gearshift controller learning . . . 37
3.1.1 Review of controller designs . . . 38
3.1.2 Review of controller parameter learning . . . 39
3.2 Experimental setup . . . 41
3.3 Gearshift controller design . . . 44
3.4 Proposed learning method . . . 47
3.4.1 Gaussian process regression . . . 49

3.4.2 Simulated policy rollouts and uncertainty propagation 50
3.4.3 Cost function and its gradients 52
3.5 Experimental results 53

II Safety guarantees for autonomous vehicles 58

4 Autonomous vehicles 59
4.1 Functional architecture of autonomous vehicles 59
4.2 Vehicle models 61
4.2.1 Pure longitudinal vehicle dynamics 61
4.2.2 Combined longitudinal and lateral vehicle dynamics 62
4.2.3 Pure lateral vehicle dynamics 64
4.3 Safe control of vehicle dynamics with learning 65

5 Learning-based synthesis of robust controllers 66
5.1 Review of learned robustly stable controllers 67
5.2 Robust control framework 70
5.3 Controller learning method 70
5.3.1 Performance gradients 72
5.3.2 Nominal stability gradients 73
5.3.3 Robust stability gradients 74
5.4 Example problem 75
5.4.1 Uncertainty cover 75
5.4.2 Trajectory and nominal command 77
5.4.3 Controller structure and initialization 79
5.4.4 Controller training 79
5.5 Results and discussion 80
5.5.1 Augmented cost 83

6 Structured learning for state constraints 84
6.1 Review of safety guarantees for state constraints 85
6.2 Safe uncertainty-learning principle 86
6.3 Learned safety condition 86
6.3.1 Robust safety condition from a control barrier function 87
6.3.2 Robustly safe controller 89
6.3.3 Learning parametric uncertainty 90
6.4 Example 1: vehicle lateral control 91
6.4.1 System model and safety condition 91
6.4.2 Results and discussion 94
6.5 Example 2: vehicle longitudinal control 96
6.5.1 System model and safety condition 96
6.5.2 Results and discussion 99
6.6 Discussion 101

Conclusion **104**
Results and contributions 104
Research findings 105
Future work 107

A Introduction to machine learning **108**

B Equations of motion for a planetary gearset **111**
B.1 Complete system model 111
B.2 Reduced system model 113

C Analytical gradients for controller learning **115**
C.1 Gradients for the kernel function 115
C.2 Gradients for the state distribution 116
C.3 Gradients for the cost function 119

D Partial derivatives for the lateral dynamics **120**

Bibliography **122**

Introduction

The transition towards electric transportation is slowly gaining momentum in Canada. Table 1 shows the proportion of new motor vehicle registrations that are battery electric in the light vehicle category—i.e., passenger cars, pickup trucks, vans, and multi-purpose vehicles [1]. The promise of electric vehicles is to reduce greenhouse gas emissions and alleviate climate change. This is important given that in Canada, the transportation sector is responsible for 25% of all greenhouse gas emissions, which amounts to 186 Mt of CO_2 equivalent [2]. Of these emissions, passenger transportation by cars and light trucks is responsible for 99 Mt, and freight transportation by heavy trucks, 65 Mt. As a technology, the electric drivetrain is still relatively new. The development of more efficient drivetrains could help accelerate the adoption of electric vehicles. Particularly important to heavy duty trucks are multi-speed transmissions. Part of developing a new transmission involves choosing its architecture, which influences whether the vehicle will be capable of uninterrupted gearshifts. It also involves synthesizing a gearshift controller.

In parallel, several companies are developing autonomous vehicles [3]. Some automakers already offer advanced driver assistance, and are gradually expanding the capabilities of these systems. Other companies aim directly at fully automated driving. Regardless of the approach, the promise of autonomous driving is to reduce the number of road collisions. In Canada, vehicle collisions cause 1800 fatalities every year [4]. Driver distraction is a contributing factor in 22% of these collisions, speed, 23%, and driving under the influence, 15%. Despite the challenges of developing autonomous vehicles [5], the potential for safety they represent is very appealing [6]. In addition to perceiving the environment and planning their trajectory, autonomous vehicles must accurately control their dynamics.

Table 1: Portion of battery electric vehicles in Canadian new-vehicle registrations.

Year	2017	2018	2019	2020	2021
Proportion [%]	0.4	1.1	1.8	2.5	3.3

Meanwhile, machine learning is becoming a popular tool in several scientific and engineering fields such as materials science [7], computer vision [8], machine fault detection [9], and process control [10]. The Appendix A presents a brief introduction to machine learning for the interested reader. In the field of control engineering, machine learning can be used to obtain better performing controllers. In effect, most controller synthesis methods rely on a model of the system dynamics, which inevitably differs from the actual system dynamics. If this modeling error is too large, the closed-loop system may become unsafe. While robust control [11] offers a principled way to account for modeling uncertainty and maintain robust stability, a large uncertainty necessitates a sacrifice in performance [12]. Thus, using machine learning to learn the actual system dynamics and reduce the modeling error is a promising approach. This approach can be seen as an extension of system identification [13], with the main distinction being that the models used in machine learning control can be more sophisticated—e.g., neural networks [14] and Gaussian processes [15]. Moreover, reinforcement learning can provide an alternative to traditional controller synthesis methods. With reinforcement learning, the controller parameters are learned iteratively as the system is being used and data gets collected. In this context, machine learning control can also be seen as a generalization of adaptive control [16].

Despite the potential benefits, the use of machine learning control in the context of electric and autonomous vehicles also raises concerns. First, learning for control has to accelerate the development of new vehicles. In other words, it should be faster to obtain controllers with learning than without it. High profile accomplishments [17] already showed that deep reinforcement learning can achieve impressive results when provided with enough data, computational power, and training time. For the control of physical systems however, the data collection is rate-limited by the need for physical experiments. Also, a method that is very computationally expensive may be deemed impractical. Therefore, the machine learning control of physical systems has an additional efficiency requirement.

Second, the control of physical systems typically requires safety guarantees, both in terms of robust stability and state constraint satisfaction. However, most theoretical guarantees that originate from control theory are incompatible with the type of learned models that are typical of modern machine learning. Therefore, the machine learning control of physical systems also requires the adaptation of safety guarantees from control theory.

This thesis investigates the following two research questions.

1. Can machine learning control accelerate the development of electric drivetrains?
2. Can machine learning control preserve safety guarantees for vehicular control?

Thesis structure

Part I relates to the first research question. It concerns the use of learning for control to accelerate the development of a multi-speed transmission for an electric vehicle. Chapter 1 reviews transmission architectures and introduces corresponding dynamical models. It also presents a new transmission design. Chapter 2 describes the fundamental limitations to uninterrupted gearshifts. In effect, not all transmission architectures are capable of such gearshifts, and motor saturation can be a significant source of limitation. This is important to consider during the design process of a new drivetrain, but also during the learning of a gearshift controller. Chapter 3 presents a method to learn a gearshift controller from experience, i.e., by performing gearshift trials on a test bench. The experimental setup consists of a transmission prototype based on the new design introduced in Chapter 1. The learning method focuses on reducing the number of gearshift trials required to tune the controller, thereby addressing the efficiency concerns associated with machine learning control.

Part II relates to the second research question. It concerns methods to preserve safety guarantees when learning to control vehicular maneuvers. Chapter 4 reviews the typical functional architecture of an autonomous vehicle. It also introduces models for the longitudinal and lateral vehicle dynamics. Chapter 5 presents a new method to learn a linear controller from experience. The method is compatible with any learned dynamical model, and it preserves robust stability guarantees. It is used to synthesize a controller for a lane change with concurrent vehicle acceleration. Chapter 6 proposes the safe uncertainty-learning principle. This principle suggests requirements for learning a safety condition to keep a dynamical system within state constraints. The principle is supported by examples of both longitudinal and lateral control. Together Chapters 5 and 6 address the concerns of maintaining safety guarantees for robust stability and state constraint satisfaction when learning vehicular control.

Claims of originality

The theorems for the fundamental limitations to uninterrupted gearshifts of Chapter 2 are an original contribution; they are published in [18]. It is the first time that the gearshift limitations resulting from electric motor saturation are explicitly defined.

Another contribution is demonstrating the efficacy of a model-based reinforcement learning algorithm to tune a gearshift controller from a physical prototype, as presented in Chapter 3. The method was published in [19]. It is the first time that a learning algorithm is used to tune both the feedforward and feedback components of a gearshift controller. The algorithm itself is not entirely new—it is adapted from the PILCO algorithm [20]. Nonetheless, our adaptation of the algorithm provides additional flexibility due to the use of automatic gradients. Also, the analytical gradients of Appendix C are new results, and can be used to verify the proper implementation of the automatic gradients.

The learning-based method to synthesize a robust linear controller presented in Chapter 5 is a contribution. It is the first time that a learning algorithm can tune a linear controller from any type of learned model while preserving robust stability. For this contribution, a manuscript was accepted for publication [21].

The safe uncertainty-learning principle of Chapter 6 is also a contribution. The principle suggests conditions such that an uncertain dynamical system is kept within state constraints during learning. It is the first time that such conditions are explicitly stated. The principle is introduced in a manuscript accepted for publication [22].

Finally, the transmission design of Chapter 1 could be considered an original contribution, as it is the object of a patent application [23]. The clever arrangement of the braking and locking mechanisms makes it possible to obtain three gear ratios with only two modulated braking elements, thereby saving cost on the assembly. Moreover, the use of locking elements in parallel to the braking elements allows to deactivate the braking elements when not shifting in order to save energy. This combination of features was never found in the literature. However, this thesis makes no attempt at justifying the superiority of this design over others, hence the reserved claim for this contribution.

Contributions of authors

The four papers [18, 19, 21, 22] supporting this thesis have Mr. Beaudoin as a first author and Prof. Boulet as the second (and last) author. For all the work presented in this thesis, Mr. Beaudoin can be credited for the research planning and execution, while Prof. Boulet provided guidance and suggestions for improving the research work.

Notation

The notation devised for this thesis aims to harmonize the conventions of different research fields, namely mechanics, control theory, probability theory, and machine learning. While the variables may change meaning between chapters, the notation remains consistent throughout.

In general, scalar variables are italicized lower-case letters (a_f), vectors are bold lower-case letters ($\mathbf{x}$), and matrices are italicized upper-case letters (A_d). There exist a few exceptions, however. Notably, $F_\square$, $T_\square$, and $I_\square$ typically refer to the scalar magnitude of a force, a torque, and a mass moment of inertia, respectively; see Figure 1.2 for instance. Note that I without a subscript is always the identity matrix. For the free-body diagrams of Figure 4.3, upper-case letters are used to contrast the inertial frame of reference from the non-inertial one. In Chapters 3 and 5, J^π and J are scalar-valued cost functions, following the traditions of optimization and machine learning.

The notations $[\mathbf{x}]_i$ and $[A]_{ij}$ are used to denote the i and (i,j) entries of a vector $\mathbf{x}$ and a matrix A. The notation $\mathbf{x}_{[i]}$ is used to denote a vector $\mathbf{x}$ with a particular index i. This can be used in the context of an indexed dataset $\mathcal{D} = \{\mathbf{x}_{[1]}, ..., \mathbf{x}_{[N]}\}$, or to represent state vectors $\mathbf{x}_{[t]}$ in discrete time. Higher-order tensors are denoted as A_{ijk}, where i, j, and k are the tensor dimensions. The Einstein summation convention is used for tensor manipulations. For instance, a simple matrix multiplication would be denoted $C_{ik} = A_{ij}B_{jk}$. It is always assumed that the indices in the resulting tensor are arranged in alphabetical order. For instance, the product $A_{ik}B_{jl}$ would yield a fourth order tensor C_{ijkl}.

In Chapter 3, state vectors $\mathbf{x}_{[t]}$ are sometimes stochastic variables. Assuming $p(\mathbf{x}_{[t]})$ follows a Gaussian distribution, $\boldsymbol{\mu}^{\mathbf{x}}_{[t]}$ represents the mean of the probability distribution, and $\Sigma^{\mathbf{x}}_{[t]}$, its covariance matrix. In other words, $\mathbf{x}_{[t]} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{x}}_{[t]}, \Sigma^{\mathbf{x}}_{[t]})$.

In Chapter 5, $y(s)$ refers to a signal in the frequency domain. This signal may be multi-dimensional despite the use of a non-bold letter. Similarly, $G(s)$ represent a linear time-invariant system, not a matrix. Note that using a partition within square brackets confers a particular meaning. Take for example

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}, \text{ and } G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

In the first case, M is a matrix composed of smaller matrices M_1 to M_4 with matching dimensions. In the second case, the expression is a shorthand for the state-space realization $C(sI + A)^{-1}B + D$ of the system $G(s)$.

The rest of the notation should be clear from context.

Part I

Development of electric vehicle drivetrains

Chapter 1

Multi-speed transmissions

This chapter reviews common electric powertrain architectures, justifies the use of a multi-speed transmissions, presents dynamical models for common transmission types, and presents a new transmission design.

Contents

1.1 Review of powertrain architectures **7**

1.2 Justification for multi-speed transmissions **9**

1.3 Vehicle and driveline models **10**

1.4 Transmission models **12**

1.4.1 Parallel shaft architecture with two frictional clutches 13

1.4.2 Parallel shaft architecture with a one-way clutch 13

1.4.3 Planetary gearset architectures 14

1.5 New multi-speed transmission design **16**

1.1 Review of powertrain architectures

The literature abounds with electric powertrain concepts, each aimed at different client needs [24–26]. Typical design specifications for electric powertrains include energy efficiency, drivability, and cost. The efficiency is important as it directly influences the driving range. Drivability is a subjective measure of the vehicle's response to driver inputs, and is typically evaluated for specific maneuvers. For a tip-in maneuver—i.e., when a driver suddenly requests a vehicle acceleration—oscillations in the acceleration is generally unappreciated [27]. For gearshifts, it is important to minimize torque interruptions and vehicle jerk [28].

Perhaps the simplest architecture is using a single motor and a fixed reduction ratio between the motor and the wheels. This concept has an excellent drivability, but it introduces significant drawbacks on the vehicle design: to meet vehicular performance specifications, the motor is often oversized, and the resulting powertrain only seldom operates in its optimal efficiency region. This becomes especially problematic for heavier vehicles, as will be shown in Section 1.2.

A natural evolution of the single-motor fixed-ratio concept is the introduction of a multi-speed transmission. A simple concept is the manual transmission [29, 30]. It consists of mounting gears on bearings, and selectively locking different gears to their transmission shafts to achieve different transmission ratios. Because electric motors do not need to idle, it is possible to use a manual transmission without a clutch between the motor and the transmission. To shift gears, the motor torque is first reduced to zero, then the first gear is disengaged, the motor speed is synchronized with the second gear, the second gear is engaged, and finally the motor driving torque is reapplied. Synchronizers may also help with the shaft synchronization and gear engagement [31–34]. If properly performed, such a gearshift can have a low jerk level, but a torque gap is inevitable, as the shifting elements have to be engaged and disengaged when no torque is passed through them. To reduce this torque gap, we can introduce a torque gap filler in the transmission architecture [35, 36]. This is typically a clutch placed between the motor and the transmission output shaft; it is used only during gearshifts.

Alternatives to manual transmissions used in electric vehicles are dual-clutch transmissions [37–39] and automatic transmissions [40–43]. Conceptually, they are almost equivalent. Both consist of offering clutching and braking devices that can be modulated such that the transmission's torque can be continuously transferred between different transmission paths. The distinction resides in that dual-clutch transmissions typically use a parallel shaft architecture and only require two clutches, while automatic transmissions typically use a planetary gearset architecture and require more clutching and braking devices if more than two gear ratios are to be offered. In both parallel shaft and planetary architectures, it can be interesting to replace a friction clutch by a one-way clutch [37, 41, 44]. A one-way clutch that transfers the transmission torque in a given ratio will automatically disengage when a friction clutch of a higher gear ratio is engaged. This automatic disengagement may also simplify the gearshift control algorithm, as one fewer clutch needs to be controlled. This concept also allows to continuously transfer the torque between the two transmission paths, but with the added benefits that a one-way clutch is cheaper and more compact than a friction clutch or a brake.

Instead of using a multi-speed transmission, one could circumvent the drawbacks of a single-motor fixed-ratio powertrain by using a plurality of motors. The different motors can be mounted on different axles, where the driving torque is shared through the road. Each motor can either power a single wheel [45] or a front or rear axle [46]. Alternatively, the different motors can also be mounted on the same axle [47]. These architectures allow the driving torque to be continuously transferred from one motor to another, thus providing excellent drivability. This is also true for multi-motor architectures with multi-speed transmissions. For instance, a planetary gearset architecture can be configured to receive inputs from two motors [48–51]. These transmission architectures are conceptually indistinguishable from power-split transmissions used in hybrid electric vehicles [52]. Alternatively, a parallel shaft architecture can be configured to receive inputs from two motors [53–55]. Such a powertrain is capable of perfectly smooth gearshifts: if the driving torque is taken exclusively from one of the two motors, the other motor transmits no torque, which allows for an easy gear change on some transmission shafts. However, a torque gap may still exist, as the torque on one motor has to be reduced to zero and the other motor may not be able to fully compensate this torque decrease.

Finally, electric powertrains can also include mechanical continuously-variable transmissions [56]. These powertrains provide excellent drivability, as the transmission ratio can be smoothly varied. However, friction losses may lower the powertrain energy efficiency, which is undesirable due to the high cost of energy storage in battery electric vehicles [57, 58].

1.2 Justification for multi-speed transmissions

For certain vehicles, a multi-speed transmission may offer the best tradeoff between the conflicting requirements of efficiency, drivability, and cost. The performance benefits of having more than one gear ratio can only be determined with a vehicle-level analysis. The promise is to compensate for the additional cost, volume, and complexity of the transmission by allowing the use of a smaller motor, and operating the motor on a higher efficiency region for a larger fraction of the time, which can reduce the battery size, and thereby save on overall vehicle cost [59]. When compared to a fixed ratio drivetrain, a multi-speed transmission can lower the energy consumption on standard drive cycles by over 20%.

The potential benefits of having multiple gear ratios can be seen through an example motor selection process. The vehicle hypothesized is a commercial vehicle with a gross ve-

Table 1.1: Design specifications considered in the motor selection.

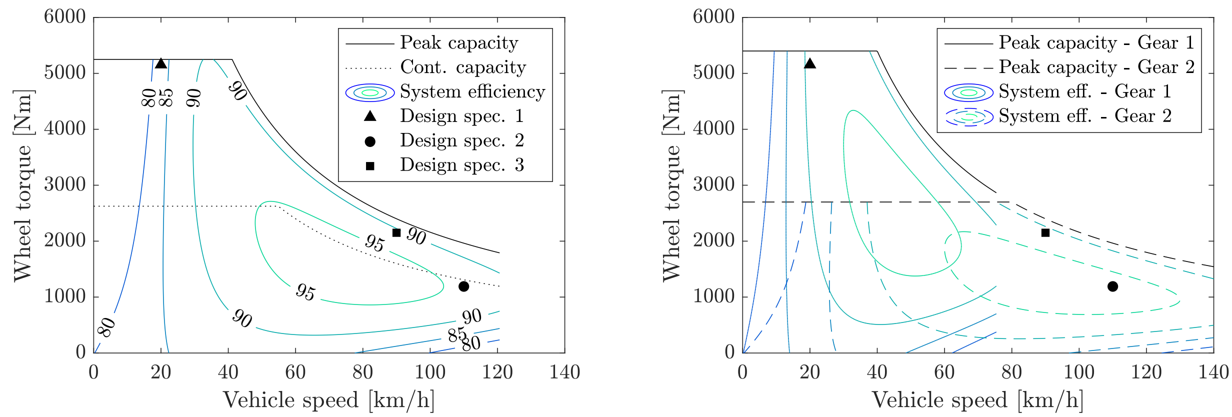
Design specification	v [km/h]	α [%]	Duration
1: extreme grade	20	20	$< 1\,\mathrm{min}$
2: highway, cruise speed	110	0	Continuous
3: highway, high grade	90	5	$< 1\,\mathrm{min}$

hicle mass of 8500 kg. It would be classified as an N2 commercial vehicle in Europe, and a Class 5 medium-duty truck in North America. The gross vehicle mass of 8500 kg is used for the motor selection process. The analysis also considers a frontal area of $6\,\mathrm{m/s^2}$ and a drag coefficient of 0.7. The three design specifications of Table 1.1 define the vehicle performance requirements. The first specification consists of an extreme grade—it sets the maximal torque requirement. This is a short duration event, so the powertrain is allowed to operate above its continuous capacity limit. The second specification consists of the vehicle cruising on the highway, which sets both a wheel speed requirement and a continuous power requirement. The third specification happens when the vehicle climbs a steep but reasonable grade on the highway, which sets the maximal power requirement.

As shown in Figure 1.1a, if the vehicle is equipped with a single-speed transmission, the vehicle requirements can be met with a 200 kW motor with 700 Nm peak torque and 8000 rpm maximum speed, using a fixed total reduction ratio of 7.5. With a two-speed transmission with ratios of 12 and 6, the peak torque requirement can be lowered to 450 Nm, while keeping the same power and speed limit requirements. In practice, this would allow vehicle designers to reduce the motor's active length [60], thereby reducing the motor peak torque capacity while maintaining the same motor power capacity. The resulting system capacity is displayed in Figure 1.1b. The vehicle now has a 36% volumetrically smaller motor and a larger high-efficiency operating region.

1.3 Vehicle and driveline models

This section introduces the vehicle and driveline models used in Chapters 2 and 3; they are shown on Figure 1.2. It is assumed that the vehicle longitudinal speed v follows the driving wheel speed $\dot{\theta}_{\mathrm{v}}$ according to $v = \dot{\theta}_{\mathrm{v}} r_{\mathrm{w}}$, where r_{w} is the wheel radius. In other words, there is no longitudinal slip at the wheel. This allows to project the vehicle mass and the vehicular forces on the wheel coordinate. The vehicle mass m becomes an equivalent rotational inertia $I_{\mathrm{v}} = m r_{\mathrm{w}}^2$. In this model, three vehicular forces are considered: the aerodynamic drag F_{a},



(a) Single speed transmission, ratio of 7.5. Motor requirements: 200 kW power, 700 Nm peak torque, and 8000 rpm maximum speed.

(b) Two-speed transmission, ratios of 12 and 6. Motor requirements: 200 kW power, 450 Nm peak torque, and 8000 rpm maximum speed.

Figure 1.1: Vehicle capacity for (a) a single-speed transmission and (b) a two-speed transmission.

the tire rolling resistance F_t, and gravity F_g.

$$F_\mathrm{v} = F_\mathrm{a} + F_\mathrm{r} + F_\mathrm{g}, \tag{1.1}$$
$$F_\mathrm{a} = \tfrac{1}{2}\rho v^2 a_\mathrm{f} c_\mathrm{d}, \tag{1.2}$$
$$F_\mathrm{t} = mgc_\mathrm{t}\cos(\alpha), \tag{1.3}$$
$$F_\mathrm{g} = mg\sin(\alpha), \tag{1.4}$$

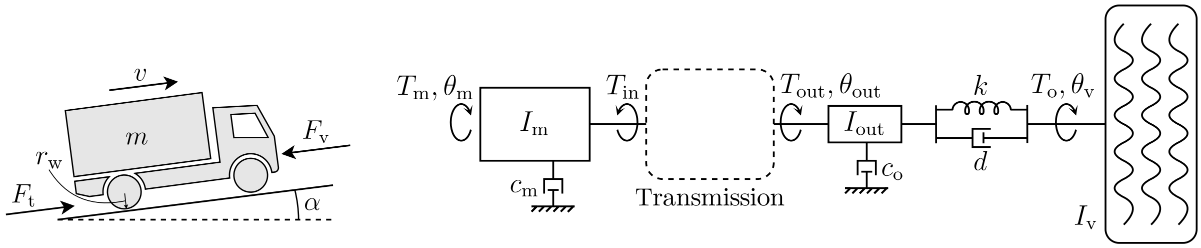
These three forces become an equivalent torque T_v applied to the vehicle wheel

$$T_\mathrm{v} = r_\mathrm{w}(\tfrac{1}{2}\rho v^2 a_\mathrm{f} c_\mathrm{d} + mgc_\mathrm{t}\cos(\alpha) + mg\sin(\alpha)), \tag{1.5}$$

where ρ is the air density, a_f is the vehicle frontal area, c_d is the aerodynamic drag coefficient, c_t is the tire rolling resistance coefficient, g is gravity, and α is the road slope.

The driveline model is shown on Figure 1.2b. It contains three rotating bodies: the electric motor I_m, the transmission output shaft I_out, and the equivalent vehicle inertia I_v. The three equations of motion are

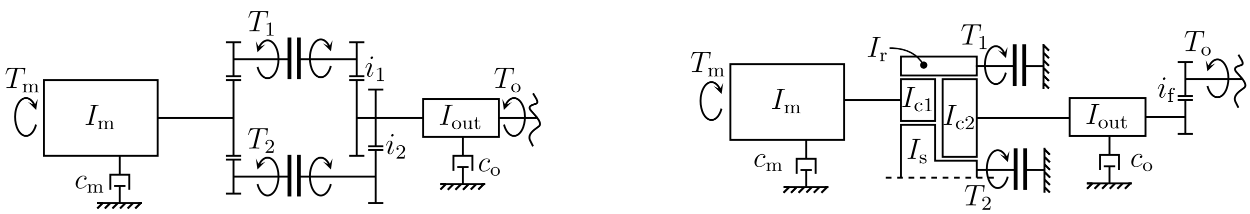
$$I_\mathrm{m}\ddot{\theta}_\mathrm{m} = -c_\mathrm{m}\dot{\theta}_\mathrm{m} + T_\mathrm{m} - T_\mathrm{in}, \tag{1.6}$$
$$I_\mathrm{out}\ddot{\theta}_\mathrm{out} = -c_\mathrm{o}\dot{\theta}_\mathrm{out} + T_\mathrm{out} - k(\theta_\mathrm{out} - \theta_\mathrm{v}) - d(\dot{\theta}_\mathrm{out} - \dot{\theta}_\mathrm{v}), \tag{1.7}$$
$$I_\mathrm{v}\ddot{\theta}_\mathrm{v} = -T_\mathrm{v} + k(\theta_\mathrm{out} - \theta_\mathrm{v}) + d(\dot{\theta}_\mathrm{out} - \dot{\theta}_\mathrm{v}), \tag{1.8}$$



(a) Vehicle model.

(b) Driveline model.

Figure 1.2: Vehicle and driveline models.



(a) Two-speed transmission with parallel shaft architecture. Short name: dual-clutch transmission.

(b) Two-speed transmission with a planetary gearset architecture. Short name: dual-brake transmission [40].

Figure 1.3: Two popular transmission architectures.

where c_m is the coefficient of viscous damping on the motor, and c_o is the coefficient of viscous damping on the transmission output. The coefficients k and d represent lumped driveline stiffness and damping, respectively. They cover phenomena such as driveshaft and tire flexibility and damping. The resistive torque T_v can be obtained from the vehicle model.

1.4 Transmission models

This section introduces dynamical models for three popular transmission architectures. In Chapter 2, the transmission models will be combined with the general driveline model of Equations (1.6)–(1.8). The resulting complete driveline models are then used to analyze gearshift trajectories.

1.4.1 Parallel shaft architecture with two frictional clutches

The first transmission architecture to be modeled is illustrated in Figure 1.3a. The equations of motion for this system are

$$I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} = -c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + T_{\mathrm{m}} - T_1 - T_2, \tag{1.9}$$

$$I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}} = -c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} - T_{\mathrm{o}} + i_1 T_1 + i_2 T_2, \tag{1.10}$$

where T_1 and T_2 are the clutch torques. For this first model, both clutches are assumed to be frictional clutches. The torque output of a friction clutch depends on its state—i.e., sticking or slipping. The Coulomb friction model is assumed for simplicity. The clutch torques can be expressed as

$$T_1 = \begin{cases} T_{\mathrm{m}} - T_2 - I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} - c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} & \dot{\theta}_{\mathrm{m}} = i_1\dot{\theta}_{\mathrm{out}} \\ F_{\mathrm{n1}}\mu_{\mathrm{d}} r_{\mathrm{a}} n_{\mathrm{p}} \operatorname{sign}(\dot{\theta}_{\mathrm{m}} - i_1\dot{\theta}_{\mathrm{out}}) & \dot{\theta}_{\mathrm{m}} \neq i_1\dot{\theta}_{\mathrm{out}} \end{cases}, \tag{1.11}$$

$$T_2 = \begin{cases} T_{\mathrm{m}} - T_1 - I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} - c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} & \dot{\theta}_{\mathrm{m}} = i_2\dot{\theta}_{\mathrm{out}} \\ F_{\mathrm{n2}}\mu_{\mathrm{d}} r_{\mathrm{a}} n_{\mathrm{p}} \operatorname{sign}(\dot{\theta}_{\mathrm{m}} - i_2\dot{\theta}_{\mathrm{out}}) & \dot{\theta}_{\mathrm{m}} \neq i_2\dot{\theta}_{\mathrm{out}} \end{cases}, \tag{1.12}$$

where F_{n} is the linear force at the clutch plates, μ_{d} is the clutch's dynamic friction coefficient, r_{a} is the mean friction radius, and n_{p} is the number of friction surfaces. The clutch starts to slip when the reaction torque at the interface reaches the clutch torque capacity $T_{\mathrm{cap}} = F_{\mathrm{n}}\mu_{\mathrm{s}} r_{\mathrm{a}} n_{\mathrm{p}}$, where μ_{s} is the static friction coefficient.

1.4.2 Parallel shaft architecture with a one-way clutch

In this second transmission model, the first-gear clutch in Figure 1.3a is a one-way clutch. The equations of motion (1.9) and (1.10) remain the same. But this time, T_1 is a reaction torque in one direction, and is null in the other direction:

$$T_1 = \begin{cases} T_{\mathrm{m}} - T_2 - I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} - c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} & \dot{\theta}_{\mathrm{m}} = i_1\dot{\theta}_{\mathrm{out}} \\ 0 & \dot{\theta}_{\mathrm{m}} < i_1\dot{\theta}_{\mathrm{out}} \end{cases}. \tag{1.13}$$

The one-way clutch also introduces the kinematic constraint

$$\dot{\theta}_{\mathrm{m}} \leq i_1\dot{\theta}_{\mathrm{out}}. \tag{1.14}$$

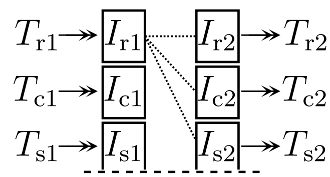


Figure 1.4: General representation of a double planetary gearset. Possible connections are shown between the ring of the first set and the elements of the second set.

1.4.3 Planetary gearset architectures

A single planetary stage can be seen as a combination of three bodies: a ring gear with inertia I_r, a planet carrier (I_c), and a sun gear (I_s). The equations of motion for each of these bodies are

$$I_\mathrm{r}\ddot{\theta}_\mathrm{r} = T_\mathrm{r} - r_\mathrm{r}F, \tag{1.15}$$

$$I_\mathrm{c}\ddot{\theta}_\mathrm{c} = T_\mathrm{c} + r_\mathrm{r}F + r_\mathrm{s}F, \tag{1.16}$$

$$I_\mathrm{s}\ddot{\theta}_\mathrm{s} = T_\mathrm{s} - r_\mathrm{s}F, \tag{1.17}$$

where $T_\square$ is the torque applied on either the ring (r), carrier (c), or sun (s), r_r is the radius of the ring gear, r_s is the radius of the sun gear, and F is the tooth force in the gearset. There is also a kinematic constraint associated with these equations:

$$r_\mathrm{s}\dot{\theta}_\mathrm{s} + r_\mathrm{r}\dot{\theta}_\mathrm{r} = (r_\mathrm{s} + r_\mathrm{r})\dot{\theta}_\mathrm{c}. \tag{1.18}$$

Equations (1.15)–(1.18) are commonly used the the analysis of automatic transmissions, see [61] for instance. It is worth noting that these equations imply to approximate as null the rotational inertia of the planet gears—see Appendix B for more details. Meanwhile, the mass of the planet gears can still be considered in the equations by projecting it into the rotational inertia for the planet carrier I_c.

It is common to combine stages of planetary gearsets in series, such as in Figure 1.4. In this case, one set of the equations (1.15) to (1.18) needs to be added to the system model for every additional planetary stage. By connecting and grounding elements, kinematic constraints are added to the system model, and the number of degrees of freedom in the system is reduced. These connections must be done carefully, as the system can become over-constrained or under-constrained. An example design is presented in Section 1.5.

The transmission of Figure 1.3b is an example of a double planetary gearset [40]. The rest of this section shows how to obtain a system model for it. The inertias I_{m} and I_{c1} are lumped into a single mass; the same is done for I_{out} and I_{c2}. The equations of motion are

$$I_{\mathrm{r}}\ddot{\theta}_{\mathrm{r}} = T_1 - r_{\mathrm{r1}}F_1 - r_{\mathrm{r2}}F_2, \tag{1.19}$$

$$I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} = -c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + T_{\mathrm{m}} + r_{\mathrm{r1}}F_1 + r_{\mathrm{s1}}F_1, \tag{1.20}$$

$$I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}} = -c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} - i_{\mathrm{f}}^{-1}T_{\mathrm{o}} + r_{\mathrm{r2}}F_2 + r_{\mathrm{s2}}F_2, \tag{1.21}$$

$$I_{\mathrm{s}}\ddot{\theta}_{\mathrm{s}} = T_2 - r_{\mathrm{s1}}F_1 - r_{\mathrm{s2}}F_2. \tag{1.22}$$

The two kinematic constraints are

$$r_{\mathrm{s1}}\dot{\theta}_{\mathrm{s}} + r_{\mathrm{r1}}\dot{\theta}_{\mathrm{r}} = (r_{\mathrm{s1}} + r_{\mathrm{r1}})\dot{\theta}_{\mathrm{m}}, \tag{1.23}$$

$$r_{\mathrm{s2}}\dot{\theta}_{\mathrm{s}} + r_{\mathrm{r2}}\dot{\theta}_{\mathrm{r}} = (r_{\mathrm{s2}} + r_{\mathrm{r2}})\dot{\theta}_{\mathrm{out}}. \tag{1.24}$$

The kinematic constraints can be used to reduce the four equations of motion into a set of only two equations. In order to solve this algebraic problem, researchers have assumed that elements other than the input and output shafts have a negligible inertia [61]. This greatly simplifies the reduction process. For the system of Figure 1.3b, it would mean that $I_{\mathrm{r}} = 0$ and $I_{\mathrm{s}} = 0$. For convenience, the parameters $\beta_1 = r_{\mathrm{r1}}/r_{\mathrm{s1}}$ and $\beta_2 = r_{\mathrm{r2}}/r_{\mathrm{s2}}$ are introduced. The result of this reduction is a set of equations that is identical in form to that of a parallel shaft architecture. This can be observed by comparing Equations (1.25) and (1.26) to Equations (1.9) and (1.10).

$$I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} = -c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + T_{\mathrm{m}} + \frac{1+\beta_1}{\beta_1-\beta_2}T_1 - \frac{\beta_2(1+\beta_1)}{\beta_1-\beta_2}T_2, \tag{1.25}$$

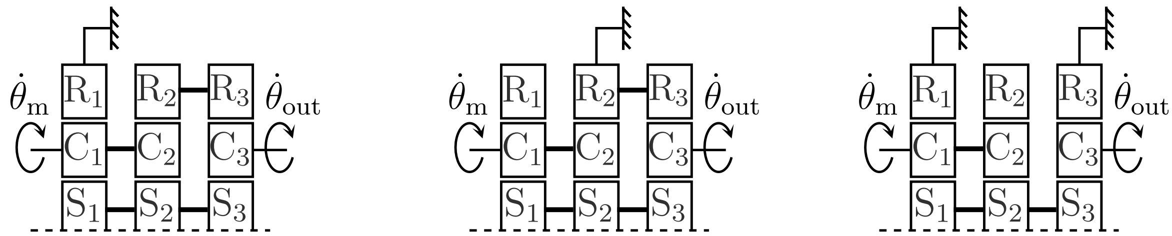
$$I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}} = -c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} - i_{\mathrm{f}}^{-1}T_{\mathrm{o}} + \frac{1+\beta_2}{\beta_2-\beta_1}T_1 - \frac{\beta_1(1+\beta_2)}{\beta_2-\beta_1}T_2. \tag{1.26}$$

If it is not assumed that $I_{\mathrm{r}} = I_{\mathrm{s}} = 0$, the reduction process results in a different set of equations; they are more coupled this time.

$$I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} = c_1(-c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + T_{\mathrm{m}}) + c_2(-c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} - i_{\mathrm{f}}^{-1}T_{\mathrm{o}}) + c_3T_1 + c_4T_2, \tag{1.27}$$

$$I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}} = c_5(-c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + T_{\mathrm{m}}) + c_6(-c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} - i_{\mathrm{f}}^{-1}T_{\mathrm{o}}) + c_7T_1 + c_8T_2. \tag{1.28}$$

The constant coefficients c_1 to c_8 are not detailed further in this thesis. They are fairly involved algebraic expressions that only pertain to the specific architecture of Figure 1.3b. More important is the observation that the motor acceleration is now coupled with the transmission output.



(a) Gear 1. **(b)** Gear 2. **(c)** Gear 3.

Figure 1.5: The three configurations of the new multi-speed transmission design.

1.5 New multi-speed transmission design

This section presents a new design for a three-speed planetary transmission. The design was manufactured and used as a transmission test bench for the experiments of Chapter 3. The transmission comprises three planetary gearsets. By varying the internal connections and the grounding connections on the different bodies, three speed ratios can be achieved. The three configurations are illustrated on Figure 1.5.

In industry, new transmission designs must be optimized with respect to several design criteria. Typically, engineers randomly generate numerous candidate designs and automate the filtering and selection process through design optimization—a process called transmission synthesis [62–64]. In this work, we simply used trial and error to iterate through a few candidates, and selected the final design with simple criteria such as ease of manufacturing and the resulting transmission ratios. The ratios can be computed by solving a linear system of equations, which needs to be repeated for each of the three configurations of Figure 1.5. The process is now shown for the Gear 1 configuration of Figure 1.5a.

Three kinematic relations are required, one for each planetary gearset,

$$r_{\mathrm{s}i}\dot{\theta}_{\mathrm{s}i} + r_{\mathrm{r}i}\dot{\theta}_{\mathrm{r}i} = (r_{\mathrm{s}i} + r_{\mathrm{r}i})\dot{\theta}_{\mathrm{c}i}, \quad i = \{1, 2, 3\}. \tag{1.29}$$

Because R_1 is grounded, $\dot{\theta}_{\mathrm{r1}} = 0$. Moreover, C_1 and C_2 are connected, R_2 and R_3 are connected, and S_1, S_2 and S_3 are connected, so $\dot{\theta}_{\mathrm{c1}} = \dot{\theta}_{\mathrm{c2}}$, $\dot{\theta}_{\mathrm{r1}} = \dot{\theta}_{\mathrm{r3}}$, and $\dot{\theta}_{\mathrm{s1}} = \dot{\theta}_{\mathrm{s2}} = \dot{\theta}_{\mathrm{s3}}$. Assuming $\dot{\theta}_{\mathrm{c3}} = 1$, a linear system of equations can be formed and the transmission ratio can be computed by solving for $\dot{\theta}_{\mathrm{c1}}$. Table 1.2 lists the chosen radius for the components, and the resulting transmission ratios.

When a candidate design yields no solution to the system of equations, this means that the transmission is either over-constrained with too many connections, or under-constrained

Table 1.2: Gear radius and transmission ratios for the new transmission design.

Sun gears		Ring gears		Ratios	
r_{s1}	28 mm	r_{r1}	100 mm	i_1	2.59
r_{s2}	50 mm	r_{r2}	100 mm	i_2	1.52
r_{s3}	28 mm	r_{r3}	100 mm	i_3	1.00

with not enough connections.

It is also possible to compute the steady-state reaction torques on the braking and locking elements, as well as the gear meshing forces. This is achieved by setting the bodies' acceleration to zero in Equations (1.15) to (1.17) which yields

$$0 = T_{\mathrm{r}i} - r_{\mathrm{r}i} F_i, \tag{1.30}$$

$$0 = T_{\mathrm{c}i} + r_{\mathrm{r}i} F_i + r_{\mathrm{s}i} F_i, \tag{1.31}$$

$$0 = T_{\mathrm{s}i} - r_{\mathrm{s}i} F_i. \tag{1.32}$$

Because there are three planetary gearsets, three sets of these three equations are required, where $i = \{1, 2, 3\}$. The connections between the elements generate additional equations. For instance, C_1 is connected to C_2, and the motor provides torque to C_1, so $T_{\mathrm{c1}} = T_{\mathrm{m}} - T_{\mathrm{c2}}$. Just as with the speed ratios, the resulting system can be solved by assuming $T_{\mathrm{out}} = 1$ for instance. The torque ratio between the input and the output of the transmission should be the inverse of the speed ratio, which is a good verification.

The next step in the design process is to verify that the three configurations of Figure 1.5 can be switched between one another, in other words, that gearshifts can take place. This requires placing braking and locking elements on the various bodies—see Figure 1.6. This design comprises two braking elements that can be modulated to provide uninterrupted gearshifts. The design also comprises three locking elements, which allows to deactivate the braking elements after gearshifts, thereby saving energy by eliminating the need to provide constant pressure on the brake friction plates. Interestingly, this design only requires two braking elements for three gear ratios—this is part of the motivations for the patent application associated with this design. Table 1.3 shows how the locking and braking elements can be activated to yield the three gear ratios. Switching between configurations 1c and 2a results in a clutch-to-clutch gearshift between gear ratios 1 and 2. Switching between configurations 2c and 3a results in a clutch-to-clutch gearshift between ratios 2 and 3.

The next chapters investigates the conditions under which clutch-to-clutch gearshifts can

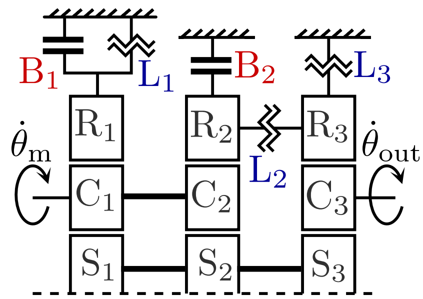


Figure 1.6: The component arrangement in the new transmission design.

Table 1.3: The configurations of the transmission and their component activation.

Gear	Config.	B_1	B_2	L_1	L_2	L_3
1	1a			x	x	
	1b	x		x	x	
	1c	x			x	
2	2a		x		x	
	2b		x		x	x
	2c		x			x
	2d				x	x
3	3a	x				x
	3b	x		x		x
	3c			x		x

result in an uninterrupted torque at the transmission output, and when motor saturation makes this fundamentally impossible.

Chapter 2

Uninterrupted gearshifts

This chapter concerns the fundamental limitations to uninterrupted gearshifts. First, common shifting processes are illustrated through example gearshift trajectories. Then, the limitations are explicitly defined by introducing three new theorems.

Contents

2.1 Limits to uninterrupted gearshifts 19

2.2 Example gearshift trajectories 22

2.3 Theorems for fundamental limitations 27

2.3.1 Theorems for a dual-clutch architecture 27

2.3.2 Adaptations for planetary gearset architectures 33

2.3.3 Other vehicle models and kinematic conditions 35

2.1 Limits to uninterrupted gearshifts

Uninterrupted gearshifts provide superior drivability to electric vehicles, but not all multi-speed transmissions are capable of it. For that, it must be possible to continuously transfer the motor torque from one transmission path to another during gearshifts. Transmission architectures with this capability were reviewed in Chapter 1, namely dual-clutch transmissions and transmissions based on planetary gearsets. But in the presence of motor and clutch saturation, even such architectures can fail to provide an uninterrupted gearshift. This chapter explores the fundamental limitations on gearshift performance that originate from actuator saturation.

These fundamental limitations should be considered early in an electric vehicle design process, such as when selecting the transmission type during the conceptual design phase. Design methodologies typically attribute a high importance to the conceptual design phase, as its outcome has a large influence on the rest of the design project, and ultimately, the product quality [65–67]. This chapter aims to provide automotive engineers with clear expectations on the potential gearshift performance of various transmission architectures before they delve into resource-intensive detailed modeling.

Moreover, the limitations should be considered when designing—or learning—a gearshift controller. In effect, the theorems describe the system trajectories that allow for an uninterrupted gearshift.

