UNIVERSITÀ DI PADOVA  FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione

Scuola di Dottorato di Ricerca in Ingegneria dell'Informazione
Indirizzo: Ingegneria Informatica ed Elettronica Industriali
XX Ciclo

Topics on Optimization Strategies for Constrained Mechanical Systems

Dottoranda: MAURA PASQUOTTI

Supervisore: Prof. RUGGERO FREZZA

Direttore della Scuola: Prof. MATTEO BERTOCCO

Padova, 31 Luglio 2008

Ai miei genitori

Ich sage euch: man muß noch Chaos in sich haben, um einen tanzenden Stern gebären zu können.[1]

Friedrich Wilhelm Nietzsche,
Also sprach Zarathustra

[1] I tell you: one must have chaos within oneself, to give birth to a dancing star
Bisogna avere un caos dentro di sé per partorire una stella danzante.

Abstract

In the first part of the work we analyze the problem of optimal control of a vehicle along a preassigned trajectory. The vehicle system is studied and simplified in order to obtain a computationally tractable model, which still presents the main characteristics of the real vehicle. The control algorithm, based on MPC techniques, is then explained and its effectiveness is proved through simulation results. The "minimum lap time problem" is afterward considered, which can be considered as an evolution of the trajectory tracking problem; its analysis is presented and is followed by the developed solution, based on pseudospectral methods.

In the second part of the thesis the problem of control of underactuated mechanical systems is discussed. The nonholonomic system classically called "rolling disk" is considered as test case; it is a wheel with punctiform contact surface that can roll on the plane without sliding laterally. Differently from the literature we consider the torque as the unique control input signal. This system is modeled through the Lagrangian formalism and then the control strategy, based on backstepping and receding horizon techniques, is shown and proved to be effective.

Keywords: Nonlinear optimal control, constrained optimization, trajectory tracking, receding Horizon, pseudospectral methods, underactuated systems, non-holonomic constraints.

Sommario

Nella prima parte del lavoro viene analizzato il problema del controllo ottimo di un veicolo lungo una traiettoria preassegnata. Il sistema in questione viene pertanto analizzato e modellizzato in maniera da renderlo computazionalmente trattabile pur garantendo la buona approssimazione del sistema reale. Successivamente viene presentato l'algoritmo di controllo ottimo sviluppato, basato su tecniche di model predictive control e viene dimostrata attraverso le prove sperimentali l'efficacia della strategia. Viene poi considerata l'estensione al caso in cui il controllo non sia solo di traiettoria, ma che vi sia presente anche un vincolo temporale, passando pertanto alla formulazione del problema del "minimum lap time", alla cui analisi segue la risoluzione attraverso l'applicazione di metodi pseudospettrali.

Nella seconda parte della tesi viene invece affrontato il problema del controllo di sistemi meccanici sottoattuati. Come esempio di sistema appartenente a tale classe viene considerato il caso del "rolling disk", ben noto in letteratura, che consiste in un disco che può muoversi nel piano e che è soggetto a un vincolo anolonomo di non scivolamento laterale. A differenza di quanto considerato classicamente in letteratura, viene aggiunto un ulteriore vincolo sul controllo; l'unico ingresso ammissibile al sistema sarà la coppia motrice alla ruota, mentre verrà tralasciata la possibilità di avere un angolo di sterzo come ingresso. In tale maniera la sottoattuazione del sistema diventa maggiore. Dopo aver modellizzato il disco utilizzando l'approccio lagrangiano, verrà sviluppato l'algoritmo di controllo, basato su backstepping and receding horizon tecniques.

Parole chiave: Controllo ottimo non lineare, ottimizzazione vincolata, inseguimento di traiettoria, metodi pseudospettrali, sistemi sottoattuati, vincoli anolonomi.

Contents

Introduction **1**

1 Vehicle Models **5**
1.1 Vehicle Analysis 5
1.2 Coordinate System 6
1.2.1 Motion Equation 7
1.3 Tire Model 10
1.3.1 Combined Pacejka Formula 12
1.3.2 Linear Approximation 14
1.4 Steady-State Handling Characteristics 15

2 Optimal Controller **23**
2.1 Lateral Control 23
2.1.1 Path Following 24
2.1.2 Connecting Contour Computation 24
2.2 Longitudinal Control 25
2.3 The Controller 27
2.3.1 Model Linearization 29
2.3.2 Optimal Control 30
2.4 Controller Implementation 30
2.5 Simulation Results 31
2.5.1 Vehicle Characteristics 31
2.5.2 Straight Path 32
2.5.3 Chicane 32
2.5.4 Mixed Path Test 36
2.5.5 Path with Vertical Dynamics 40
2.6 Further Developments 44

3 Pseudospectral Methods **47**
3.1 Review of Solution Methods 47
3.2 Pseudospectral Methods 49

3.2.1 Toy Problem 51
3.3 Minimum Lap Time 59
3.4 Conclusions 61

4 Rolling Disk **65**
4.1 Introduction 65
4.2 Dynamical Model 67

5 Rolling Disk: Control Strategy and Results **73**
5.1 Lean angle control design 73
5.2 Receding horizon control 75
5.3 Simulation results 79
5.4 Conclusions and future work 79

A One Track Model: Motion Equation **87**
A.1 Lagrangian Motion Equation 87
A.2 The Model 89
A.2.1 Change of coordinates 91
A.2.2 The Whole Model 92

B Optimal Controller Matlab Implementation **95**
B.1 MSC.ADAMS 12.0 95
B.2 The Controller 96
B.2.1 Trajectory Generation 96
B.2.2 Model Linearization 98
B.2.3 Optimal Control Calculation 100
B.2.4 Control Application 100
B.3 Simulator Input Signals 100
B.3.1 Gearshift Mechanism 103
B.3.2 RPM Controller 106

Acknowledgements **111**

Bibliography **113**

List of Figures

1.1 Coordinate system. 6
1.2 (a) 2 Track Model, (b) Bicycle Model. 9
1.3 Tire model. 11
1.4 Relation between slip and generated forces. 13
1.5 Directional responses of neutral steer, under and over steer vehicle to a lateral force disturbance F. 18
1.6 Relation between steering angle and lateral speed. 20
1.7 Relation between steering angle and yaw rate ω. 20

2.1 Vehicle distances from reference. 24
2.2 Example of relation between the engine speed, the maximum traction couple T_M and the maximum braking couple T_m. . . . 26
2.3 Control scheme. 31
2.4 Straight test: speed reference. 33
2.5 Straight test: speed and steering angle. 33
2.6 Straight test: lateral acceleration. 34
2.7 Chicane test: path reference. 34
2.8 Low speed chicane test: speed and steering angle. 35
2.9 Low speed chicane test: lateral acceleration. 37
2.10 Medium speed chicane test: speed and steering angle. 37
2.11 Medium speed chicane test: wheels slip angles. 38
2.12 Medium speed chicane test: Differences between a well tuned controller and a worse tuned one. 38
2.13 Medium speed chicane test: Differences between two differently tuned controllers. 39
2