The existence of fundamental limitations to uninterrupted gearshifts that originate from motor saturation was hinted in [37], but they were never formulated explicitly. Some studies on gearshift jerk reduction are framed around gearshift trajectory optimization [68,69]. Typically, researchers model a driveline, formulate a cost function that balances vehicle jerk and clutch energy dissipation, then solve a trajectory optimization problem. The main caveat with this approach is that often the optimization problem is non-convex, so a global optimum is not guaranteed. Moreover, it is hard to apply the results to other vehicles or to small alterations of the transmission design. Other studies address the design of an optimal gearshift controller [70–73]. But similarly, this approach does not allow to generalize, as it is not known whether the torque gap during gearshift is a result of an imperfect controller, or an unavoidable fundamental limitation. Instead, this chapter aims to provide explicit definitions of these limitations in the form of theorems.

The work in this chapter is done under the assumption of a perfect state feedback, a perfect control of actuator forces, and a perfect model of the system dynamics. Formally, this translates into Assumption 1. In practice, this assumption would never hold; other system limitations inevitably lead to torque interruptions. Nevertheless, Assumption 1 is appropriate for this chapter given that the theorems are only intended to predict when actuator saturation is an inevitable contributor to torque interruptions.

Assumption 1. Only the following two system limitations can lead to unavoidable torque interruption during gearshift:

1. a limit on the motor torque, which can be characterized both in terms of maximal torque $T_{\max}$, or maximal power $p_{\max}$;
2. a limit on the torque application rate of a friction clutch, $\dot{T}$.

Furthermore, uninterrupted gearshifts need to be properly defined. In this work, a gearshift is considered uninterrupted when the vehicle acceleration a_{r} remains constant for the duration of the gearshift. This is also the definition of a no-jerk gearshift, therefore in this chapter, these two concepts are treated as equivalent and the terms are used interchangeably. The definition for an uninterrupted gearshift can be specified using the driveline model of Chapter 1. A constant acceleration implies that $\ddot{\theta}_{\mathrm{v}} = a_{\mathrm{r}}/r_{\mathrm{w}}$. And by extension, $\dot{\theta}_{\mathrm{v}} = (a_{\mathrm{r}}t + v_{\mathrm{i}})/r_{\mathrm{w}}$, where v_{i} is the initial vehicle speed at the beginning of the gearshift ($t = 0$). Since the gearshifts are short in duration—approximately 0.5 s—it can be assumed that the vehicular forces T_{v} are constant. Given the driveline model in Equations (1.6)–(1.8), the constraints introduced on $\ddot{\theta}_{\mathrm{v}}$, $\dot{\theta}_{\mathrm{v}}$, and T_{v} also restrict $\dot{\theta}_{\mathrm{out}}$ and $\ddot{\theta}_{\mathrm{out}}$. These variables can be solved for explicitly. First, the vehicular forces and acceleration are grouped into a single constant output torque $T_{\mathrm{o}} = I_{\mathrm{v}}\ddot{\theta}_{\mathrm{v}} + T_{\mathrm{v}}$. Substituting T_{o} in Equation (1.8) yields

$$T_{\mathrm{o}} = k(\theta_{\mathrm{out}} - \theta_{\mathrm{v}}) + d(\dot{\theta}_{\mathrm{out}} - \dot{\theta}_{\mathrm{v}}). \tag{2.1}$$

Taking the time derivative of Equation (2.1) gives

$$0 = k(\dot{\theta}_{\mathrm{out}} - \dot{\theta}_{\mathrm{v}}) + d(\ddot{\theta}_{\mathrm{out}} - \ddot{\theta}_{\mathrm{v}}). \tag{2.2}$$

Substituting $\ddot{\theta}_{\mathrm{v}}$ and $\dot{\theta}_{\mathrm{v}}$ with their no-jerk constraints in Equation (2.2), a linear differential is obtained, where

$$\ddot{\theta}_{\mathrm{out}}(t) + \frac{k}{d}\dot{\theta}_{\mathrm{out}}(t) = \frac{k}{dr_{\mathrm{w}}}(a_{\mathrm{r}}t + v_{\mathrm{i}}) + \frac{a_{\mathrm{r}}}{r_{\mathrm{w}}}. \tag{2.3}$$

Assuming the initial condition $\dot{\theta}_{\mathrm{out}}(0) = v_{\mathrm{i}}/r_{\mathrm{w}}$, the differential equation can be solved, and the trajectory for the output shaft is obtained as

$$\dot{\theta}_{\mathrm{out}}(t) = \frac{a_{\mathrm{r}}t + v_{\mathrm{i}}}{r_{\mathrm{w}}}. \tag{2.4}$$

The prescribed trajectories on $\dot{\theta}_{\mathrm{v}}$, $\dot{\theta}_{\mathrm{out}}$, and T_{o} can be used to formally define an no-jerk gearshift.

Definition 1. A no-jerk gearshift is obtained if, for the duration of the gearshift,

$$\dot{\theta}_{\mathrm{out}} = \dot{\theta}_{\mathrm{v}} = (a_{\mathrm{r}}t + v_{\mathrm{i}})/r_{\mathrm{w}}, \tag{2.5}$$

$$T_{\mathrm{o}} = I_{\mathrm{v}}a_{\mathrm{r}}/r_{\mathrm{w}} + T_{\mathrm{v}}. \tag{2.6}$$

Table 2.1: Example-vehicle parameters.

Param.	Value	Param.	Value	Param.	Value
m	6500 kg	I_{out}	0.05 kg m^2	i_2	6
r_{w}	0.3 m	c_{m}	0.02 Nm s/rad	I_{r}	0.03 kg m^2
a_{f}	6 m^2	c_{o}	0.04 Nm s/rad	I_{s}	0.03 kg m^2
c_{d}	0.7	k	10 kNm/rad	β_1	2
c_{t}	0.007	d	75 Nm s/rad	β_2	4
I_{m}	0.3 kg m^2	i_1	12	i_{f}	7.2

Table 2.2: Gearshift scenarios.

Scenario	Direction	Motor quadrant	Region	v_{i} [km/h]	a_{r} [m/s^2]	DTD
1	Upshift	Driving	Power-limited	65	1.0	80%
2	Downshift	Driving	Torque-limited	18	1.0	80%
3	Downshift	Braking	Torque-limited	45	-1.5	-

2.2 Example gearshift trajectories

This section presents five example gearshift trajectories where the no-jerk conditions of Definition 1 are met. The examples illustrate the implications of the no-jerk condition on the input torques—the motor torque, and the clutch torques. The vehicle parameters are listed in Table 2.1. The three gearshift scenarios of interest are that of Table 2.2. Scenario 1 is an upshift during vehicle acceleration, where the driver torque demand (DTD) at the beginning of the shift is 80% of the available torque. This gearshift takes places in the power-limited region of the motor map. Scenario 2 is a downshift when the vehicle is accelerating, also with a DTD of 80%. The downshift takes place in the torque-limited region of the motor map. This gearshift is motivated by the desire to have a greater wheel torque after the downshift. Scenario 3 is a downshift when the motor is used for regenerative braking in the torque-limited region.

The example trajectories are computed using the transmission models of Section 1.4: Equations (1.9)–(1.10) for a dual-clutch transmission, Equations (1.25)–(1.26) for a dual-brake transmission with neglected gear inertias, and Equations (1.27)–(1.28) for a dual-brake transmission with the gear inertias considered. First, T_{o} is computed from (2.6) and (1.5) given the acceleration a_{r} and initial velocity v_{i} of the scenario. Similarly, $\dot{\theta}_{\text{out}}(t)$ and $\ddot{\theta}_{\text{out}}(t)$ can be computed from (2.4). A trajectory is chosen for $\dot{\theta}_{\text{m}}(t)$. And finally, the input torques can be computed from the transmission model.

Figure 2.1 shows an upshift during vehicle acceleration (Scenario 1) with a dual-clutch transmission where the first clutch is a one-way clutch. The gearshift starts at 0.05 s and

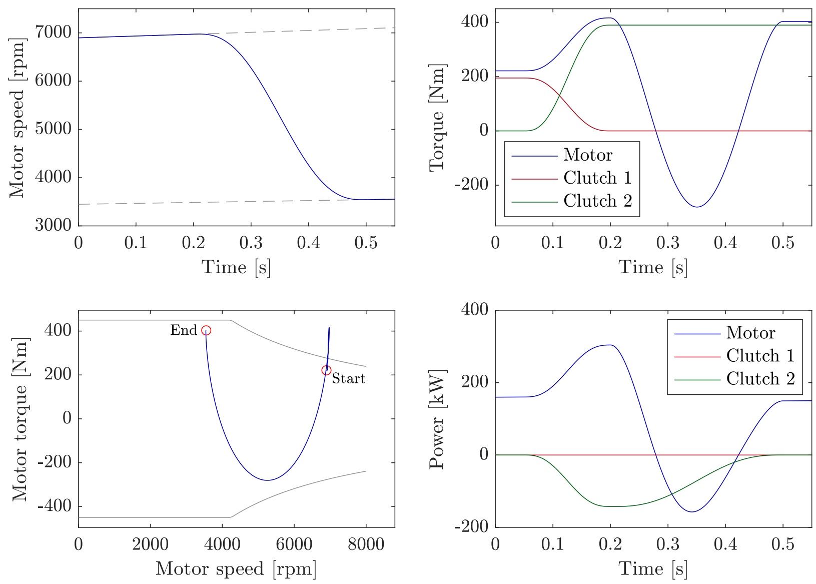


Figure 2.1: Example trajectory for the gearshift scenario 1 with a dual-clutch transmission, where the first clutch is a one-way clutch.

ends at 0.50 s. It begins with a torque phase—where the transmission torque is transferred from Clutch 1 to Clutch 2—followed by an inertia phase, where the motor is synchronized with the Gear 2 speed. During the torque phase, T_2 is gradually increased from zero to the torque required at Gear 2 following an arbitrary trajectory. This has the effect of gradually reducing the reaction torque on the one-way clutch, T_1, all the way down to zero at the end of the torque phase. In order to maintain a constant output torque T_o, the motor torque T_m has to increase according the to transmission model. Interestingly, because the first clutch is a one-way clutch, the torque transfer phase just described is the only possible trajectory. But since the motor power exceeds its 200 kW limit in the example of Figure 2.1, the example trajectory is in fact infeasible. In reality, the vehicle would inevitably experience a torque gap. This is the first fundamental limitation examined in this chapter—Theorem 1 in Section 2.3 formalizes it. The rest of the gearshift consists of the inertia phase, which is not as prone to motor saturation as the torque phase. T_1 and T_2 are kept constant, and the motor is synchronized with the second-gear speed following an arbitrary trajectory. It is possible to reduce the variation in motor power during the inertia phase by simply making it longer.

Figure 2.2 also shows an upshift during vehicle acceleration with a dual-clutch transmission, but Clutch 1 is a friction clutch this time. This introduces two new possibilities: the motor speed can be increased above the Gear 1 speed, and T_1 can be modulated when the clutch is slipping. By taking advantage of these possibilities, a new trajectory can be

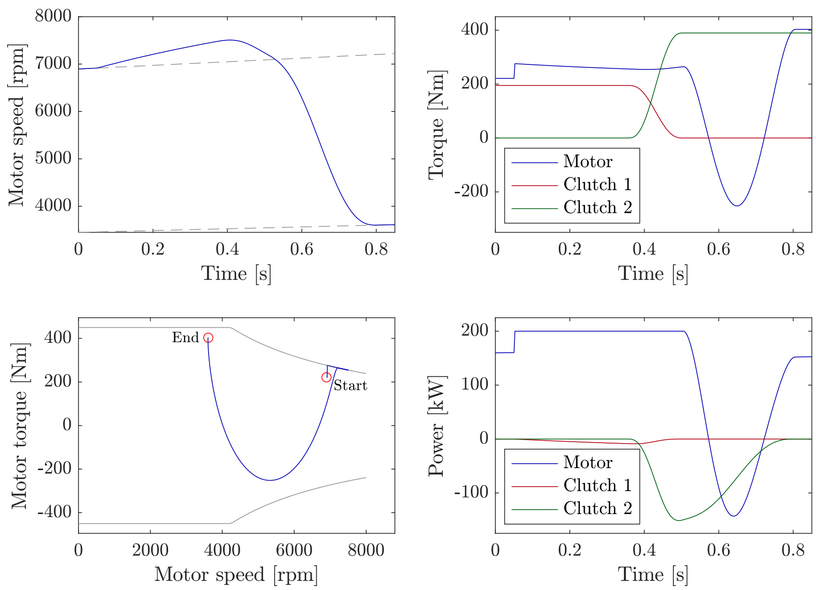


Figure 2.2: Example trajectory for the gearshift scenario 1 with a dual-clutch transmission, where the first clutch is a friction clutch.

crafted to obtain a no-jerk gearshift. Prior to transferring the clutch torques, the motor speed is increased above Gear 1 speed. Then the torque transfer begins, which has the effect of decreasing the motor speed. It is important that the torque transfer completes before the motor crosses Gear 1 speed, as this would result in a sign reversal on T_1. In effect, when a friction clutch slips, the clutch torque opposes the clutch slipping velocity as per Equation (1.11). With a torque reversal on T_1, T_2 would have to instantaneously compensate in order to maintain the constant T_o condition. This would violate any constraint on the clutch torque application rate. Therefore, this trajectory also contains a fundamental limitation, which is formulated in Theorem 2.

Figure 2.3 shows a downshift during motor acceleration (Scenario 2) with a dual-clutch transmission. This trajectory can be obtained whether Clutch 1 is a one-way clutch or a friction clutch. This time the inertia phase precedes the torque phase. In the inertia phase, the motor speed is increased using maximal motor power. When the motor reaches Gear 1 speed, the torque transfer begins. The fundamental limitation for this scenario concerns the capacity to synchronize the motor speed within acceptable time, while maintaining the output torque T_o constant. This limitation is formalized in Theorem 3.

Figure 2.4 shows a downshift during vehicle deceleration (Scenario 3) with a dual-clutch transmission. The motor operates in regenerative braking mode. Because $T_\mathrm{o} < 0$, the clutch torques T_1 and T_2 must be negative. And since $T_1 < 0$, this trajectory is only possible

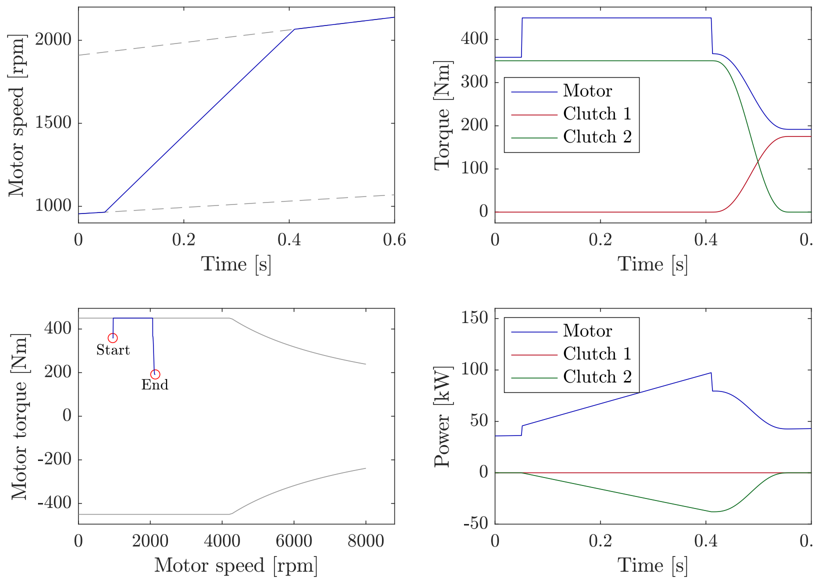


Figure 2.3: Example trajectory for the gearshift scenario 2 with a dual-clutch transmission. This trajectory is valid for both one-way clutch and dual-friction-clutch architectures.

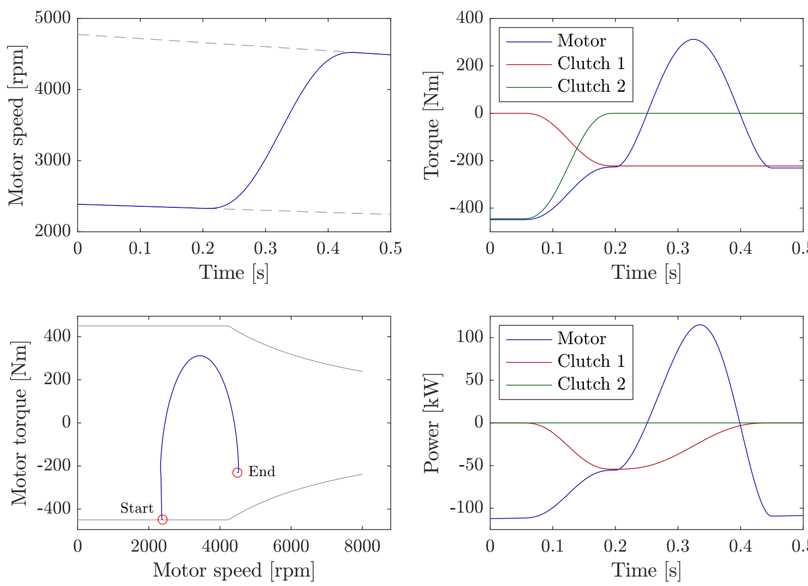


Figure 2.4: Example trajectory for the gearshift scenario 3 with a dual-clutch transmission. This trajectory is only possible if Clutch 1 is of friction type.

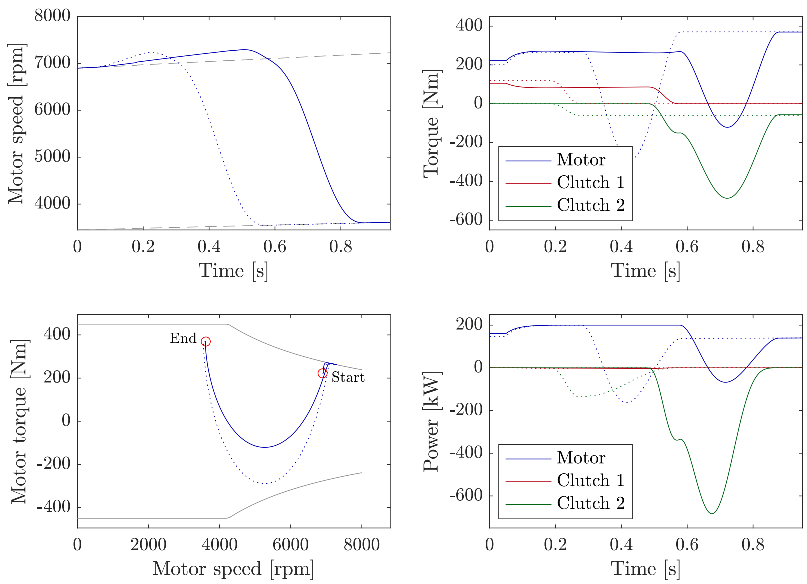


Figure 2.5: Example trajectories for the gearshift scenario 1 with a dual-brake transmission, namely that of Figure 1.3b. The dotted lines represent the situation where the trajectory is computed with Equations (1.25)–(1.26), thereby assuming $I_r = I_s = 0$. The solid lines represent the situation where the inertias of I_r and I_s are considered, and the trajectory is computed with Equations (1.27)–(1.28).

if Clutch 1 is a friction clutch—by design, a one-way clutch can only carry torque in one direction. Often, transmissions with a one-way clutch are also equipped with a locking mechanism in parallel to the one-way clutch [37]. This allows for regenerative braking when the transmission operates in first gear. However, the locking mechanism typically cannot be engaged when there is a significant speed difference between the mating elements, and it cannot be modulated. Therefore, it is impossible for a dual-clutch transmission with a one-way clutch to provide uninterrupted shifting in regenerative braking mode. Figure 2.4 shows that for a dual friction clutch architecture, such a gearshift can be initiated even when the motor is essentially on the saturation limit. For transmissions with two friction clutches, the system limitations included in Assumption 1 will not result in unavoidable gearshift jerk.

Figure 2.5 shows a gearshift of Scenario 1, but with the dual-brake transmission of Figure 1.3b this time. The gearshift strategy is the same as in Figure 2.2, i.e., increasing the motor speed above the Gear 1 speed just before the torque transfer. The trajectory in dotted lines are obtained by computing the trajectory with Equations (1.25)–(1.26), which implies assuming $I_r = I_s = 0$. The trajectories in solid lines are obtained considering $I_r = I_s = I_m/10$ and using Equations (1.27)–(1.28). Introducing a small inertia on I_r and I_s has a strong influence on the resulting trajectory. First, T_m becomes coupled with T_1—when the motor torque

Table 2.3: Summary of fundamental limitations to uninterrupted gearshifts for the dual-clutch transmission.

Scenario	One-way clutch	Dual-friction clutch
1: Upshift, driving motor	Limited by Theorem 1	Limited by Theorem 2
2: Downshift, driving motor	Limited by Theorem 3	Limited by Theorem 3
3: Downshift, braking motor	Impossible	Not limited

is increased in the first phase of the gearshift, T_1 must be decreased in order to maintain the same output torque T_o. This means T_m cannot increase arbitrarily quickly due to the clutch torque application rate limitation. Also, the additional inertia increases the time required to attain a given motor speed during the first phase. Finally, Clutch 2 takes a larger portion of the load during the inertia phase, and the motor takes a smaller portion of the load. From a design perspective, this means that the maximal torque requirement on Clutch 2 is higher, which has to be accounted for in the sizing of this component. In summary, even small gear inertias can have a significant influence on the gearshift trajectory of a transmission with a planetary gearset. Engineers should take precaution before neglecting them in the equations of motion for their systems.

2.3 Theorems for fundamental limitations

This section introduces three theorems that define the fundamental limitations to no-jerk gearshift for a dual-clutch transmission. Table 2.3 summarizes which theorem pertains to which combination of gearshift scenario and Clutch 1 type. Theorems for Scenario 3 are not provided as the conclusions follow naturally from Figure 2.4 and the associated discussion in Section 2.2. Then in Section 2.3.2, the theorems are adapted for transmissions based on planetary gearsets.

2.3.1 Theorems for a dual-clutch architecture

Theorem 1 presents a necessary and sufficient condition for a no-jerk upshift in the power-limited region of the motor map when Clutch 1 is a one-way clutch. Together Equation (2.7) and the assumptions that $\dot{\theta}_\mathrm{out}(t) = (a_\mathrm{r} t + v_\mathrm{i})/r_\mathrm{w}$ and $\ddot{\theta}_\mathrm{out}(t) = a_\mathrm{r}/r_\mathrm{w}$ can be used to predict whether a specific driving condition allows for a no-jerk gearshift. Interestingly, it does not require to simulate the complete gearshift.

Theorem 1. For a dual-clutch architecture where the first-gear clutch is a one-way clutch, and when the motor operates in a power-limited region, a no-jerk upshift of Scenario 1 can be obtained if and only if $p_{\mathrm{m}}(t_{\mathrm{tr}}) \leq p_{\max}$, where $p_{\mathrm{m}}(t_{\mathrm{tr}})$ is the required motor power at the end of the torque transfer phase, and $p_{\max}$ is the motor power limit; $p_{\mathrm{m}}(t_{\mathrm{tr}})$ is evaluated as

$$p_{\mathrm{m}}(t_{\mathrm{tr}}) = i_1\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}})\left(\left(i_1 I_{\mathrm{m}} + i_2^{-1} I_{\mathrm{out}}\right)\ddot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + \left(i_1 c_{\mathrm{m}} + i_2^{-1} c_{\mathrm{o}}\right)\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + i_2^{-1} T_{\mathrm{o}}\right). \tag{2.7}$$

Proof. (Necessity): The gearshift begins with a torque transfer phase that spans from $t = 0$ to $t = t_{\mathrm{tr}}$, which is defined as follows: the clutch torques are smoothly varied from $T_1(0) \neq 0$ and $T_2(0) = 0$ to $T_1(t_{\mathrm{tr}}) = 0$ and $T_2(t_{\mathrm{tr}}) \neq 0$, while $\dot{\theta}_{\mathrm{m}} = i_1\dot{\theta}_{\mathrm{out}}$ and $\ddot{\theta}_{\mathrm{m}} = i_1\ddot{\theta}_{\mathrm{out}}$ for all $t \in [0, t_{\mathrm{tr}}]$, and T_{o}, $\dot{\theta}_{\mathrm{out}}$, and $\ddot{\theta}_{\mathrm{out}}$ follow the conditions for a no-jerk gearshift as per Definition 1. The condition $\dot{\theta}_{\mathrm{m}} = i_1\dot{\theta}_{\mathrm{out}}$ is necessary because the first-gear clutch is a one-way clutch: Equation (1.13) indicates that $T_1 \neq 0 \rightarrow \dot{\theta}_{\mathrm{m}} = i_1\dot{\theta}_{\mathrm{out}}$, so this imposes $\dot{\theta}_{\mathrm{m}} = i_1\dot{\theta}_{\mathrm{out}}$ at least until $T_1 = 0$. The condition $\ddot{\theta}_{\mathrm{m}} = i_1\ddot{\theta}_{\mathrm{out}}$ is necessary for a no-jerk gearshift: Equation (1.13) indicates that if the clutch opens before $T_1 = 0$, which means that $\ddot{\theta}_{\mathrm{m}} < i_1\ddot{\theta}_{\mathrm{out}}$, then the clutch torque immediately drops to 0, and Equation (1.10) indicates that for T_{o}, $\dot{\theta}_{\mathrm{out}}$, and $\ddot{\theta}_{\mathrm{out}}$ to follow the conditions for a no-jerk gearshift, T_2 would have to immediately jump to a higher value, which violates any limit on clutch torque application rate. Moreover, the one-way clutch imposes the kinematic constraint that $\dot{\theta}_{\mathrm{m}} \leq i_1\dot{\theta}_{\mathrm{out}}$, so it is impossible that $\ddot{\theta}_{\mathrm{m}} > i_1\ddot{\theta}_{\mathrm{out}}$. Therefore, the torque transfer phase as defined above is necessary for a no-jerk gearshift, as any deviation from it either implies a vehicle jerk, or is physically impossible.

The required motor power at the end of the torque phase can then be computed. Substituting $T_1(t_{\mathrm{tr}}) = 0$ in Equation (1.10) yields

$$T_2(t_{\mathrm{tr}}) = i_2^{-1}\left(I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + T_{\mathrm{o}}\right), \tag{2.8}$$

which can be substituted in Equation (1.9) to get the motor torque at t_{tr}:

$$T_{\mathrm{m}}(t_{\mathrm{tr}}) = I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}}(t_{\mathrm{tr}}) + c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}}(t_{\mathrm{tr}}) + i_2^{-1}\left(I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + T_{\mathrm{o}}\right). \tag{2.9}$$

The motor power at t_{tr} is computed using $p_{\mathrm{m}}(t_{\mathrm{tr}}) = \dot{\theta}_{\mathrm{m}}(t_{\mathrm{tr}})T_{\mathrm{m}}(t_{\mathrm{tr}})$, Equation (2.9), and the conditions that $\dot{\theta}_{\mathrm{m}}(t_{\mathrm{tr}}) = i_1\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}})$ and $\ddot{\theta}_{\mathrm{m}}(t_{\mathrm{tr}}) = i_1\ddot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}})$; Equation (2.7) is obtained. Naturally, this requires that the motor be capable of producing $p_{\mathrm{m}}(t_{\mathrm{tr}})$, therefore a no-jerk gearshift implies that $p_{\mathrm{m}}(t_{\mathrm{tr}}) \leq p_{\max}$.

(Sufficiency): Assumption 1 indicates that if a no-jerk gearshift cannot be obtained, it is because either the motor saturates, or the clutch saturates. Further assuming that t_{tr} is large enough such that the clutch does not saturate, if a no-jerk gearshift cannot be obtained, it is because the motor saturates. By showing that $p_{\mathrm{m}}(t_{\mathrm{tr}})$ is the maximal required motor power during the gearshift, it can be shown that if the motor saturates, then $p_{\mathrm{m}}(t_{\mathrm{tr}}) > p_{\max}$. To do this, Equations (1.9) and (1.10) are rearranged to eliminate T_2 and isolate T_m, and the motor power is obtained as follows

$$p_\mathrm{m} = \dot{\theta}_\mathrm{m}\left(I_\mathrm{m}\ddot{\theta}_\mathrm{m} + c_\mathrm{m}\dot{\theta}_\mathrm{m} + T_1\left(1 - \frac{i_1}{i_2}\right) + i_2^{-1}\left(I_\mathrm{out}\ddot{\theta}_\mathrm{out} + c_\mathrm{o}\dot{\theta}_\mathrm{out} + T_\mathrm{o}\right)\right). \tag{2.10}$$

Assuming that the motor begins to synchronize with Gear 2 speed exactly at t_tr, the maximal $\dot{\theta}_\mathrm{m}(t)$ is at $t = t_\mathrm{tr}$. Furthermore, every term on the right-hand side of Equation (2.10) is maximized at t_tr. In particular, since $i_1 > i_2$, $T_1(t_\mathrm{tr}) = 0$ maximizes p_m, as $T_1 \geq 0$. Therefore, if a no-jerk gearshift cannot be obtained, then $p_\mathrm{m}(t_\mathrm{tr}) > p_\mathrm{max}$. □

Theorem 2 presents a necessary and sufficient condition for a no-jerk upshift in the power-limited region of the motor map when Clutch 1 is a friction clutch. The gearshift strategy is outlined in Figure 2.6. Several instances of this strategy are simulated and the result is shown in Figure 2.7. In practice, Theorem 2 can be used to predict whether a specific driving condition allows for a no-jerk gearshift. Because Equation (2.14) renders the transmission model nonlinear, there may not be a convenient closed-form solution to find a sufficient Δ_m for a no-jerk gearshift. Perhaps the best way to do so is to simulate the first phase of the gearshift up to a given Δ_m, and then validate if this Δ_m is sufficient to allow a complete torque transfer before $\Delta_\mathrm{s} = 0$. If the given Δ_m is not sufficient, then the process can be repeated for a higher Δ_m, until no higher Δ_m can be reached, at which point it is concluded that a no-jerk trajectory is infeasible.

Theorem 2. Consider a dual-clutch architecture with two friction clutches as shown in Figure 1.3a. Referring to Figure 2.6, suppose that at $t = 0$ the motor speed is synchronized with Gear 1 speed. Let $t_\mathrm{tr} > 0$ be the set torque transfer duration and t_1, the time at which the torque transfer is initiated. The gearshift is intended to complete at t_2. When the motor operates in a power-limited region, a no-jerk upshift of Scenario 1 can be obtained if and only if

$$\exists\ \Delta_\mathrm{m} := \dot{\theta}_\mathrm{m}(t_1) - i_1\dot{\theta}_\mathrm{out}(t_1) \geq 0, \tag{2.11}$$

$$\text{such that } \Delta_\mathrm{s} := \dot{\theta}_\mathrm{m}(t_1 + t_\mathrm{tr}) - i_1\dot{\theta}_\mathrm{out}(t_1 + t_\mathrm{tr}) \geq 0, \tag{2.12}$$

$$\dot{\theta}_\mathrm{m}(t_1) \leq \dot{\theta}_\mathrm{max}, \tag{2.13}$$

$$\text{where } T_\mathrm{m}(t) = p_\mathrm{max}/\dot{\theta}_\mathrm{m}(t), \quad 0 \leq t \leq t_1 + t_\mathrm{tr}, \tag{2.14}$$

$$T_1(t) = 0, \quad t_1 + t_\mathrm{tr} \leq t \leq t_2, \tag{2.15}$$

$$T_2(t) = 0, \quad 0 \leq t \leq t_1. \tag{2.16}$$

Proof. (Necessity): The gearshift begins with a speed phase ($0 \leq t \leq t_1$), where the motor speed is increased above $i_1\dot{\theta}_{\mathrm{out}}$, with $T_2 = 0$ and $T_{\mathrm{m}} = p_{\max}/\dot{\theta}_{\mathrm{m}}$. Then the gearshift continues with a torque transfer phase ($t_1 \leq t \leq t_1 + t_{\mathrm{tr}}$), where the clutch torques are smoothly varied from $T_1(t_1) \neq 0$ and $T_2(t_1) = 0$ to $T_1(t_1 + t_{\mathrm{tr}}) = 0$ and $T_2(t_1 + t_{\mathrm{tr}}) \neq 0$, while $T_{\mathrm{m}} = p_{\max}/\dot{\theta}_{\mathrm{m}}$. Finally, the gearshift ends with an inertia phase ($t_1 + t_{\mathrm{tr}} \leq t \leq t_2$), where the motor speed is brought down to $i_2\dot{\theta}_{\mathrm{out}}$, while $T_1 = 0$. During all three phases, the conditions for a no-jerk gearshift in Definition 1 can be maintained by modulating T_1 and T_2 as per Equation (1.10). Motor saturation can be avoided as the motor torque is set to $T_{\mathrm{m}} = p_{\max}/\dot{\theta}_{\mathrm{m}}$ for the first two phases, and the motor synchronization of the inertia phase requires that $T_{\mathrm{m}} < p_{\max}/\dot{\theta}_{\mathrm{m}}$. The torque application rate on both clutches can be maintained within limits during the speed phase, as $T_2 = 0$ and the variations on T_{o}, $\dot{\theta}_{\mathrm{out}}$, and $\ddot{\theta}_{\mathrm{out}}$ are small enough such that $\mathrm{d}T_1/\mathrm{d}t$ is within limits. The same argument can be made for the inertia phase, with $T_1 = 0$ this time. During the torque transfer phase, appropriate clutch torque profiles must be chosen such that limits on $\dot{T}$ are respected. Moreover, a torque reversal on Clutch 1 must be avoided.

In effect, the clutch torque T_1 is necessarily positive when the motor is driving the vehicle through the first gear and Clutch 1 sticks. When Clutch 1 slips, the direction of T_1 is dependent on the slip direction, as described in Equation (1.11). If $\dot{\theta}_{\mathrm{m}} < i_1\dot{\theta}_{\mathrm{out}}$, then the torque T_1 suddenly reverses direction and becomes negative. In order to maintain the conditions for a no-jerk gearshift, Equation (1.10) indicates that the torque reversal on T_1 must be instantaneously compensated by an increase in T_2, which necessarily violates any application rate limitation. Consequently, it must be that $\dot{\theta}_{\mathrm{m}} \geq i_1\dot{\theta}_{\mathrm{out}}$ as long as $T_1 \neq 0$. Once the torque transfer phase begins, $\ddot{\theta}_{\mathrm{m}}(t) < \ddot{\theta}_{\mathrm{m}}(t_1)$, given that $i_1 > i_2$, which can be seen from Equations (1.9) and (1.10). Therefore, the torque transfer needs to start at a sufficiently high Δ_{m} such that $\Delta_{\mathrm{s}} \geq 0$. Moreover, Δ_{m} must correspond to a motor speed that is within the motor's limit $\dot{\theta}_{\max}$. However, nothing guarantees the system can reach such a Δ_{m}. If it fails to reach a satisfactory Δ_{m}, then one of the conditions for a no-jerk gearshift will not be satisfied due to system limitations—there will be vehicle jerk. This proves the necessity part by contraposition.

(Sufficiency): By construction, the actuation strategy described in Equations (2.14)–(2.16) is sufficient for a no-jerk gearshift given that Assumption 1 holds.

This actuation strategy is not unique, but any deviation would only make it harder to achieve a sufficiently high Δ_{m}. Consequently, if a no-jerk gearshift can be obtained using such a deviation from Equations (2.14) to (2.16), it can also be obtained using the actuation strategy described in these equations. Therefore, if a no-jerk gearshift cannot be obtained, there is no Δ_{m} such that $\Delta_{\mathrm{s}} \geq 0$ when the Equations (2.14)–(2.16) hold. $\square$

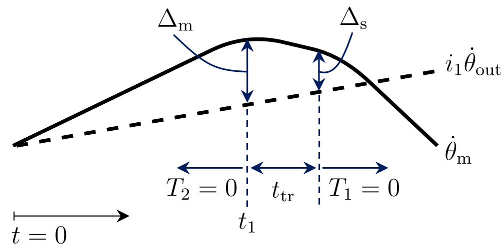
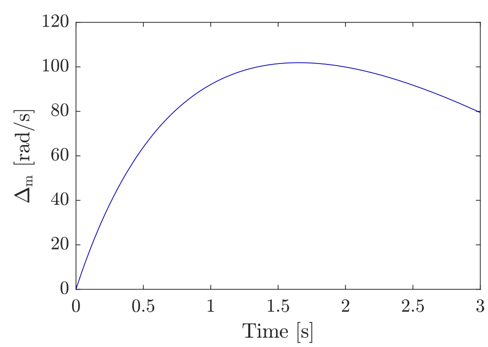
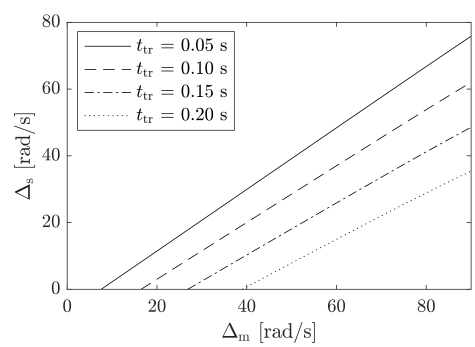


Figure 2.6: Strategy for a power-on upshift with a dual-friction clutch transmission. First the motor speed is increased to an increment Δ_{m} above Gear 1 speed. Once Δ_{m} is reached, the torque transfer begins, which takes a set time t_{tr}. At the end of the torque transfer, the motor will have decelerated to a different speed whose increment over Gear 1 speed is defined as Δ_{s}. In order to avoid torque reversal at Clutch 1, which would imply an unavoidable jerk, it must be that $\Delta_{\text{s}} \geq 0$.



(a) Even when using the maximum available motor power, not all Δ_{m} can be reached.



(b) The Δ_{s} after the torque transfer is a function of the starting Δ_{m} and the transfer time t_{tr}.

Figure 2.7: Example limitations for a power-on upshift with the example vehicle and a dual-friction-clutch transmission, for a gearshift initiated at $v_{\text{i}} = 65\,\text{km/h}$ and $a_{\text{r}} = 1.0\,\text{m/s}^2$.

Comparing Theorems 1 and 2 highlights that transmissions with a one-way clutch are more prone to motor saturation. In effect, the constraint on the maximal motor power imposed in Theorem 1 does not exits in Theorem 2, and as a result, transmissions with two friction clutches have a wider set of possible no-jerk gearshift trajectories. For example, both gearshift trajectories of Figure 2.1 (one-way clutch) and Figure 2.2 (two friction clutches) are initiated under the same driving scenario, namely an upshift at 80% DTD in the motor's power-limited region. The trajectory of Figure 2.1 results in motor saturation and Theorem 1 would indicate that it is inevitable, meanwhile the trajectory of Figure 2.2 completes without motor saturation and a no-jerk gearshift is obtained.

Theorem 3 presents a necessary and sufficient condition for a no-jerk downshift in the torque limited region of the motor map. Several instances of this gearshift scenario were simulated with a different initial vehicle acceleration a_r. The time t_s required for the motor to synchronize with Gear 1 speed is shown on Figure 2.8. When a_r is too high, either t_s is impractically long, or the motor never synchronizes with Gear 1 speed.

Theorem 3. Consider a dual-clutch transmission where the motor operates in a torque-limited region. Assuming a constant T_2 during the inertia phase, a no-jerk power-on downshift can be obtained if and only if

$$\exists\ t_\mathrm{s} > 0 \quad \text{s.t.} \quad 0 = -i_1 \frac{v_\mathrm{i} + a_\mathrm{r} t_\mathrm{s}}{r_\mathrm{w}} + \exp\left(-\frac{c_\mathrm{m}}{I_\mathrm{m}} t_\mathrm{s}\right) i_2 \frac{v_\mathrm{i}}{r_\mathrm{w}} + \frac{T_\mathrm{max} - T_2}{c_\mathrm{m}} \left[1 - \exp\left(-\frac{c_\mathrm{m}}{I_\mathrm{m}} t_\mathrm{s}\right)\right]. \tag{2.17}$$

Proof. (Necessity): The gearshift begins with an inertia phase ($0 \le t \le t_\mathrm{s}$), where $\dot{\theta}_\mathrm{m}$ is accelerated from Gear 2 speed to Gear 1 speed, while $T_\mathrm{m} = T_\mathrm{max}$ and $T_1 = 0$. The gearshift ends with a torque phase ($t_\mathrm{s} \le t \le t_\mathrm{s} + t_\mathrm{tr}$), where the transmission torque is transferred from Clutch 2 to Clutch 1. During both phases, the conditions for a no-jerk gearshift in Definition 1 can be maintained by modulating T_2 (and T_1 when applicable) as per Equation (1.10). Motor saturation can be avoided by restricting the motor power to $T_\mathrm{m} \le T_\mathrm{max}$. Clutch saturation can be avoided by using an appropriate torque transfer trajectory during the torque phase. However, it is not guaranteed that the motor will eventually synchronize with Gear 1 speed when these conditions are maintained. If the synchronization time t_s exists, it can be computed from the system model. Substituting $T_\mathrm{m} = T_\mathrm{max}$ and $T_1 = 0$ in Equation (1.9), the motor acceleration is

$$\ddot{\theta}_\mathrm{m} = I_\mathrm{m}^{-1} \left(-c_\mathrm{m}\dot{\theta}_\mathrm{m} + T_\mathrm{max} - T_2\right). \tag{2.18}$$

Assuming that T_2 is constant during the inertia phase, Equation (2.18) becomes a linear ordinary differential equation. It can be solved to obtain the evolution of $\dot{\theta}_\mathrm{m}(t)$ from the initial condition $\dot{\theta}_\mathrm{m}(0) = i_2\dot{\theta}_\mathrm{out}(0)$. The result is

$$\dot{\theta}_\mathrm{m}(t) = \exp\left(-\frac{c_\mathrm{m}}{I_\mathrm{m}} t\right) i_2 \dot{\theta}_\mathrm{out}(0) + \frac{T_\mathrm{max} - T_2}{c_\mathrm{m}} \left[1 - \exp\left(-\frac{c_\mathrm{m}}{I_\mathrm{m}} t\right)\right]. \tag{2.19}$$

When the motor synchronizes with Gear 1 speed, $\dot{\theta}_\mathrm{m} = i_1\dot{\theta}_\mathrm{out}$. Substituting this relation into Equation (2.19), and using the fact that $\dot{\theta}_\mathrm{out}(t) = (v_\mathrm{i} + a_\mathrm{r} t)/r_\mathrm{w}$, Equation (2.17) is obtained. For the motor to synchronize with Gear 1 speed, Equation (2.17) must have a solution, otherwise a no-jerk gearshift cannot be obtained. This proves the necessity part by

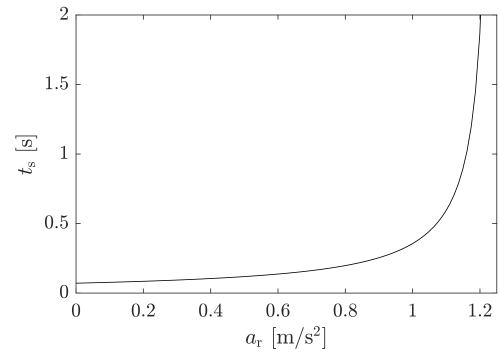


Figure 2.8: Example limitations for a power-on downshift with the example vehicle, when the gearshift is initiated at $v_i = 18\,\mathrm{km/h}$, and a_r is varied. When the desired vehicle acceleration a_r is increased, the time t_s required for the inertia phase to complete also increases.

contraposition.

(Sufficiency): By construction, the actuation strategy described in the first paragraph of this proof is sufficient for a no-jerk gearshift, given that Assumption 1 holds.

This actuation strategy is not unique. It could be that $T_m < T_{max}$, but it would only make it harder to synchronize the motor with Gear 1 speed. This can be seen from Equation (2.18). If Clutch 1 is a one-way clutch, then $T_1 = 0$ until the motor synchronizes, as per Equation (1.13). If Clutch 1 is a friction clutch, then it is possible to activate T_1 before t_s. But Equation (1.11) indicates that T_1 would be negative, so Equation (1.10) dictates that T_2 increases by $|T_1|(i_1/i_2)$ to respect the no-jerk conditions, and since $i_1 > i_2$, $\ddot{\theta}_m$ would again be smaller than if $T_1 = 0$. Therefore, if a no-jerk gearshift can be completed with a different actuation strategy than the one presented in the necessity part of the proof, it can also be completed with this strategy. As a result, the existence of a solution to Equation (2.17) is a sufficient condition for the possibility of a no-jerk gearshift. □

2.3.2 Adaptations for planetary gearset architectures

In this section, Theorems 1–3 are adapted for planetary gearset architectures described by the general Equations (1.27)–(1.28).

Theorem 1 adaptation

With the new transmission model, the motor power at the end of the torque transfer phase—originally described by Equation (2.7)—now becomes

$$p_{\mathrm{m}}(t_{\mathrm{tr}}) = i_1\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}})\Bigg(c_{\mathrm{m}}i_1\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + \left(c_1 - \frac{c_4c_5}{c_8}\right)^{-1}\Bigg[\left(i_1 I_{\mathrm{m}} - \frac{c_4}{c_8}I_{\mathrm{out}}\right)\ddot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + \left(c_2 - \frac{c_4c_6}{c_8}\right)\left(c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}}(t_{\mathrm{tr}}) + i_{\mathrm{f}}^{-1}T_{\mathrm{o}}\right)\Bigg]\Bigg). \tag{2.20}$$

The proof for the necessity condition remains valid, as it it based on the limitations imposed by the one-way clutch, and these limitations still apply. The proof for the sufficiency condition requires to demonstrate that $p_{\mathrm{m}}(t_{\mathrm{tr}})$ is the maximal motor power required during the gearshift, which now ultimately depends on the coefficients c_1 to c_8, as can be seen by adapting Equation (2.10) into a new expression for the motor power:

$$p_{\mathrm{m}} = \dot{\theta}_{\mathrm{m}}\Bigg(c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + \left(c_1 - \frac{c_4c_5}{c_8}\right)^{-1}\Bigg[I_{\mathrm{m}}\ddot{\theta}_{\mathrm{m}} - \frac{c_4}{c_8}I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}} + \left(c_2 - \frac{c_4c_6}{c_8}\right)\left(c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} + i_{\mathrm{f}}^{-1}T_{\mathrm{o}}\right) - \left(c_3 - \frac{c_4c_7}{c_8}\right)T_1\Bigg]\Bigg). \tag{2.21}$$

In particular, for $T_1 = 0$ to maximize p_{m}, it must be that $-(c_3 - c_4c_7/c_8)(c_1 - c_4c_5/c_8)^{-1} < 0$. For the architecture in Figure 1.3b, this expression reduces to $-(\beta_1+1)/\beta_1$, so that indeed $T_1 = 0$ maximizes p_{m} since $-(\beta_1+1)/\beta_1 < 0$. Assuming that the other variables—i.e., $\dot{\theta}_{\mathrm{m}}$, $\ddot{\theta}_{\mathrm{m}}$, $\dot{\theta}_{\mathrm{out}}$, $\ddot{\theta}_{\mathrm{out}}$, and T_{o}—remain approximately constant during the torque phase, $p_{\mathrm{m}}(t_{\mathrm{tr}})$ is the maximal motor power required during the gearshift for the case of the architecture in Figure 1.3b.

Theorem 2 adaptation

The existence of a sufficient Δ_{m} such that $\Delta_{\mathrm{s}} \geq 0$ remains a necessary and sufficient condition for a no-jerk gearshift. The only difference is that T_{m} cannot jump to $p_{\max}/\dot{\theta}_{\mathrm{m}}$ at $t = 0$, as prescribed in Equation (2.14). In effect, T_{m} now appears in the second equation of motion—Equation (1.28). The effect of an increase in T_{m} on T_1 can be seen in Figure 2.5. To maintain T_{o}, $\dot{\theta}_{\mathrm{m}}$, and $\ddot{\theta}_{\mathrm{m}}$ such that the no-jerk conditions in Definition 1 are met, a sudden increase in T_{m} implies a sudden increase in T_1. This would violate any torque application rate limitation. Therefore, Theorem 2 remains valid, but T_{m} must be increased such that $\dot{T}$ is within limits.

Theorem 3 adaptation

The proof for Theorem 3 remains valid, but the necessary and sufficient condition described by Equation (2.17) must be adapted to the new transmission model. First, $\dot{\theta}_{\mathrm{m}}$ can be computed for the inertia phase by imposing $T_1 = 0$ in Equations (1.27) and (1.28).

$$\ddot{\theta}_{\mathrm{m}} = I_{\mathrm{m}}^{-1}\left(-\gamma c_{\mathrm{m}}\dot{\theta}_{\mathrm{m}} + \gamma T_{\mathrm{m}} - \tau\right), \tag{2.22}$$

$$\gamma = \left(c_1 - \frac{c_4 c_5}{c_8}\right), \tag{2.23}$$

$$\tau = \left(c_2 - \frac{c_4 c_6}{c_8}\right)\left(c_{\mathrm{o}}\dot{\theta}_{\mathrm{out}} + i_{\mathrm{f}}^{-1}T_{\mathrm{o}}\right) + \frac{c_4}{c_8}I_{\mathrm{out}}\ddot{\theta}_{\mathrm{out}}. \tag{2.24}$$

Following the argument in Section 2.3.2, T_{m} cannot jump to $T_{\max}$ at $t = 0$, as this would imply a jump in T_2, which would violate any rate limitation. Therefore, T_{m} should be gradually increased from $T_{\mathrm{m}}(0)$ to $T_{\max}$ at the beginning of the gearshift. For this proof adaptation however, this effect is neglected as the ramp up is assumed very fast, and $T_{\mathrm{m}} \approx T_{\max}$. Further, it is assumed that $\dot{\theta}_{\mathrm{out}}(t) = \dot{\theta}_{\mathrm{out}}(0)$ and $T_{\mathrm{o}}(t) = T_{\mathrm{o}}(0)$ for the duration of the inertia phase, which was also assumed in Theorem 3 by imposing T_2 constant for the duration of the inertia phase. Equation (2.22) is now a linear ordinary differential equation of the same form than that of Theorem 3. It can also be solved imposing $\dot{\theta}_{\mathrm{m}}(0) = i_2\dot{\theta}_{\mathrm{out}}(0)$. The adapted condition for the possibility of a no-jerk gearshift is

$$\begin{aligned}\exists\ t_{\mathrm{s}} > 0 \quad \text{s.t.} \quad 0 = -i_1\frac{v_{\mathrm{i}} + a_{\mathrm{r}}t_{\mathrm{s}}}{r_{\mathrm{w}}} &+ \exp\left(-\frac{\gamma c_{\mathrm{m}}}{I_{\mathrm{m}}}t_{\mathrm{s}}\right)i_2\frac{v_{\mathrm{i}}}{r_{\mathrm{w}}} \\ &+ \frac{\gamma T_{\max} - \tau}{\gamma c_{\mathrm{m}}}\left[1 - \exp\left(-\frac{\gamma c_{\mathrm{m}}}{I_{\mathrm{m}}}t_{\mathrm{s}}\right)\right].\end{aligned} \tag{2.25}$$

2.3.3 Other vehicle models and kinematic conditions

The theorems presented in this article are based on the general driveline model described in Section 1.3, which does not include tire slip and other nonlinear behaviors likely to occur on real-world vehicles. The theorems could be adapted to other vehicle models. This implies solving for the transmission output torque $T_{\mathrm{o}}(t)$ and velocity $\dot{\theta}_{\mathrm{out}}(t)$ using the new equations for the vehicle model and the no-vehicle-jerk constraint. Effectively, this would result in different conditions for a no-jerk gearshift than that of Definition 1. The theorems could be adapted according to the new definition, perhaps with additional mathematical complexity as more terms may become time-dependent.

Similarly, the theorems could be adapted to another kinematic conditions than the one imposed in this chapter, i.e., $\dddot{\theta}_{\mathrm{v}} = 0$. This also requires solving for new conditions on $T_{\mathrm{o}}(t)$ and $\dot{\theta}_{\mathrm{out}}(t)$, and adapting the theorems accordingly. The same remark holds: this may increase the mathematical complexity as more terms may become time-dependent.

Chapter 3

Learning gearshift controllers

This chapter presents a model-based reinforcement learning approach to tune a gearshift controller from gearshift trials. First the problem is introduced, and the experimental setup and corresponding system models are presented. The gearshift controller design is then outlined, followed by the learning algorithm, where PILCO is introduced along with Gaussian process regression. The chapter closes with the experimental results.

Contents

3.1 Motivations for gearshift controller learning 37

3.1.1 Review of controller designs 38

3.1.2 Review of controller parameter learning 39

3.2 Experimental setup 41

3.3 Gearshift controller design 44

3.4 Proposed learning method 47

3.4.1 Gaussian process regression 49

3.4.2 Simulated policy rollouts and uncertainty propagation 50

3.4.3 Cost function and its gradients 52

3.5 Experimental results 53

3.1 Motivations for gearshift controller learning

To perform smooth and consistent gearshifts, a multi-speed transmission requires a well-designed and well-calibrated gearshift controller whose development can be a challenge. For that, engineers first synthesize a controller: they define its mathematical structure and parametrization. Then, they find controller parameters that maximize a chosen performance objective. An initial set of parameters can be obtained from a principled method that rely on an approximate model of the transmission and vehicle dynamics. However, the final set of controller parameters is usually calibrated from gearshift trials on a physical transmission test bench. At this point, engineers rely solely on statistics to infer the best combination of parameters from the recorded gearshift trials, typically using design of experiments (DOE) [74]. This method can be time and resource consuming, sometimes requiring thousands of gearshift trials [75–77]. This is because despite modern advances [78], DOE-like methods treat the mapping from the controller parameters to the gearshift performance indicator as a black box. This leaves no choice but to generate a lot of data points by performing multiple gearshift trials with varied parameters, do statistical inference on the collected data set, and try to optimize the parameters with this information. This leads to wonder whether modern approaches in reinforcement learning can be used to assist in the development of gearshift controllers by better leveraging the data generated during the gearshift trials. In particular, it should be possible to better exploit prior knowledge of the approximate system dynamics.

To be interesting in practice, a learning approach to gearshift controller calibration should drastically reduce the number of gearshift trials required. Moreover, the method should yield controllers that perform well under varying operating conditions—not just the specific conditions under which the training data was obtained. Because the number of gearshift trials would be reduced, engineers should be able to synthesize and iterate through a wide range of controller types with varying parametrization. This brings about the last requirement: the learning method should be easily amendable to various controller designs. This is important for the development of multi-speed transmissions for electric vehicles, as this emerging technology may not have converged to well established gearshift control strategies. This chapter presents a gearshift controller tuning method based on reinforcement learning. The method is argued to be interesting in practice for automotive engineers on the basis of the considerations introduced above.

3.1.1 Review of controller designs

In terms of feedback controller type and principled design method, several approaches are reported in the literature. In [70], researchers first linearized the system along the reference gearshift trajectory, then formulated an optimal control problem and used dynamic programming to solve it. The controller is a feedforward plus linear feedback controller. Researchers in [40] first obtained an open-loop optimal controller using the Pontryagin's minimum principle, and closed the feedback loop with the design of a backstepping controller. Article [79] presents a backstepping controller that integrates lookup tables for the strongly nonlinear elements of the powertrain model, such as the torque converter. In [80], the same powertrain nonlinearities are considered for the design of a feedforward controller this time, which is used in combination with a linear feedback controller. In [37], the motor torque is controlled with a proportional-integral-derivative (PID) feedback controller during the inertia phase. The PID gains were tuned by shaping the closed-loop transfer function between the motor torque and the motor speed. In [81], researchers designed a robust feedforward-feedback controller for the inertia phase using μ-synthesis. The solution has guaranteed robust stability and robust performance given the parametric uncertainty included in the model. Similarly, researchers in [82] reported a complete gearshift solution which includes a multi-variable feedback controller designed with the robust $\mathcal{H}_\infty$ method. Finally, model predictive control was also used for clutch-to-clutch gearshift control [83].

In this research, we used a linear feedback controller and tuned its initial parameters with the linear quadratic regulator (LQR) method based on a nominal model of the linear system dynamics. This yields a feedback controller structure with an interesting number of parameters to tune—the eight entries of the K_c matrix, see Section 3.3. The feedback controller is also complemented with a feedforward signal, which adds four more parameters to be learned.

3.1.2 Review of controller parameter learning

None of the methods reviewed in Section 3.1.1 learns the feedback controller from iterated trials with a transmission test bench. The closest to it would be the work reported in [75], where researchers use iterative learning control (ILC) to tune the parametrization of a feedforward signal for the closure of Clutch 2. The experimental results show that very few trials are required to learn appropriate parameters. However, ILC directly iterates on control signals [84], therefore this method is ill-suited for tuning the feedback portion of gearshift controllers.

The principled control design methods introduced in Section 3.1.1 all rely on a system model to compute the controller parameters. It is customary to use system identification methods to obtain, calibrate, or validate system models [13]. With an identified model, one can hope to get a superior controller since the design method now relies on a model that better represents the true system dynamics. However, this research aims to go beyond this approach and proposes using reinforcement learning to concurrently and iteratively gain knowledge about the system dynamics and tune the control parameters.

As discussed in Appendix A, model-free approaches to reinforcement learning tend to require many more interactions with the environment [85]. Thus, we chose a model-based approach for this work. More specifically, we chose PILCO [20, 86], which uses Gaussian processes (GP) [15,87] to efficiently learn the system dynamics. To name a few, this method was used to efficiently tune linear controllers [88] and multivariate PID controllers [89] for robotic arm applications. In PILCO, the control policy is iterated with analytic gradients obtained from simulated policy rollouts using the learned model. However in this work, we make use of the automatic gradient functionalities of TensorFlow [90] for additional speed and flexibility in the implementation of the method. The primary criticism of PILCO is that the method scales poorly for problems of higher dimensions [91], which should not be an issue in this work.

For higher dimensional problems, an alternative would be guided policy search [92]. In this method, control policies are randomly searched in a model-free fashion, but the search is guided by optimal control solutions obtained with differential dynamic programming [93], using a nominal model of the system dynamics. This method is interesting for avoiding local minima in complex high-dimensional control problems. But because our control problem is quite small, this method is likely to be less efficient than PILCO and provide little added benefit.

Another alternative would be Coarse-ID control [94]. This method starts with the identification of a linear system dynamics with least squares estimation. Then a bootstrap technique it proposed to bound the error between the real dynamics and the identified model. Finally, a controller is synthesized by solving a robust optimization problem. Researchers introduced a method for LQR controller synthesis, but the work could be extended to other controller types, and perhaps feedforward signals as well. Coarse-ID is very close to the traditional system identification plus principled controller synthesis method discussed at the beginning of this section, with the only difference being that machine learning is used twice: once for the system identification, and again for the uncertainty estimation. We subscribe to the idea that researchers should strive to reduce the gap between reinforcement learning and control theory [95]. This study is an attempt to do so.

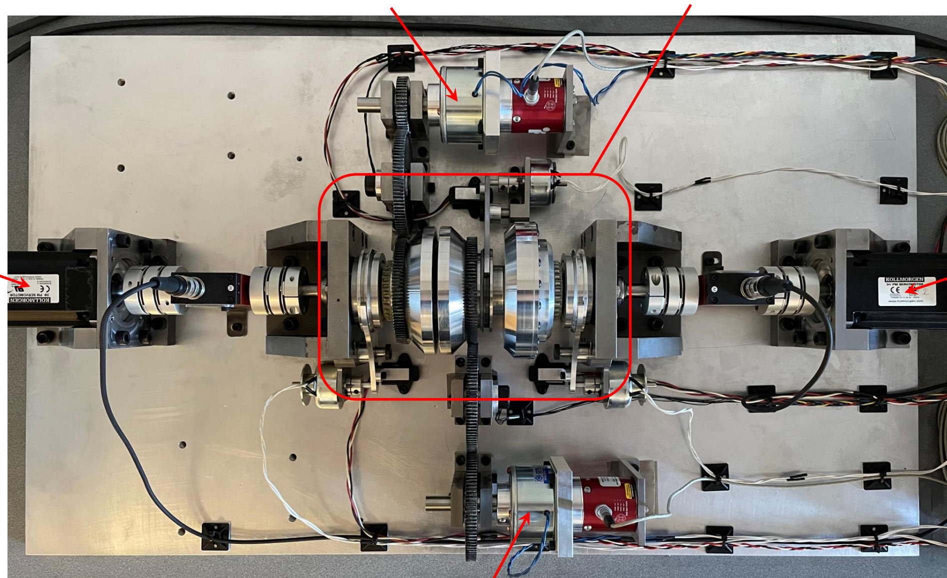


Figure 3.1: Reduced-scale electric vehicle transmission prototype.

3.2 Experimental setup

The experimental setup consists of a multi-speed transmission prototype—it implements the transmission design of Section 1.5. The test bench is shown on Figure 3.1. A section view of the planetary gearset is shown on Figure 3.2. The input motor simply represents the electric motor in an electrical drivetrain. Both clutches are friction-plate electromagnetic brakes: a spring keeps the plates apart, until the electromagnet is activated, which magnetizes the floating plate and brings the braking surface into contact. A section view of a clutch is shown on Figure 3.3. The load motor on Figure 3.1 inputs a torque that corresponds to a simulated driveline torque. This resistive torque is determined by a driveline model that is simulated in real-time. The driveline model is shown on Figure 3.6 and further discussed in Section 3.3.

A schematic representation of the control loop is shown on Figure 3.4. The real-time controller is implemented on a CompactRIO. It includes the gearshift controller whose parameters are to be tuned with the proposed learning method. The gearshift controller outputs three torque commands: the motor torque, and the two clutch torques. The motor torque command is sent directly to a motor drive. The motor drives manage the closed-loop control of the actual motor torques based on current feedback.

The clutch drives are simple H-bridge drivers however, and take as an input a pulse-width modulated (PWM) signal. In this setup, we implemented a closed-loop control of the actual

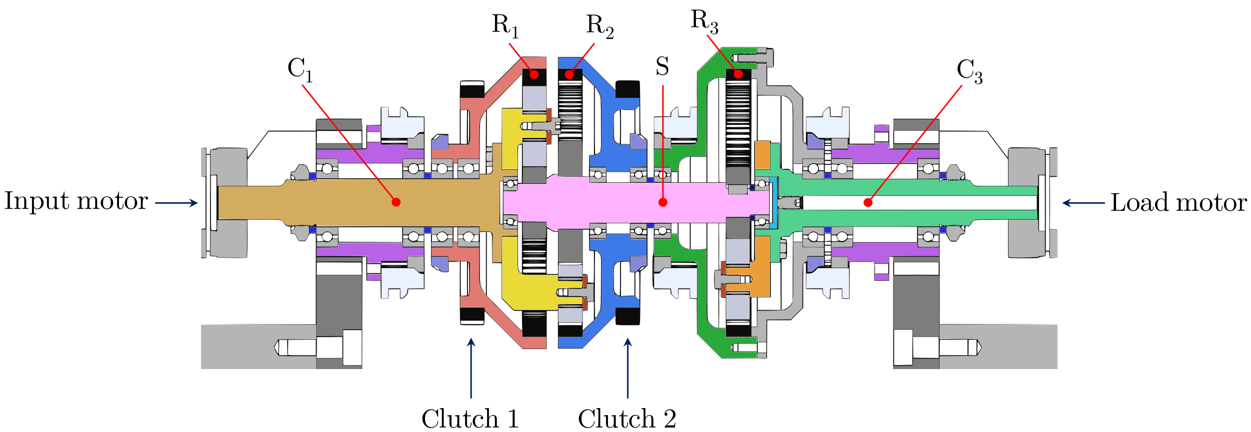


Figure 3.2: Section view of the planetary gearset.

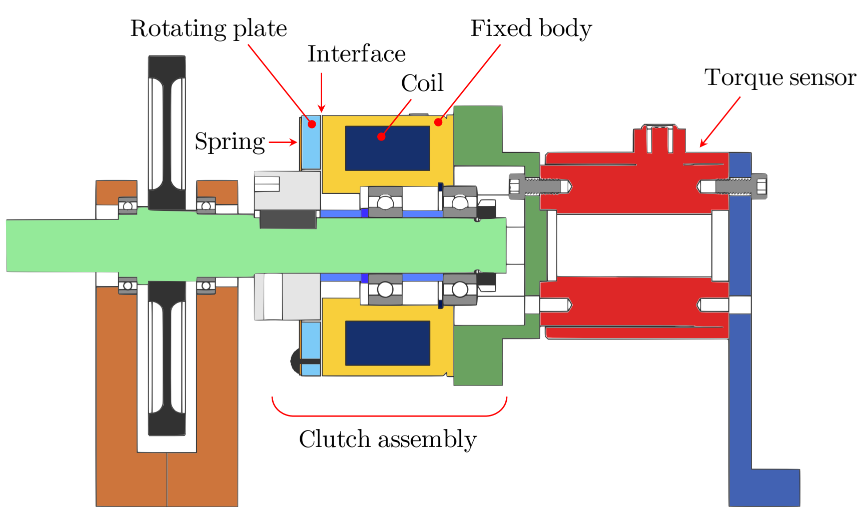


Figure 3.3: Section view of the clutch assembly.

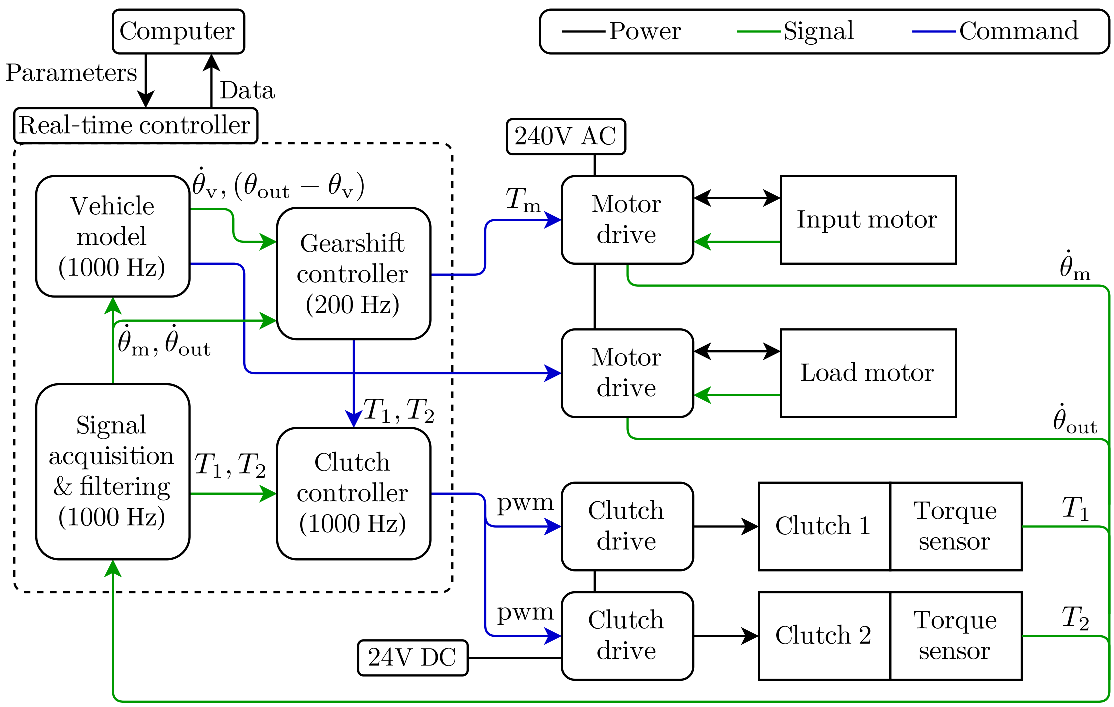
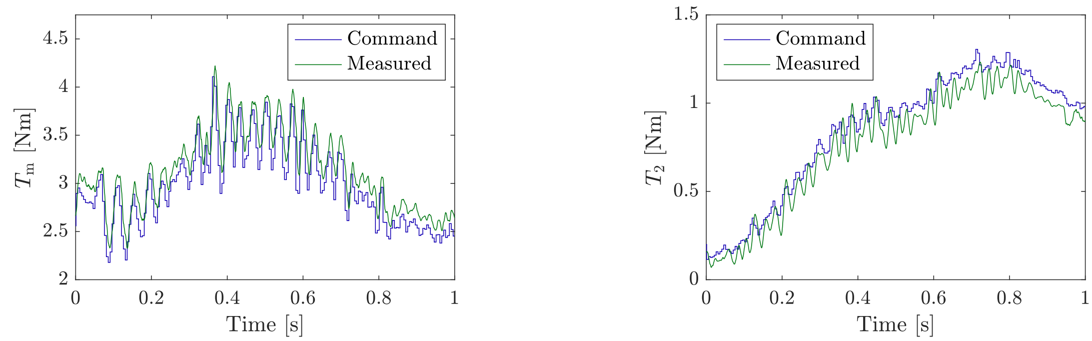


Figure 3.4: Experimental setup.

clutch torque output. Torque sensors are installed between the fixed part of the brakes and the ground plate, as shown on Figure 3.3. The clutch controller modulates the PWM signals to track the desired torque values given by the gearshift controller. Typically, automotive transmissions do not include such torque sensors, so this kind of tracking is impossible in practice. However, automotive clutches are also typically better suited for the task at hand, and engineers spend significant efforts to characterize them. Given that the research objective is to learn a gearshift controller from reinforcement learning, the closed-loop control of the friction clutches was deemed acceptable. The clutch controller is a simple proportional integral controller tuned heuristically until an acceptable tracking is obtained.

Figure 3.5 shows both the commanded torque and the measured torque for the motor and Clutch 2, for an example gearshift. Unsurprisingly, the motor torque is well tracked by the motor drive. Figure 3.1 shows that there is also a torque sensor between the input motor and the planetary gearset, which was used for the torque measurement of Figure 3.5a and for debugging purposes. However, this torque signal is not used in the control loop of Figure 3.4. From Figure 3.5, the clutch torque can also be considered sufficiently tracked.

The signals are acquired at 1000 Hz. A fourth-order Butterworth filter with a cutoff frequency of 450 Hz was implemented for anti-aliasing purposes. The gearshift controller runs at 200 Hz, which is sufficient to cover the dynamics of the problem. An increase in the con-



(a) Motor torque. **(b)** Clutch 2 torque.

Figure 3.5: Comparison of the command signals and the measured torques.

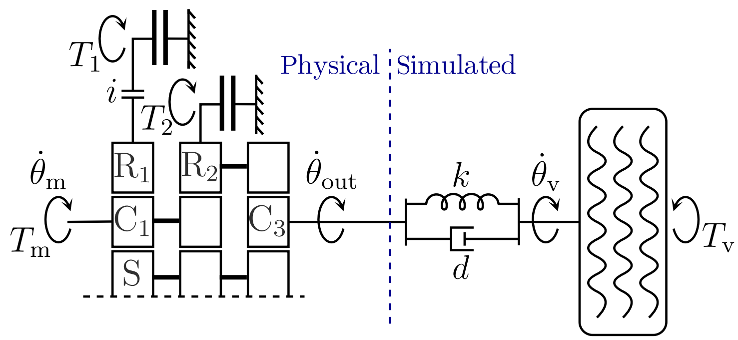


Figure 3.6: Driveline and vehicle model. The components on the left—the motor, planetary gearset, and clutches—are the physical components of Figure 3.1. The vehicle model on the right is simulated in real-time and used to generate the torque command on the load motor.

troller frequency results in an increase in the number of time steps that must be simulated during controller learning. Here 200 Hz was deemed an appropriate balance.

3.3 Gearshift controller design

The system model used for the controller design is shown on Figure 3.6. The system is in part physically realized with the input motor, the clutches, and the planetary gearset, and in part simulated in real-time with a simple driveline model. In the displayed configuration, the planetary gearset is composed of the five rotating bodies labeled on Figure 3.6: S, C$_1$, C$_3$, R$_1$, R$_2$. Choosing the motor speed $\dot{\theta}_m$, the output shaft speed $\dot{\theta}_{out}$ and the vehicle speed

$\dot{\theta}_{\mathrm{v}}$ as the general coordinates, the equations of motion are

$$\ddot{\theta}_{\mathrm{m}} = c_1 T_{\mathrm{m}} + c_2\big(k(\theta_{\mathrm{out}} - \theta_{\mathrm{v}}) + d(\dot{\theta}_{\mathrm{out}} - \dot{\theta}_{\mathrm{v}})\big) + c_3 T_1 + c_4 T_2, \tag{3.1}$$

$$\ddot{\theta}_{\mathrm{out}} = c_5 T_{\mathrm{m}} + c_6\big(k(\theta_{\mathrm{out}} - \theta_{\mathrm{v}}) + d(\dot{\theta}_{\mathrm{out}} - \dot{\theta}_{\mathrm{v}})\big) + c_7 T_1 + c_8 T_2, \tag{3.2}$$

$$\ddot{\theta}_{\mathrm{v}} = I_{\mathrm{v}}^{-1}\big(k(\theta_{\mathrm{out}} - \theta_{\mathrm{v}}) + d(\dot{\theta}_{\mathrm{out}} - \dot{\theta}_{\mathrm{v}})\big) - I_{\mathrm{v}}^{-1} T_{\mathrm{v}}, \tag{3.3}$$

where c_1 to c_8 are constants that regroup parameters such as the inertias of the rotating elements and the number of teeth on the meshing ones. Following the driveline model of Chapter 1, I_{v} represents the equivalent vehicle inertia, k is the equivalent driveline stiffness and d, its damping. By choosing the set of states $\mathbf{x} = [\dot{\theta}_{\mathrm{m}}, \dot{\theta}_{\mathrm{out}}, \dot{\theta}_{\mathrm{v}}, (\theta_{\mathrm{out}} - \theta_{\mathrm{v}})]^\top$ and control inputs $\mathbf{u} = [T_{\mathrm{m}}, T_1, T_2]^\top$ for the system, a linear state-space representation of its dynamics can be obtained, where

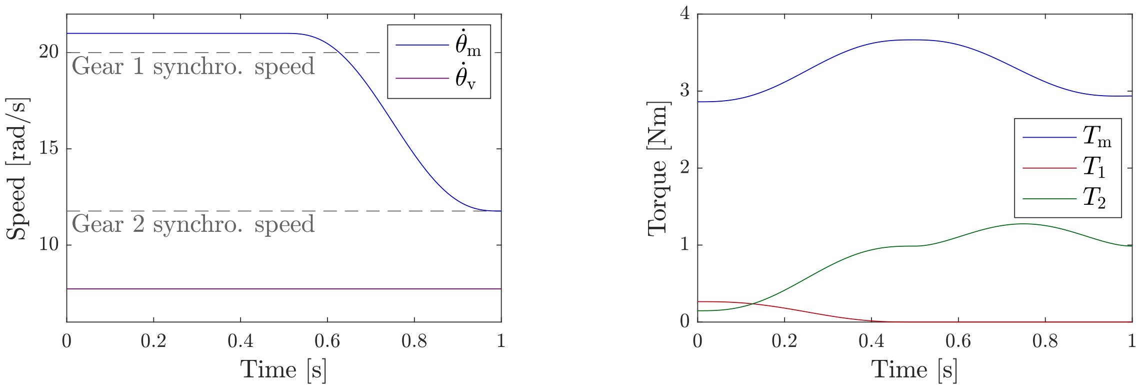
$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + \mathbf{t}_0, \tag{3.4}$$

$$A = \begin{bmatrix} 0 & c_2 d & -c_2 d & c_2 k \\ 0 & c_6 d & -c_6 d & c_6 k \\ 0 & I_{\mathrm{v}}^{-1} d & -I_{\mathrm{v}}^{-1} d & I_{\mathrm{v}}^{-1} k \\ 0 & 1 & -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} c_1 & c_3 & c_4 \\ c_5 & c_7 & c_8 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.5}$$

$$\mathbf{t}_0 = [0, 0, -I_{\mathrm{v}}^{-1} T_{\mathrm{v}}, 0]^\top. \tag{3.6}$$

Since they are simulated, the parameters I_{v}, k, and d must be defined. First, I_{v} is set such that, in first gear, the equivalent inertia of the motor and transmission projected at the vehicle level is 10% that of I_{v}. This is a realistic ratio for a real-world vehicle [37, 96, 97]. Then, k and d are set such that, in first gear again, the natural frequency of the driveline is 5 Hz, and its damping ratio is 0.15, which are also typical values of driveline dynamics.

Following the clutch-to-clutch gearshift strategies discussed in Chapter 2, a prescribed gearshift trajectory $\bar{\mathbf{x}}$ and a nominal torque command $\bar{\mathbf{u}}$ can be computed using Equations (3.4)–(3.6). The rest of this paragraph outlines the procedure, and the results are shown on Figure 3.7. The controller's objective is to track the prescribed state trajectory $\bar{\mathbf{x}}$. In this work, the main gearshift performance indicator is maintaining a constant vehicle speed $\dot{\theta}_{\mathrm{v}}$. The gearshift begins under the following conditions: $\dot{\theta}_{\mathrm{m}} = 20\,\mathrm{rad/s}$ and $T_{\mathrm{v}} = 3.5\,\mathrm{Nm}$. For this study, T_{v} is kept constant throughout the gearshift, which is a common assumption in gearshift control research [40, 82]. The motor speed $\dot{\theta}_{\mathrm{m}}$ is kept 1 rad/s above Gear 1 synchronization speed during the torque phase, and is then smoothly brought down to Gear 2 speed during the inertia phase. A constant output shaft speed $\dot{\theta}_{\mathrm{out}}$ that matches $\dot{\theta}_{\mathrm{v}}$ is prescribed,



(a) Prescribed trajectory $\bar{\mathbf{x}}$

(b) Nominal torque command $\bar{\mathbf{u}}$

Figure 3.7: The gearshift begins with a torque phase [0 s, 0.5 s], where $\dot{\theta}_\mathrm{m}$ is kept above the Gear 1 synchronization speed and T_1 is gradually reduced to zero. It ends with an inertia phase [0.5 s, 1 s], where $\dot{\theta}_\mathrm{m}$ is brought down to the Gear 2 synchronization speed.

as well as a constant driveshaft elongation $(\theta_\mathrm{out} - \theta_\mathrm{v}) = T_\mathrm{v}/k$. With $\bar{\mathbf{x}}$ defined, the next step is to compute $\bar{\mathbf{u}}$. First, an idealized nominal torque command $\bar{\mathbf{u}}_0 = [T_{\mathrm{m},0}, T_{1,0}, T_{2,0}]^\top$ is computed by solving the system of Equations (3.4)–(3.6). The system trajectory $\bar{\mathbf{x}}$ is used to define the state variables. Then, an arbitrary trajectory is imposed for $T_{1,0}$: it starts at the Clutch 1 torque required at the beginning of the gearshift, and ends at zero at the end of the torque phase. The rest of $\bar{\mathbf{u}}$, i.e., $T_{\mathrm{m},0}$ and $T_{2,0}$, can be computed by solving for the remaining terms in Equations (3.4)–(3.6). Finally, the actual $\bar{\mathbf{u}}$ is defined as follows:

$$T_\mathrm{m} = T_{\mathrm{m},0} + a_1\dot{\theta}_\mathrm{m}/\dot{\theta}_\mathrm{out} + a_2, \tag{3.7}$$

$$T_1 = a_3 T_{1,0}, \tag{3.8}$$

$$T_2 = a_4 T_{2,0}, \tag{3.9}$$

where $\{a_1, \ldots, a_4\}$ are parameters for the feedforward signal $\bar{\mathbf{u}}$ of the gearshift controller. These parameters help to account for missing terms in Equations (3.4)–(3.6) such as friction in the planetary gearset, as well as other discrepancies between the nominal model and the real system dynamics. These are the four parameters that the learning algorithm will vary in order to tune $\bar{\mathbf{u}}$. An initial value for these parameters is defined heuristically from simple measurements done on the test bench, such as estimating friction from constant-speed runs.

The complete controller has the form

$$\mathbf{u} = \bar{\mathbf{u}} + K_\mathrm{c}(\bar{\mathbf{x}} - \mathbf{x}), \tag{3.10}$$

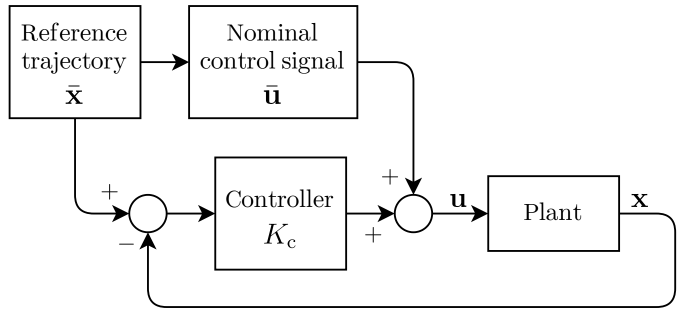


Figure 3.8: Gearshift controller setup. The control signal $\mathbf{u} = \bar{\mathbf{u}} + K_c(\bar{\mathbf{x}} - \mathbf{x})$ is the sum of feedforward ($\bar{\mathbf{u}}$) and state-feedback components.

where $\mathbf{u}$ is the control signal, $\bar{\mathbf{u}}$ is the (feedforward) nominal control signal, and $K_c(\bar{\mathbf{x}} - \mathbf{x})$ is the feedback term. The control loop is pictured on Figure 3.8. The linear controller K_c is obtained by solving an LQR problem where the second column of B is removed. In effect, only T_m and T_2 are feedback controlled. Even when removing the second column of B, the system is still controllable. The resulting controller is a 2×4 matrix, which adds eight more controller parameters to tune during training. The next section presents the algorithm used to tune the 12 controller parameters $\boldsymbol{\psi}$ from gearshift trials.

3.4 Proposed learning method

The learning algorithm used in this work is an altered version of PILCO. The method is schematically represented in Figure 3.9, and outlined in Algorithm 1. The learning problem is formulated in discrete time; the states and control actions are still continuous. The system model used for the simulated policy rollouts is

$$\mathbf{x}_{[t]} = A_d \mathbf{x}_{[t-1]} + B_d \mathbf{u}_{[t-1]} + f(\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]}) + \mathbf{t}_0, \tag{3.11}$$

$$\mathbf{u}_{[t-1]} = \pi(\mathbf{x}_{[t-1]}, \boldsymbol{\psi}) = \bar{\mathbf{u}}_{[t-1]} + K_c(\bar{\mathbf{x}}_{[t-1]} - \mathbf{x}_{[t-1]}), \tag{3.12}$$

where A_d, B_d, and $\mathbf{t}_0$ are obtained by discretizing the system in Equations (3.4)–(3.6). The dynamics of Equation (3.11) is composed of a known nominal model ($A_d \mathbf{x}_{[t-1]} + B_d \mathbf{u}_{[t-1]} + \mathbf{t}_0$) and an unknown function $f(\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]})$. This unknown dynamics $f(\mathbf{x}, \mathbf{u})$ is to be learned from gearshift trials on the test bench, which is addressed in Section 3.4.1. The control policy $\pi(\mathbf{x}, \boldsymbol{\psi})$ is deterministic, and $\boldsymbol{\psi}$ regroups the 12 policy parameters. The tuning of these parameters is the subject of Section 3.4.3.

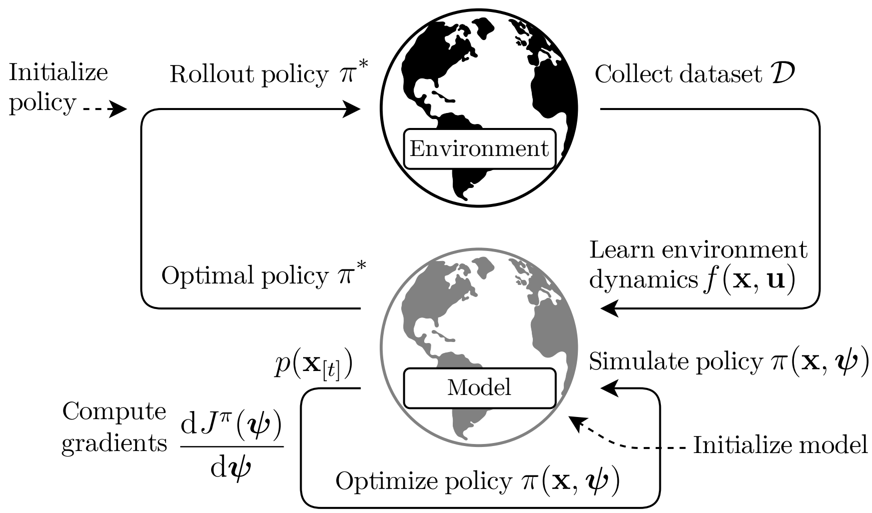


Figure 3.9: The model-based reinforcement learning method used to tune the parameters $\boldsymbol{\psi}$ of the control policy π.

Algorithm 1: PILCO for gearshift controllers

Result: Learned policy $\pi(\mathbf{x}, \boldsymbol{\psi})$.

Initialize $\boldsymbol{\psi}$: initialize K_c from LQR with a nominal model, initialize $\bar{\mathbf{u}}$ heuristically.

while π *not learned* **do**

- Rollout the policy π on the test bench and collect a dataset $\mathcal{D}$.
- Learn the unknown system dynamics $f(\mathbf{x}, \mathbf{u})$ with Gaussian processes.
- **while** π *not optimized* **do**
  - Simulate a policy rollout: compute the state probability distribution $p(\mathbf{x}_{[t]})\,\forall t \in \{0, \ldots, T\}$, the cost function $J^\pi(\boldsymbol{\psi})$, and the gradients $\frac{\mathrm{d}J^\pi(\boldsymbol{\psi})}{\mathrm{d}\boldsymbol{\psi}}$.
  - Iterate the policy parameters $\boldsymbol{\psi}$ using the gradients $\frac{\mathrm{d}J^\pi(\boldsymbol{\psi})}{\mathrm{d}\boldsymbol{\psi}}$.

3.4.1 Gaussian process regression

The unknown dynamics $f(\mathbf{x}, \mathbf{u})$ is learned using Gaussian processes. GPs approximate functions with a scalar output, so D functions $f_d(\mathbf{x}, \mathbf{u})$ need to be learned, i.e., one for every state we wish to predict. The structure of the unknown function is $f(\mathbf{x}, \mathbf{u}) = [f_1(\mathbf{x}, \mathbf{u}), \dots, f_D(\mathbf{x}, \mathbf{u})]^\top$. For each dimension d, n training targets and feature vectors are obtained with

$$y_{d\,[i]} = x_{d\,[i]} - [A_\mathrm{d}\mathbf{x}_{[i-1]} + B_\mathrm{d}\mathbf{u}_{[i-1]} + \mathbf{t}_0]_d, \tag{3.13}$$

$$\mathbf{z}_{[i]} = [\mathbf{x}_{[i-1]}^\top, \mathbf{u}_{[i-1]}^\top]^\top. \tag{3.14}$$

Each dimension d has its own target vector $\mathbf{y}_d$ composed of the $y_{d\,[i]}$ elements computed above. All dimensions share the same set of corresponding feature vectors regrouped in $Z = [\mathbf{z}_{[1]}, \dots, \mathbf{z}_{[n]}]$. Effectively, the GP learns the difference between the nominal model and the real (measured) system dynamics. GPs are stochastic processes characterized by a mean $m(\mathbf{z})$ and a kernel function $k(\mathbf{z}, \mathbf{z}')$. Here, we choose the mean function $m(\mathbf{z}) := 0$, and the square exponential kernel function

$$k(\mathbf{z}, \mathbf{z}') := \sigma_\mathrm{f}^2 \exp(-\tfrac{1}{2}(\mathbf{z} - \mathbf{z}')^\top \Lambda^{-1} (\mathbf{z} - \mathbf{z}')), \tag{3.15}$$

where σ_f^2 is the signal variance and $\Lambda = \mathrm{diag}([l_1^2, \dots, l_{D+F}^2])$ is a diagonal matrix composed of the characteristics length-scales. These are the hyper-parameters of the Gaussian process, and each dimension d has its own set of hyper-parameters. The last hyper-parameter in this problem is the noise variance σ_ϵ^2, which will appear in Equations (3.16) and (3.17).

In this work, the system dynamics are modeled with a linear nominal model and a Gaussian process with a zero mean function $m(\mathbf{z}) = 0$. This is equivalent to modeling the dynamics with no nominal model and a Gaussian process with a linear mean function. In the original implementation of PILCO [20], no nominal model is used and the mean function is also zero. This means that the entirety of the system dynamics has to be learned from data. Unsurprisingly, researchers in [98] showed that using a linear model as a mean function accelerates the learning process, which motivates the use of the linear nominal model in our case.

For a deterministic test point $\mathbf{z}_*$, the output of $f_d(\mathbf{z})$ will be normally distributed with mean and variance

$$\mu_d(\mathbf{z}_*) = K_{\mathbf{z}_*\mathbf{z}}(K_{\mathbf{zz}} + \sigma_\epsilon^2 I)^{-1}\mathbf{y}_d, \tag{3.16}$$

$$\Sigma_d(\mathbf{z}_*) = k(\mathbf{z}_*, \mathbf{z}_*) - K_{\mathbf{z}_*\mathbf{z}}(K_{\mathbf{zz}} + \sigma_\epsilon^2 I)^{-1}K_{\mathbf{z}_*\mathbf{z}}^\top, \tag{3.17}$$

$$\text{where} \quad K_{\mathbf{zz}} = \begin{bmatrix} k(\mathbf{z}_{[1]}, \mathbf{z}_{[1]}) & \cdots & k(\mathbf{z}_{[1]}, \mathbf{z}_{[n]}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{z}_{[n]}, \mathbf{z}_{[1]}) & \cdots & k(\mathbf{z}_{[n]}, \mathbf{z}_{[n]}) \end{bmatrix}, \tag{3.18}$$

$$K_{\mathbf{z}_*\mathbf{z}} = \begin{bmatrix} k(\mathbf{z}_*, \mathbf{z}_{[1]}) & \cdots & k(\mathbf{z}_*, \mathbf{z}_{[n]}) \end{bmatrix}. \tag{3.19}$$

The accuracy of the prediction $f_d(\mathbf{z}_*)$ depends on having appropriate GP hyper-parameters. It is possible to tune the set of hyper-parameters $\boldsymbol{\theta}_d$ by maximizing the logarithm of the marginal likelihood of the observed data points $\mathbf{y}_d$, such as suggested in [15], with

$$\log p(\mathbf{y}_d|Z, \boldsymbol{\theta}_d) = -\frac{1}{2}\mathbf{y}_d^\top K^{-1}\mathbf{y}_d - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi, \tag{3.20}$$

where $K = K_{\mathbf{zz}} + \sigma_\epsilon^2 I$. The likelihood is maximized with a gradient-based optimization algorithm using

$$\frac{\partial}{\partial \theta_{d\,[j]}} \log p(\mathbf{y}_d|Z, \boldsymbol{\theta}_d) = \frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - K^{-1}\right)\frac{\partial K}{\partial \theta_{d\,[j]}}\right), \tag{3.21}$$

$$\boldsymbol{\alpha} = K^{-1}\mathbf{y}_d. \tag{3.22}$$

3.4.2 Simulated policy rollouts and uncertainty propagation

Algorithm 1 requires to simulate a rollout from an initial state $\mathbf{x}_0$, under a given control policy π, from time $t = 0$ to $t = T$. This means computing the state distributions $\{p(\mathbf{x}_{[0]}), \ldots, p(\mathbf{x}_{[T]})\}$. Computing $p(\mathbf{x}_{[t]})$ by performing the integration

$$p(\mathbf{x}_{[t]}) = \iint p(\mathbf{x}_{[t]}|\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]})p(\mathbf{u}_{[t-1]}|\mathbf{x}_{[t-1]})p(\mathbf{x}_{[t-1]})\,\mathrm{d}\mathbf{x}_{[t-1]}\,\mathrm{d}\mathbf{u}_{[t-1]} \tag{3.23}$$

is generally intractable. In particular, $p(\mathbf{x}_{[t]}|\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]})$ is quite challenging. Recall that the results of Equations (3.16) and (3.17) were for a deterministic test input $\mathbf{z}_*$. When $\mathbf{x}_{[t-1]}$ and $\mathbf{u}_{[t-1]}$ are non-deterministic, the output of a Gaussian process is in general not Gaussian, even if $\mathbf{x}_{[t-1]}$ and $\mathbf{u}_{[t-1]}$ are themselves normally distributed. In the original PILCO implementation, they approximate the output distribution as Gaussian, and they obtain the mean and variance of the distribution using exact moment matching [20]. In this work, we take a some-what simpler approach: we also approximate the output as a Gaussian distribution, but we get the mean and variance using an approximate solution. Assuming $\mathbf{z}_* \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{z}_*}, \Sigma^{\mathbf{z}_*})$, and using a first order Taylor expansion around $\boldsymbol{\mu}^{\mathbf{z}_*}$—see [99]—the mean and variance become

$$\mu_d(\mathbf{z}_*) = \mu_d(\boldsymbol{\mu}^{\mathbf{z}_*}), \tag{3.24}$$

$$\Sigma_d(\mathbf{z}_*) = \Sigma_d(\boldsymbol{\mu}^{\mathbf{z}_*}) + \left.\frac{\partial \mu_d(\mathbf{z}_*)}{\partial \mathbf{z}_*}\right|_{\mathbf{z}_*=\boldsymbol{\mu}^{\mathbf{z}_*}} \Sigma^{\mathbf{z}_*} \left.\frac{\partial \mu_d(\mathbf{z}_*)}{\partial \mathbf{z}_*}\right|_{\mathbf{z}_*=\boldsymbol{\mu}^{\mathbf{z}_*}}^{\top}. \tag{3.25}$$

This approach is also chosen by researchers in [100], where GPs are used in the context of model predictive control. This requires to introduce the derivative of the mean of the GP prediction with respect to a deterministic test point $\mathbf{z}_*$

$$\frac{\partial \mu_d(\mathbf{z}_*)}{\partial \mathbf{z}_*} = -(K_{\mathbf{z}_*\mathbf{z}} \odot \mathbf{y}_d^{\top}(K_{\mathbf{z}\mathbf{z}} + \sigma_\epsilon^2 I)^{-1})\tilde{Z}_*^{\top}\Lambda^{-1}, \tag{3.26}$$

where $\tilde{Z}_* = [\mathbf{z}_* - \mathbf{z}_{[1]}, \ldots, \mathbf{z}_* - \mathbf{z}_{[n]}]$, and $\odot$ represents an element-wise product.

Assuming that $\mathbf{x}_{[t-1]} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}, \Sigma^{\mathbf{x}}_{[t-1]})$, the mean and variance of the control signals are

$$\boldsymbol{\mu}^{\mathbf{u}}_{[t-1]} = \bar{\mathbf{u}}_{[t-1]} + K_{\mathrm{c}}(\bar{\mathbf{x}}_{[t-1]} - \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}), \tag{3.27}$$

$$\Sigma^{\mathbf{u}}_{[t-1]} = K_{\mathrm{c}}\Sigma^{\mathbf{x}}_{[t-1]}K_{\mathrm{c}}^{\top}. \tag{3.28}$$

Then $p(\mathbf{z}_{[t-1]})$ can be expressed as

$$\mathbf{z}_{[t-1]} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]} \\ \bar{\mathbf{u}}_{[t-1]} + K_{\mathrm{c}}(\bar{\mathbf{x}}_{[t-1]} - \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}) \end{bmatrix}, \begin{bmatrix} \Sigma^{\mathbf{x}}_{[t-1]} & \Sigma^{\mathbf{x}}_{[t-1]}K_{\mathrm{c}}^{\top} \\ K_{\mathrm{c}}\Sigma^{\mathbf{x}}_{[t-1]} & K_{\mathrm{c}}\Sigma^{\mathbf{x}}_{[t-1]}K_{\mathrm{c}}^{\top} \end{bmatrix}\right). \tag{3.29}$$

For convenience, Equations (3.11) and (3.12) are regrouped into

$$\mathbf{x}_{[t]} = (A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})\mathbf{x}_{[t-1]} + f(\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]}) + B_{\mathrm{d}}(\bar{\mathbf{u}}_{[t-1]} + K_{\mathrm{c}}\bar{\mathbf{x}}_{[t-1]}) + \mathbf{t}_0, \tag{3.30}$$

from which an expression for $p(\mathbf{x}_{[t]})$ can be obtained as

$$\boldsymbol{\mu}^{\mathbf{x}}_{[t]} = (A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})\boldsymbol{\mu}^{\mathbf{x}}_{[t-1]} + \mu_{\mathrm{f}}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}) + B_{\mathrm{d}}(\bar{\mathbf{u}}_{[t-1]} + K_{\mathrm{c}}\bar{\mathbf{x}}_{[t-1]}) + \mathbf{t}_0, \tag{3.31}$$

$$\Sigma^{\mathbf{x}}_{[t]} = (A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})\Sigma^{\mathbf{x}}_{[t-1]}(A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})^{\top} + \Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]}), \tag{3.32}$$

where

$$\mu_{\mathrm{f}}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}) = [\mu_1(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}), \dots, \mu_D(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})]^\top, \tag{3.33}$$

$$\Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]}) = \mathrm{diag}([\Sigma_1(\mathbf{z}_{[t-1]}), \dots, \Sigma_D(\mathbf{z}_{[t-1]})]). \tag{3.34}$$

3.4.3 Cost function and its gradients

As is customary in reinforcement learning, the goal of Algorithm 1 is to minimize the expected long-term cost of following a policy π over a finite horizon of T time steps

$$J^\pi(\boldsymbol{\psi}) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]. \tag{3.35}$$

Following PILCO's original paper, the cost function used in this study is the saturating immediate cost

$$c(\mathbf{x}_{[t]}) = 1 - \exp\big(-\tfrac{1}{2}(\mathbf{x}_{[t]} - \bar{\mathbf{x}}_{[t]})^\top L^{-1}(\mathbf{x}_{[t]} - \bar{\mathbf{x}}_{[t]})\big), \tag{3.36}$$

where L^{-1} is a diagonal matrix whose elements dictate the width of the cost function for each of the state dimensions. For $\mathbf{x}_{[t]} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{x}}_{[t]}, \Sigma^{\mathbf{x}}_{[t]})$, the expectation of this cost function is

$$\mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})] = 1 - |I + \Sigma^{\mathbf{x}}_{[t]} L^{-1}|^{-1/2} \exp\big(-\tfrac{1}{2}(\boldsymbol{\mu}^{\mathbf{x}}_{[t]} - \bar{\mathbf{x}}_{[t]})^\top \tilde{S}(\boldsymbol{\mu}^{\mathbf{x}}_{[t]} - \bar{\mathbf{x}}_{[t]})\big), \tag{3.37}$$

$$\tilde{S} = L^{-1}(I + \Sigma^{\mathbf{x}}_{[t]} L^{-1})^{-1}, \tag{3.38}$$

where $|\cdot|$ denotes the determinant of a matrix. The cost can be minimized by following the gradients given by

$$\frac{\mathrm{d}J^\pi(\boldsymbol{\psi})}{\mathrm{d}\boldsymbol{\psi}} = \sum_{t=0}^{T} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\psi}} \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]. \tag{3.39}$$

The two most viable options for computing these gradients are analytical differentiation and automatic differentiation. The original implementation of PILCO computes the gradients analytically, which consists of expanding Equation (3.39) with the chain rule until it becomes an analytical expression that can be computed directly. The first expansion is

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\psi}} \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})] = \left(\frac{\partial}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t]}} \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]\right) \frac{\mathrm{d}\boldsymbol{\mu}^{\mathbf{x}}_{[t]}}{\mathrm{d}\boldsymbol{\psi}} + \left(\frac{\partial}{\partial \Sigma^{\mathbf{x}}_{[t]}} \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]\right) \frac{\mathrm{d}\Sigma^{\mathbf{x}}_{[t]}}{\mathrm{d}\boldsymbol{\psi}}, \tag{3.40}$$

where the derivatives of the mean and variance of the state distribution with respect to the controller parameters can be further expanded as

$$\frac{\mathrm{d}\boldsymbol{\mu}^{\mathbf{x}}_{[t]}}{\mathrm{d}\boldsymbol{\psi}} = \frac{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}} \frac{\mathrm{d}\boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}{\mathrm{d}\boldsymbol{\psi}} + \frac{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t]}}{\partial \Sigma^{\mathbf{x}}_{[t-1]}} \frac{\mathrm{d}\Sigma^{\mathbf{x}}_{[t-1]}}{\mathrm{d}\boldsymbol{\psi}} + \frac{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\psi}}, \tag{3.41}$$

$$\frac{\mathrm{d}\Sigma^{\mathbf{x}}_{[t]}}{\mathrm{d}\boldsymbol{\psi}} = \frac{\partial \Sigma^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}} \frac{\mathrm{d}\boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}{\mathrm{d}\boldsymbol{\psi}} + \frac{\partial \Sigma^{\mathbf{x}}_{[t]}}{\partial \Sigma^{\mathbf{x}}_{[t-1]}} \frac{\mathrm{d}\Sigma^{\mathbf{x}}_{[t-1]}}{\mathrm{d}\boldsymbol{\psi}} + \frac{\partial \Sigma^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\psi}}. \tag{3.42}$$

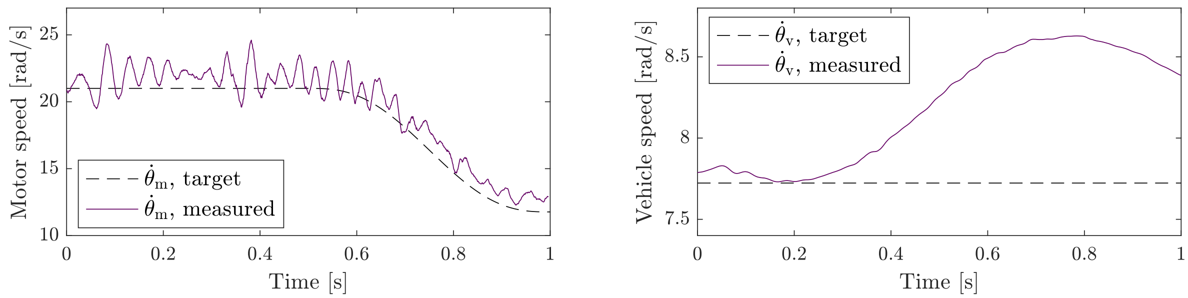
The derivatives $\frac{\mathrm{d}\boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}{\mathrm{d}\boldsymbol{\psi}}$ and $\frac{\mathrm{d}\Sigma^{\mathbf{x}}_{[t-1]}}{\mathrm{d}\boldsymbol{\psi}}$ are given from the previous time step, while the rest of the gradients have to be further expanded. This approach quickly becomes cumbersome, and more importantly, several steps would need to be redone if the parametrization of the controller changed.

For that reason, we implemented automatic differentiation [101]. This method uses the fact that all computations are ultimately compositions of elementary operations with known derivatives. Automatic differentiation consists of augmenting the computation of the elementary operations leading to a result—here, the cost function—with the computation of the derivative of these operations. Then the stored derivatives can be combined with the chain rule to yield the derivative of the result with respect any chosen constituent of the computation. Several frameworks exists for implementing automatic differentiation; here we chose TensorFlow. In TensorFlow, computational graphs are used to perform the forward computations and the automatic differentiation efficiently.

In this study, we still worked out the analytical solution for the derivative of the cost function with respect to the matrix K_c. The results are presented in Appendix C. The analytical solution were used to verify our implementation of automatic differentiation, as programming mistakes are easy to make and hard to detect otherwise. The results of Appendix C can be reused by the interested researchers to verify their own implementation of PILCO with automatic gradients. Alternatively, numerical gradients can also be used for gradient validation. However, numerical gradients are never exact, and it can be hard to decipher whether the discrepancies are caused by numerical imprecision or programming errors.

3.5 Experimental results

Measurements of $\dot{\theta}_\mathrm{m}$ and $\dot{\theta}_\mathrm{v}$ during a gearshift with the initialized (untrained) controller is shown on Figure 3.10. The rest of the result section focuses on the improvement of the



(a) Measured motor speed $\dot{\theta}_{\mathrm{m}}$.

(b) Measured vehicle speed $\dot{\theta}_{\mathrm{v}}$.

Figure 3.10: Experimental results for the untrained control policy.

tracking performance for $\dot{\theta}_{\mathrm{v}}$ with the proposed learning method. For the model-based learning method to be effective, the learned model must accurately represent the actual system dynamics. This was verified every time a new model iteration was obtained.

Figure 3.11 shows the evolution of the trajectories of $\dot{\theta}_{\mathrm{v}}$ through the iterations of the bigger loop in Figure 3.9 and Algorithm 1. Every trace is a measurement of $\dot{\theta}_{\mathrm{v}}$ on the test bench where the gearshift is performed with a newly optimized policy π^*. Table 3.1 shows the reduction of various norms of the tracking error signal $e(t)$ for $\dot{\theta}_{\mathrm{v}}$. The results show that very few gearshift trial—in this case, only about four—are required to tune the 12 parameters of the gearshift controller. The computation of each iterated policy π^* only takes about 100 s on a laptop computer. Of course, the various measures of error reduction presented in Table 3.1 heavily depend on the quality of the initialized controller. After all, PILCO was shown to be capable of learning controllers starting from randomly initialized parameters. In the context of a gearshift controller development process however, it may be counterproductive to randomly initialize the controllers given that several principled design methods exist in the literature, and engineers typically have good approximate models for the driveline dynamics. Therefore, it is interesting to see that the method still improves the performance of a reasonably initialized gearshift controller, and does so using only a few gearshift trials.

Moreover, Figure 3.12 shows the repeatability of the results. Figure 3.12a shows 10 gearshift trials with the initialized policy (in purple), and 10 gearshift trials with the learned policy (in blue). This indicates that the improvement reported in Figure 3.11 and Table 3.1 are not due to mere variations in the measurements. Figures 3.12b and 3.12c show that the learned parameters also improve the gearshift quality for conditions that were never used during training. Figure 3.12b shows a gearshift with a shortened duration, i.e., 0.6 s instead

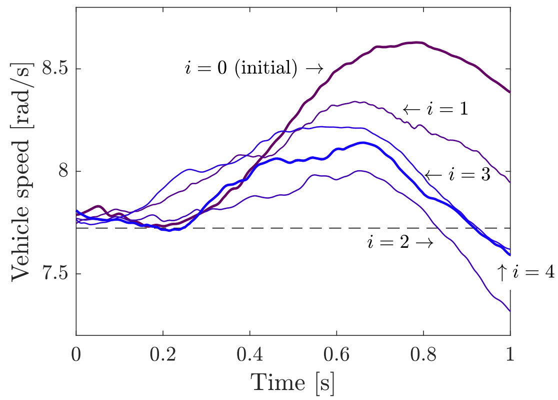


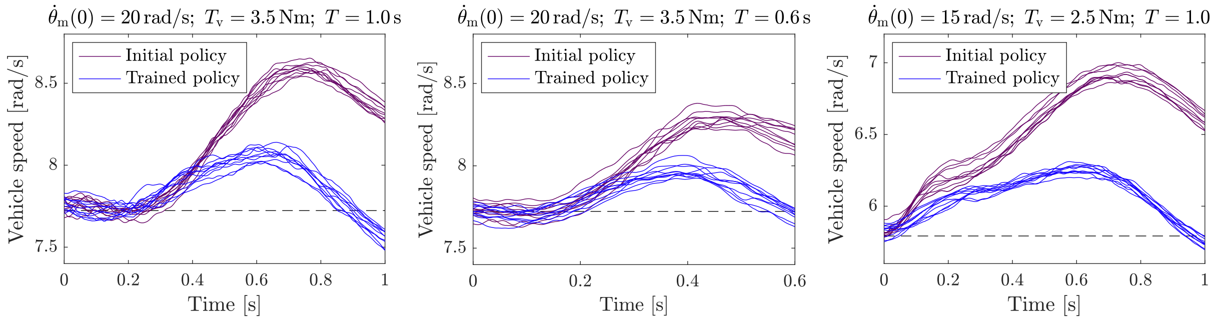
Figure 3.11: Evolution of vehicle speed $\dot{\theta}_{\mathrm{v}}$ trajectories through the iterations.

Table 3.1: Reduction in the tracking error of $\dot{\theta}_{\mathrm{v}}$ through the iterations.

iter. nb. (i)	$\|e\|_\infty$	$\|e(1)\|$	$\|e\|_2$
0	0.91	0.66	18.5
1	0.62	0.22	12.4
2	0.41	0.41	5.4
3	0.50	0.10	10.0
4	0.42	0.13	7.3
reduction	54 %	80 %	61 %

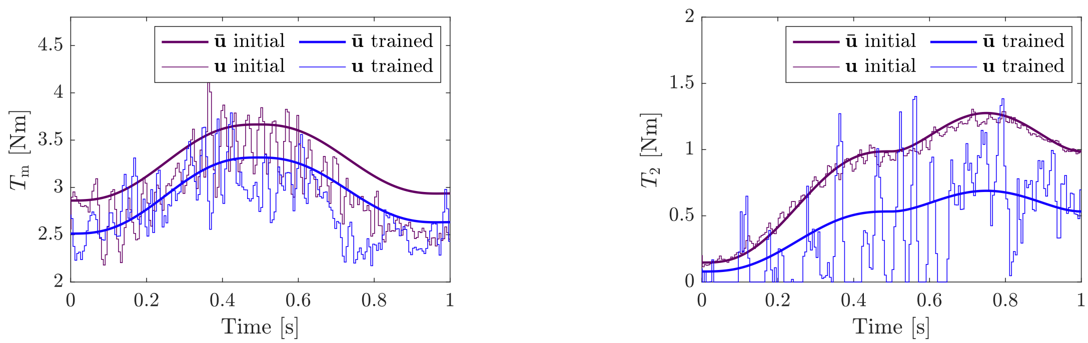
of the original 1.0 s. Figure 3.12c shows a 1-s gearshift initiated at reduced motor speed and reduced vehicle load. This suggests that the automatic tuning of gearshift controller parameters using the proposed method does not require trying a myriad of operating conditions, which greatly accelerates the tuning process.

Figure 3.13 shows how the learning process affects the torque commands. The two gearshift trials displayed—the initial policy and the trained policy—correspond to the trials with $i = 0$ and $i = 4$ in Figure 3.11, respectively. Figure 3.13a shows that with the initial policy, the nominal motor torque ($\bar{\mathbf{u}}$, thick purple line) is likely set too high, as the total controller output ($\mathbf{u}$, thin purple line) is almost always lower than the nominal torque. The learning process reduces the nominal torque, which makes the controller's output more centered around the nominal value. This suggests that the learning method appropriately corrects feedforward parameters. Figure 3.13b shows that the feedback gains for the command of T_2 are greatly increased. The initial policy barely deviates from the nominal torque command, and the trained policy does so significantly. Note that the torque command is saturated at zero, since it is impossible to command a negative torque on a friction clutch. This large increase in the feedback gains, combined with the fact that it improves the trajectory tracking—see Figure 3.11 and Table 3.1—suggests that the feedback did not have enough



(a) **(b)** **(c)**

Figure 3.12: Comparison of repeated gearshifts between the initial (purple) and trained (blue) policies (a) under the same operating conditions and gearshift trajectory used during the learning process, (b) under the same operating conditions, but with a shorter gearshift duration, and (c) at reduced motor speed and vehicle torque.



(a) Motor torque **(b)** Clutch 2 torque

Figure 3.13: Comparison of the control signals for the initial policy (purple) and the trained policy (blue). The thick lines represent the feedforward component of the control signals ($\bar{\mathbf{u}}$), and the thin lines represent the total controller output ($\mathbf{u}$), as in Equation (3.10) and Figure 3.8.

authority in the initial policy, which the learning process corrects. It also belies unknown dynamics in the physical transmission, and motivates the learning approach.

Finally, Figure 3.11 and Table 3.1 show that the policy seems to converge to a local minimum. In practice, engineers could restart the learning process with different initial values for the controller parameters, which would help determine whether the local minimum is also a global one given the current controller parametrization. Moreover, engineers could vary the parametrization of the controller, and determine whether it is possible to further improve the gearshift performance. After all, the chosen feedback controller and the arbitrary parametrization of the feedforward signal in this study may not be the best that one can devise. This highlights the importance of a flexible approach that can learn quickly. The method proposed in this work is made more flexible by the use of automatic gradients. Since the method can learn from few gearshift trials, several parametrizations can be tried. This is a key advantage of the proposed method over traditional ones, which are mainly suitable for fine-tuning a given parametrization.

Part II

Safety guarantees for autonomous vehicles

Chapter 4

Autonomous vehicles

This chapter reviews the functional architecture of autonomous vehicles. It also presents the vehicle models to be used in the example problems of Chapters 5 and 6.

Contents

4.1 Functional architecture of autonomous vehicles 59

4.2 Vehicle models 61

4.2.1 Pure longitudinal vehicle dynamics 61

4.2.2 Combined longitudinal and lateral vehicle dynamics 62

4.2.3 Pure lateral vehicle dynamics 64

4.3 Safe control of vehicle dynamics with learning 65

4.1 Functional architecture of autonomous vehicles

Modern vehicles have varying degrees of autonomy, ranging from simple driver assistance features—e.g., lane centering or adaptive cruise control—to more sophisticated autonomous capabilities, such as navigating completely autonomously in an urban environment as a driverless taxi [102]. Several system architectures were proposed to structure the problem of autonomous driving [103–105]. Typically, the architectures are structured around the thee main functions presented in Figure 4.1: perception, planning, and control.

The perception problem consists of transforming sensorial inputs into useful information for the motion planner and the controller. Depending on the autonomous features, these inputs may come from one or several of the following sensor types: GPS receivers, cameras, radars, lidars, sonars, inertial measurement units, and other sensors typical of vehicle chassis,

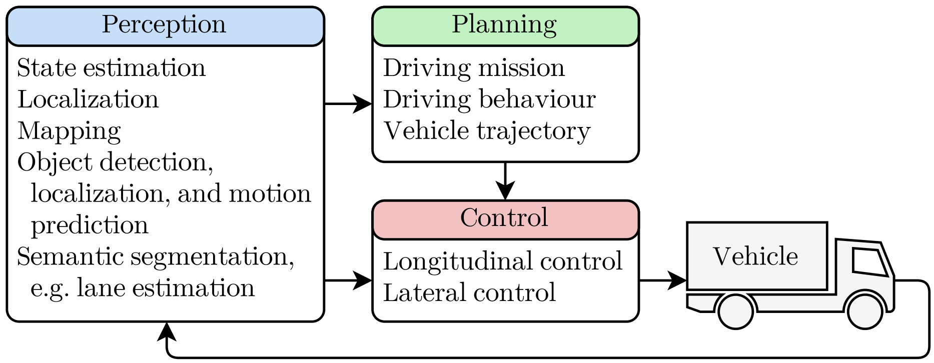
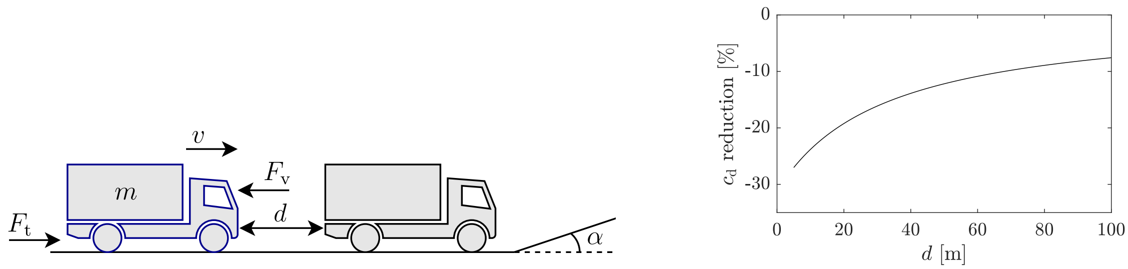


Figure 4.1: Functional architecture of autonomous vehicles.

such as wheel speed sensors, steering angle sensors, and other motor and drivetrain sensors. Part of the perception task is vehicle state estimation, commonly done through sensor fusion and the use of a Kalman filter [106]. More specific to the autonomous driving problem is the rest of the perception task: localizing the vehicle on a road map, creating or adapting a map of road surroundings [107], detecting and identifying objects such as road signs, pedestrians, and other vehicles [108], along with predicting their future trajectory, and finally, identifying the drivable areas and lane divisions [109].

The planning task is structured hierarchically [110, 111]. The mission planner is responsible for navigating the road network toward the intended destination. The behavioral planner [112] is responsible for selecting the short term maneuver to be executed, such as stopping for a red light, or changing lane. The planning task finished with computing a smooth and safe dynamical trajectory for the vehicle to follow [113, 114], which is then sent to the vehicle controller. Both the behavioral and trajectory planning can be updated in real-time to adapt to the motion of other vehicles [115].

The controller is the lowest level task in the autonomous driving problem. It is the controller that sends signals to the various vehicle actuators: mostly the motor(s), brakes, and actuated steering wheel, but also potentially other actuators such as transmission actuators or active suspension components. To simplify the control task, it seems common to treat the longitudinal and lateral control problems separately—see [105] for instance—although this is not necessary. Control methods typically rely on a model of the vehicle dynamics, which is the subject of the next section. Section 4.3 reviews popular control approaches for autonomous vehicles.



(a) Vehicle dynamics. **(b)** Drag reduction.

Figure 4.2: Longitudinal vehicle dynamics.

4.2 Vehicle models

This section presents dynamical models for three vehicular control tasks: pure longitudinal control, combined longitudinal and lateral control, and pure lateral control.

4.2.1 Pure longitudinal vehicle dynamics

The free-body diagram for the longitudinal control problem is shown on Figure 4.2a. The vehicular forces considered in F_{v} are the aerodynamic drag F_{a}, the tire rolling resistance F_{r}, and the effect of gravity due to a road slope F_{g}.

$$F_{\mathrm{v}} = F_{\mathrm{a}} + F_{\mathrm{r}} + F_{\mathrm{g}}, \tag{1.1}$$

$$F_{\mathrm{a}} = \tfrac{1}{2}\rho v^2 a_{\mathrm{f}} c_{\mathrm{d}}(d), \tag{4.1}$$

$$F_{\mathrm{t}} = mgc_{\mathrm{t}}\cos(\alpha), \tag{1.3}$$

$$F_{\mathrm{g}} = mg\sin(\alpha), \tag{1.4}$$

This is mostly the same model as in Section 1.3, with the exception that F_{a} depends on the distance d between the vehicles [116]—see Figure 4.2b. The drag coefficient c_{d} is modeled with

$$c_{\mathrm{d}}(d) = c_{\mathrm{d0}}\left(1 - \frac{c_1}{c_2 + d}\right), \tag{4.2}$$

where c_{d0} is the value for $d \to \infty$. The road slope is assumed small enough so that the small angle approximation holds, where $\cos(\alpha) = 1$ and $\sin(\alpha) = \alpha$ in Equations (1.3) and (1.4). The force F_{t} represents the tractive or braking force on the vehicle. It is therefore an input to this model. The magnitude of F_{t} is assumed to be low enough so that the effect of longitudinal tire slip can be neglected. Similarly, the effect of driveline and suspension dynamics are neglected as well. The model for the longitudinal dynamics is

$$\dot{v} = -\tfrac{1}{2m}\rho v^2 a_{\mathrm{f}} c_{\mathrm{d}}(d) - g c_{\mathrm{t}} - g\alpha + F_{\mathrm{t}}. \tag{4.3}$$

4.2.2 Combined longitudinal and lateral vehicle dynamics

The free-body diagram for the combined longitudinal and lateral control problem is shown on Figure 4.3a. The equations of motion for this system are

$$\ddot{x} = \tfrac{1}{m}\left(F_{\mathrm{r}x} + F_{\mathrm{f}x}\cos(\phi) - F_{\mathrm{f}y}\sin(\phi) - F_{\mathrm{v}}\right) + \dot{y}\dot{\psi}, \tag{4.4}$$

$$\ddot{y} = \tfrac{1}{m}\left(F_{\mathrm{f}x}\sin(\phi) + F_{\mathrm{r}y} + F_{\mathrm{f}y}\cos(\phi)\right) - \dot{x}\dot{\psi}, \tag{4.5}$$

$$\ddot{\psi} = \tfrac{1}{I_{\mathrm{z}}}\left(F_{\mathrm{f}x} l_{\mathrm{f}}\sin(\phi) - F_{\mathrm{r}y} l_{\mathrm{r}} + F_{\mathrm{f}y} l_{\mathrm{f}}\cos(\phi)\right). \tag{4.6}$$

The steering dynamics are modeled with a first-order linear dynamical system [117], where

$$\dot{\phi} = \lambda_{\mathrm{s}}(\phi_{\mathrm{r}} - \phi). \tag{4.7}$$

The kinematic relations between the inertial frame of reference (X, Y)—the road—and the non-inertial frame (x, y)—the vehicle—are described by

$$\dot{X} = \dot{x}\cos(\psi) - \dot{y}\sin(\psi), \tag{4.8}$$

$$\dot{Y} = \dot{x}\sin(\psi) - \dot{y}\cos(\psi). \tag{4.9}$$

The states of interest for this problem are $\mathbf{x} = [\dot{x}, \dot{y}, \dot{\psi}, \psi, Y, \phi]^\top$, and the control inputs are $\mathbf{u} = [F_{\mathrm{r}x}, F_{\mathrm{f}x}, \phi_{\mathrm{r}}]^\top$. The vehicular forces can be expanded until they are expressed solely in terms of the state variables. In this problem, F_{v} only considers aerodynamic drag, thus

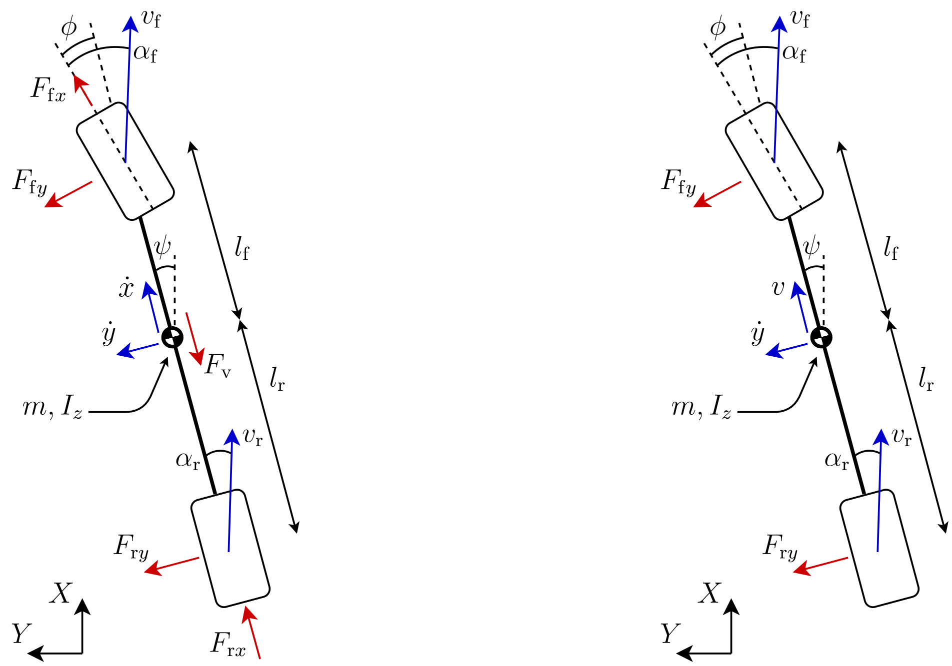
$$F_{\mathrm{v}} = \tfrac{1}{2}\rho\dot{x}^2 a_{\mathrm{f}} c_{\mathrm{d}}. \tag{4.10}$$

Several tire models exist to describe the lateral forces $F_{\mathrm{f}y}$ and $F_{\mathrm{r}y}$. These models typically rely on a notion of tire slip angles, illustrated on Figure 4.3a and described with

$$\alpha_{\mathrm{f}} = \phi - \arctan\left(\frac{\dot{\psi} l_{\mathrm{f}} + \dot{y}}{\dot{x}}\right), \tag{4.11}$$

$$\alpha_{\mathrm{r}} = \arctan\left(\frac{\dot{\psi} l_{\mathrm{r}} - \dot{y}}{\dot{x}}\right). \tag{4.12}$$

A popular choice of tire model is the Pacejka model [118]. However, this model is highly



(a) Combined longitudinal and lateral dynamics. **(b)** Pure lateral dynamics.

Figure 4.3: Free-body diagrams of bicycle models.

nonlinear and sometimes unnecessarily cumbersome. For small slip angles, a commonly used model is the linear tire model [119], where

$$F_{\mathrm{f}y} = c_\mathrm{f}\alpha_\mathrm{f}, \tag{4.13}$$

$$F_{\mathrm{r}y} = c_\mathrm{r}\alpha_\mathrm{r}. \tag{4.14}$$

In this work, the small slip angle approximation is assumed to hold because the vehicle maneuvers are gentle. In effect, this study focuses on lane-change maneuvers, not racing-style cornering or emergency collision avoidance. Consequently, the slip angle definition can also be simplified, where

$$\alpha_\mathrm{f} = \phi - \frac{\dot{\psi} l_\mathrm{f} + \dot{y}}{\dot{x}}, \tag{4.15}$$

$$\alpha_\mathrm{r} = \frac{\dot{\psi} l_\mathrm{r} - \dot{y}}{\dot{x}}. \tag{4.16}$$

Moreover, assuming that the vehicle ψ and steering ϕ angles remain small, the equations of motion can be further simplified. With the tire model, the equations of motion become

$$\ddot{x} = \tfrac{1}{m}\left(F_{\mathrm{r}x} + F_{\mathrm{f}x} - c_{\mathrm{f}}\left(\phi^2 - \frac{\phi(\dot{\psi}l_{\mathrm{f}} + \dot{y})}{\dot{x}}\right) - \tfrac{1}{2}\rho\dot{x}^2 a_{\mathrm{f}} c_{\mathrm{d}}\right) + \dot{y}\dot{\psi}, \tag{4.17}$$

$$\ddot{y} = \tfrac{1}{m}\left(F_{\mathrm{f}x}\phi + c_{\mathrm{r}}\left(\frac{\dot{\psi}l_{\mathrm{r}} - \dot{y}}{\dot{x}}\right) + c_{\mathrm{f}}\left(\phi - \frac{\dot{\psi}l_{\mathrm{f}} + \dot{y}}{\dot{x}}\right)\right) - \dot{x}\dot{\psi}, \tag{4.18}$$

$$\ddot{\psi} = \tfrac{1}{I_{\mathrm{z}}}\left(F_{\mathrm{f}x}l_{\mathrm{f}}\phi - c_{\mathrm{r}}l_{\mathrm{r}}\left(\frac{\dot{\psi}l_{\mathrm{r}} - \dot{y}}{\dot{x}}\right) + c_{\mathrm{f}}l_{\mathrm{f}}\left(\phi - \frac{\dot{\psi}l_{\mathrm{f}} + \dot{y}}{\dot{x}}\right)\right), \tag{4.19}$$

The equation for the steering dynamics (4.7) remains the same. The equation for $\dot{Y}$ becomes

$$\dot{Y} = \dot{x}\psi - \dot{y}. \tag{4.20}$$

Together Equations (4.7), (4.17)–(4.20) describe the dynamics of the system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$. The dynamics can be linearized to conform to $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ by taking the Jacobian of $f(\mathbf{x}, \mathbf{u})$ with respect to the states $\mathbf{x}$ and inputs $\mathbf{u}$. The partial derivatives are presented in Appendix D. A nominal linear model is chosen by evaluating the partial derivatives at a nominal point $\mathbf{x}_0 = [\dot{x}_0, 0, 0, 0, 0, 0]^\top$, $\mathbf{u}_0 = [0, 0, 0]^\top$, where most of the partial derivatives vanish.

$$A = \begin{bmatrix} \frac{-\rho\dot{x}_0 a_{\mathrm{f}} c_{\mathrm{d}}}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-(c_{\mathrm{f}}+c_{\mathrm{r}})}{m\dot{x}_0} & \frac{-c_{\mathrm{f}}l_{\mathrm{f}}+c_{\mathrm{r}}l_{\mathrm{r}}}{m\dot{x}_0} - \dot{x}_0 & 0 & 0 & \frac{c_{\mathrm{f}}}{m} \\ 0 & \frac{-c_{\mathrm{f}}l_{\mathrm{f}}+c_{\mathrm{r}}l_{\mathrm{r}}}{I_z\dot{x}_0} & \frac{-(c_{\mathrm{f}}l_{\mathrm{f}}^2+c_{\mathrm{r}}l_{\mathrm{r}}^2)}{I_z\dot{x}_0} & 0 & 0 & \frac{c_{\mathrm{f}}l_{\mathrm{f}}}{I_z} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \dot{x}_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\lambda_{\mathrm{s}} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{m} & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_{\mathrm{s}} \end{bmatrix}. \tag{4.21}$$

Chapter 5 considers two types of modeling uncertainty for this control problem: the uncertainty related to vehicle parameters—e.g., c_{f} and c_{r}—and the uncertainty related to the linearization around a given set of states. To obtain an envelope of possible system dynamics, the partial derivatives can be evaluated for different states and different parameter values, thereby obtaining different A and B matrices for the system equation.

4.2.3 Pure lateral vehicle dynamics

For the purely lateral control problem, the free-body diagram is reduced to 4.3b. The vehicle speed is assumed constant, i.e. $\dot{x} = v$, and the longitudinal dynamics are neglected. In other words, the vehicle is assumed to be closed-loop controlled such that the longitudinal speed remains v, and the problem only concerns the lateral control. The states of interest for this problem are $\mathbf{x} = [\dot{y}, \dot{\psi}, \psi, Y, \phi]^\top$, and the only control input is $\mathbf{u} = [\phi_\mathrm{r}]$. The effect of the front tractive force on the lateral dynamics is also neglected. Consequently, the equations of motion (4.18)–(4.19) and the kinematic relation (4.20) become linear.

$$\ddot{y} = c_\mathrm{r}\left(\frac{\dot{\psi}l_\mathrm{r} - \dot{y}}{mv}\right) + c_\mathrm{f}\left(\phi - \frac{\dot{\psi}l_\mathrm{f} + \dot{y}}{mv}\right) - v\dot{\psi}, \tag{4.22}$$

$$\ddot{\psi} = -c_\mathrm{r}l_\mathrm{r}\left(\frac{\dot{\psi}l_\mathrm{r} - \dot{y}}{I_\mathrm{z}v}\right) + c_\mathrm{f}l_\mathrm{f}\left(\phi - \frac{\dot{\psi}l_\mathrm{f} + \dot{y}}{I_\mathrm{z}v}\right), \tag{4.23}$$

$$\dot{Y} = v\psi - \dot{y}. \tag{4.24}$$

The same steering model (4.7) is used for this problem. The dynamics can easily be expressed in the form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$, where

$$A = \begin{bmatrix} \frac{-(c_\mathrm{f}+c_\mathrm{r})}{mv} & \frac{-c_\mathrm{f}l_\mathrm{f}+c_\mathrm{r}l_\mathrm{r}}{mv} - v & 0 & 0 & \frac{c_\mathrm{f}}{m} \\ \frac{-c_\mathrm{f}l_\mathrm{f}+c_\mathrm{r}l_\mathrm{r}}{I_z v} & \frac{-(c_\mathrm{f}l_\mathrm{f}^2+c_\mathrm{r}l_\mathrm{r}^2)}{I_z v} & 0 & 0 & \frac{c_\mathrm{f}l_\mathrm{f}}{I_z} \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_\mathrm{s} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \lambda_\mathrm{s} \end{bmatrix}. \tag{4.25}$$

4.3 Safe control of vehicle dynamics with learning

In the functional architecture of Figure 4.1, the objective of the control task is to generate actuator commands so that the vehicle follows the trajectory given by the motion planner. For a vehicle to remain safe, the controller has to both stabilize the closed-loop dynamics and keep the vehicle within state constraints. Example constraints for ground vehicles are lane keeping and maintaining a safe distance with other vehicles. Both the stability and the constraint satisfaction problems can be rendered more difficult by the presence of unknown adversarial disturbances or uncertainty in the vehicle dynamics. This motivates the use of learning for control. Machine learning is ubiquitous for the perception task and is also commonly used for the planning task, but it is not as prevalent in vehicular control [120,121].

Numerous control methods were used in the context of autonomous vehicles [122]. These include model predictive control [123], linear control [124], classical control [125], and Lyapunov-based control [126]. Vehicular control generally benefits from the addition of a feedforward component [127]. Approaches such as tube-MPC [128] explicitly consider modeling uncertainty. While some recent studies [100] make use of machine learning to counter modeling uncertainty, this remains a challenge. This is the object of Chapters 5 and 6.

Chapter 5

Learning-based synthesis of robust controllers

This chapter presents a new approach to obtain a robustly stable linear controller from a learned dynamical model of arbitrary type. It begins with a review of robust stability and learning methods. Then the new method is presented, first by introducing the robust control framework it is based on, followed by the learning algorithm. The method is demonstrated with a vehicle control problem: a lane-change maneuver with vehicle acceleration.

Contents

5.1 Review of learned robustly stable controllers 67

5.2 Robust control framework 70

5.3 Controller learning method 70

5.3.1 Performance gradients 72

5.3.2 Nominal stability gradients 73

5.3.3 Robust stability gradients 74

5.4 Example problem 75

5.4.1 Uncertainty cover 75

5.4.2 Trajectory and nominal command 77

5.4.3 Controller structure and initialization 79

5.4.4 Controller training 79

5.5 Results and discussion 80

5.5.1 Augmented cost 83

5.1 Review of learned robustly stable controllers

Machine learning can be used to improve the performance of controllers for dynamical systems. Typically the performance increase is enabled by a reduction in the modeling uncertainty. In effect, the fundamental tradeoff [12] between robustness and performance in control reveals how modeling uncertainty can oftentimes be the limiting factor in system performance. However, it can be hard to maintain theoretical guarantees of robust stability when incorporating learning components in control methods.

For complex problems such as robotic hand manipulation [129], domain randomization [130, 131] seems to provide an interesting level of robustness. The idea is to randomly alter the physics of the simulated environment used by the learning algorithm, which better prepares the controller to be deployed in the real world. This method was even proposed for the end-to-end learning of control policies for autonomous driving [132]—i.e., learning a direct mapping from camera pixels to control signals. But it remains that whenever possible, it is preferable to maintain theoretical guarantees of robust stability, and domain randomization does not.

Recently, several lines of research presented successful approaches to such guarantees. One approach [133] consists of using Lyapunov theory to characterize the system's region of attraction (ROA), and tune a neural network (NN) controller with gradient descent while considering the constraints imposed by the ROA. Gaussian processes (GP) are used to learn the system dynamics. The size of the ROA and the controller performance both gradually increase as a result of controller tuning and data collection. In another approach [134], a quadratic programming controller is used to adjust a nominal controller's output to enforce a Lyapunov-based stability constraint. Given that the safety of this approach is contingent on having an accurate model of the system dynamics, a NN is trained from collected data and used to improve the model's accuracy. In [135], researchers also devised an outer-loop controller to adjust the output of a nominal controller, but they update in real-time the model uncertainty estimates using the variance of a GP regression in order to better adjust the controller's aggressiveness with respect to the local system configuration. Researchers in [94] use machine learning to estimate a linear system model along with error bounds, from which they can obtain a robust linear controller. The method can also be used to identify systems having linear dynamics with respect to nonlinear features of states and inputs [136]. It can also be used to design robust linear controllers for systems with complex high-dimensional sensorial inputs like camera-based autonomous vehicles [137], thereby making progress toward formal guarantees even for the most complex problems.

This chapter introduces a control learning method that is appropriate for linear time-invariant (LTI) controllers, which typically have the state-space representation

$$\dot{\mathbf{x}}_{\mathrm{c}} = A_{\mathrm{K}}\mathbf{x}_{\mathrm{c}} + B_{\mathrm{K}}\mathbf{e}, \tag{5.1}$$

$$\mathbf{u} = C_{\mathrm{K}}\mathbf{x}_{\mathrm{c}} + D_{\mathrm{K}}\mathbf{e}, \tag{5.2}$$

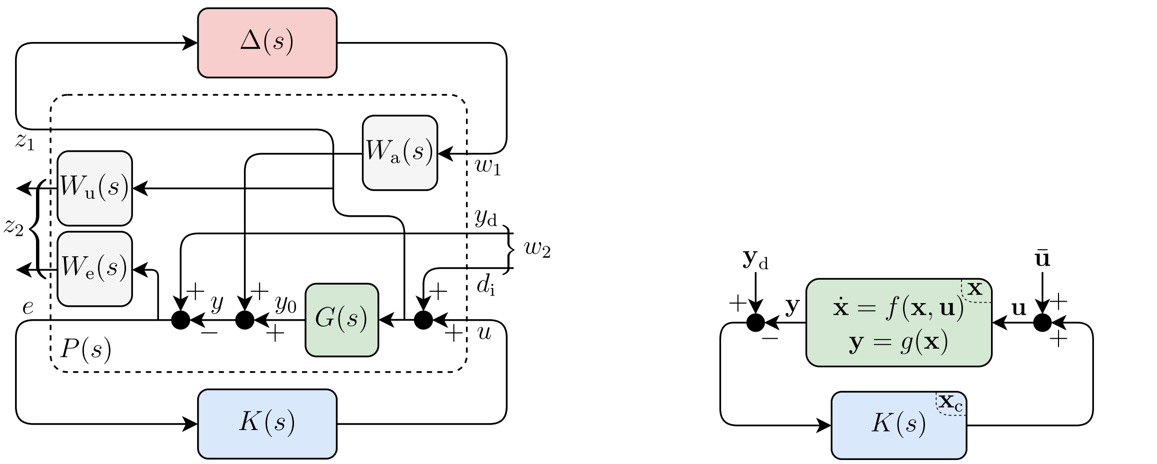
where $\mathbf{e}$ is the error signal fed to the controller, $\mathbf{x}_{\mathrm{c}}$ are the controller states, and $\mathbf{u}$ are the controller output signals. A time-proven method to synthesize such a controller is $\mathcal{H}_\infty$ loop shaping within a robust control framework [11], see [138, 139] for instance. To do so, one assumes a nominal linear system model along with uncertainty bounds, defines performance criteria in terms of filters on the input and output signals, and solves a norm minimization problem. However, $\mathcal{H}_\infty$ loop shaping can be difficult to implement in practice. Notably, the method is restricted to linear system models, so it can be hard to optimize closed-loop performance when the system is best described with a nonlinear model. As a result, the method is ill-suited for taking advantage of system identification through modern machine learning techniques, as the learned models are often nonlinear. Finally, it can be hard to translate the desire to track specific temporal reference trajectories into design filters. Thus, the objective in this research is to develop a LTI controller synthesis method that can take advantage of nonlinear learned models while ensuring robust closed-loop stability through a robust control framework.

Such a method does not exists in the literature. The approaches in [134, 135] consist of adjusting a base controller output according to a safety criterion computed in real-time, but do not tune the base controller parameters. Moreover, the approach of [133] suffers from the curse of dimensionality, which makes it ill-suited for multi-input multi-output systems with several states—a typical application for LTI controllers. The approach of [94] could be used to synthesize such controllers, but it assumes a linear system model, which defeats the objective of this work. Researchers in [140] use a robust control framework to ensure robust stability. By learning a probabilistic dynamical model with GPs, they reduce the modeling uncertainty by translating the GP posterior variance into the robust control framework, and ultimately increase the closed-loop performance. However, this approach is still a norm minimization problem with design filters. Another approach [141, 142] is to attach a NN to a LTI controller. In this chapter however, the objective is to tune a LTI controller without requiring a NN.

The proposed learning method is outlined in Algorithm 2. It requires to define a nominal linear system model $G(s)$ and uncertainty bounds. Robust stability criteria are derived from a robust control framework and the uncertain system model. An initial controller K_{i} is obtained from $\mathcal{H}_\infty$ loop shaping. Then, the controller is tuned with gradients obtained from simulated rollouts using any nominal model $f_{\mathrm{n}}(\mathbf{x}, \mathbf{u})$, thereby obtaining a tuned controller K_{t}. The dynamics $f_{\mathrm{n}}(\mathbf{x}, \mathbf{u})$ can be the same as $G(s)$, but can also be any more accurate nonlinear model. During the training, the robust stability criteria are enforced through the gradients. Finally, the tuned controller K_{t} can be adjusted to the actual system dynamics, thereby obtaining the final adjusted controller K_{a}. This is done by implementing K_{t} on the actual system, collecting data, learning a system model $f_{\mathrm{l}}(\mathbf{x}, \mathbf{u})$, and then retraining the controller from the same gradient descent method. The same robust stability criteria are enforced during this second controller tuning.

Similar to the method in Chapter 3, the control parameter tuning in Algorithm 2 is based on simulated policy rollouts. This choice was motivated by the results obtained when learning the gearshift controller, as well as the previously discussed merits of the PILCO algorithm—Section 3.1.2. However, PILCO does not explicitly enforce robust stability though constraints in the optimization problem. Instead, it propagates state uncertainty during rollouts, which is used to penalize controller behaviors leading to regions of the state space with high uncertainty. Because of the robust stability criteria, Algorithm 2 can rely on deterministic policy rollouts, which is interesting given that uncertainty propagation is not a trivial problem.

The proposed method synthesizes a controller that is already robustly stabilizing when first implemented on the actual system. This is not the case for some of the methods reviewed above, such as [134]. Moreover, the controller remains stable even in the event of a sudden change in system dynamics, assuming the change remains within the chosen uncertainty bounds. By design, the learning methods reviewed above all rely on the assumption that the system dynamics are fixed. This allows to learn the uncertainty bounds from data, and increase the closed-loop performance. But it leaves no formal guarantee for the system safety when a sudden change in dynamics occur, which could be a concern for certain applications. On the contrary, if for a given application it is judged that the fixed uncertainty bounds are too conservative, our methods still makes it possible to learn the bounds such as in [140], and retrain the controller.



(a) Robust control framework. **(b)** Closed-loop system for policy rollouts.

Figure 5.1: Systems used in this work: a) is used to derive the criteria for robust stability, and b) is used to tune the controller performance.

5.2 Robust control framework

A typical robust control problem is formulated based on system models such as that of Figure 5.1a. The goal is to find a stabilizing controller $K(s)$ that maximizes the performance of the system, which means minimizing $\|T_{z_2w_2}(s)\|_\infty$. Design filters are used to weight the penalty on the various signals of z_2 over frequency ranges of interest. In Figure 5.1a, $W_{\mathrm{u}}(s)$ penalizes the control signal and $W_{\mathrm{e}}(s)$, the error signal. Because the actual system dynamics are not perfectly known, they are assumed to be within a set of perturbed plants $\mathcal{G}$. Thus, nominal stability is not sufficient; $K(s)$ must stabilize the system for all possible plants in $\mathcal{G}$. In Figure 5.1a, additive uncertainty is used to describe $\mathcal{G} := \{G(s) + W_{\mathrm{a}}(s)\Delta(s)\}$, where $\Delta(s) \in \mathcal{RH}_\infty$ with $\|\Delta(s)\|_\infty < 1$, $G(s)$ is the nominal plant dynamics, and $W_{\mathrm{a}}(s)$ is a design filter bounding the uncertainty. Also commonly used are output multiplicative uncertainty, where $\mathcal{G} := \{(I + W_{\mathrm{m}}(s)\Delta(s))G(s)\}$, and input multiplicative uncertainty, where $\mathcal{G} := \{G(s)(I + W_{\mathrm{i}}(s)\Delta(s))\}$. From the small-gain theorem [11, 143], if $\Delta(s) \in \mathcal{RH}_\infty$ and $T_{z_1w_1}(s) \in \mathcal{RH}_\infty$, the system of Figure 5.1a is robustly stabilizing if and only if $\|\Delta(j\omega)T_{z_1w_1}(j\omega)\| < 1 \ \forall\omega$. And because $\|\Delta(s)\|_\infty < 1$ by design, robust stability is guaranteed if $K(s)$ stabilizes $G(s)$ and makes $\|T_{z_1w_1}(s)\|_\infty \leq 1$.

5.3 Controller learning method

This section explains Algorithm 2. It requires to define a nominal system model $G(s)$ along with uncertainty bounds such that all possible system dynamics are contained in a set $\mathcal{G}$.

Using the robust control framework of Section 5.2, the set $\mathcal{G}$ is used to obtain the robust stability criteria. The method also requires to define a (possibly nonlinear) system model

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \tag{5.3}$$

$$\mathbf{y} = g(\mathbf{x}), \tag{5.4}$$

to be used during the simulated policy rollouts under the closed-loop configuration of Figure 5.1b. The simulations consist of tracking N_t trajectories in rollouts of T time steps. The while loops starting at lines 6 and 14 tune the controllers by solving the constrained optimization problem

$$\arg\min_{\boldsymbol{\phi}} \; c_\mathrm{p}(\boldsymbol{\phi}) = \arg\min_{\boldsymbol{\phi}} \sum_{n_\mathrm{t}=1}^{N_\mathrm{t}} \sum_{t=0}^{T} c(\mathbf{y}_{[t]}), \tag{5.5}$$

$$\text{s.t.} \quad K(s) \text{ stabilizes } G(s), \tag{5.6}$$

$$\|T_{z_1 w_1}(s)\|_\infty \leq 1, \tag{5.7}$$

where $\boldsymbol{\phi}$ contains the control parameters to be tuned, namely the entries of the matrices A_K, B_K, C_K, and D_K. The two constraints of Equations (5.6) and (5.7) are turned into penalties with easily computable gradients. As such, a more general cost function is

$$J(\boldsymbol{\phi}) = c_\mathrm{p}(\boldsymbol{\phi}) + \xi_\mathrm{s} c_\mathrm{s}(\boldsymbol{\phi}) + \xi_\mathrm{r} c_\mathrm{r}(\boldsymbol{\phi}), \tag{5.8}$$

where c_p is the performance cost, c_s is the nominal stability penalty, and c_r is the robust stability penalty. The controller parameters $\boldsymbol{\phi}$ are iterated through either a simple stochastic gradient descent scheme—where $\boldsymbol{\phi}_{[i+1]} \leftarrow \boldsymbol{\phi}_{[i]} + \alpha \nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi})$ and α is the learning rate—or any other suitable optimization algorithm. Constants $\xi_\mathrm{s}, \xi_\mathrm{r} \in \mathbb{R}^+$ are used to weight the penalties and help the gradient descent method. The following sections demonstrate how to obtain the gradients of the performance cost and penalties.

Algorithm 2: Learning-based synthesis of robustly stabilizing controllers

Result: Learned controller parameters $\boldsymbol{\phi} = \{A_\mathrm{K}, B_\mathrm{K}, C_\mathrm{K}, D_\mathrm{K}\}$.

Define nominal linear system dynamics $G(s)$.
Define set of all possible system dynamics $\mathcal{G}$, bounding the uncertainty on $G(s)$.
Define weighting functions $W_\mathrm{e}(s)$ and $W_\mathrm{u}(s)$; obtain $K_\mathrm{i}(s)$ from $\mathcal{H}_\infty$ loop shaping.
Define nominal (possibly nonlinear) system dynamics $f_\mathrm{n}(\mathbf{x}, \mathbf{u})$.
Initialize $K_\mathrm{t}(s) = K_\mathrm{i}(s)$.
while $K_\mathrm{t}(s)$ *not tuned* **do**
    Simulate a policy rollout with $K_\mathrm{t}(s)$ on $f_\mathrm{n}(\mathbf{x}, \mathbf{u})$ as per Figure 5.1b.
    Compute the total gradients $\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} c_\mathrm{p}(\boldsymbol{\phi}) + \xi_\mathrm{s} \nabla_{\boldsymbol{\phi}} c_\mathrm{s}(\boldsymbol{\phi}) + \xi_\mathrm{r} \nabla_{\boldsymbol{\phi}} c_\mathrm{r}(\boldsymbol{\phi})$.
    Iterate the parameters $\boldsymbol{\phi}$ with $\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi})$.
Initialize $K_\mathrm{a}(s) = K_\mathrm{t}(s)$.
while $K_\mathrm{a}(s)$ *not learned* **do**
    Implement $K_\mathrm{a}(s)$ on actual system, collect dataset $\mathcal{D}$.
    Learn a model $f_\mathrm{l}(\mathbf{x}, \mathbf{u})$ for the system dynamics from $\mathcal{D}$.
    while $K_\mathrm{a}(s)$ *not tuned* **do**
        Simulate a policy rollout with $K_\mathrm{a}(s)$ on $f_\mathrm{l}(\mathbf{x}, \mathbf{u})$ as per Figure 5.1b.
        Compute the total gradients $\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} c_\mathrm{p}(\boldsymbol{\phi}) + \xi_\mathrm{s} \nabla_{\boldsymbol{\phi}} c_\mathrm{s}(\boldsymbol{\phi}) + \xi_\mathrm{r} \nabla_{\boldsymbol{\phi}} c_\mathrm{r}(\boldsymbol{\phi})$.
        Iterate the parameters $\boldsymbol{\phi}$ with $\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi})$.

5.3.1 Performance gradients

In discrete time, the system of Figure 5.1b transitions with

$$
\begin{aligned}
\mathbf{y}_{[t]} &= g(\mathbf{x}_{[t]}), && (5.9)\\
\mathbf{x}_{[t]} &= \mathbf{x}_{[t-1]} + T_\mathrm{s} f(\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]}), && (5.10)\\
\mathbf{u}_{[t-1]} &= C_\mathrm{K} \mathbf{x}_{\mathrm{c}[t-1]} + D_\mathrm{K}(\mathbf{y}_{\mathrm{d}[t-1]} - \mathbf{y}_{[t-1]}) + \bar{\mathbf{u}}_{[t-1]}, && (5.11)\\
\mathbf{x}_{\mathrm{c}[t-1]} &= (I + T_\mathrm{s} A_\mathrm{K}) \mathbf{x}_{\mathrm{c}[t-2]} + T_\mathrm{s} B_\mathrm{K}(\mathbf{y}_{\mathrm{d}[t-2]} - \mathbf{y}_{[t-2]}), && (5.12)
\end{aligned}
$$

where T_s is the sampling period, $\bar{\mathbf{u}}$ is a nominal (feedforward) control command, and $\mathbf{y}_\mathrm{d}$ is the desired output trajectory. The performance gradients are obtained by first applying the chain rule

$$
\frac{\mathrm{d}c_\mathrm{p}(\boldsymbol{\phi})}{\mathrm{d}\boldsymbol{\phi}} = \sum_{n_\mathrm{t}=1}^{N_\mathrm{t}} \sum_{t=0}^{T} \frac{\mathrm{d}c(\mathbf{y}_{[t]})}{\mathrm{d}\boldsymbol{\phi}} \tag{5.13}
$$

$$
\frac{\mathrm{d}c(\mathbf{y}_{[t]})}{\mathrm{d}\boldsymbol{\phi}} = \frac{\mathrm{d}c(\mathbf{y}_{[t]})}{\mathrm{d}\mathbf{y}_{[t]}} \frac{\mathrm{d}\mathbf{y}_{[t]}}{\mathrm{d}\boldsymbol{\phi}} \tag{5.14}
$$

The first term on the right-hand side of Equation (5.14) depends on the chosen cost function, the other term can be further expanded as

$$\frac{\mathrm{d}\mathbf{y}_{[t]}}{\mathrm{d}\boldsymbol{\phi}} = \frac{\mathrm{d}g(\mathbf{x}_{[t]})}{\mathrm{d}\mathbf{x}_{[t]}} \frac{\mathrm{d}\mathbf{x}_{[t]}}{\mathrm{d}\boldsymbol{\phi}} \tag{5.15}$$

$$\frac{\mathrm{d}\mathbf{x}_{[t]}}{\mathrm{d}\boldsymbol{\phi}} = \frac{\mathrm{d}\mathbf{x}_{[t-1]}}{\mathrm{d}\boldsymbol{\phi}} + T_{\mathrm{s}} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\phi}} f(\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]}) \tag{5.16}$$

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\phi}} f(\mathbf{x}_{[t-1]}, \mathbf{u}_{[t-1]}) = \frac{\partial f}{\partial \mathbf{x}_{[t-1]}} \frac{\mathrm{d}\mathbf{x}_{[t-1]}}{\mathrm{d}\boldsymbol{\phi}} + \frac{\partial f}{\partial \mathbf{u}_{[t-1]}} \frac{\mathrm{d}\mathbf{u}_{[t-1]}}{\mathrm{d}\boldsymbol{\phi}} \tag{5.17}$$

$$\begin{aligned} \frac{\mathrm{d}\mathbf{u}_{[t-1]}}{\mathrm{d}\boldsymbol{\phi}} &= C_{\mathrm{K}} \frac{\mathrm{d}\mathbf{x}_{\mathrm{c}[t-1]}}{\mathrm{d}\boldsymbol{\phi}} + (I)_{ij}(\mathbf{x}_{\mathrm{c}[t-1]})_k \frac{\mathrm{d}C_{\mathrm{K}}}{\mathrm{d}\boldsymbol{\phi}} \\ &\quad - D_{\mathrm{K}} \frac{\mathrm{d}\mathbf{y}_{[t-1]}}{\mathrm{d}\boldsymbol{\phi}} + (I)_{ij}(\mathbf{y}_{\mathrm{d}[t-1]} - \mathbf{y}_{[t-1]})_k \frac{\mathrm{d}D_{\mathrm{K}}}{\mathrm{d}\boldsymbol{\phi}} \end{aligned} \tag{5.18}$$

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_{\mathrm{c}[t-1]}}{\mathrm{d}\boldsymbol{\phi}} &= (I + T_{\mathrm{s}}A_{\mathrm{K}}) \frac{\mathrm{d}\mathbf{x}_{\mathrm{c}[t-2]}}{\mathrm{d}\boldsymbol{\phi}} + T_{\mathrm{s}}(I)_{ij}(\mathbf{x}_{\mathrm{c}[t-2]})_k \frac{\mathrm{d}A_{\mathrm{K}}}{\mathrm{d}\boldsymbol{\phi}} \\ &\quad - T_{\mathrm{s}}B_{\mathrm{K}} \frac{\mathrm{d}\mathbf{y}_{[t-2]}}{\mathrm{d}\boldsymbol{\phi}} + T_{\mathrm{s}}(I)_{ij}(\mathbf{y}_{\mathrm{d}[t-2]} - \mathbf{y}_{[t-2]})_k \frac{\mathrm{d}B_{\mathrm{K}}}{\mathrm{d}\boldsymbol{\phi}} \end{aligned} \tag{5.19}$$

where $\frac{\mathrm{d}\mathbf{x}_{[t-1]}}{\mathrm{d}\boldsymbol{\phi}}$ and $\frac{\mathrm{d}\mathbf{x}_{\mathrm{c}[t-2]}}{\mathrm{d}\boldsymbol{\phi}}$ are obtained from the previous time steps.

5.3.2 Nominal stability gradients

In the configuration of Figure 3.8, $K(s)$ stabilizes $G(s)$ if and only if the following system is stable

$$\begin{bmatrix} y(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} G(s)K(s)\big(I + G(s)K(s)\big)^{-1} & G(s)\big(I + K(s)G(s)\big)^{-1} \\ K(s)\big(I + G(s)K(s)\big)^{-1} & K(s)G(s)\big(I + K(s)G(s)\big)^{-1} \end{bmatrix} \begin{bmatrix} y_{\mathrm{d}}(s) \\ d_{\mathrm{i}}(s) \end{bmatrix}. \tag{5.20}$$

For transfer functions with the state-space realizations

$$G(s) = \left[\begin{array}{c|c} A_{\mathrm{G}} & B_{\mathrm{G}} \\ \hline C_{\mathrm{G}} & 0 \end{array}\right], \quad \text{and} \quad K(s) = \left[\begin{array}{c|c} A_{\mathrm{K}} & B_{\mathrm{K}} \\ \hline C_{\mathrm{K}} & D_{\mathrm{K}} \end{array}\right] \tag{5.21}$$

it suffices to verify that the eigenvalues λ of the four matrices

$$M_1 = \begin{bmatrix} A_{\mathrm{G}} & B_{\mathrm{G}}C_{\mathrm{K}} & B_{\mathrm{G}}D_{\mathrm{K}}C_{\mathrm{G}} \\ 0 & A_{\mathrm{K}} & -B_{\mathrm{K}}C_{\mathrm{G}} \\ 0 & B_{\mathrm{G}}C_{\mathrm{K}} & A_{\mathrm{G}} - B_{\mathrm{G}}D_{\mathrm{K}}C_{\mathrm{G}} \end{bmatrix}, \tag{5.22}$$

$$M_2 = \begin{bmatrix} A_{\mathrm{G}} - B_{\mathrm{G}}D_{\mathrm{K}}C_{\mathrm{G}} & -B_{\mathrm{G}}C_{\mathrm{K}} \\ B_{\mathrm{K}}C_{\mathrm{G}} & A_{\mathrm{K}} \end{bmatrix}, \tag{5.23}$$

$$M_3 = \begin{bmatrix} A_{\mathrm{K}} & -B_{\mathrm{K}}C_{\mathrm{G}} \\ B_{\mathrm{G}}C_{\mathrm{K}} & A_{\mathrm{G}} - B_{\mathrm{G}}D_{\mathrm{K}}C_{\mathrm{G}} \end{bmatrix}, \tag{5.24}$$

$$M_4 = \begin{bmatrix} A_{\mathrm{K}} & B_{\mathrm{K}}C_{\mathrm{G}} & 0 \\ 0 & A_{\mathrm{G}} - B_{\mathrm{G}}D_{\mathrm{K}}C_{\mathrm{G}} & -B_{\mathrm{G}}C_{\mathrm{K}} \\ 0 & B_{\mathrm{K}}C_{\mathrm{G}} & A_{\mathrm{K}} \end{bmatrix}, \tag{5.25}$$

are all in the open left-half plane. In effect, these are the A matrices of the state-space realizations of the transfer functions in Equation (5.20). In the learning algorithm, a penalty is applied whenever the system is unstable. The cost function for the nominal stability is therefore defined as

$$c_{\mathrm{s}}(\boldsymbol{\phi}) = \sum_{i=1}^{4} \max\left\{0, \max_j \left\{\Re\big(\lambda_j(M_i)\big)\right\}\right\}. \tag{5.26}$$

The gradient of c_{s} is easily obtainable with automatic differentiation [101]. In this work, we used TensorFlow [90].

5.3.3 Robust stability gradients

A state-space representation for $T_{z_1 w_1}(s)$ is obtained with the linear fractional transformation $T_{z_1 w_1}(s) = \mathcal{F}_L[P_{\mathrm{red}}(s), K(s)]$, where $P_{\mathrm{red}}(s)$ represents $[z_1, e]^\top = P_{\mathrm{red}}(s)[w_1, u]^\top$, and is partitioned as follows

$$P_{\mathrm{red}}(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{5.27}$$

Then,

$$T_{z_1 w_1}(s) = \left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array}\right] = \left[\begin{array}{cc|c} A + B_2 D_{\mathrm{K}} C_2 & B_2 C_{\mathrm{K}} & B_1 + B_2 D_{\mathrm{K}} D_{21} \\ B_{\mathrm{K}} C_2 & A_{\mathrm{K}} & B_{\mathrm{K}} D_{21} \\ \hline C_1 + D_{12} D_{\mathrm{K}} C_2 & D_{12} C_{\mathrm{K}} & D_{11} + D_{12} D_{\mathrm{K}} D_{21} \end{array}\right], \tag{5.28}$$

and the complex transfer matrices for the various frequencies ω can be obtained with

$$T_{z_1 w_1}(j\omega) = \bar{C}(j\omega I - \bar{A})^{-1}\bar{B} + \bar{D}. \tag{5.29}$$

The robust stability criterion is $\|T_{z_1 w_1}(s)\|_\infty < 1$. This can be verified by computing the maximum singular value $\bar{\sigma}$ of the complex transfer matrices $T_{z_1 w_1}(j\omega)$ on a sufficiently dense frequency grid. Thus, $c_\mathrm{r}(\boldsymbol{\phi})$ becomes

$$c_\mathrm{r}(\boldsymbol{\phi}) = \max\left\{1, \sup_\omega \bar{\sigma}\left(T_{z_1 w_1}(j\omega)\right)\right\}, \tag{5.30}$$

$$\text{where} \quad \bar{\sigma}(M) = \left[\bar{\lambda}(M^* M)\right]^{\frac{1}{2}}, \tag{5.31}$$

and $\bar{\lambda}$ is the largest eigenvalue, which is positive real. Here again, the gradients are obtained through automatic differentiation.

5.4 Example problem

The method in Algorithm 2 is applied to the problem of simultaneous lane change and vehicle acceleration. The vehicle model used is that of Section 4.2.2, also pictured on Figure 4.3a. Full state feedback is assumed, so $\mathbf{y} = \mathbf{x}$. The states are $\mathbf{x} = [\dot{x}, \dot{y}, \dot{\psi}, \psi, Y, \phi]^\top$ and controls are $\mathbf{u} = [F_{\mathrm{r}x}, F_{\mathrm{f}x}, \phi_\mathrm{r}]^\top$. The full system dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ are described by Equations (4.7), (4.17)–(4.20) reproduced below.

$$\ddot{x} = \tfrac{1}{m}\left(F_{\mathrm{r}x} + F_{\mathrm{f}x} - c_\mathrm{f}\left(\phi^2 - \frac{\phi(\dot{\psi} l_\mathrm{f} + \dot{y})}{\dot{x}}\right) - \tfrac{1}{2}\rho \dot{x}^2 a_\mathrm{f} c_\mathrm{d}\right) + \dot{y}\dot{\psi}, \tag{4.17}$$

$$\ddot{y} = \tfrac{1}{m}\left(F_{\mathrm{f}x}\phi + c_\mathrm{r}\left(\frac{\dot{\psi} l_\mathrm{r} - \dot{y}}{\dot{x}}\right) + c_\mathrm{f}\left(\phi - \frac{\dot{\psi} l_\mathrm{f} + \dot{y}}{\dot{x}}\right)\right) - \dot{x}\dot{\psi}, \tag{4.18}$$

$$\ddot{\psi} = \tfrac{1}{I_\mathrm{z}}\left(F_{\mathrm{f}x} l_\mathrm{f}\phi - c_\mathrm{r} l_\mathrm{r}\left(\frac{\dot{\psi} l_\mathrm{r} - \dot{y}}{\dot{x}}\right) + c_\mathrm{f} l_\mathrm{f}\left(\phi - \frac{\dot{\psi} l_\mathrm{f} + \dot{y}}{\dot{x}}\right)\right), \tag{4.19}$$

$$\dot{Y} = \dot{x}\psi - \dot{y}, \tag{4.20}$$

$$\dot{\phi} = \lambda_\mathrm{s}(\phi_\mathrm{r} - \phi). \tag{4.7}$$

5.4.1 Uncertainty cover

The system model is linearized to obtain the nominal model $G(s)$ following the method discussed in Section 4.2.3: taking the Jacobian of $f(\mathbf{x}, \mathbf{u})$ with respect to both states and input vectors, and evaluating the expression for nominal states $\mathbf{x}_0 = [\dot{x}_0, 0, 0, 0, 0, 0]^\top$ and inputs

Table 5.1: Vehicle parameters.

Parameter	Unit	Nominal	Uncertainty
m	kg	2000	±200
I_z	$\mathrm{kg\,m^2}$	3200	±200
c_f	kN/rad	50	±10
c_r	kN/rad	50	±10
l_f	m	1.1	–
l_r	m	1.7	–
c_d	–	0.24	–
a_f	$\mathrm{m^2}$	2.4	–
ρ	$\mathrm{kg/m^3}$	1.225	–
λ_s	$\mathrm{s^{-1}}$	8	±2
$\dot{x}$	m/s	25	+5
$\dot{y}$	m/s	0	±1.5
$\dot{\psi}$	rad/s	0	±0.15
ψ	rad	0	±0.15
δ	rad	0	±0.06
$F_{\mathrm{f}x}$	kN	0	-0.1/+2

$\mathbf{u}_0 = [0, 0, 0]^\top$. This yields the A_G and B_G matrices of $G(s)$. Assuming full state feedback, $C_\mathrm{G} = I$.

Input multiplicative uncertainty is used to cover the set of all possible system dynamics, where $\mathcal{G} := \{G(s)(I + W_1(s)\Delta(s)W_2(s))\}$ and $\|\Delta(s)\|_\infty < 1$. The filters $W_1(s)$ and $W_2(s)$ must be designed such that $\mathcal{G}$ covers $f(\mathbf{x}, \mathbf{u})$. This can be achieved by first sampling multiple possible system linearizations of $f(\mathbf{x}, \mathbf{u})$. For that, the partial derivatives of Appendix D—which form the Jacobian of the system—are evaluated at different points within the assumed range for the parameters and states presented in Table 5.1. Then, the MATLAB command `ucover` can be used to obtain $W_1(s)$ and $W_2(s)$ such that all the sampled systems are covered. However, given that $G(s)$ has 6 outputs, it is extremely difficult to cover the system without introducing too much conservatism, even when the covers are fitted with optimization techniques as is the case with the `ucover` command. Therefore, only the $\dot{y}$, $\dot{\psi}$, and ψ outputs are covered with the proposed method—a different C matrix is introduced in the sampled systems. The $W_1(s)$ and $W_2(s)$ filters obtained with this method can still be applied to the complete system. The resulting full-system uncertainty cover can be seen on Figure 5.2.

The filters are able to cover almost all the uncertain dynamics, except the input–output relation from ϕ_r to $\dot{x}$. This is deemed acceptable given the structure of the controller presented in Section 5.4.3. In effect, the longitudinal control is decoupled from the lateral control, and it mostly consists of a feedforward component. The feedback gains on the

Table 5.2: Scenario description.

	Scenarios			
Parameter	1	2	3	4
$\dot{x}_{\mathrm{f}}$ [m/s]	30.0	26.5	28.0	30.0
t_{f1} [s]	3.0	3.5	3.5	5.0
a_{max} [m/s^2]	2.0	0.5	1.0	1.8
Y_{f} [m]	3.7	3.7	3.2	3.8
t_{f2} [s]	4.0	4.5	4.5	4.5

longitudinal control are small, thus the risk of instability resulting from the incomplete uncertainty cover is negligible. The risk of instability in the lateral dynamics is considered more important.

5.4.2 Trajectory and nominal command

The trajectories $\mathbf{x}_{\mathrm{d}}$ consist of concurrent lane change and vehicle acceleration. Only the states $\dot{x}$ and Y are prescribed and the rest of the entries in $\mathbf{x}_{\mathrm{d}}$ are zeros. Four scenarios are used in this work; they are parameterized with the values presented in Table 5.2. The vehicle is accelerated from $\dot{x}_0 = 25\,\mathrm{m/s}$ and reaches $\dot{x}_{\mathrm{f}}$ at time t_{f1}. The acceleration begins at $0\,\mathrm{m/s^2}$, then is gradually increased up to a constant value a_{max}, which is held for a moment until it is brought back to $0\,\mathrm{m/s^2}$ at time t_{f1}. The lateral position begins at $Y_0 = 0\,\mathrm{m}$ and ends at Y_{f} at time t_{f2}. The trajectory of Y follows a 3-4-5 polynomial where the first and second derivatives are null at both ends of the trajectory.

The nominal command $\bar{\mathbf{u}}$ is computed in two independent steps. First, the front and rear tractive forces are computed from the reference velocity profile in $\mathbf{x}_{\mathrm{d}}$, where

$$F(t) = m\ddot{x}(t) + \tfrac{1}{2}\rho\dot{x}(t)^2 a_{\mathrm{f}} c_{\mathrm{d}}, \tag{5.32}$$

$$F_{\mathrm{f}x}(t) = \tfrac{1}{3}F(t), \tag{5.33}$$

$$F_{\mathrm{r}x}(t) = \tfrac{2}{3}F(t). \tag{5.34}$$

Then, the steering input is computed considering only the lateral dynamics of Section 4.2.3 using the model of Equation (4.25). This forms the reduced system with matrices A_{R} and B_{R}. The nominal steering input is then computed with

$$\phi_{\mathrm{r}}(t) = B_{\mathrm{R}}^{\top} \mathrm{e}^{A_{\mathrm{R}}^{\top}(t_{\mathrm{f2}}-t)} W_{\mathrm{c}}^{-1}(t_{\mathrm{f2}})[0,0,0,Y_{\mathrm{f}},0]^{\top}, \tag{5.35}$$

where $W_{\mathrm{c}}(t)$ is the controllability Gramian.

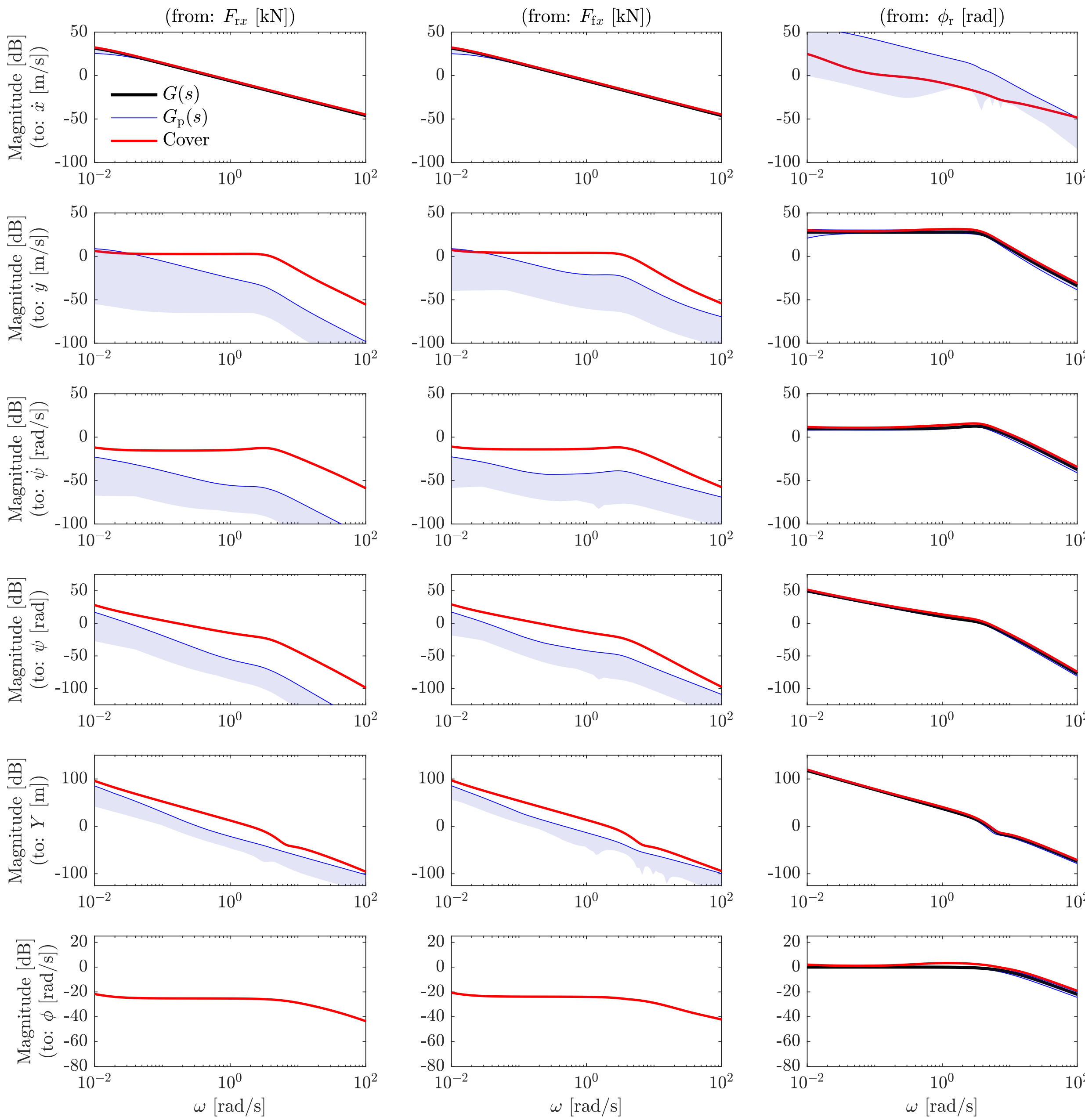


Figure 5.2: Uncertainty Cover.

5.4.3 Controller structure and initialization

The feedback controller is structured to decouple the control of the longitudinal and lateral dynamics. The front $F_{\mathrm{f}x}$ and rear $F_{\mathrm{r}x}$ tractive forces are used only for the feedback control of the vehicle velocity $\dot{x}$, while the steering input ϕ_{r} is used only for controlling the lateral dynamics. As will become apparent in the result section, a simple integrator is sufficient for the vehicle speed control. Moreover, the tracking error is so negligible that no further controller tuning is required for this integrator.

The control of the lateral dynamics is more difficult, however. An initial controller is obtained with $\mathcal{H}_\infty$ loop shaping. For that, a nominal linear model $G(s)$ is built from the same A_{R} and B_{R} matrices introduced in the previous section. Then, a controller that stabilizes $G(s)$ and minimizes the tracking error is obtained using the MATLAB command `hinfsyn`. This $\mathcal{H}_\infty$ controller is assembled with the simple integrator obtained previously to form the complete initial feedback controller K_{i}, which has 24 internal states, and has $D_{\mathrm{K}} = 0$. Together the A_{K}, B_{K}, and C_{K} matrices of K_{i} contain 792 entries.

5.4.4 Controller training

Training with nominal model

The controller K_{t} is tuned with simulated rollouts, see line 6 in Algorithm 2. The system model $f_{\mathrm{n}}(\mathbf{x}, \mathbf{u})$ is that of Equations (4.7), (4.17)–(4.20) with the nominal parameters shown in Table 5.1. The simulations have a time step of $T_{\mathrm{s}} = 0.02\,\mathrm{s}$. The cost function used is

$$c(\mathbf{x}_{[t]}) = (\mathbf{x}_{[t]} - \mathbf{x}_{\mathrm{d}[t]})^\top Q(\mathbf{x}_{[t]} - \mathbf{x}_{\mathrm{d}[t]}), \tag{5.36}$$

where Q is a diagonal matrix weighting the relative importance of the state errors.

Only the first three scenarios of Table 5.2 are used during training. Scenario 4 is reserved for testing that the controller also does well on a scenario never seen during training.

Training with learned model

The controller K_{a} is also tuned from simulated policy rollouts, but with a learned model of the system dynamics this time, see line 14 in Algorithm 2. In this example problem, the actual (unknown) system has an altered tire stiffness coefficient of $c_{\mathrm{f}} = c_{\mathrm{r}} = 40\,\mathrm{kN/rad}$. We chose to learn the dynamics using a model that has the form

$$f_{\mathrm{l}} = f_{\mathrm{n}} + [g_1(\mathbf{z}), g_2(\mathbf{z}), g_3(\mathbf{z}), 0, 0, 0]^\top \tag{5.37}$$

where $g_d(\mathbf{z})$ are functions that represent the unknown dynamics and $\mathbf{z} = [\alpha_{\mathrm{f}}, \alpha_{\mathrm{r}}]^\top$ is the feature vector used in the prediction models. To learn f_{l}, first a lane change is simulated for Scenario 1 with K_{t} (line 12 in Algorithm 2), and a dataset $\mathcal{D} = \{\mathbf{x}_{[i]}, \mathbf{u}_{[i]}\}$ composed of $n_{\mathrm{p}}+1$ data points is collected. Then Gaussian processes are used to learn the difference between the nominal system dynamics f_{n} and the actual dynamics. The targets to be learned are issued from the prediction error

$$\mathbf{e}_{\mathrm{p}[i]} = \mathbf{x}_{[i+1]} - f_{\mathrm{n}}(\mathbf{x}_{[i]}, \mathbf{u}_{[i]}). \tag{5.38}$$

More specifically, each of the three dimensions to be learned has its own target vector $\mathbf{t}_d \in \mathbb{R}^{n_{\mathrm{p}}}$, which are composed of the d-th component of the error vectors, such that

$$\mathbf{t}_d = \left[[\mathbf{e}_{\mathrm{p}[1]}]_d, \ldots, [\mathbf{e}_{\mathrm{p}[n_{\mathrm{p}}]}]_d\right]^\top. \tag{5.39}$$

All three dimensions share the same set of feature vectors $\{\mathbf{z}_{[i]}\}$. The Gaussian processes used in this work has a zero mean function $m(\mathbf{z}) := 0$ and a square exponential kernel function $k(\mathbf{z}, \mathbf{z}')$, just as in Section 3.4.1. The predictive functions $g_d(\mathbf{z})$ are simply the mean of the Gaussian processes evaluated at the test inputs $\mathbf{z}_*$

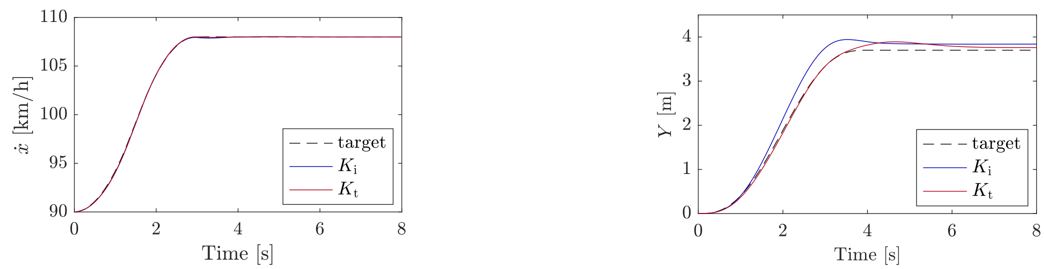
$$g_d(\mathbf{z}_*) = \mu_d(\mathbf{z}_*) = K_{\mathbf{z}_*\mathbf{z}}(K_{\mathbf{zz}} + \sigma_\epsilon^2 I)^{-1}\mathbf{t}_d, \tag{5.40}$$

where $K_{\mathbf{zz}} = [k(\mathbf{z}_{[i]}, \mathbf{z}_{[j]})]_{ij}$, and $K_{\mathbf{z}_*\mathbf{z}} = [k(\mathbf{z}_*, \mathbf{z}_{[i]})]_i$, also following Section 3.4.1. Here again, the hyper-parameters $\boldsymbol{\theta}$ are tuned by maximizing the logarithm of the marginal likelihood of the observed data points in $\mathcal{D}$.

5.5 Results and discussion

Figure 5.3 shows that the tracking error is negligible for the vehicle speed $\dot{x}$, but not for the lateral position Y. Therefore, the result section only discusses the tracking of the lateral position. For every stimulation result shown in this section, the tire slip angle was verified to confirm that the linear tire model remains appropriate.

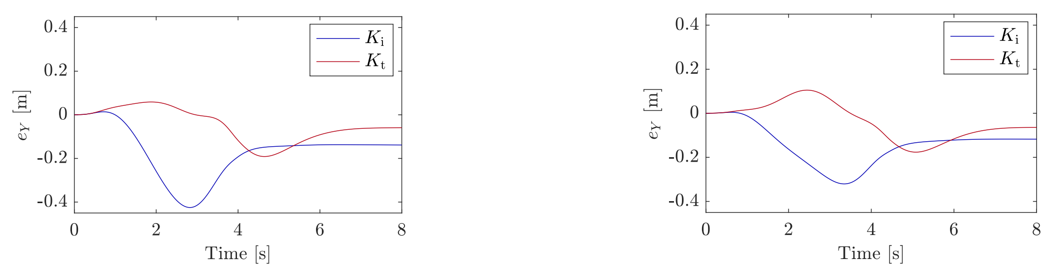
Figure 5.3b shows that the controller tuned from simulated policy rollouts (K_{t}) improves



(a) Vehicle speed, Scenario 1.

(b) Lateral position, Scenario 1.

Figure 5.3: Simulated policy rollouts on the nominal model ($c_\mathrm{f} = c_\mathrm{r} = 50\,\mathrm{kN/rad}$). The tuned controller K_t improves the lateral position tracking compared to initial controller K_i.



(a) Lateral position error, Scenario 1.

(b) Lateral position error, Scenario 4.

Figure 5.4: Simulated policy rollouts on the nominal model ($c_\mathrm{f} = c_\mathrm{r} = 50\,\mathrm{kN/rad}$). The tuned controller K_t improves the tracking performance for both a scenario used during training (Scenario 1) and a scenario not used during training (Scenario 4).

the tracking performance when compared to the initial controller (K_i) obtained from $\mathcal{H}_\infty$ loop-shaping. Figure 5.4a shows the tracking error e_Y on the lateral position for this run, and Table 5.3 presents the $\mathcal{L}_2$-norm of this error signal. The tuning method also reduces the tracking error for a lane-change scenario that was not used during training. In effect, both Figure 5.4b and Table 5.3 show that K_t does better than K_i for Scenario 4. This suggests that the tuning method does not overfit on the training scenarios, and can generalize to others. Finally, the proposed method keeps the tuned controller K_t robustly stabilizing, which can be seen in Figure 5.5.

The tuning of K_t requires 4000 training epochs. The first 2000 epochs are done using simulated rollouts with a duration of 8 s (simulated time), and the last 2000 epochs, with 40-s rollouts. The longer rollouts are used to ensure that the low-frequency dynamics are also considered during the controller tuning. The total training time for the first 2000 epochs is 1083 s, and for the last 2000 epochs, 5245 s. Figure 5.6 shows the evolution of the performance cost c_p in (a), the maximum real part of all eigenvalues in the matrices M_1 to M_4

Table 5.3: $\mathcal{L}_2$-norm of the tracking error on Y (e_Y) for the 0–8 s time span.

	Scenarios			
Nominal model	1	2	3	4
K_i	3.85	2.22	2.92	3.39
K_t	1.78	1.53	1.33	1.80
Actual dynamics				
K_t	3.84	3.85	3.11	3.87
K_a	2.21	2.55	1.90	2.27

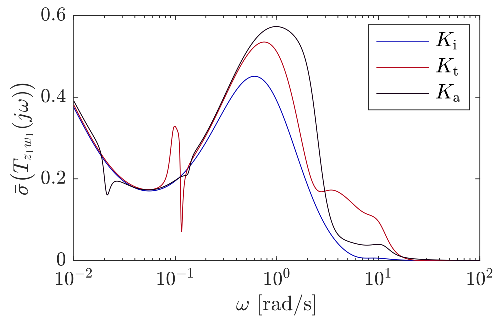
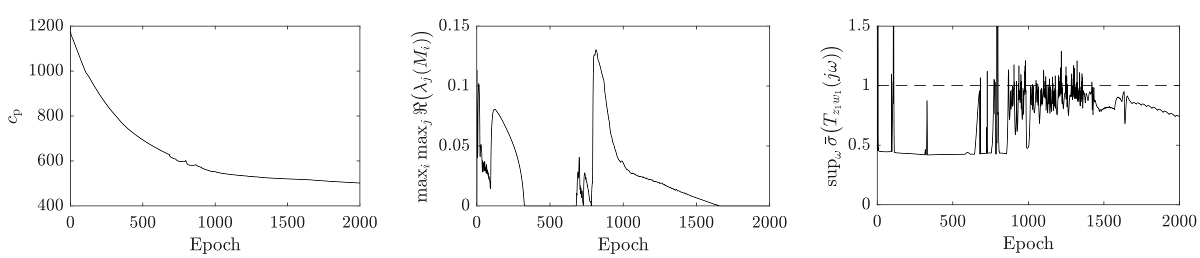


Figure 5.5: The maximum singular value $\bar{\sigma}\big(T_{z_1w_1}(j\omega)\big)$ for K_i, K_t, and K_a.

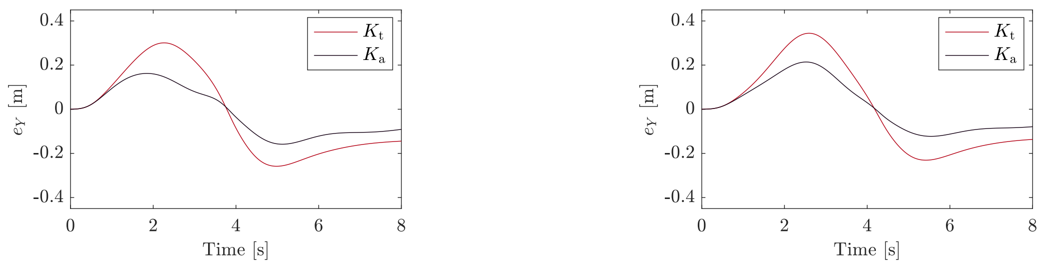
in (b), and the maximum singular value of $T_{z_1w_1}$ in (c). This indicates that the penalties $\xi_\mathrm{s}c_\mathrm{s}$ and $\xi_\mathrm{r}c_\mathrm{r}$ were sufficient to enforce both the nominal and robust stability constraints. Figure 5.6 shows the training behavior when the weights are $\xi_\mathrm{s} = 10{,}000$ and $\xi_\mathrm{r} = 1000$. For smaller values of ξ_s and ξ_r however, the penalties were not strong enough to enforce the stability constraints. The gradient descent algorithm used during training is Adam [144], which helps stabilize the gradients compared to a simple stochastic gradient descent algorithm.

Figure 5.7 and Table 5.3 show that the tracking error of K_t increases for policy rollouts on the actual dynamics, i.e., when $c_\mathrm{f} = c_\mathrm{r} = 40\,\mathrm{kN/rad}$. However, the controller remains stable, as the actual dynamics are still within the assumed uncertainty bounds. This allows



(a) Performance cost c_p. **(b)** Nominal stability criteria. **(c)** Robust stability criteria.

Figure 5.6: Evolution of the performance cost, nominal stability criterion, and robust stability criterion during the first 2000 training epochs for K_t.



(a) Lateral position, Scenario 1.

(b) Lateral position, Scenario 4.

Figure 5.7: Simulated policy rollouts on the actual system dynamics ($c_{\mathrm{f}} = c_{\mathrm{r}} = 40\,\mathrm{kN/rad}$). The controller K_{a} has a better tracking performance than K_{t}. This is because K_{a} was tuned on a learned model f_{l} while K_{t} was only tuned on a nominal model f_{n}.

to bring back the performance to a similar level by re-tuning the controller with a learned model. As a result, the adjusted controller K_{a} reduces the tracking error, even for the unseen Scenario 4. Training with the learned model is longer, but still reasonable. The same 2000 epochs of 8-s rollouts takes 2071 s of training time, which is approximately twice the time that was required for the nominal model.

5.5.1 Augmented cost

It is possible to augment the total cost $J(\boldsymbol{\phi})$ of Equation (5.8) with $c_{\mathrm{t}}(\boldsymbol{\phi}) = \|T_{z_2 w_2}(s)\|_\infty$ to help shape the closed-loop behavior with frequency-based specifications. The gradients $\nabla_{\boldsymbol{\phi}} c_{\mathrm{t}}(\boldsymbol{\phi})$ can be computed with the method of Section 5.3.3—which was used to obtain $\nabla_{\boldsymbol{\phi}} c_{\mathrm{r}}(\boldsymbol{\phi})$—but with $T_{z_2 w_2}(j\omega)$ instead of $T_{z_1 w_1}(j\omega)$. This additional cost c_{t} could be used in problems where scenario-based tuning alone is inadequate. One such example could be a controller that must account for a large frequency range. In effect, if T_{s} is small enough to allow the simulation of high-frequency dynamics, it also means that the policy rollouts must have a long time span to also include the effects of low-frequency dynamics. Therefore, the cost c_{t} could be used to penalize the low-frequency errors, while the scenario-based performance cost c_{p} tunes the high-frequency behavior. In the lane-change problem however, this was found unnecessary. As previously discussed, it was sufficient to simulate 40-s rollouts. As a final remark, the nominal model $G(s)$ is not learned in the proposed method, so the additional cost c_{t} could be less beneficial if the actual system deviates significantly from $G(s)$.

Chapter 6

Structured learning for state constraints

This chapter presents the safe uncertainty-learning principle, which can be used to structure the learning component of a safety controller. After the principle is motivated and introduced, a learning method based on control barrier functions is presented. Then, two vehicular control examples are demonstrated and the principle is discussed.

Contents

6.1 Review of safety guarantees for state constraints **85**

6.2 Safe uncertainty-learning principle **86**

6.3 Learned safety condition **86**

6.3.1 Robust safety condition from a control barrier function 87

6.3.2 Robustly safe controller 89

6.3.3 Learning parametric uncertainty 90

6.4 Example 1: vehicle lateral control **91**

6.4.1 System model and safety condition 91

6.4.2 Results and discussion 94

6.5 Example 2: vehicle longitudinal control **96**

6.5.1 System model and safety condition 96

6.5.2 Results and discussion 99

6.6 Discussion **101**

6.1 Review of safety guarantees for state constraints

For a large class of dynamical systems—e.g., autonomous vehicles and unmanned aerial vehicles—the notion of safety implies keeping the system within state constraints. The general approach is to use a model of the system dynamics, and derive a condition on the control signals that guarantees system safety. Three such methods are: model predictive control (MPC) [145], control barrier function (CBF) [146], and Hamilton-Jacobi reachability analysis (HJR) [147].

In MPC, the control signal is optimized at every time step by predicting the future system states for a finite time-horizon. This optimization can account for state constraints, and therefore keep the system safe [123, 148, 149]. A CBF is a Lyapunov-like scalar function whose output depends on whether the system states are within a given set, and how far they are from the set boundary. A CBF and its time derivatives can be used to obtain a condition on the control signals for the system to remain within the given set, thereby enforcing forward invariance on that set. If the chosen set is within state constraints, then the CBF can be used to keep the system safe [150, 151]. In HJR, a target set is defined as the set of states that violate the constraints, and a backward reachable (BR) set is defined as the set of states that could lead the system to the target set despite the system's best possible actions. The BR set, the optimal action, and a value function can all be computed by solving the Hamilton-Jacobi partial differential equation for this system, typically through dynamic programming. The system can be kept safe by applying the optimal action whenever the system comes close to the BR set [152, 153]. These methods can also be combined: for instance, the value function obtained though HJR can be used as a CBF [154].

For systems with uncertain dynamics, all three methods outlined above can be adapted to consider bounded system uncertainty [155–157]. Generally, this is nontrivial and often leads to a higher computational burden for the controller. This is because the controller must now predict the system evolution considering the set of possible system dynamics. Also, this can lead to overly conservative controllers if the uncertainty description is itself too conservative. For instance, this can arise if the method is formulated for a rather general uncertainty description, and the actual system uncertainty is in fact more restricted—more structured. Naturally, this motivates using machine learning to adapt the system model from data collected during the system's operation. An approach is where instead of using uncertainty bounds, a nominal model is learned from data, which reduces the risk associated with a discrepancy between the system model and the actual dynamics [158]. Another approach is where uncertain system dynamics are considered, and the uncertainty bounds are learned from data, thereby increasing the performance of the system by obtaining tighter uncertainty bounds [100, 159–162].

6.2 Safe uncertainty-learning principle

However, the use of machine learning to derive the safety condition also introduces certain risks. These risks are not sufficiently discussed in the literature, or at least, not explicitly enough. In this chapter, the following principle is introduced.

Safe uncertainty-learning principle (1). Take an uncertain dynamical system with safety defined by state constraints, which must be respected at all time during operation. Suppose machine learning is used to obtain a condition on the control signal that will guarantee the system's safety. For the learning-based control method to preserve safety guarantees:

1. A robust safety condition is necessary, where uncertainty bounds are considered.
2. The uncertainty bounds must be initialized conservatively.
3. The uncertainty bounds should only be tightened if it can be assumed that the collected data sufficiently captures the future behavior of the system. Particularly challenging adversarial events are a sudden change in the system dynamics or a rare disturbance. If expected, these events must be distinctly accounted for, since it could be impossible to model them from previously collected data.

The argument is based on contraposition: through two example problems, it is shown that when one of these conditions is not respected, the safety guarantee is lost. The examples are 1) the lateral control of an autonomous vehicle through a lane-change maneuver, and 2) the longitudinal control of an autonomous vehicle in a two-vehicle platoon, also commonly called adaptive cruise control. Both these problems were investigated several times in the literature, including with some of the approaches presented above [146, 148, 163]. In this work, the example problems are solved using robust exponential control barrier functions for the safety condition, and maximum likelihood estimation for the uncertainty learning.

6.3 Learned safety condition

This section presents the approach to derive the learned safety condition for the example problems of Sections 6.4 and 6.5. It begins with the introduction of CBFs and exponential-CBFs. Then uncertainty is considered and robust exponential-CBFs are introduced. The definitions and theorems are largely adapted from [151, 160]. Finally, maximum likelihood estimation is briefly discussed.

6.3.1 Robust safety condition from a control barrier function

Control barrier function

Take a system with states $\mathbf{x} \in \mathbb{R}^n$, controls $\mathbf{u} \in \mathbb{R}^m$, and dynamics

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}. \tag{6.1}$$

If the dynamics are locally Lipschitz, then given an initial condition $\mathbf{x}_0$, there exists a maximum time interval $I(\mathbf{x}_0) = [t_0, T)$ such that $\mathbf{x}(t)$ is a unique solution on $I(\mathbf{x}_0)$. Let a closed convex set $\mathcal{S} \subset \mathbb{R}^n$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ where

$$\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^n | h(\mathbf{x}) \geq 0\}, \tag{6.2}$$

$$\partial\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^n | h(\mathbf{x}) = 0\}, \tag{6.3}$$

$$\mathrm{int}(\mathcal{S}) := \{\mathbf{x} \in \mathbb{R}^n | h(\mathbf{x}) > 0\}. \tag{6.4}$$

Definition 2. The set $\mathcal{S}$ is forward invariant if for every $\mathbf{x}_0 \in \mathcal{S}$, $\mathbf{x}(t) \in \mathcal{S}\ \forall t \in I(\mathbf{x}_0)$.

Definition 3. A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is an extended class $\mathcal{K}_\infty$ function if it is strictly increasing, $\alpha(0) = 0$, and is defined on the entire real line.

Definition 4. Let $\mathcal{S}$ be a 0-superlevel set for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, then h is a control barrier function if there exists an extended class $\mathcal{K}_\infty$ function α such that for all $\mathbf{x} \in \mathcal{S}$

$$\sup_{\mathbf{u} \in \mathcal{U}} \dot{h}(\mathbf{x}, \mathbf{u}) \geq -\alpha(h(\mathbf{x})). \tag{6.5}$$

Theorem 4. Let $\mathcal{S}$ be a 0-superlevel set for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, if h is a control barrier function for (6.1) on $\mathcal{S}$, then any Lipschitz continuous controller satisfying (6.5) renders the set $\mathcal{S}$ forward invariant [151].

As a result of Theorem 4, if the set $\mathcal{S}$ is defined as the safe set, then any controller that meets Condition (6.5) keeps the system safe.

Exponential control barrier function

For systems where $\dot{h}(\mathbf{x})$ does not depend on $\mathbf{u}$, Definition 4 cannot be used to find a control input $\mathbf{u}$ that keeps the system safe. It is possible, however, to use higher derivatives of $h(\mathbf{x})$. For a system where the r^{th} time derivative of $h(\mathbf{x})$ depends on $\mathbf{u}$, but not any of the lower derivatives, a new system with states $\eta(\mathbf{x}) := [h(\mathbf{x}), \dot{h}(\mathbf{x}), \ddot{h}(\mathbf{x}), \dots, h^{(r-1)}(\mathbf{x})]^\top$, input $\mu = h^{(r)}(\mathbf{x}, \mathbf{u})$, and output $h(\mathbf{x})$ can be formed. The dynamics of this system are

$$\dot{\eta}(\mathbf{x}) = F\eta(\mathbf{x}) + G\mu, \tag{6.6}$$

$$h(\mathbf{x}) = C\eta(\mathbf{x}), \tag{6.7}$$

with

$$F = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{6.8}$$

If the input $\mu = -K_\alpha \eta(\mathbf{x})$ is chosen, then $h(\mathbf{x}(t)) = Ce^{(F-GK_\alpha)t}\eta(\mathbf{x}_0)$. By the comparison lemma, if $\mu \geq -K_\alpha \eta(\mathbf{x})$, then $h(\mathbf{x}(t)) \geq Ce^{(F-GK_\alpha)t}\eta(\mathbf{x}_0)$.

Assumption 2. K_α is chosen such that the poles $p_i,\ i \in \{1, \dots, r\}$ of $(F - GK_\alpha)$ are real and negative.

Assumption 3. Define a family of recursive functions $\nu_i : \mathbb{R}^n \to \mathbb{R},\ i \in \{0, \dots, r\}$, and corresponding superlevel sets $\mathcal{S}_i$ as

$$\nu_i(\mathbf{x}) = \dot{\nu}_{i-1}(\mathbf{x}) - p_i \nu_{i-1}(\mathbf{x}), \tag{6.9}$$

$$\mathcal{S}_i = \{\mathbf{x} \in \mathbb{R}^n \mid \nu_i(\mathbf{x}) \geq 0\}, \tag{6.10}$$

and define $\nu_0(\mathbf{x}) := h(\mathbf{x})$. Then, K_α is chosen such that $\nu_i(\mathbf{x}_0) \geq 0,\ i \in \{1, \dots, r\}$.

Definition 5. Let $\mathcal{S}$ be a 0-superlevel set for a r-times continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, then h is an exponential control barrier function if there exists a row vector $K_\alpha \in \mathbb{R}^r$ constrained by Assumptions 2 and 3, such that for all $\mathbf{x} \in \mathcal{S}$

$$\sup_{\mathbf{u} \in \mathcal{U}} h^{(r)}(\mathbf{x}, \mathbf{u}) \geq -K_\alpha \eta(\mathbf{x}). \tag{6.11}$$

Robust exponential control barrier function

For systems where the dynamics are uncertain, Definition 5 is inappropriate to find a control input $\mathbf{u}$ that keeps the system safe. Take an uncertain system model where the dynamics depend explicitly on an unknown parameter vector $\boldsymbol{\delta}$, with each parameter bounded in magnitude such that $\boldsymbol{\delta} \in \mathcal{D}$, where

$$\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\delta}) + g(\mathbf{x}, \boldsymbol{\delta})\mathbf{u}. \tag{6.12}$$

Now, the control barrier function $h(\mathbf{x}, \boldsymbol{\delta})$ and its derivatives depend on the unknown parameters $\boldsymbol{\delta}$. This requires to adapt Assumption 3 and Definition 5.

Assumption 4. Define a family of recursive functions $\nu_i : \mathbb{R}^n \to \mathbb{R},\ i \in \{0, \dots, r\}$, and corresponding superlevel sets $\mathcal{S}_i$ as

$$\nu_i(\mathbf{x}, \boldsymbol{\delta}) = \dot{\nu}_{i-1}(\mathbf{x}, \boldsymbol{\delta}) - p_i \nu_{i-1}(\mathbf{x}, \boldsymbol{\delta}), \tag{6.13}$$

$$\mathcal{S}_i = \{\mathbf{x} \in \mathbb{R}^n \mid \nu_i(\mathbf{x}, \boldsymbol{\delta}) \geq 0\}, \tag{6.14}$$

and define $\nu_0(\mathbf{x}, \boldsymbol{\delta}) := h(\mathbf{x}, \boldsymbol{\delta})$. Then, K_α is chosen such that $\nu_i(\mathbf{x}_0, \boldsymbol{\delta}) \geq 0,\ i \in \{1, \dots, r\},\ \forall \boldsymbol{\delta} \in \mathcal{D}$.

Definition 6. Let $\mathcal{S}$ be a 0-superlevel set for a r-times continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, then h is a robust exponential control barrier function if there exists a row vector $K_\alpha \in \mathbb{R}^r$ constrained by Assumptions 2 and 4 such that for all $\mathbf{x} \in \mathcal{S}$ and for all $\boldsymbol{\delta} \in \mathcal{D}$

$$\sup_{\mathbf{u} \in \mathcal{U}} h^{(r)}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}) \geq -K_\alpha \eta(\mathbf{x}, \boldsymbol{\delta}). \tag{6.15}$$

6.3.2 Robustly safe controller

The robustly safe controller used in this work consists of correcting a nominal $k_{\mathrm{n}}(\mathbf{x})$ controller's output whenever Condition (6.15) is not respected. This can be expressed as the following optimization problem.

$$\mathbf{u} = \underset{\mathbf{u} \in \mathcal{U}}{\arg\min}\ \ \tfrac{1}{2}\|\mathbf{u} - k_{\mathrm{n}}(\mathbf{x})\|_2^2, \tag{6.16}$$

$$\text{s.t. } h^{(r)}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}) \geq -K_\alpha \eta(\mathbf{x}, \boldsymbol{\delta}),\ \forall \boldsymbol{\delta} \in \mathcal{D}. \tag{6.17}$$

This optimization problem is in general nontrivial. It may be possible to further structure the problem depending on the system equations, the uncertainty description, and the chosen control barrier function.

6.3.3 Learning parametric uncertainty

When the controller is initialized, conservative bounds $\mathcal{D}$ are estimated for the parameters $\boldsymbol{\delta}$. As the controller is deployed and data is collected, it may be possible to update the bounds for $\boldsymbol{\delta}$. In this work, the following structure is assumed

$$\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}) + g(\mathbf{x}, \boldsymbol{\theta})\mathbf{u}, \tag{6.18}$$

$$\mathbf{y} = l(\mathbf{x}, \boldsymbol{\theta}), \tag{6.19}$$

$$\mathbf{z}_{[n]} = \mathbf{y}_{[n]} + G\mathbf{w}_{[n]}, \tag{6.20}$$

$$\mathbf{w}_{[n]} \sim \mathcal{N}\big(\mathbf{0}, \Sigma_{\mathbf{w}}\big), \tag{6.21}$$

where $\boldsymbol{\theta}$ are uncertain model parameters, which include the parameters $\boldsymbol{\delta}$ of interest, $\mathbf{y}$ are the system output, $\mathbf{z}_{[n]}$ are the n-indexed measurements contaminated by Gaussian noise $\mathbf{w}_{[n]}$. The noise is assumed to be zero-mean and to have a known and diagonal covariance matrix $\Sigma_{\mathbf{w}}$. To obtain the best estimate for the parameters $\boldsymbol{\theta}$, the likelihood of the observed data $p(z|\boldsymbol{\theta})$ is maximized. For that, $p(z|\boldsymbol{\theta})$ is assumed Gaussian. Maximizing $p(z|\boldsymbol{\theta})$ is then equivalent to maximizing

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{z}_{[n]} - \mathbf{y}_{[n]})^\top \Sigma_{\mathbf{w}}^{-1}(\mathbf{z}_{[n]} - \mathbf{y}_{[n]}), \tag{6.22}$$

where N is the number of data points collected. In this work, $J(\boldsymbol{\theta})$ is maximized through gradient ascent. Finally, the following covariance matrix is used to estimate the uncertainty bounds around the parameters

$$P = \left\{\sum_{n=1}^{N}\left[\frac{\mathrm{d}\mathbf{y}_{[n]}}{\mathrm{d}\boldsymbol{\theta}}\right]^\top \Sigma_{\mathbf{w}}^{-1}\left[\frac{\mathrm{d}\mathbf{y}_{[n]}}{\mathrm{d}\boldsymbol{\theta}}\right]\right\}^{-1}. \tag{6.23}$$

This method is commonly used in practice to identify system parameters, such as for flight dynamics [164]. A notable caveat, however, is that the recorded data is not guaranteed to provide enough information to identify the parameters with enough accuracy to be useful for the given application. This is mitigated by the fact that we also estimate uncertainty bounds around the parameters, and can therefore decide not to update the parameters when the identification is not precise enough, i.e., when the variance of the parameter estimation in P is too high.

6.4 Example 1: vehicle lateral control

The first example problem is to control a lane-change maneuver performed by an autonomous vehicle. The goal is to bring the vehicle form the middle of a lane where the lateral position is defined as $Y = 0\,\mathrm{m}$, to the middle of a neighboring lane where $Y = 3.7\,\mathrm{m}$. However, the vehicle must not exceed $Y_{\max} = 3.85\,\mathrm{m}$ in order to avoid a potential collision with other vehicles. In this problem, the vehicle speed is assumed to remain constant throughout the maneuver and the dynamics of Section 4.2.3 are used.

6.4.1 System model and safety condition

The vehicle is modeled with a bicycle model as shown on Figure 4.3a. The states of the system are $\mathbf{x} = [\dot{y}, \dot{\psi}, \psi, Y, \phi]^\top$, and the control is $\mathbf{u} = [\phi_\mathrm{r}]$. The dynamics are as per Equation (4.25) reproduced below.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}. \tag{6.24}$$

$$A = \begin{bmatrix} \frac{-(c_\mathrm{f}+c_\mathrm{r})}{mv} & \frac{-c_\mathrm{f}l_\mathrm{f}+c_\mathrm{r}l_\mathrm{r}}{mv} - v & 0 & 0 & \frac{c_\mathrm{f}}{m} \\ \frac{-c_\mathrm{f}l_\mathrm{f}+c_\mathrm{r}l_\mathrm{r}}{I_z v} & \frac{-(c_\mathrm{f}l_\mathrm{f}^2+c_\mathrm{r}l_\mathrm{r}^2)}{I_z v} & 0 & 0 & \frac{c_\mathrm{f}l_\mathrm{f}}{I_z} \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_\mathrm{s} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \lambda_\mathrm{s} \end{bmatrix}. \tag{4.25}$$

The actual vehicle parameters are unknown to the controller. The parametric uncertainty shown in Table 6.1 is used to bound the system behavior. The vehicle is assumed to be a medium-duty truck; its mass m may vary significantly depending on the payload it carries. The vehicle's moment of inertia I_z is assumed to vary proportionally to the mass of the vehicle, as well as an unknown parameter δ_{I_z} that accounts for an unknown mass distribution. The wheelbase of the vehicle is known to be $l = 4.5\,\mathrm{m}$. The distance of the front axle to the center of mass is unknown, but assumed to vary around 55% of the wheelbase. The front and rear tire cornering stiffness—c_f and c_r, respectively—are assumed to be proportional to the weight on the given axle, as well as a parameter c_α. This c_α parameter is dependent on the tire construction and its loading condition [165, 166], thus is parameterized as well. Finally, the decay constant λ_s in the steering mechanism is also unknown. This parametrization allows to express the A and B matrices of Equation (4.25) explicitly in terms of the uncertain parameters. Below are the non-zero matrix components, where a_{ij} represents $[A]_{ij}$. As a

Table 6.1: Uncertainty parametrization for Example 1.

Parametrization		Unit	Param.	Nom.	Range
m	$m_\mathrm{n}\delta_m$	kg	m_n	6500	-
I_z	$d_\mathrm{n} m \delta_{I_z}$	$\mathrm{kg\,m^2}$	d_n	4.8	-
l_f	$a_\mathrm{n} l \delta_1$	m	l	4.5	-
l_r	$l - l_\mathrm{f}$	m	a_n	0.55	-
c_α	$c_\mathrm{n}\delta_2$	rad^{-1}	c_n	8	-
c_f	$c_\alpha m g l_\mathrm{r}/l$	N/rad	λ_n	8	-
c_r	$c_\alpha m g l_\mathrm{f}/l$	N/rad	δ_m	1	± 0.3
λ_s	$\lambda_\mathrm{n}\delta_3$	s^{-1}	δ_{I_z}	1	± 0.3
			δ_1	1	± 0.4
			δ_2	1	± 0.4
			δ_3	1	± 0.4

result of the parametrization, δ_m does not appear in these terms.

$$a_{11} = -c_\mathrm{n}\delta_2 g/v, \tag{6.25}$$
$$a_{12} = -v, \tag{6.26}$$
$$a_{15} = c_\mathrm{n}\delta_2 g(1 - a_\mathrm{n}\delta_1), \tag{6.27}$$
$$a_{21} = 0, \tag{6.28}$$
$$a_{22} = -c_\mathrm{n}\delta_2 g l(1 - a_\mathrm{n}\delta_1)a_\mathrm{n}\delta_1/(d_\mathrm{n}\delta_{I_z} v), \tag{6.29}$$
$$a_{25} = c_\mathrm{n}\delta_2 g(1 - a_\mathrm{n}\delta_1)a_\mathrm{n}\delta_1/(d_\mathrm{n}\delta_{I_z}), \tag{6.30}$$
$$a_{55} = -\lambda_\mathrm{n}\delta_3, \tag{6.31}$$
$$b_{51} = \lambda_\mathrm{n}\delta_3. \tag{6.32}$$

The safety condition is turned into an exponential control barrier function $h(\mathbf{x})$ with the following derivatives

$$h(\mathbf{x}) = Y_{\max} - Y, \tag{6.33}$$
$$\dot{h}(\mathbf{x}) = -\dot{y} - v\psi, \tag{6.34}$$
$$\ddot{h}(\mathbf{x}) = -a_{11}\dot{y} - a_{15}\phi, \tag{6.35}$$
$$\dddot{h}(\mathbf{x}, \mathbf{u}) = -a_{11}^2\dot{y} + a_{11}v\dot{\psi} - a_{15}(a_{11} + a_{55})\phi - a_{15}b_{51}\phi_\mathrm{r}. \tag{6.36}$$

A system with states $\eta(\mathbf{x}) = [h(\mathbf{x}), \dot{h}(\mathbf{x}), \ddot{h}(\mathbf{x})]^\top$, input $\mu = \dddot{h}(\mathbf{x}, \mathbf{u})$, and dynamics that follow Equations (6.6)–(6.8) is composed. A controller $K_\alpha = [k_1, k_2, k_3]$ is devised with pole placement. As a result of Definition 6, the safety of the system is ensured with the following condition

$$\phi_{\mathrm{r}} \leq \min_{\boldsymbol{\delta} \in \mathcal{D}} s(\mathbf{x}, \boldsymbol{\delta}), \tag{6.37}$$

$$\begin{aligned} s(\mathbf{x}, \boldsymbol{\delta}) := \frac{1}{a_{15} b_{51}} \Big(& k_1 Y_{\max} - \big(k_2 + a_{11} k_3 + a_{11}^2\big) \dot{y} + a_{11} v_0 \dot{\psi} - k_2 v_0 \psi \\ & - k_1 Y - a_{15} (k_3 + a_{11} + a_{55}) \phi \Big), \end{aligned} \tag{6.38}$$

$$\boldsymbol{\delta} = [\delta_1, \delta_2, \delta_3]^\top. \tag{6.39}$$

This safety condition is formulated as a direct limitation on the controller's output signal ϕ_{r}. Therefore, a safe controller must ensure that Condition (6.37) is respected at every time step. This optimization problem is not trivial, as the parameters in $\boldsymbol{\delta}$ appear multiple times in $s(\mathbf{x}, \boldsymbol{\delta})$. In this work, it was found sufficient to discretize the search space $\mathcal{D}$ and evaluate $s(\mathbf{x}, \boldsymbol{\delta})$ at every point of the resulting 3D grid. With 10 points per axis, this requires 1000 function evaluations every time step, which is easily manageable in real-time.

The base controller is a linear state-feedback controller.

$$k_{\mathrm{n}}(\mathbf{x}) = K(\mathbf{x}_{\mathrm{r}} - \mathbf{x}), \tag{6.40}$$

where $\mathbf{x}_{\mathrm{r}}$ is the desired reference state, namely $\mathbf{x}_{\mathrm{r}} = [0, 0, 0, 3.7, 0]^\top$. In this work, K is obtained by solving a LQR problem. Finally, ϕ_{r} is saturated at 0.08 rad to counter the large input resulting when the desired state is suddenly switched to 3.7 m at $t = 0$ s.

In order to reduce the uncertainty bounds around the components of $\boldsymbol{\delta}$, parameter identification is performed according to the method presented in Section 6.3.3, where

$$\mathbf{y}_{[n]} = \dot{\mathbf{x}}_{[n]}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}), \tag{6.41}$$

$$\mathbf{z}_{[n]} = \mathbf{y}_{[n]} + \mathbf{w}_{[n]}, \tag{6.42}$$

$$\Sigma_{\mathbf{w}} = 0.1^2 I_5, \tag{6.43}$$

$$\boldsymbol{\theta} = [\delta_{I_z}, \delta_1, \delta_2, \delta_3]^\top. \tag{6.44}$$

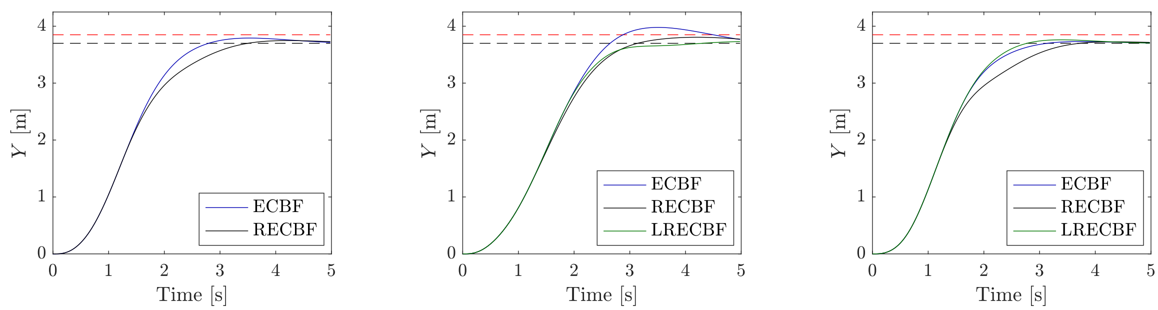
Because the measurements are obtained from a simulated environment, random noise is injected in the measurements as per Equations (6.42) and (6.43) to have a more realistic identification problem.

6.4.2 Results and discussion

Figure 6.1 shows the lateral position of the vehicle for various simulated runs. In Figure 6.1a the actual system parameters are at nominal value, and in Figures 6.1b and 6.1c, the systems have the actual parameters of Vehicles #1 and #2 respectively, which are listed in Table 6.2. In all three plots, the blue lines originate from a safety controller that does not account for uncertainty bounds around the system parameters; it is an exponential control barrier function (ECBF) as presented in Section 6.3.1. This is sufficient to keep the nominal vehicle safe, but not Vehicle #1. When considering uncertainty bounds around the system parameters and using the robust-ECBF (RECBF) presented in Section 6.3.1, all three systems remain safe. Figure 6.2a shows the effect of the RECBF on the control action during the simulation with the nominal vehicle model. The black line represents the controller signal ϕ_r, and the red line, the maximal value that ϕ_r could take and still respect Condition (6.37). When the two lines coincide, it means that the RECBF limits the control action; otherwise, only the nominal controller $k_\mathrm{n}(\mathbf{x})$ is in effect. Figures 6.2b–6.2d show the values of $\boldsymbol{\delta} \in \mathcal{D}$ that minimize $s(\mathbf{x}, \boldsymbol{\delta})$. Most of the time, these values are located at the boundary of $\mathcal{D}$, which supports the assumption that the rather crude optimization method chosen for this problem is adequate.

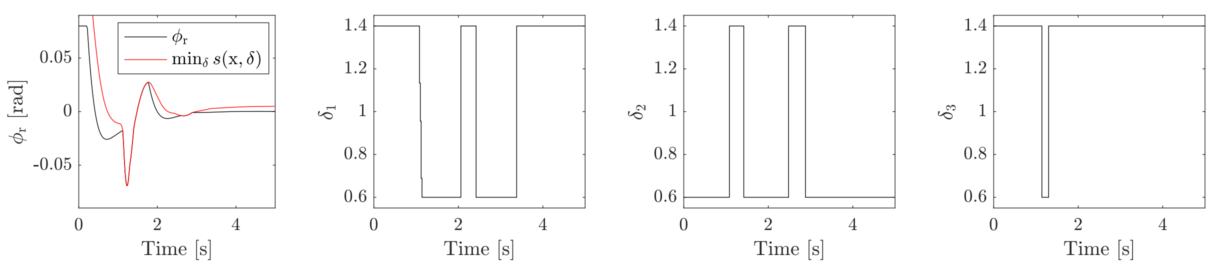
Figures 6.1a and 6.1c show that for some systems, the robust formulation of the safety controller may hinder the system performance by delaying the completion of the lane change. This motivates learning the actual system parameters, and hopefully estimating smaller uncertainty bounds. The system parameters are estimated for the Vehicles #1 and #2 by collecting data issued from the RECBF runs and then applying the learning method. The estimations are shown in Table 6.2. The results for the learned-RECBF (LRECBF) are shown in green in Figure 6.1b and 6.1c. The new parameter bounds for δ_1, δ_2, and δ_3 are updated as $\mu \pm 3\sigma$, where μ is the estimated mean and σ, the estimated standard deviation.

Figure 6.3 highlights the risk associated with sudden changes in system dynamics. The figure shows a simulation when the actual system behaves as Vehicle #1, but the RECBF uses parameter estimates from data collected with Vehicle #2. In practice, this could happen if the controller learns the vehicle parameters, then the vehicle is unloaded at a stop and driven to another location, and the controller does not reset the parameter estimates. In effect, given that the vehicle must perform a lane change to gather the data required to estimate the parameters, and that this lane change must always be performed safely, the only solution would be to reset the uncertainty bounds to the initial conservative values every time the vehicle stops.



(a) Nominal model. **(b)** Vehicle #1. **(c)** Vehicle #2.

Figure 6.1: Simulation results for the lane-change maneuvers of Example 1.



(a) **(b)** **(c)** **(d)**

Figure 6.2: Details of the RECBF simulation for the nominal vehicle model in Figure 6.1a.

Table 6.2: Parameter estimation for Example 1.

	Vehicle #1			Vehicle #2		
Param.	Actual	μ	σ	Actual	μ	σ
δ_m	0.80	-	-	1.20	-	-
δ_{I_z}	1.15	1.202	0.037	1.05	1.048	0.022
δ_1	0.70	0.706	0.009	1.35	1.357	0.005
δ_2	0.60	0.595	0.003	1.35	1.351	0.006
δ_3	1.35	1.386	0.073	1.35	1.354	0.062

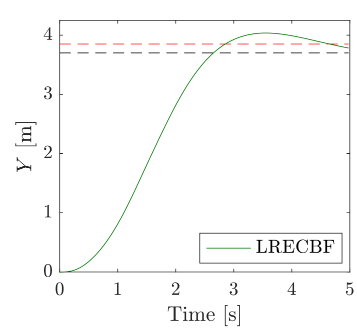


Figure 6.3: Lane-change maneuver with Vehicle #1 using parameters identified from data obtained with Vehicle #2.

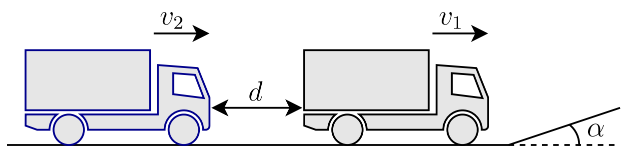


Figure 6.4: Problem setup for Example 2.

6.5 Example 2: vehicle longitudinal control

The second example problem is controlling the trailing vehicle in a two-vehicle platoon, as pictured in Figure 6.4. It is assumed that there is no communication between the two vehicles, and that the leading vehicle operates independently of the trailing vehicle. The trailing vehicle can measure its own speed v_2, the distance d between the two vehicles, and the road grade α. Recall from Section 4.2.1 that the aerodynamic drag on the trailing vehicle reduces when d reduces. Therefore, the control objective is to minimize the distance between the two vehicles, without ever risking a collision.

6.5.1 System model and safety condition

The states are $\mathbf{x} = [v_1, v_2, d]^\top$, the only control action $\mathbf{u} = [u]$ is either a tractive force or a braking force on the second vehicle, and the disturbances are $\mathbf{v} = [a_1, \alpha]^\top$. Following the model of Equation (4.3), the dynamics for the actual system are

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{d} \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{2m}\rho v_2^2 a_\mathrm{f} c_\mathrm{d}(d) - g c_\mathrm{t} \\ v_1 - v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} u + \begin{bmatrix} 1 & 0 \\ 0 & -g \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ \alpha \end{bmatrix}, \tag{6.45}$$

where

$$c_\mathrm{d}(d) = c_\mathrm{d0}\left(1 - \frac{c_1}{c_2 + d}\right). \tag{4.2}$$

In this model, a_1 represents the acceleration of the leading vehicle. Full state measurement is assumed for this problem. The actual vehicle parameters are not exactly known, but they are assumed to be centered around nominal values as shown in Table 6.3. The actual vehicle exhibits a behavior where $c_\mathrm{d}(d)$ decreases when d decreases, as per Equation (4.2) and Figure 4.2b, with the c_1 and c_2 parameters of Table 6.3. However, for the system model used in the computation of the nominal and safety controllers, this effect is neglected. Instead, it is assumed that

$$c_\mathrm{d} = c_\mathrm{d0}. \tag{6.46}$$

Table 6.3: Uncertain parameters for Example 2.

Parameter	Nominal	Actual	Range	Unit
m	6500	5000	[4500, 8500]	kg
$a_\mathrm{f}c_\mathrm{d0}$	4.9	4.2	[3.4, 5.6]	m^2
c_1	-	10	-	m
c_2	-	32	-	m
c_t	0.006	0.007	[0.005, 0.007]	-
Disturbance	**Nominal**	**Actual**	**Range**	**Unit**
a_1	0	-	[-9, 2]	$\mathrm{m/s}^2$
α	0	-	[-0.06, 0.06]	rad

This assumption is compensated by considering a larger uncertainty on the $a_\mathrm{f}c_\mathrm{d0}$ parameter of Table 6.3. Finally, the tractive force is also bound by a maximal power of $p_\mathrm{max} = 250\,\mathrm{kW}$.

The safety condition is based in part on the notion of a safe distance d_min. This distance represents the initial distance d such that, if both vehicles come to a full stop with their respective maximum deceleration rate, the final distance d is zero. This distance d_min is a function of the vehicle speeds, as well as the maximum deceleration rate for the first $a_{1,\mathrm{e}}(\alpha)$ and second $a_{2,\mathrm{e}}(m,\alpha)$ vehicles. It is assumed that the leading vehicle is capable of exploiting a static friction coefficient of $\mu_\mathrm{s} = 0.9$. Moreover, an additional $0.17\,\mathrm{m/s}^2$ of deceleration due to aerodynamic drag is considered for this vehicle. It is assumed that the trailing vehicle is only capable of exploiting a friction coefficient of $\mu_\mathrm{s} = 0.7$, and that it is also constrained by a maximum braking force $F_\mathrm{b,max}$. This force is computed assuming that at maximum payload, the vehicle has a maximum deceleration rate of only $4\,\mathrm{m/s}^2$, which is just enough to comply with the Federal Motor Vehicle Safety Standards [167]. Aerodynamic drag is neglected for the trailing vehicle, which is a conservative assumption. Figure 6.5 shows how both the trailing vehicle mass m and the road slope α influence d_min.

$$d_\mathrm{min}(v_1, v_2, m, \alpha) = \frac{-v_2^2}{2a_{2,\mathrm{e}}(m,\alpha)} + \frac{v_1^2}{2a_{1,\mathrm{e}}(\alpha)}, \tag{6.47}$$

$$a_{1,\mathrm{e}}(\alpha) = -g\alpha - 9, \tag{6.48}$$

$$a_{2,\mathrm{e}}(m,\alpha) = -g\alpha - \min\{F_\mathrm{b,max}m^{-1}, 0.7g\}. \tag{6.49}$$

Safety is enforced through two control barrier functions, $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$. The first ensures that the trailing vehicle stays behind d_min, and the second enforces a maximum

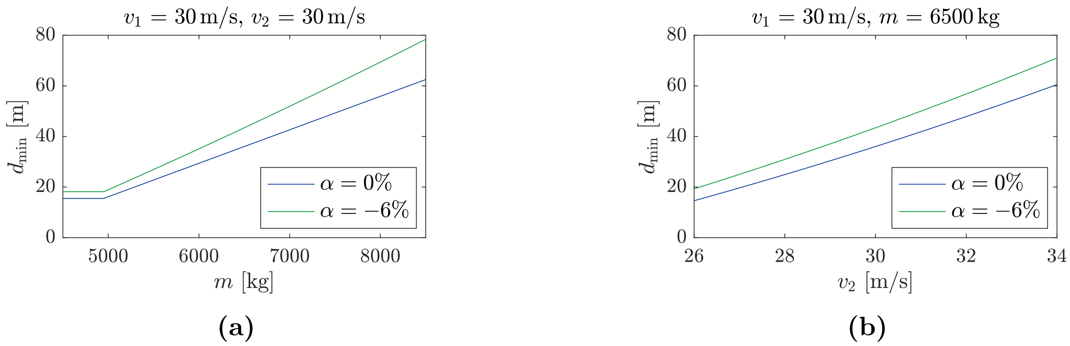


Figure 6.5: Safe distance $d_{\min}$ as a function of vehicle mass, road grade, and vehicle speed.

speed $v_{\max} = 32\,\mathrm{m/s}$.

$$h_1(\mathbf{x}) = d - d_{\min}, \tag{6.50}$$

$$h_2(\mathbf{x}) = v_{\max} - v_2, \tag{6.51}$$

For $h_1(\mathbf{x})$, a system $\eta(\mathbf{x}) = [h_1(\mathbf{x}), \dot{h}_1(\mathbf{x})]^\top$ is defined. Then a controller $K_\alpha = [k_1, k_2]$ is obtained from pole placement. This results in the following condition on the control action

$$u \leq \min_{\boldsymbol{\delta} \in \mathcal{D}} s_1(\mathbf{x}, \boldsymbol{\delta}), \tag{6.52}$$

$$s_1(\mathbf{x}, \boldsymbol{\delta}) := m\big(k_1(d - d_{\min}) + k_2(v_1 - v_2) + \tfrac{1}{2m}\rho v_2^2 a_{\mathrm{f}} c_{\mathrm{d0}} + g(c_{\mathrm{t}} + \alpha) + a_1\big), \tag{6.53}$$

$$\boldsymbol{\delta} = [m, a_{\mathrm{f}} c_{\mathrm{d0}}, c_{\mathrm{t}}, a_1, \alpha]. \tag{6.54}$$

For $h_2(\mathbf{x})$, an extended class $\mathcal{K}_\infty$ function is defined as $-k_3 h_2(\mathbf{x})$ where $k_3 > 0$. Using the condition that $\dot{h}_2(\mathbf{x}, \mathbf{u}) \geq -k_3 h_2(\mathbf{x}),\ \forall \boldsymbol{\delta} \in \mathcal{D}$, a second condition is obtained for the control action

$$u \leq \min_{\boldsymbol{\delta} \in \mathcal{D}} s_2(\mathbf{x}, \boldsymbol{\delta}), \tag{6.55}$$

$$s_2(\mathbf{x}, \boldsymbol{\delta}) := m\big(k_3(v_{\max} - v_2) + \tfrac{1}{2m}\rho v_2^2 a_{\mathrm{f}} c_{\mathrm{d0}} + g(c_{\mathrm{t}} + \alpha)\big), \tag{6.56}$$

$$\boldsymbol{\delta} = [m, a_{\mathrm{f}} c_{\mathrm{d0}}, c_{\mathrm{t}}, \alpha]. \tag{6.57}$$

For both Conditions (6.52) and (6.55), the minimization problem is much simpler than for Example 1; it can be solved exactly.

The base controller $k_{\mathrm{n}}(\mathbf{x})$ is a combination of feedforward and feedback signals, where

$$k_{\mathrm{n}}(\mathbf{x}) = u_{\mathrm{ff}} + u_{\mathrm{fb}}(\mathbf{x}), \tag{6.58}$$

$$u_{\mathrm{ff}} = -\tfrac{1}{2}\rho v_{\mathrm{nom}}^2 a_{\mathrm{f}} c_{\mathrm{d0}}, \tag{6.59}$$

$$u_{\mathrm{fb}}(\mathbf{x}) = k_{\mathrm{p}}(d - d_{\mathrm{min}}). \tag{6.60}$$

The speed $v_{\mathrm{nom}} = 30\,\mathrm{m/s}$ is an approximate nominal speed for this problem. k_{p} is a simple proportional gain.

As with the previous example, online parameter identification is performed according to the method presented in Section 6.3.3, where

$$y_{[n]} = -\tfrac{1}{2m}\rho(v_{2[n]})^2 a_{\mathrm{f}} c_{\mathrm{d}} - g c_{\mathrm{t}} + \tfrac{1}{m} u_{[n]}, \tag{6.61}$$

$$z_{[n]} = \dot{v}_{2[n]} + g\alpha_{[n]} + w_{[n]}, \tag{6.62}$$

$$w_{[n]} \sim \mathcal{N}(0, 0.05^2), \tag{6.63}$$

$$\boldsymbol{\theta} = [m, a_{\mathrm{f}} c_{\mathrm{d}}, c_{\mathrm{t}}]^\top. \tag{6.64}$$

Again, noise is injected into the measurements to obtain a more realistic system identification problem.

6.5.2 Results and discussion

For this problem, the road profile is taken from a portion of the Fleet DNA Long-Haul Representative drive cycle from the National Renewable Energy Laboratory [168]. This defines the road slope to be used during simulation; it has a minimum grade of -6%. Then, the $v_1(t)$ speed profile is obtained from simulating a vehicle where a PID controller attempts to track a $v_1 = 30\,\mathrm{m/s}$ velocity. Finally, the simulation for the trailing vehicle—the problem of interest—can begin.

For this problem, only the robust formulation of the control barrier functions $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ are implemented. The simulation begins with $v_2 = 29.5\,\mathrm{m/s}$ and $d = 100\,\mathrm{m}$. As per Table 5.1, the initial range for the mass parameter is 4500–8500 kg. This results in a conservative behavior, where $d_{\mathrm{min}} \approx 80\,\mathrm{m}$ for the first section of the simulation, see Figure 6.6. Using 25 s of recorded data, the vehicle parameters are estimated online; the results are shown in Table 6.4. The resulting variance on the $a_{\mathrm{f}} c_{\mathrm{d}}$ and c_{t} parameters are too high to extract useful information. This is caused by the noise level being too high for the parameters

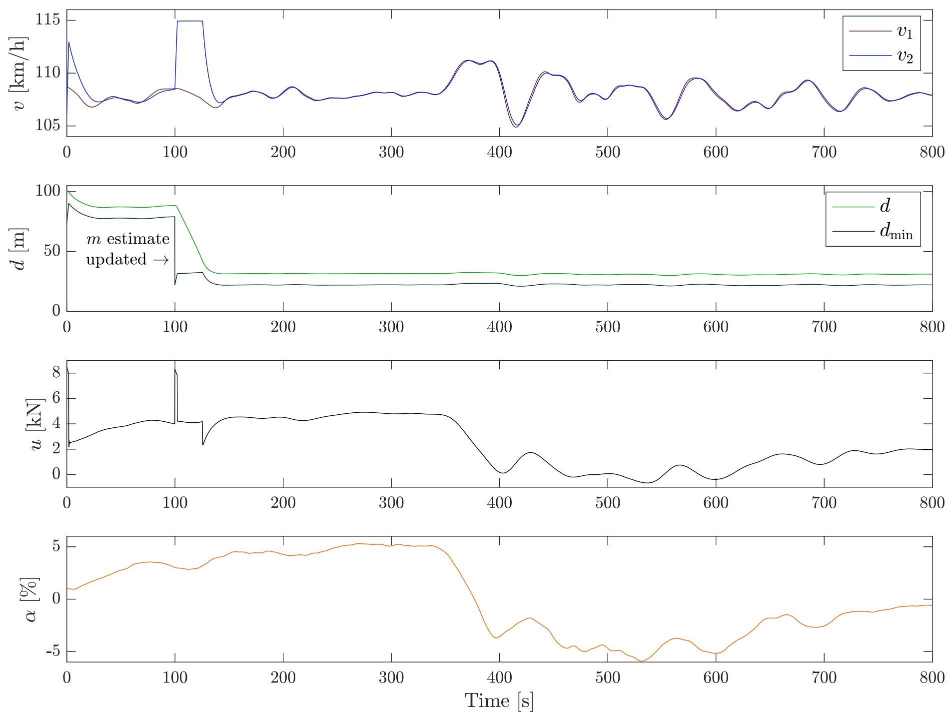


Figure 6.6: Simulation results for Example 2.

Table 6.4: Parameter estimation for Example 2.

Parameter	Actual	Estimation μ	Estimation σ	Unit
m	5000	5026	57	kg
$a_{\mathrm{f}}c_{\mathrm{d}}$	≈ 3.83	3.92	0.95	m^2
c_{t}	0.007	0.006	0.011	-

to be identified properly. However, the mass m can be updated to $\mu \pm 3\sigma$, which effectively reduces the range of the parameter. In the simulation, this uncertainty bound is updated at time $t = 100\,\mathrm{s}$, which results in a large increase in system performance.

In the simulation, the range for the two disturbances a_1 and α are not updated. Likewise, the functions $a_{1,\mathrm{e}}(\alpha)$ and $a_{2,\mathrm{e}}(m, \alpha)$ required to compute d_{min} are not updated either. Figure 6.7 shows a histogram of the recorded a_1 for the whole duration of the simulation; it also shows a normal distribution fitted on the data. Given that the d_{min} and $h_1(\mathbf{x})$ safety conditions are defined based on an extreme event—an emergency braking scenario—and that the recorded data does not include such a scenario, it would be impossible to safely update the range of a_1 from online learning. Similarly, updating the functions $a_{1,\mathrm{e}}(\alpha)$ and $a_{2,\mathrm{e}}(m, \alpha)$ would require the two vehicles to perform one or several brake tests. Finally, the range for

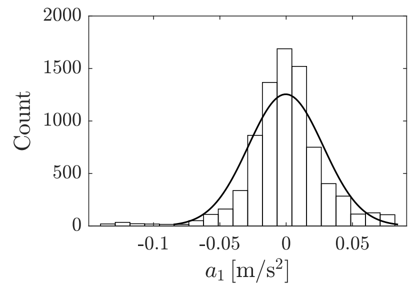


Figure 6.7: Histogram of the a_1 disturbance during the simulation of Figure 6.6.

the road slope α is not updated either. This is because even if the vehicle can record the previous values of α, this gives very little information with regard to the future values. In this problem, it is insufficient to react to the current and previous levels of the disturbances a_1 and α, as a sudden increase in disturbance could cause a sudden increase in $d_{\min}$, for which the system would be unable to react to, thereby causing the system to suddenly leave the safe set. For the road slope α, one could think of a map-based solution where, given its current position on a predefined trajectory, the system could predict the expected disturbance range for the next 30 s or so, which leaves enough time to react to.

6.6 Discussion

The first point in Principle 1 is supported by the results of Example 1. The CBF based on the nominal model is insufficient to keep Vehicle #1 within state constraints. The results in [158] suggest that a CBF based on a learned nominal model could keep a uncertain system safe, as long as the learned model closely matches the actual system behavior. While this may be the case for some scenarios, this method yields no formal guarantees. In practice, CBFs include a certain amount of conservatism, which depends on the chosen α function. In effect, this function influences the rate at which the system is allowed to evolve toward the set boundary. Therefore, a more conservative CBF may compensate for modeling error. Moreover, as exemplified by Figure 6.1c, not all model perturbation results in an unsafe situation. Nevertheless, it remains that a nominal model cannot provide safety guarantees, even if it is learned from data.

The second point in Principle 1 is trivially supported by the first point: if a safety condition based on a nominal model can lead the system outside state constraints, then too small an uncertainty bound can certainly do so.

But perhaps a more interesting discussion concerns learned safety conditions based on non-parametric models. Such models are interesting because the shape of the uncertainty need not be specified in advance, which can lead to a less conservative uncertainty description. Researchers in [169] use data to construct a point-wise uncertainty set that bounds the dynamics, which they use in a robust-CBF setting. Several studies use a Gaussian process (GP) [15, 87] as a non-parametric model to both learn (or correct) a nominal model of the dynamics as well as bound its uncertainty [100, 161, 170–172]. In effect, GP models provide a probability distribution as state prediction. This distribution can be used to bound the uncertainty on the dynamics [173]. Despite being a non-parametric model, the distribution obtained from a GP is still dependent on its hyper-parameters, namely the type of kernel function and its chosen parameters. Thus, such hyper-parameters are often optimized through maximum likelihood estimation, see [100, 161], much like the parameter estimation presented in this chapter.

Safety conditions based on non-parametric models require special care when first initialized, given that no data is yet collected, thus no safety condition can be computed. One approach is to initially use a nominal model, but restrict the initial safe set to a smaller subset than the entire state constraints [161, 172]. In [161] for instance, the initial safe set is computed though HJR with assumed conservative bounds on the disturbance. As data gets collected, it is then possible to use the robust safety condition and to iteratively increase the size of the safe set. Therefore, approaches based on non-parametric models can still meet the condition of Point 2 in Principle 1, but additional assumptions are required for the method's initialization.

The third point in Principle 1 is supported by both Examples 1 and 2. In Example 1, the system could become unsafe if the actual dynamics suddenly shift outside the set covered by the estimated uncertainty. In Example 2, the system could become unsafe due to an adversarial disturbance not covered by the estimated uncertainty. These are particularly difficult situations to account for, and they share the same root cause: the data previously collected does not sufficiently describe the future behavior of the system. In Sections 6.4 and 6.5, example solutions were proposed for this type of problem. For uncertainty sources that are assumed to remain mostly constant for periods of time, but that are still expected to vary—e.g., the dynamics in Example 1—the controller should predict when a significant change may occur, and reset the corresponding uncertainty bounds to more conservative values. For uncertainty sources that represent rare and extreme events—e.g., the a_1 disturbance in Example 2—the controller should never attempt to learn the corresponding uncertainty parameters.

The change prediction and adaptation mechanism proposed for Example 1 is simple: the controller resets the uncertainty bounds every time the vehicle stops. Moreover, the controller does not need to continually update the uncertainty parameters as the dynamics are assumed to remain constant when the vehicle operates. For some systems, however, the dynamics could evolve continually, or the sudden change in dynamics could be impossible to predict in advance. These scenarios require a more sophisticated adaptation law if machine learning is to be used to define the safety condition. In [161], researchers propose to constantly evaluate the validity of the learned uncertainty bounds. When the learned model is deemed invalid, a pre-computed safety action is applied, the uncertainty bounds are reevaluated, and finally the updated safe controller comes back online and resumes normal operation. In [160], the controller constantly identifies the dynamics with set membership identification, and updates the uncertainty parameters accordingly. However, using one of these adaptive methods to guarantee safety in the face of a possibly changing dynamics still requires to make additional assumptions, notably on the magnitude and the rate of these possible changes. Certainly not every scenario is recoverable.

Conclusion

This research concerns the control of uncertain dynamical systems using machine learning. It concentrates on the control of electric and autonomous vehicles. The research questions motivating the thesis are

1. Can machine learning control accelerate the development of electric drivetrains?
2. Can machine learning control preserve safety guarantees for vehicular control?

This conclusion summarizes the results and contributions presented in the thesis, suggests answers to the research questions, and suggests future work based on the remaining ambiguity.

Results and contributions

In Chapter 1, a novel transmission design is presented. The transmission offers three gear ratios and a possibility of uninterrupted gearshifts, while containing only two friction brakes. The use of locking mechanisms also allows to deactivate the braking elements between the gearshifts and save energy. A reduced-scale prototype of this transmission was designed and built. The prototype was used as an experimental test bench for the research of Chapter 3.

Chapter 2 provides theorems for the fundamental limitations to uninterrupted gearshifts that are associated with motor saturation. These theorems can be used during the conceptual design phase of a new electric drivetrain, which can help engineers to choose between transmission architectures. The theorems also help guiding the development of gearshift controllers by restricting the gearshift trajectories. This is especially useful for guiding a learning algorithm, as the algorithm does not spend time exploring known bad trajectories.

Chapter 3 presents a method to learn a gearshift controller from gearshift trials on a transmission test bench. The typical approach in industry uses a statistical method called design of experiments, which requires thousands of gearshift trials. The proposed learning algorithm converged to a tuned controller in about four trials. The method was able to improve gearshift performance by tuning the parameters of both a feedback controller and feedforward signals. The method is implemented using automatic gradients, which means that it can easily be adapted to different controllers. In an engineering context, this can foster the exploration of many more controller types and parametrization, thereby accelerating the development of a multi-speed transmission for an electric vehicle.

Chapter 4 presents models for the longitudinal and lateral dynamics of road vehicles, along with simplifying assumptions that make the models better suited for controller design. These models are used in the example problems of Chapters 5 and 6.

Chapter 5 presents a new algorithm to tune a linear controller from a learned model of arbitrary type, while preserving guarantees of robust stability. This is especially interesting given that modern machine learning typically employs nonlinear models; this allows to fully harness the predictive capabilities of machine learning. The proposed approach was used to tune a controller for autonomous vehicle maneuvers, namely a lane change with concurrent vehicle acceleration. The approach improved the trajectory tracking performance compared to $\mathcal{H}_\infty$ loop shaping. The controller also remained stable when the vehicle dynamics was altered, thereby demonstrating the robustness of the approach.

Chapter 6 introduces the safe uncertainty-learning principle. This principle allows to quickly evaluate whether a learning algorithm preserves safety guarantees of state constraint satisfaction. To emphasize the importance of the principle, two vehicular control examples were presented, where a robust control barrier function was used to maintain the uncertain dynamical systems within state constraints. It was shown that when the proposed principle is not respected, the safety guarantees are lost.

Research findings

Learning for control can indeed accelerate the development of electric drivetrains. In this work, this was directly demonstrated by learning a gearshift controller, but it should also hold true for other drivetrain-related control tasks such as anti-jerk controllers [174–176]. For such applications, it is recommended to use efficient models to learn the uncertain system dynamics. In effect, the need to perform physical measurements limits the data collection rate. In this work, Gaussian processes were shown to be good candidates. Also, if reinforcement learning is to be used, it is recommended to choose a model-based approach. This allows to reduce the number of interactions with the physical world because the controller is iterated in a simulated environment. The model-based reinforcement learning approach used in this work was able to learn a gearshift controller with very little information.

Depending on the application, it may not be productive to randomly initialize controller parameters. In this work, the gearshift controller was initialized such that the first gearshift trial actually completed. This way, the first instance of data collection was relevant. This would likely not have been the case for a randomly initialized controller. In effect, the non-linearities present in multi-speed transmissions restrict the set of dynamical trajectories the system should follow. It is important to properly define the set of interesting trajectories so that the learning algorithm does not spend time exploring bad ones. Sometimes, this can be achieved by formulating theorems such as those of Chapter 2.

It can also be faster to learn the difference between a nominal model and the real system dynamics, instead of learning the entire system dynamics. For the control of engineered system, there should always be a nominal model. In this work, Gaussian processes were shown to be a good model type to do so. Because they are a non-parametric model, they do not require the assumption of a specific form of uncertainty, which is convenient to capture the unknown discrepancy between a nominal model and the actual system dynamics.

The use of automatic differentiation is also recommended. It allows to quickly iterate on the formulation of the control law. However, automatic gradients can be cumbersome to program, and great care should be taken to ensure the validity of the code. It is recommended to also derive some gradients analytically and compare the results with the automatic method.

It is possible to preserve safety guarantees while learning to control uncertain dynamics. To maintain robust stability, it was shown sufficient to combine a robust control framework with a gradient-based optimization method, as presented in Chapter 5. To generalize on the proposed method, the idea is to utilize known theorems of control engineering—in our case, the small-gain theorem—and to adapt learning methods around them. Given the richness of the control literature, this may be easier than developing new theorems for existing learning algorithms.

To maintain guarantees of state constraint satisfaction, several methods can be used: Hamilton-Jacobi reachability analysis, model predictive control, and control barrier functions. When a learning component is incorporated into these methods in an attempt to reduce modeling uncertainty and improve performance, the safe uncertainty-learning principle of Chapter 6 suggests how to safely vary the uncertainty bounds. It is important to ensure the validity of the uncertainty bounds.

Both the considerations of Chapters 5 and 6 should be applied to vehicular control. In effect, the dynamics of vehicles may be uncertain at times, or may be expected to vary. Thus, machine learning can be employed to reduce this uncertainty, but not at the expense of safety guarantees.

Future work

It would be interesting to apply the learning method of Chapter 3 to a full-scale transmission prototype with actual transmission actuators. A more complete vehicle model could be simulated as well, one with a nonlinear tire model for instance. Perhaps, additional considerations could be learned from this experiment, further refining the method.

The third point in the safe uncertainty-learning principle stipulates that when tightening uncertainty bounds, challenging adversarial events should be distinctly accounted for. The examples of Chapter 6 suggest two ways to do so. When the event is a sudden change in dynamics that can be predicted in advance, the controller should reset the uncertainty bounds prior to the change in dynamics. When the event is based on rare and extreme disturbances, the controller should never try to model the disturbances based on collected data. When the dynamics are slowly varying however, it should be possible to vary the uncertainty bounds accordingly and preserve safety guarantees, but there is yet no method to judge when this approach is acceptable and when it is not. This should be the object of further studies.

Finally, it would be interesting to investigate the safe uncertainty-learning principle in the context of vehicle trajectory planning. A notoriously difficult problem in this area is to model the statistical distribution of human agent trajectories [177]. Perhaps the need to abruptly adjust the desired vehicle trajectory due to an erratic driver behavior could be treated as a vehicular control problem, and be translated into in additional uncertainty bounds in the system dynamics.

Appendix A

Introduction to machine learning

The output of a machine learning algorithm is a function $f(\mathbf{x})$ that was learned from data [178]. The three main categories of learning algorithms are: supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning, the algorithm is given a dataset $\mathcal{D}$ composed of outputs $\mathbf{y}_{[i]}$ and corresponding inputs $\mathbf{x}_{[i]}$. The algorithm learns a function f that approximate $\mathbf{y} = f(\mathbf{x})$ as closely as possible. The goal of machine learning is to generalize $f(\mathbf{x})$, which means $f(\mathbf{x})$ would return an appropriate output $\mathbf{y}$ when given an unseen input $\mathbf{x}$—an input not contained in $\mathcal{D}$. Thus, the algorithm must avoid overfitting to the data points in $\mathcal{D}$. A common approach to verify whether the algorithm overfitted $f(\mathbf{x})$ is to separate $\mathcal{D}$ into a training set and a test set. The function $f(\mathbf{x})$ is learned from the data in the training set, and it is validated with the data in the test set. If $f(\mathbf{x})$ accurately predicts the outputs $\mathbf{y}$ of the test set, then it is not overfitted. This goal of approximating a model that generalizes well is what sets machine learning apart from statistics. In statistics, the objective is to describe a population from sampled data. In machine learning, the predicted outputs $\mathbf{y} = f(\mathbf{x})$ constitute the information of interest, whereas in statistics, it is the properties of the function $f(\mathbf{x})$ that are of interest.

Supervised learning problems are categorized with respect to the type of output they approximate. If the output $\mathbf{y}$ is a continuous variable, the learning problem is a regression problem. If $\mathbf{y}$ is instead a discrete variable—representing distinct categories for instance—then the learning problem is a classification problem.

It is commonplace to distinguish between parametric and non-parametric models. Learning a parametric model consists of tuning its various parameters. A neural network [14] is an example of a parametric model: the parameters are the weights connecting the nodes. As the name suggests, a non-parametric model does not contain model parameters. Gaussian processes [15] are a good example. Every time an output $\mathbf{y}$ is to be estimated from a test input $\mathbf{x}_*$, the input $\mathbf{x}_*$ has to be compared to all the other inputs $\mathbf{x}$ contained in $\mathcal{D}$. Therefore, non-parametric models do not need to be "trained" like parametric models do. Both parametric and non-parametric models contain hyper-parameters, however. These are parameters that are not learned during the training process (if applicable), but that still influence the prediction $\mathbf{y}$. Examples of hyper-parameters for neural networks are the number of nodes in the hidden layers, and the choice of activation function. Examples of hyper-parameters for Gaussian processes are the choice of kernel function and its parameters. For Gaussian processes, the hyper-parameters are typically tuned to maximize the likelihood of the observed data—essentially a form of parameter tuning—thereby blurring the line between parametric and non-parametric approaches.

In unsupervised learning, the algorithm is not given target outputs $\mathbf{y}$; it is only given a set of inputs $\mathbf{x}$. The goal of unsupervised learning can be to find patterns in the dataset, or to reduce the dimensionality of the input space by finding a useful projection onto a lower-dimensional space. Unsupervised learning is not used in this thesis.

Reinforcement learning algorithms use a particular problem setup. They assume that an agent interacts with an environment. The agent makes observations of the environment, from which it must takes appropriate actions. The agent is also fed a reward signal as an indicator of its performance. The goal of reinforcement learning is to learn a function that maps observations to actions, with the objective of maximizing a cumulative reward [179].

Reinforcement learning and control engineering are very similar. In both fields, the goal is to obtain an agent (a controller) that behaves optimally in a given environment by maximizing a reward (minimizing a cost function). Notably, central to both optimal control and reinforcement learning are Bellman's Principle of Optimality [180] and dynamic programming. Perhaps the most important distinction is that in reinforcement learning, the dynamics of the environment are unknown [95].

Methods in reinforcement learning are typically classified in two categories: model-based and model-free approaches. In the former, the agent progressively builds an internal model of the environment (learning), then uses this model to design a control policy (planning). In the latter, the agent directly learns a control policy from interacting with the environment. Model-free approaches tend to require many more interactions with the environment [85].

The field of artificial intelligence concerns the design of rational agents that behaves optimally in unknown environments [181]. Again, this is very similar to reinforcement learning and control theory. Artificial intelligence is more general, however. For instance, not every intelligent algorithm involves learning. Also, artificial intelligence can handle a more diverse set of problems than control theory. In effect, control theory is typically applied to dynamical systems expressed with differential equations, whereas artificial intelligence can be applied to other kinds of mathematical models. It is not wrong, but not particularly useful to see machine learning and control theory as a subset of artificial intelligence. For example, plenty of learning problems—especially supervised—do not benefit from this notion of an autonomous agent. Also, control theory is typically taught without the framework of artificial intelligence. However, the overlaps between the different fields are notable, and should be exploited. In fact, this is one of the objectives in this thesis.

Appendix B

Equations of motion for a planetary gearset

This appendix shows how to obtain the dynamical model for a planetary gearset that was used in this thesis, namely the three equations of motion (1.15)–(1.17), and the kinematic constraint (1.18). The process begins with a complete system model that considers the four components of a planetary gearset: the ring gear, the planet carrier, the planet gears, and the sun gear. This model is obtained from the free-body diagrams of Figure B.1. Then, approximations are made in order to avoid having to consider the motion of the planet gears. This simplification is convenient for transmission design and gearshift analysis, since the planet gears never have external torques directly applied to them—it is the planet carrier that does.

For simplicity, this process is done only considering one planet gear. Also, in Figure B.1, the normal component of the meshing forces are omitted from the free-body diagram, as they are irrelevant to this analysis.

B.1 Complete system model

For the gearset to mesh and assemble properly, the radius of the various components must match. This results in two geometric constraints

$$r_{\mathrm{r}} = r_{\mathrm{p}} + r_{\mathrm{c}}, \tag{B.1}$$

$$r_{\mathrm{c}} = r_{\mathrm{s}} + r_{\mathrm{p}}. \tag{B.2}$$

In addition, there exist three kinematic constraints. First, the gear mesh does not allow for slippage at the contact point between meshing gears. There are two contact points to

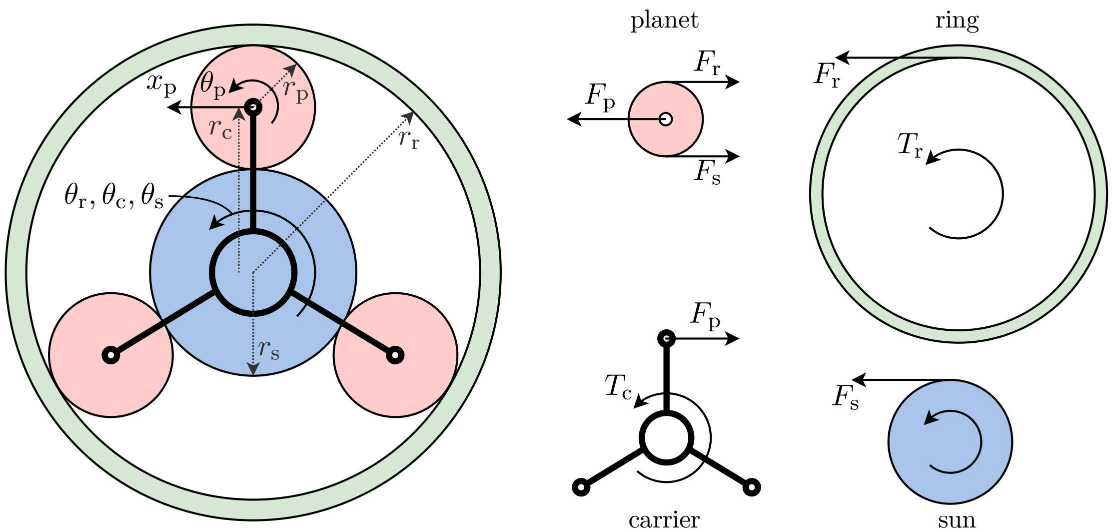


Figure B.1: Free-body diagrams for the four elements of a planetary gearset.

consider: the contact between the planet gear and the ring gear, and the contact between the planet gear and the sun gear. Also, the center of the planet gear is fixed with respect to the carrier body. The resulting three constraints are

$$r_r\dot{\theta}_r = r_c\dot{\theta}_c + r_p\dot{\theta}_p, \tag{B.3}$$

$$r_s\dot{\theta}_s = r_c\dot{\theta}_c - r_p\dot{\theta}_p, \tag{B.4}$$

$$\dot{x}_p = r_c\dot{\theta}_c. \tag{B.5}$$

From the free-body diagrams of Figure B.1, five equations of motion can be obtained. They are

$$I_r\ddot{\theta}_r = T_r + r_rF_r, \tag{B.6}$$

$$I_c\ddot{\theta}_c = T_c - r_cF_p, \tag{B.7}$$

$$I_s\ddot{\theta}_s = T_s + r_sF_s, \tag{B.8}$$

$$m_p\ddot{x}_p = F_p - F_r - F_s, \tag{B.9}$$

$$I_p\ddot{\theta}_p = -r_pF_r + r_pF_s. \tag{B.10}$$

The Equations (B.1)–(B.10) are sufficient to describe the complete motion of the planetary gearset. In the next section, these 10 equations are reduced to a more practical set of four equations.

B.2 Reduced system model

To begin, the Equations (B.1) to (B.4) are reduced to obtain the kinematic constraint (1.18) by eliminating r_p, r_c, and $\dot{\theta}_\mathrm{p}$. This is a trivial process of variable elimination. For instance, Equations (B.1) and (B.2) can be combined to obtain and expression for r_c, where

$$r_\mathrm{c} = \frac{r_\mathrm{r} + r_\mathrm{s}}{2}. \tag{B.11}$$

Also, $r_\mathrm{p}\dot{\theta}_\mathrm{p}$ can be eliminated by combining Equations (B.3) and (B.4), which gives

$$r_\mathrm{r}\dot{\theta}_\mathrm{r} + r_\mathrm{s}\dot{\theta}_\mathrm{s} = 2r_\mathrm{c}\dot{\theta}_\mathrm{c}. \tag{B.12}$$

Combining (B.11) with (B.12) yields the kinematic constraint (1.18).

$$\boxed{r_\mathrm{r}\dot{\theta}_\mathrm{r} + r_\mathrm{s}\dot{\theta}_\mathrm{s} = (r_\mathrm{r} + r_\mathrm{s})\dot{\theta}_\mathrm{c}.} \tag{1.18}$$

Note that no approximation was required to obtain this equation.

Next, Equations (B.6)–(B.10) are reduced to obtain the final equations of motion (1.15)–(1.17), and eliminate the need for $\ddot{x}_\mathrm{p}$ and $\ddot{\theta}_\mathrm{p}$. The first step is to isolate F_p in Equation (B.9), and substitute the expression in Equation (B.7) to obtain

$$I_\mathrm{c}\ddot{\theta}_\mathrm{c} = T_\mathrm{c} - r_\mathrm{c}(m_\mathrm{p}\ddot{x}_\mathrm{p} + F_\mathrm{r} + F_\mathrm{s}). \tag{B.13}$$

Taking the time derivative of Equation (B.5), one gets $\ddot{x}_\mathrm{p} = r_\mathrm{c}\ddot{\theta}_\mathrm{c}$, which can be substituted into Equation (B.13) to obtain

$$(I_\mathrm{c} + r_\mathrm{c}^2 m_\mathrm{p})\ddot{\theta}_\mathrm{c} = T_\mathrm{c} - r_\mathrm{c}(F_\mathrm{r} + F_\mathrm{s}). \tag{B.14}$$

By introducing the equivalent inertia $I_\mathrm{c,eq} = I_\mathrm{c} + r_\mathrm{c}^2 m_\mathrm{p}$, Equation (B.14) can be further simplified into

$$I_\mathrm{c,eq}\ddot{\theta}_\mathrm{c} = T_\mathrm{c} - r_\mathrm{c}(F_\mathrm{r} + F_\mathrm{s}). \tag{B.15}$$

The Equations (B.7) and (B.9) were effectively reduced into (B.15), and the need for $\ddot{x}_\mathrm{p}$ was eliminated. Again, no approximation was required thus far.

The last step is to eliminate $\ddot{\theta}_\mathrm{p}$ from the equations of motion. This can be done by approximating $I_\mathrm{p} = 0$. Doing so, Equation (B.10) reduces to $F_\mathrm{r} = F_\mathrm{s}$. A new variable F

can be introduced to designate both F_r and F_s, as they are now approximated as equal. Using Equations (B.6), (B.15), (B.8), the expression for r_c in Equation (B.11), as well as the approximation that $F = F_\mathrm{r} = F_\mathrm{s}$, the final set of equations of motion is obtained, where

$$\boxed{\begin{aligned} I_\mathrm{r}\ddot{\theta}_\mathrm{r} &= T_\mathrm{r} + r_\mathrm{r}F, && (1.15)\\ I_\mathrm{c,eq}\ddot{\theta}_\mathrm{c} &= T_\mathrm{c} - r_\mathrm{r}F - r_\mathrm{s}F, && (1.16)\\ I_\mathrm{s}\ddot{\theta}_\mathrm{s} &= T_\mathrm{s} + r_\mathrm{s}F. && (1.17)\end{aligned}}$$

Appendix C

Analytical gradients for controller learning

This appendix provides analytical expressions for the computation of

$$\frac{\mathrm{d}J^{\pi}(\boldsymbol{\psi})}{\mathrm{d}\boldsymbol{\psi}} = \sum_{t=0}^{T} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\psi}} \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})], \tag{3.39}$$

where $\boldsymbol{\psi}$ is simply K_{c}. This work uses the numerator layout when displaying the Jacobian of a function. While a few of the results in this appendix can be found in the literature [182], most are new results that pertain to our specific implementation of PILCO.

C.1 Gradients for the kernel function

The square exponential kernel can be differentiated as follows:

$$\frac{\partial k(\mathbf{z}_*, \mathbf{z}_{[1]})}{\partial \mathbf{z}_*} = -(\mathbf{z}_* - \mathbf{z}_{[1]})^{\top} \Lambda^{-1} k(\mathbf{z}_*, \mathbf{z}_{[1]}), \tag{C.1}$$

$$\frac{\partial^2 k(\mathbf{z}_*, \mathbf{z}_{[1]})}{\partial \mathbf{z}_*^2} = -\Lambda^{-1}(I - (\mathbf{z}_* - \mathbf{z}_{[1]})(\mathbf{z}_* - \mathbf{z}_{[1]})^{\top} \Lambda^{-1}) k(\mathbf{z}_*, \mathbf{z}_{[1]}). \tag{C.2}$$

This allows to obtain derivatives of the mean and variance functions with respect to a deterministic test point $\mathbf{z}_*$.

$$\frac{\partial \mu_d(\mathbf{z}_*)}{\partial \mathbf{z}_*} = -(K_{\mathbf{z}_*\mathbf{z}} \odot \mathbf{y}_d^{\top} (K_{\mathbf{z}\mathbf{z}} + \sigma_\epsilon^2 I)^{-1}) \tilde{Z}_*^{\top} \Lambda^{-1}, \tag{3.26}$$

$$\frac{\partial \Sigma_d(\mathbf{z}_*)}{\partial \mathbf{z}_*} = 2(K_{\mathbf{z}_*\mathbf{z}} \odot K_{\mathbf{z}_*\mathbf{z}} (K_{\mathbf{z}\mathbf{z}} + \sigma_\epsilon^2 I)^{-1}) \tilde{Z}_*^{\top} \Lambda^{-1}, \tag{C.3}$$

where $\tilde{Z}_* = [\mathbf{z}_* - \mathbf{z}_{[1]}, \ldots, \mathbf{z}_* - \mathbf{z}_{[n]}]$, and $\odot$ represents an element-wise product. The second derivative of the mean function can also be obtained. The index notation is used since a third order tensor needs to be introduced.

$$\frac{\partial^2 \mu_d(\mathbf{z}_*)}{\partial \mathbf{z}_*^2} = (\hat{Z}_*)_{ijk} (K_{\mathbf{z}_*\mathbf{z}}^\top \odot (K_{\mathbf{zz}} + \sigma_\epsilon^2 I)^{-1} \mathbf{y}_d)_k, \tag{C.4}$$

$$\text{where} \quad (\hat{Z}_*)_{ijk} = -\Lambda^{-1}(I - (\mathbf{z}_* - \mathbf{z}_k)(\mathbf{z}_* - \mathbf{z}_k)^\top \Lambda^{-1}). \tag{C.5}$$

C.2 Gradients for the state distribution

The equations for the state transition are reproduced below:

$$\boldsymbol{\mu}_{[t]}^{\mathbf{x}} = (A_\mathrm{d} - B_\mathrm{d} K_\mathrm{c}) \boldsymbol{\mu}_{[t-1]}^{\mathbf{x}} + \mu_\mathrm{f}(\boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}) + B_\mathrm{d}(\bar{\mathbf{u}}_{[t-1]} + K_\mathrm{c} \bar{\mathbf{x}}_{[t-1]}) + \mathbf{t}_0, \tag{3.31}$$

$$\Sigma_{[t]}^{\mathbf{x}} = (A_\mathrm{d} - B_\mathrm{d} K_\mathrm{c}) \Sigma_{[t-1]}^{\mathbf{x}} (A_\mathrm{d} - B_\mathrm{d} K_\mathrm{c})^\top + \Sigma_\mathrm{f}(\mathbf{z}_{[t-1]}), \tag{3.32}$$

where $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{u} \in \mathbb{R}^F$.

The first gradient for the state distribution of Equations (3.41)–(3.42) is

$$\frac{\partial \boldsymbol{\mu}_{[t]}^{\mathbf{x}}}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{x}}} = A_\mathrm{d} - B_\mathrm{d} K_\mathrm{c} + \frac{\partial \mu_\mathrm{f}(\boldsymbol{\mu}_{[t-1]}^{\mathbf{z}})}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}} \frac{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{x}}}, \tag{C.6}$$

where

$$\frac{\partial \mu_\mathrm{f}(\boldsymbol{\mu}_{[t-1]}^{\mathbf{z}})}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}} = \begin{bmatrix} \dfrac{\partial \mu_1(\boldsymbol{\mu}_{[t-1]}^{\mathbf{z}})}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}} \\ \vdots \\ \dfrac{\partial \mu_D(\boldsymbol{\mu}_{[t-1]}^{\mathbf{z}})}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}} \end{bmatrix}, \tag{C.7}$$

$$\frac{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{z}}}{\partial \boldsymbol{\mu}_{[t-1]}^{\mathbf{x}}} = \begin{bmatrix} I_{D\times D} \\ -K_\mathrm{c} \end{bmatrix}. \tag{C.8}$$

The next term in Equation (3.41) is trivial:

$$\frac{\partial \boldsymbol{\mu}_{[t]}^{\mathbf{x}}}{\partial \Sigma_{[t-1]}^{\mathbf{x}}} = 0. \tag{C.9}$$

Next, there is

$$\frac{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\psi}} = (B_{\mathrm{d}})_{ij}(\bar{\mathbf{x}}_{[t-1]} - \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]})^{\top}_{k} + \frac{\partial \mu_{\mathrm{f}}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}} \frac{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\psi}}\,, \tag{C.10}$$

$$\frac{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\psi}} = \begin{bmatrix} \mathbf{0}_{D\times D\times F} \\ (I_{F\times F})_{ij}(\bar{\mathbf{x}}_{[t-1]} - \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]})^{\top}_{k} \end{bmatrix}. \tag{C.11}$$

Next, there is

$$\frac{\partial \Sigma^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}} = \frac{\partial \Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}\,, \tag{C.12}$$

$$\left(\frac{\partial \Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}\right)_{iij} = \left(\frac{\partial \Sigma_{i}(\mathbf{z}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}\right)_{1j1}\,, \tag{C.13}$$

$$\frac{\partial \Sigma_{d}(\mathbf{z}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}} = \frac{\partial}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}} \left(\Sigma_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}) + \underbrace{\left(\frac{\partial \mu_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}} \Sigma^{\mathbf{z}}_{[t-1]} \frac{\partial \mu_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}^{\top}\right)}_{\square}\right) \frac{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t-1]}}\,, \tag{C.14}$$

$$\frac{\partial(\square)}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}} = 2\,\frac{\partial \mu_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}} \Sigma^{\mathbf{z}}_{[t-1]} \frac{\partial^{2} \mu_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial {\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}^{2}}\,. \tag{C.15}$$

Next, there is

$$\left(\frac{\partial \Sigma^{\mathbf{x}}_{[t]}}{\partial \Sigma^{\mathbf{x}}_{[t-1]}}\right)_{ijkl} = (A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})_{ik}(A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})_{jl} + \frac{\partial \Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]})}{\partial \Sigma^{\mathbf{x}}_{[t-1]}}\,, \tag{C.16}$$

$$\left(\frac{\partial \Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]})}{\partial \Sigma^{\mathbf{x}}_{[t-1]}}\right)_{iijk} = \left(\frac{\partial \Sigma_{i}(\mathbf{z}_{[t-1]})}{\partial \Sigma^{\mathbf{x}}_{[t-1]}}\right)_{1jk1}, \tag{C.17}$$

$$\frac{\partial \Sigma_{d}(\mathbf{z}_{[t-1]})}{\partial \Sigma^{\mathbf{x}}_{[t-1]}} = \frac{\partial(\Box)}{\partial \Sigma^{\mathbf{z}}_{[t-1]}}\,\frac{\partial \Sigma^{\mathbf{z}}_{[t-1]}}{\partial \Sigma^{\mathbf{x}}_{[t-1]}}\,, \tag{C.18}$$

$$\frac{\partial(\Box)}{\partial \Sigma^{\mathbf{z}}_{[t-1]}} = \left(\frac{\partial \mu_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}\right)^{\top}\left(\frac{\partial \mu_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}\right), \tag{C.19}$$

$$\frac{\partial \Sigma^{\mathbf{z}}_{[t-1]}}{\partial \Sigma^{\mathbf{x}}_{[t-1]}} = \begin{bmatrix} (I_{D\times D})_{ik}(I_{D\times D})_{jl} & (I_{D\times D})_{ik}(K_{\mathrm{c}})_{jl} \\ (K_{\mathrm{c}})_{ik}(I_{D\times D})_{jl} & (K_{\mathrm{c}})_{ik}(K_{\mathrm{c}})_{jl} \end{bmatrix}. \tag{C.20}$$

Finally, there is

$$\frac{\partial \Sigma^{\mathbf{x}}_{[t]}}{\partial \boldsymbol{\psi}} = \frac{\partial}{\partial \boldsymbol{\psi}}\Big(\underbrace{(A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})\Sigma^{\mathbf{x}}_{[t-1]}(A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})^{\top}}_{\Diamond}\Big) + \frac{\partial \Sigma_{\mathrm{f}}(\mathbf{z}_{[t-1]})}{\partial \boldsymbol{\psi}}\,, \tag{C.21}$$

$$\begin{aligned}\frac{\partial(\Diamond)}{\partial \boldsymbol{\psi}} = &-\Big(\big((I_{D\times D})_{ik}(\Sigma^{\mathbf{x}}_{[t-1]})_{lj}\big)_{imkl}\big(A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}}\big)_{jm} \\ &+ \big((A_{\mathrm{d}} - B_{\mathrm{d}}K_{\mathrm{c}})\Sigma^{\mathbf{x}}_{[t-1]}\big)_{im}\big((I_{D\times D})_{jk}(I_{D\times D})_{il}\big)_{mjkl}\Big)_{ijkl}\Big((B_{\mathrm{d}})_{ik}(I_{D\times D})_{jl}\Big)_{kl},\end{aligned} \tag{C.22}$$

$$\frac{\partial \Sigma_{d}(\mathbf{z}_{[t-1]})}{\partial \boldsymbol{\psi}} = \frac{\partial \Sigma_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\psi}} + \frac{\partial(\Box)}{\partial \boldsymbol{\psi}}\,, \tag{C.23}$$

$$\frac{\partial \Sigma_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\psi}} = \frac{\partial \Sigma_{d}(\boldsymbol{\mu}^{\mathbf{z}}_{[t-1]})}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}\,\frac{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\psi}}\,, \tag{C.24}$$

$$\frac{\partial(\Box)}{\partial \boldsymbol{\psi}} = \frac{\partial(\Box)}{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}\,\frac{\partial \boldsymbol{\mu}^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\psi}} + \frac{\partial(\Box)}{\partial \Sigma^{\mathbf{z}}_{[t-1]}}\,\frac{\partial \Sigma^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\psi}}\,, \tag{C.25}$$

$$\frac{\partial \Sigma^{\mathbf{z}}_{[t-1]}}{\partial \boldsymbol{\psi}} = \begin{bmatrix} \mathbf{0}_{D\times D\times F\times D} & (I_{F\times F})_{jk}(\Sigma^{\mathbf{x}}_{[t-1]})_{il} \\ (I_{F\times F})_{ik}(\Sigma^{\mathbf{x}}_{[t-1]})_{lj} & \triangle \end{bmatrix}, \tag{C.26}$$

$$\triangle = \big((I_{F\times F})_{ik}(\Sigma^{\mathbf{x}}_{[t-1]})_{lj}\big)_{imkl}\big(K_{\mathrm{c}}\big)_{jm} + \big(K_{\mathrm{c}}\Sigma^{\mathbf{x}}_{[t-1]}\big)_{im}\big((I_{F\times F})_{jk}(I_{D\times D})_{il}\big)_{mjkl}. \tag{C.27}$$

C.3 Gradients for the cost function

The derivatives of the expected cost with respect to the state distribution are

$$\frac{\partial \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]}{\partial \boldsymbol{\mu}^{\mathbf{x}}_{[t]}} = \big(1 - \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]\big)(\boldsymbol{\mu}^{\mathbf{x}}_{[t]} - \bar{\mathbf{x}}_{[t]})^\top \tilde{S}, \tag{C.28}$$

$$\frac{\partial \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]}{\partial \Sigma^{\mathbf{x}}_{[t]}} = \tfrac{1}{2}\big(1 - \mathbb{E}_{\mathbf{x}_{[t]}}[c(\mathbf{x}_{[t]})]\big)\Big(\tilde{S} - \big((\boldsymbol{\mu}^{\mathbf{x}}_{[t]} - \bar{\mathbf{x}}_{[t]})(\boldsymbol{\mu}^{\mathbf{x}}_{[t]} - \bar{\mathbf{x}}_{[t]})^\top\big)_{kl}\big(\tilde{S}_{ik}\tilde{S}_{jl}\big)_{lkij}\Big). \tag{C.29}$$

Appendix D

Partial derivatives for the lateral dynamics

This appendix lists the non-zero partial derivative for the nonlinear equations of the lateral dynamics, namely

$$\ddot{x} = \tfrac{1}{m}\left(F_{\mathrm{r}x} + F_{\mathrm{f}x} - c_{\mathrm{f}}\left(\phi^2 - \frac{\phi(\dot{\psi}l_{\mathrm{f}} + \dot{y})}{\dot{x}}\right) - \tfrac{1}{2}\rho\dot{x}^2 a_{\mathrm{f}} c_{\mathrm{d}}\right) + \dot{y}\dot{\psi}, \tag{4.17}$$

$$\ddot{y} = \tfrac{1}{m}\left(F_{\mathrm{f}x}\phi + c_{\mathrm{r}}\left(\frac{\dot{\psi}l_{\mathrm{r}} - \dot{y}}{\dot{x}}\right) + c_{\mathrm{f}}\left(\phi - \frac{\dot{\psi}l_{\mathrm{f}} + \dot{y}}{\dot{x}}\right)\right) - \dot{x}\dot{\psi}, \tag{4.18}$$

$$\ddot{\psi} = \tfrac{1}{I_{\mathrm{z}}}\left(F_{\mathrm{f}x}l_{\mathrm{f}}\phi - c_{\mathrm{r}}l_{\mathrm{r}}\left(\frac{\dot{\psi}l_{\mathrm{r}} - \dot{y}}{\dot{x}}\right) + c_{\mathrm{f}}l_{\mathrm{f}}\left(\phi - \frac{\dot{\psi}l_{\mathrm{f}} + \dot{y}}{\dot{x}}\right)\right), \tag{4.19}$$

$$\dot{Y} = \dot{x}\psi - \dot{y}. \tag{4.20}$$

The partial derivatives for $\ddot{x}$ are

$$\frac{\partial \ddot{x}}{\partial \dot{x}} = -\frac{c_{\mathrm{f}}\phi(\dot{\psi}l_{\mathrm{f}} + \dot{y})}{m\dot{x}^2} - \frac{\rho\dot{x}a_{\mathrm{f}}c_{\mathrm{d}}}{m}, \tag{D.1}$$

$$\frac{\partial \ddot{x}}{\partial \dot{y}} = \frac{c_{\mathrm{f}}\phi}{m\dot{x}} + \dot{\psi}, \tag{D.2}$$

$$\frac{\partial \ddot{x}}{\partial \dot{\psi}} = \frac{c_{\mathrm{f}}l_{\mathrm{f}}\phi}{m\dot{x}} + \dot{y}, \tag{D.3}$$

$$\frac{\partial \ddot{x}}{\partial \phi} = -\frac{2c_{\mathrm{f}}\phi}{m} + \frac{c_{\mathrm{f}}(\dot{\psi}l_{\mathrm{f}} + \dot{y})}{m\dot{x}}, \tag{D.4}$$

$$\frac{\partial \ddot{x}}{\partial F_{\mathrm{r}x}} = \frac{\partial \ddot{x}}{\partial F_{\mathrm{f}x}} = \frac{1}{m}. \tag{D.5}$$

For $\ddot{y}$,

$$\frac{\partial \ddot{y}}{\partial \dot{x}} = \frac{(c_\mathrm{r} + c_\mathrm{f})\dot{y} + (-c_\mathrm{r} l_\mathrm{r} + c_\mathrm{f} l_\mathrm{f})\dot{\psi}}{m\dot{x}^2} - \dot{\psi}, \tag{D.6}$$

$$\frac{\partial \ddot{y}}{\partial \dot{y}} = -\frac{(c_\mathrm{r} + c_\mathrm{f})}{m\dot{x}}, \tag{D.7}$$

$$\frac{\partial \ddot{y}}{\partial \dot{\psi}} = \frac{c_\mathrm{r} l_\mathrm{f} - c_\mathrm{f} l_\mathrm{f}}{m\dot{x}} - \dot{x}, \tag{D.8}$$

$$\frac{\partial \ddot{y}}{\partial \phi} = \frac{F_{\mathrm{f}x} + c_\mathrm{f}}{m}, \tag{D.9}$$

$$\frac{\partial \ddot{y}}{\partial F_{\mathrm{f}x}} = \frac{\phi}{m}. \tag{D.10}$$

For $\ddot{\psi}$,

$$\frac{\partial \ddot{\psi}}{\partial \dot{x}} = \frac{(-c_\mathrm{r} l_\mathrm{r} + c_\mathrm{f} l_\mathrm{f})\dot{y} + (c_\mathrm{r} l_\mathrm{r}^2 + c_\mathrm{f} l_\mathrm{f}^2)\dot{\psi}}{I_\mathrm{z}\dot{x}^2}, \tag{D.11}$$

$$\frac{\partial \ddot{\psi}}{\partial \dot{y}} = \frac{(c_\mathrm{r} l_\mathrm{r} - c_\mathrm{f} l_\mathrm{f})}{I_\mathrm{z}\dot{x}}, \tag{D.12}$$

$$\frac{\partial \ddot{\psi}}{\partial \dot{y}} = -\frac{(c_\mathrm{r} l_\mathrm{r}^2 + c_\mathrm{f} l_\mathrm{f}^2)}{I_\mathrm{z}\dot{x}}, \tag{D.13}$$

$$\frac{\partial \ddot{\psi}}{\partial \phi} = \frac{F_{\mathrm{f}x} l_\mathrm{f} + c_\mathrm{f} l_\mathrm{f}}{I_\mathrm{z}\dot{x}}, \tag{D.14}$$

$$\frac{\partial \ddot{\psi}}{\partial F_{\mathrm{f}x}} = \frac{l_\mathrm{f}\phi}{I_\mathrm{z}}. \tag{D.15}$$

And finally, for $\dot{Y}$,

$$\frac{\partial \dot{Y}}{\partial \dot{x}} = \psi, \tag{D.16}$$

$$\frac{\partial \dot{Y}}{\partial \dot{y}} = 1, \tag{D.17}$$

$$\frac{\partial \dot{Y}}{\partial \dot{\psi}} = \dot{x}. \tag{D.18}$$

Bibliography

[1] Statistics Canada, "New motor vehicle registrations: Quarterly data visualization tool," 2021. [Online]. https://www150.statcan.gc.ca/n1/pub/71-607-x/71-607-x2021019-eng.htm

[2] Environment and Climate Change Canada, "Environmental Sustainability Indicators: Greenhouse gas emissions," Tech. Rep. En4-144/18-2021E-PDF, 2021.

[3] California Department of Motor Vehicles, "Autonomous vehicle testing permit holders," 2021. [Online]. https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-testing-permit-holders/

[4] Transport Canada, "Canadian Motor Vehicle Traffic Collision Statistics: 2019," 2021. [Online]. https://tc.canada.ca/en/road-transportation/statistics-data/canadian-motor-vehicle-traffic-collision-statistics-2019

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020. doi:10.1109/ACCESS.2020.2983149.

[6] National Highway Traffic Safety Administration, "Automated Vehicles for Safety," 2021. [Online]. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[7] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 1, p. 83, 2019. doi:10.1038/s41524-019-0221-0.

[8] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 2018. doi:10.1155/2018/7068349.

[9] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, 2020. doi:10.1016/j.ymssp.2019.106587.

[10] R. Nian, J. Liu, and B. Huang, "A review On reinforcement learning: Introduction and applications in industrial process control," *Computers & Chemical Engineering*, vol. 139, p. 106886, 2020. doi:10.1016/j.compchemeng.2020.106886.

[11] K. Zhou and J. C. Doyle, *Essentials of robust control*. Upper Saddle River: Prentice Hall, 1998.

[12] B. Boulet and Y. Duan, "The fundamental tradeoff between performance and robustness - A new perspective on loop shaping - Classic control revisited part II," *IEEE Control Systems*, vol. 27, no. 3, pp. 30–44, 2007. doi:10.1109/MCS.2007.365002.

[13] L. Ljung, *System identification: theory for the user*, 2nd ed. Upper Saddle River: Prentice Hall, 1999.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge: The MIT Press, 2016.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge: MIT Press, 2006.

[16] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, "Reinforcement learning and optimal adaptive control: An overview and implementation examples," *Annual Reviews in Control*, vol. 36, no. 1, pp. 42–59, 2012. doi:10.1016/j.arcontrol.2012.03.004.

[17] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. doi:10.1038/nature16961.

[18] M.-A. Beaudoin and B. Boulet, "Fundamental limitations to no-jerk gearshifts of multi-speed transmission architectures in electric vehicles," *Mechanism and Machine Theory*, vol. 160, p. 104290, 2021. doi:10.1016/j.mechmachtheory.2021.104290.

[19] ——, "Improving gearshift controllers for electric vehicles with reinforcement learning," *Mechanism and Machine Theory*, vol. 169, p. 104654, 2022. doi:10.1016/j.mechmachtheory.2021.104654.

[20] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A Model-Based and Data-Efficient Approach to Policy Search," in *28th International Conference on Machine Learning (ICML)*, Bellevue, WA, USA, Jun. 2011.

[21] M.-A. Beaudoin and B. Boulet, "Learning-Based Synthesis of Robust Linear Time-Invariant Controllers," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022. doi:10.1109/TIV.2022.3174029.

[22] ——, "Structured learning of safety guarantees for the control of uncertain dynamical systems," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022. doi:10.1109/TIV.2022.3148212.

[23] M.-A. Beaudoin, "Vehicle transmission system and method for operation of the transmission system," US Patent US20 220 034 385A1, Feb., 2022.

[24] G. Wu, X. Zhang, and Z. Dong, "Powertrain architectures of electrified vehicles: Review, classification and comparison," *Journal of the Franklin Institute*, vol. 352, no. 2, pp. 425–448, 2015. doi:10.1016/j.jfranklin.2014.04.018.

[25] Y. Yang, K. A. Ali, J. Roeleveld, and A. Emadi, "State-of-the-art electrified powertrains - hybrid, plug-in, and electric vehicles," *International Journal of Powertrains*, vol. 5, no. 1, p. 1, 2016. doi:10.1504/IJPT.2016.075181.

[26] F. A. Machado, P. Kollmeyer, D. G. Barroso, and A. Emadi, "Multi-speed Gearboxes for Battery Electric Vehicles: Current Status and Future Trends," *IEEE Open Journal of Vehicular Technology*, pp. 1–1, 2021. doi:10.1109/OJVT.2021.3124411.

[27] R. E. Dorey and C. B. Holmes, "Vehicle Driveability - Its Characterisation and Measurement," *SAE Technical Paper*, no. 1999-01-0949, 1999. doi:10.4271/1999-01-0949.

[28] A. Sorniotti, E. Galvagno, A. Morgando, M. Velardocchia, and F. Amisano, "An Objective Evaluation of the Comfort During the Gear Change Process," in *SAE World Congress & Exhibition*, Detroit, MI, USA, Apr. 2007. doi:10.4271/2007-01-1584.

[29] P. E. Resele and O. Bitsche, "Advanced Fully Automatic Two-Speed Transmission for Electric Automobiles," *SAE Technical Paper*, no. 951885, 1995. doi:10.4271/951885.

[30] Y. Lei, J. Hu, Y. Fu, S. Sun, X. Li, W. Chen, L. Hou, and Y. Zhang, "Control strategy of automated manual transmission based on active synchronisation of driving motor in electric bus," *Advances in Mechanical Engineering*, vol. 11, no. 4, 2019. doi:10.1177/1687814019846734.

[31] H. V. Alizadeh and B. Boulet, "Robust control of synchromesh friction in an electric vehicle's clutchless automated manual transmission," in *2014 IEEE Conference on Control Applications (CCA)*, Juan Les Antibes, France, Oct. 2014. doi:10.1109/CCA.2014.6981407.

[32] C.-Y. Tseng and C.-H. Yu, "Advanced shifting control of synchronizer mechanisms for clutchless automatic manual transmission in an electric vehicle," *Mechanism and Machine Theory*, vol. 84, pp. 37–56, 2015. doi:10.1016/j.mechmachtheory.2014.10.007.

[33] W. Mo, J. Wu, P. D. Walker, and N. Zhang, "Shift characteristics of a bilateral Harpoon-shift synchronizer for electric vehicles equipped with clutchless AMTs," *Mechanical Systems and Signal Processing*, vol. 148, p. 107166, 2021. doi:10.1016/j.ymssp.2020.107166.

[34] W. Mo, P. D. Walker, Y. Tian, and N. Zhang, "Dynamic analysis of unilateral harpoon-shift synchronizer for electric vehicles," *Mechanism and Machine Theory*, vol. 157, p. 104173, 2021. doi:10.1016/j.mechmachtheory.2020.104173.

[35] F. Amisano, E. Galvagno, M. Velardocchia, and A. Vigliani, "Automated manual transmission with a torque gap filler Part 1: kinematic analysis and dynamic analysis," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 228, no. 11, pp. 1247–1261, 2014. doi:10.1177/0954407014527739.

[36] E. Galvagno, M. Velardocchia, and A. Vigliani, "Analysis and simulation of a torque assist automated manual transmission," *Mechanical Systems and Signal Processing*, vol. 25, no. 6, pp. 1877–1886, 2011. doi:10.1016/j.ymssp.2010.12.014.

[37] A. Sorniotti, T. Holdstock, G. L. Pilone, F. Viotto, S. Bertolotto, M. Everitt, R. J. Barnes, B. Stubbs, and M. Westby, "Analysis and simulation of the gearshift methodology for a novel two-speed transmission system for electric powertrains with a central motor," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 7, pp. 915–929, 2012. doi:10.1177/0954407011431415.

[38] S. Hong, H. Son, S. Lee, J. Park, K. Kim, and H. Kim, "Shift control of a dry-type two-speed dual-clutch transmission for an electric vehicle," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, no. 3, pp. 308–321, 2016. doi:10.1177/0954407015585686.

[39] B. Gao, Q. Liang, Y. Xiang, L. Guo, and H. Chen, "Gear ratio optimization and shift control of 2-speed I-AMT in electric vehicle," *Mechanical Systems and Signal Processing*, vol. 50-51, pp. 615–631, 2015. doi:10.1016/j.ymssp.2014.05.045.

[40] M. S. Mousavi, A. Pakniyat, T. Wang, and B. Boulet, "Seamless dual brake transmission for electric vehicles: Design, control and experiment," *Mechanism and Machine Theory*, vol. 94, pp. 96–118, 2015. doi:10.1016/j.mechmachtheory.2015.08.003.

[41] Y. Tian, J. Ruan, N. Zhang, J. Wu, and P. Walker, "Modelling and control of a novel two-speed transmission for electric vehicles," *Mechanism and Machine Theory*, vol. 127, pp. 13–32, 2018. doi:10.1016/j.mechmachtheory.2018.04.023.

[42] S. Fang, J. Song, H. Song, Y. Tai, F. Li, and T. Sinh Nguyen, "Design and control of a novel two-speed Uninterrupted Mechanical Transmission for electric vehicles," *Mechanical Systems and Signal Processing*, vol. 75, pp. 473–493, 2016. doi:10.1016/j.ymssp.2015.07.006.

[43] M. Roozegar, Y. Setiawan, and J. Angeles, "Design, modelling and estimation of a novel modular multi-speed transmission system for electric vehicles," *Mechatronics*, vol. 45, pp. 119–129, 2017. doi:10.1016/j.mechatronics.2017.06.002.

[44] J. Ye, X. Huang, K. Zhao, and Y. Liu, "Optimal coordinating control for the overlapping shift of a seamless 2-speed transmission equipped in an electric vehicle," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 10, pp. 797–811, 2017. doi:10.1177/0959651817730176.

[45] S. Murata, "Innovation by in-wheel-motor drive unit," *Vehicle System Dynamics*, vol. 50, no. 6, pp. 807–830, 2012. doi:10.1080/00423114.2012.666354.

[46] Y. Tang, "Control system for an all-wheel drive electric vehicle," US Patent US7 739 005B1, Jun., 2010.

[47] ——, "Dual motor drive and control system for an electric vehicle," US Patent US8 453 770B2, Jun., 2013.

[48] J. Wu, J. Liang, J. Ruan, N. Zhang, and P. D. Walker, "A robust energy management strategy for EVs with dual input power-split transmission," *Mechanical Systems and Signal Processing*, vol. 111, pp. 442–455, 2018. doi:10.1016/j.ymssp.2018.04.007.

[49] ——, "Efficiency comparison of electric vehicles powertrains with dual motor and single motor input," *Mechanism and Machine Theory*, vol. 128, pp. 569–585, 2018. doi:10.1016/j.mechmachtheory.2018.07.003.

[50] M. Hu, J. Zeng, S. Xu, C. Fu, and D. Qin, "Efficiency Study of a Dual-Motor Coupling EV Powertrain," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2252–2260, 2015. doi:10.1109/TVT.2014.2347349.

[51] J. Wu and N. Zhang, "Driving mode shift control for planetary gear based dual motor powertrain in electric vehicles," *Mechanism and Machine Theory*, vol. 158, p. 104217, 2021. doi:10.1016/j.mechmachtheory.2020.104217.

[52] J. Miller, "Hybrid electric vehicle propulsion system architectures of the e-CVT type," *IEEE Transactions on Power Electronics*, vol. 21, no. 3, pp. 756–767, 2006. doi:10.1109/TPEL.2006.872372.

[53] A. Sorniotti, T. Holdstock, M. Everitt, M. Fracchia, F. Viotto, C. Cavallino, and S. Bertolotto, "A novel clutchless multiple-speed transmission for electric axles," *International Journal of Powertrains*, vol. 2, no. 2/3, p. 103, 2013. doi:10.1504/IJPT.2013.054155.

[54] J. Liang, H. Yang, J. Wu, N. Zhang, and P. D. Walker, “Shifting and power sharing control of a novel dual input clutchless transmission for electric vehicles,” *Mechanical Systems and Signal Processing*, vol. 104, pp. 725–743, 2018. doi:10.1016/j.ymssp.2017.11.033.

[55] C. T. Nguyen, P. D. Walker, and N. Zhang, “Shifting strategy and energy management of a two-motor drive powertrain for extended-range electric buses,” *Mechanism and Machine Theory*, vol. 153, p. 103966, 2020. doi:10.1016/j.mechmachtheory.2020.103966.

[56] F. Bottiglione, S. De Pinto, G. Mantriota, and A. Sorniotti, “Energy Consumption of a Battery Electric Vehicle with Infinitely Variable Transmission,” *Energies*, vol. 7, no. 12, pp. 8317–8337, 2014. doi:10.3390/en7128317.

[57] I. I. Mazali, Z. H. C. Daud, M. K. A. Hamid, V. Tan, P. M. Samin, A. Jubair, K. A. Ibrahim, M. S. C. Kob, W. Xinrui, and M. H. A. Talib, “Review of the Methods to Optimize Power Flow in Electric Vehicle Powertrains for Efficiency and Driving Performance,” *Applied Sciences*, vol. 12, no. 3, p. 1735, 2022. doi:10.3390/app12031735.

[58] A. Sorniotti, S. Subramanyan, A. Turner, C. Cavallino, F. Viotto, and S. Bertolotto, “Selection of the Optimal Gearbox Layout for an Electric Vehicle,” *SAE International Journal of Engines*, vol. 4, no. 1, pp. 1267–1280, 2011. doi:10.4271/2011-01-0946.

[59] F. Di Nicola, A. Sorniotti, T. Holdstock, F. Viotto, and S. Bertolotto, “Optimization of a Multiple-Speed Transmission for Downsizing the Motor of a Fully Electric Vehicle,” *SAE International Journal of Alternative Powertrains*, vol. 1, no. 1, pp. 134–143, 2012. doi:10.4271/2012-01-0630.

[60] J. Goss, R. Wrobel, P. Mellor, and D. Staton, “The design of AC permanent magnet motors for electric vehicles: A design methodology,” in *2013 International Electric Machines & Drives Conference (IEMDC)*, Chicago, IL, USA, May 2013. doi:10.1109/IEMDC.2013.6556200.

[61] S. Bai, J. Maguire, and H. Peng, *Dynamic analysis and control system design of automatic transmissions*. Warrendale: SAE, 2013.

[62] O. H. Dagci, H. Peng, and J. W. Grizzle, “Hybrid Electric Powertrain Design Methodology With Planetary Gear Sets for Performance and Fuel Economy,” *IEEE Access*, vol. 6, pp. 9585–9602, 2018. doi:10.1109/ACCESS.2018.2796939.

[63] J. Liu, L. Yu, Q. Zeng, and Q. Li, “Synthesis of multi-row and multi-speed planetary gear mechanism for automatic transmission,” *Mechanism and Machine Theory*, vol. 128, pp. 616–627, 2018. doi:10.1016/j.mechmachtheory.2018.07.007.

[64] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, “Topology Generation for Hybrid Electric Vehicle Architecture Design,” *Journal of Mechanical Design*, vol. 138, no. 8, p. 081401, 2016. doi:10.1115/1.4033656.

[65] K. T. Ulrich and S. D. Eppinger, *Product design and development*, 5th ed. New York: McGraw-Hill/Irwin, 2012.

[66] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, *Engineering Design*. London: Springer, 2007.

[67] M. J. French, *Conceptual Design for Engineers*. London: Springer, 1999.

[68] M. Golkani, M. Steinberger, M. Bachinger, J. Rumetshofer, M. Stolz, and M. Horn, “Optimal Gear Shift Strategy for Dual Clutch Transmissions,” in *20th IFAC World Congress*, Toulouse, France, Jul. 2017. doi:10.1016/j.ifacol.2017.08.964.

[69] J. Ye, K. Zhao, Y. Liu, X. Huang, and H. Lin, “Multi-stage global trajectory optimization for the overlapping shift of a seamless two-speed transmission using Legendre pseudo-spectral method,” *Advances in Mechanical Engineering*, vol. 9, no. 12, 2017. doi:10.1177/1687814017741737.

[70] A. Haj-Fraj and F. Pfeiffer, “Optimal control of gear shift operations in automatic transmissions,” *Journal of the Franklin Institute*, vol. 338, no. 2-3, pp. 371–390, 2001. doi:10.1016/S0016-0032(00)00091-0.

[71] P. Walker, B. Zhu, and N. Zhang, “Powertrain dynamics and control of a two speed dual clutch transmission for electric vehicles,” *Mechanical Systems and Signal Processing*, vol. 85, pp. 1–15, 2017. doi:10.1016/j.ymssp.2016.07.043.

[72] B. Gao, Y. Xiang, H. Chen, Q. Liang, and L. Guo, “Optimal Trajectory Planning of Motor Torque and Clutch Slip Speed for Gear Shift of a Two-Speed Electric Vehicle,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 6, p. 061016, 2015. doi:10.1115/1.4029469.

[73] S. Kim, J. Oh, and S. Choi, “Gear shift control of a dual-clutch transmission using optimal control allocation,” *Mechanism and Machine Theory*, vol. 113, pp. 109–125, 2017. doi:10.1016/j.mechmachtheory.2017.02.013.

[74] R. A. Fisher, *The design of experiments*, 1st ed. Edinburgh: Oliver and Boyd, 1935.

[75] K. D. Mishra, G. Cardwell, and K. Srinivasan, “Automated calibration of gearshift controllers using iterative learning control for hybrid systems,” *Control Engineering Practice*, vol. 111, p. 104786, 2021. doi:10.1016/j.conengprac.2021.104786.

[76] F. Kucukay, T. Kassel, G. Alvermann, and T. Gartung, “Efficient Calibration of Automatic Transmissions on the Roller Dynamometer,” *ATZ worldwide*, vol. 111, no. 3, pp. 46–52, 2009. doi:10.1007/BF03225221.

[77] F. Boissinot, J. Bellavoine, A. Shabashevich, and S. Puster, “Automated Calibration for Transmission on Powertrain Dynamometers,” *SAE Technical Paper*, no. 2015-01-1625, 2015. doi:10.4271/2015-01-1625.

[78] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998. doi:10.1023/A:1008306431147.

[79] B. Z. Gao, H. Chen, K. Sanada, and Y. Hu, “Design of Clutch-Slip Controller for Automatic Transmission Using Backstepping,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 3, pp. 498–508, 2011. doi:10.1109/TMECH.2010.2045391.

[80] B. Gao, H. Chen, Y. Hu, and K. Sanada, “Nonlinear feedforward–feedback control of clutch-to-clutch shift technique,” *Vehicle System Dynamics*, vol. 49, no. 12, pp. 1895–1911, 2011. doi:10.1080/00423114.2011.552617.

[81] K. Sanada and A. Kitagawa, "A study of two-degree-of-freedom control of rotating speed in an automatic transmission, considering modeling errors of a hydraulic system," *Control Engineering Practice*, vol. 6, no. 9, pp. 1125–1132, 1998. doi:10.1016/S0967-0661(98)00105-1.

[82] S. Kim and S. B. Choi, "Cooperative Control of Drive Motor and Clutch for Gear Shift of Hybrid Electric Vehicles With Dual-Clutch Transmission," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1578–1588, 2020. doi:10.1109/TMECH.2020.2980120.

[83] F. Mesmer, T. Szabo, and K. Graichen, "Embedded Nonlinear Model Predictive Control of Dual-Clutch Transmissions With Multiple Groups on a Shrinking Horizon," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 2156–2168, 2019. doi:10.1109/TCST.2018.2856191.

[84] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006. doi:10.1109/MCS.2006.1636313.

[85] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to Trust Your Model: Model-Based Policy Optimization," in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2019.

[86] M. P. Deisenroth, "Efficient reinforcement learning using Gaussian processes," Ph.D. thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2010.

[87] M. Liu, G. Chowdhary, B. Castra da Silva, S.-Y. Liu, and J. P. How, "Gaussian Processes for Learning and Control: A Tutorial with Examples," *IEEE Control Systems*, vol. 38, no. 5, pp. 53–86, 2018. doi:10.1109/MCS.2018.2851010.

[88] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning," in *International Conference on Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2011. doi:10.15607/RSS.2011.VII.008.

[89] A. Doerr, D. Nguyen-Tuong, A. Marco, S. Schaal, and S. Trimpe, "Model-based policy search for automatic tuning of multivariate PID controllers," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May 2017. doi:10.1109/ICRA.2017.7989622.

[90] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Savannah, GA, USA, Nov. 2016.

[91] T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba, "Benchmarking Model-Based Reinforcement Learning," *arXiv:1907.02057*, 2019.

[92] S. Levine and V. Koltun, "Guided Policy Search," in *30th International Conference on Machine Learning (ICML)*, Atlanta, GA, USA, Jun. 2013.

[93] E. Todorov and Weiwei Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference (ACC)*, Portland, OR, USA, 2005. doi:10.1109/ACC.2005.1469949.

[94] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the Sample Complexity of the Linear Quadratic Regulator," *Foundations of Computational Mathematics*, vol. 20, pp. 633–679, 2019. doi:10.1007/s10208-019-09426-y.

[95] B. Recht, "A Tour of Reinforcement Learning: The View from Continuous Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 253–279, 2019. doi:10.1146/annurev-control-053018-023825.

[96] E. Galvagno, D. Morina, A. Sorniotti, and M. Velardocchia, "Drivability analysis of through-the-road-parallel hybrid vehicles," *Meccanica*, vol. 48, no. 2, pp. 351–366, 2013. doi:10.1007/s11012-012-9606-6.

[97] T. Holdstock, A. Sorniotti, N. Suryanto, L. Shead, F. Viotto, C. Cavallino, and S. Bertolotto, "Linear and non-linear methods to analyse the drivability of a through-the-road parallel hybrid electric vehicle," *International Journal of Powertrains*, vol. 2, no. 1, p. 52, 2013. doi:10.1504/IJPT.2013.052664.

[98] B. Bischoff, D. Nguyen-Tuong, H. van Hoof, A. McHutchon, C. E. Rasmussen, A. Knoll, J. Peters, and M. Deisenroth, "Policy search for learning robot control using sparse data," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014. doi:10.1109/ICRA.2014.6907422.

[99] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian Process Priors with Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting," in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2002.

[100] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious Model Predictive Control Using Gaussian Process Regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 1–8, 2020. doi:10.1109/TCST.2019.2949757.

[101] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *Journal of machine learning research*, vol. 18, no. 153, pp. 1–43, 2018.

[102] Society of Automotive Engineers, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," Standard J3016_202104, 2021.

[103] O. S. Tas, F. Kuhnt, J. M. Zollner, and C. Stiller, "Functional system architectures towards fully automated driving," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gotenburg, Sweden, Jun. 2016. doi:10.1109/IVS.2016.7535402.

[104] A. Serban, E. Poll, and J. Visser, "A Standard Driven Software Architecture for Fully Autonomous Vehicles:," *Journal of Automotive Software Engineering*, vol. 1, no. 1, p. 20, 2020. doi:10.2991/jase.d.200212.001.

[105] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of Autonomous Car—Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5119–5132, 2015. doi:10.1109/TIE.2015.2410258.

[106] A. Wischnewski, T. Stahl, J. Betz, and B. Lohmann, "Vehicle Dynamics State Estimation and Localization for High Performance Race Cars," in *10th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, Gdansk, Poland, Jul. 2019. doi:10.1016/j.ifacol.2019.08.064.

[107] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017. doi:10.1109/TIV.2017.2749181.

[108] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A Survey on 3D Object Detection Methods for Autonomous Driving Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019. doi:10.1109/TITS.2019.2892405.

[109] D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, C. Glaser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2021. doi:10.1109/TITS.2020.2972974.

[110] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016. doi:10.1109/TITS.2015.2498841.

[111] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015. doi:10.1016/j.trc.2015.09.011.

[112] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A Behavioral Planning Framework for Autonomous Driving," in *2014 IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA, Jun. 2014. doi:10.1109/IVS.2014.6856582.

[113] Y. Zhang, H. Chen, S. L. Waslander, T. Yang, S. Zhang, G. Xiong, and K. Liu, "Speed Planning for Autonomous Driving via Convex Optimization," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, Nov. 2018. doi:10.1109/ITSC.2018.8569414.

[114] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-Time Trajectory Planning for Autonomous Urban Driving: Framework, Algorithms, and Verifications," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 740–753, 2016. doi:10.1109/TMECH.2015.2493980.

[115] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical Game-Theoretic Planning for Autonomous Vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019. doi:10.1109/ICRA.2019.8794007.

[116] B. McAuliffe, P. Smith, A. Raeesi, M. Hoffman, and D. M. Bevly, "Track-Based Aerodynamic Testing of a Two-Truck Platoon," in *SAE WCX Digital Summit*, Apr. 2021. doi:10.4271/2021-01-0941.

[117] S.-S. You and S.-K. Jeong, "Vehicle dynamics and control synthesis for four-wheel steering passenger cars," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 212, no. 6, pp. 449–461, 1998. doi:10.1243/0954407981526109.

[118] H. B. Pacejka and I. Besselink, *Tire and vehicle dynamics*, 3rd ed. Oxford: Butterworth-Heinemann, 2012.

[119] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, Jun. 2015. doi:10.1109/IVS.2015.7225830.

[120] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and Decision-Making for Autonomous Vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 2018. doi:10.1146/annurev-control-060117-105157.

[121] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, Planning, Control, and Coordination for Autonomous Vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017. doi:10.3390/machines5010006.

[122] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016. doi:10.1109/TIV.2016.2578706.

[123] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive Active Steering Control for Autonomous Vehicle Systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007. doi:10.1109/TCST.2007.894653.

[124] U. Kiencke and L. Nielsen, *Automotive control systems: for engine, driveline, and vehicle*, 2nd ed. Berlin: Springer, 2005.

[125] R. Marino, S. Scalzi, and M. Netto, "Nested PID steering control for lane keeping in autonomous vehicles," *Control Engineering Practice*, vol. 19, no. 12, pp. 1459–1467, 2011. doi:10.1016/j.conengprac.2011.08.005.

[126] E. J. Rossetter and J. C. Gerdes, "Lyapunov Based Performance Guarantees for the Potential Field Lane-keeping Assistance System," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 3, pp. 510–522, 2006. doi:10.1115/1.2192835.

[127] S. Xu and H. Peng, "Design, Analysis, and Experiments of Preview Path Tracking Control for Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48–58, 2020. doi:10.1109/TITS.2019.2892926.

[128] A. Wischnewski, T. Herrmann, F. Werner, and B. Lohmann, "A Tube-MPC Approach to Autonomous Multi-Vehicle Racing on High-Speed Ovals," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022. doi:10.1109/TIV.2022.3169986.

[129] M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020. doi:10.1177/0278364919887447.

[130] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, Sep. 2017. doi:10.1109/IROS.2017.8202133.

[131] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active Domain Randomization," in *Conference on Robot Learning (CoRL)*, Osaka, Japan, Oct. 2019.

[132] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning Robust Control Policies for End-to-End Autonomous Driving From Data-Driven Simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020. doi:10.1109/LRA.2020.2966414.

[133] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe Model-Based Reinforcement Learning with Stability Guarantees," in *31st International Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, Dec. 2017.

[134] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic Learning with Control Lyapunov Functions for Uncertain Robotic Systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019. doi:10.1109/IROS40897.2019.8967820.

[135] M. K. Helwa, A. Heins, and A. P. Schoellig, "Provably Robust Learning-Based Approach for High-Accuracy Tracking Control of Lagrangian Systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1587–1594, 2019. doi:10.1109/LRA.2019.2896728.

[136] H. Mania, M. I. Jordan, and B. Recht, "Active Learning for Nonlinear System Identification with Guarantees," *Journal of Machine Learning Research*, vol. 23, pp. 1–30, 2022.

[137] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust Guarantees for Perception-Based Control," in *2nd Conference on Learning for Dynamics and Control (L4DC)*, Berkeley, CA, USA, Jun. 2020.

[138] R. Hyde and K. Glover, "The application of scheduled $\mathcal{H}_\infty$ controllers to a VSTOL aircraft," *IEEE Transactions on Automatic Control*, vol. 38, no. 7, pp. 1021–1039, 1993. doi:10.1109/9.231458.

[139] B. Boulet, B. Francis, P. Hughes, and T. Hong, "Uncertainty modeling and experiments in $\mathcal{H}_\infty$ control of large flexible space structure," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 5, pp. 504–519, 1997. doi:10.1109/87.623036.

[140] C. Fiedler, C. W. Scherer, and S. Trimpe, "Learning-enhanced robust controller synthesis with rigorous statistical and control-theoretic guarantees," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Austin, TX, USA, Dec. 2021. doi:10.1109/CDC45484.2021.9682836.

[141] R. M. Kretchmar, P. M. Young, C. W. Anderson, D. C. Hittle, M. L. Anderson, and C. C. Delnero, "Robust reinforcement learning control with static and dynamic stability," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 15, pp. 1469–1500, 2001. doi:10.1002/rnc.670.

[142] M. Jin and J. Lavaei, "Stability-Certified Reinforcement Learning: A Control-Theoretic Perspective," *IEEE Access*, vol. 8, pp. 229 086–229 100, 2020. doi:10.1109/ACCESS.2020.3045114.

[143] G. Zames, "On the input-output stability of time-varying nonlinear feedback systems Part one: Conditions derived using concepts of loop gain, conicity, and positivity," *IEEE Transactions on Automatic Control*, vol. 11, no. 2, pp. 228–238, 1966. doi:10.1109/TAC.1966.1098316.

[144] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," San Diego, CA, USA, May 2015.

[145] E. F. Camacho and C. Bordons, *Model Predictive control*, 2nd ed. London: Springer, 2007.

[146] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017. doi:10.1109/TAC.2016.2638961.

[147] M. Chen and C. J. Tomlin, "Hamilton–Jacobi Reachability: Some Recent Theoretical Advances and Applications in Unmanned Airspace Management," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 333–358, 2018. doi:10.1146/annurev-control-060117-104941.

[148] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads," in *ASME 2010 Dynamic Systems and Control Conference*, Cambridge, MA, USA, Jan. 2010. doi:10.1115/DSCC2010-4263.

[149] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015. doi:10.1002/oca.2123.

[150] Q. Nguyen and K. Sreenath, "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference (ACC)*, Boston, MA, USA, Jul. 2016. doi:10.1109/ACC.2016.7524935.

[151] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *18th European Control Conference (ECC)*, Napoli, Italy, Jun. 2019. doi:10.23919/ECC.2019.8796030.

[152] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *18th International Conference on Hybrid Systems: Computation and Control*, Seattle, WA, USA, Apr. 2015. doi:10.1145/2728606.2728612.

[153] M. Chen, Q. Hu, J. F. Fisac, K. Akametalu, C. Mackin, and C. J. Tomlin, "Reachability-Based Safety and Goal Satisfaction of Unmanned Aerial Platoons on Air Highways," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 6, pp. 1360–1373, 2017. doi:10.2514/1.G000774.

[154] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust Control Barrier-Value Functions for Safety-Critical Control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Austin, TX, USA, Dec. 2021. doi:10.1109/CDC45484.2021.9683085.

[155] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. London, UK: Springer, 1999, vol. 245, pp. 207–226.

[156] H. Seo, D. Lee, C. Y. Son, C. J. Tomlin, and H. J. Kim, "Robust Trajectory Planning for a Multirotor against Disturbance based on Hamilton-Jacobi Reachability Analysis," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019. doi:10.1109/IROS40897.2019.8968126.

[157] Q. Nguyen and K. Sreenath, "Robust Safety-Critical Control for Dynamic Robotics," *IEEE Transactions on Automatic Control*, pp. 1–1, 2021. doi:10.1109/TAC.2021.3059156.

[158] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *2nd Conference on Learning for Dynamics and Control (L4DC)*, Berkeley, CA, USA, Jun. 2020.

[159] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *American Control Conference (ACC)*, Denver, CO, USA, Jul. 2020. doi:10.23919/ACC45564.2020.9147463.

[160] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Robust Adaptive Control Barrier Functions: An Adaptive and Data-Driven Approach to Safety," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1031–1036, 2021. doi:10.1109/LCSYS.2020.3005923.

[161] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019. doi:10.1109/TAC.2018.2876389.

[162] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020. doi:10.1146/annurev-control-090419-075625.

[163] A. Gattami, A. Al Alam, K. H. Johansson, and C. J. Tomlin, "Establishing Safety for Heavy Duty Vehicle Platooning: A Game Theoretical Approach," in *18th IFAC World Congress*, Milano, Italy, Jan. 2011. doi:10.3182/20110828-6-IT-1002.02071.

[164] R. V. Jategaonkar, *Flight vehicle system identification: a time-domain methodology*, 2nd ed. Reston: AIAA, 2015.

[165] R. Ervin, C. Winkler, J. Bernard, and R. Gupta, "Effects of tire properties on truck and bus handling," Highway Safety Research Institute, The University of Michigan, Ann Arbor, MI, USA, Tech. Rep. UM-HSRI-76-11, 1976.

[166] G. Tapia, "Extended Tire Testing," Calspan Corporation, Buffalo, NY, USA, Tech. Rep. 6871-V-1, 1983.

[167] U.S. Department of Transportation, "Laboratory Test Procedure for FMVSS 105: Hydraulic And Electric Brake Systems," Tech. Rep. TP-105-03, 2005.

[168] C. Zhang, A. Kotz, K. Kelly, and L. Rippelmeyer, "Development of heavy-duty vehicle representative driving cycles via decision tree regression," *Transportation Research Part D: Transport and Environment*, vol. 95, p. 102843, 2021. doi:10.1016/j.trd.2021.102843.

[169] A. J. Taylor, V. D. Dorobantu, S. Dean, B. Recht, Y. Yue, and A. D. Ames, "Towards Robust Data-Driven Control Synthesis for Nonlinear Systems with Actuation Uncertainty," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Austin, TX, USA, Dec. 2021. doi:10.1109/CDC45484.2021.9683511.

[170] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks," in *33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, Jan. 2019. doi:10.1609/aaai.v33i01.33013387.

[171] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, "Probabilistic Safety Constraints for Learned High Relative Degree System Dynamics," in *2nd Conference on Learning for Dynamics and Control (L4DC)*, Berkeley, CA, USA, Jun. 2020.

[172] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe Learning of Quadrotor Dynamics Using Barrier Certificates," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018. doi:10.1109/ICRA.2018.8460471.

[173] L. Hewing, E. Arcari, L. P. Fröhlich, and M. N. Zeilinger, "On Simulation and Trajectory Prediction with Gaussian Process Dynamics," in *2nd Conference on Learning for Dynamics and Control (L4DC)*, Berkeley, CA, USA, Jun. 2020.

[174] M. Batra, J. McPhee, and N. L. Azad, "Anti-Jerk Dynamic Modeling and Parameter Identification of an Electric Vehicle Based on Road Tests," *Journal of Computational and Nonlinear Dynamics*, vol. 13, no. 10, p. 101005, 2018. doi:10.1115/1.4040870.

[175] M. Batra, A. Maitland, J. McPhee, and N. L. Azad, "Non-Linear Model Predictive Anti-Jerk Cruise Control for Electric Vehicles with Slip-Based Constraints," in *American Control Conference (ACC)*, Milwaukee, WI, USA, Jun. 2018. doi:10.23919/ACC.2018.8431389.

[176] J. Buerger and J. Anderson, "Robust control for electric vehicle powertrains," *Control Theory and Technology*, vol. 17, no. 4, pp. 382–392, 2019. doi:10.1007/s11768-019-8279-2.

[177] R. Cheng, R. M. Murray, and J. W. Burdick, "Limits of Probabilistic Safety Guarantees when Considering Human Uncertainty," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, May 2021. doi:10.1109/ICRA48506.2021.9561843.

[178] C. M. Bishop, *Pattern recognition and machine learning.* New York: Springer, 2006.

[179] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, 2nd ed. Cambridge: The MIT Press, 2018.

[180] R. Bellman, *Dynamic programming.* Princeton: Princeton University Press, 1957.

[181] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, 4th ed. Hoboken: Pearson, 2021.

[182] A. McHutchon, "Nonlinear Modelling and Control using Gaussian Processes," Ph.D. thesis, University of Cambridge, Cambridge, UK, 2014.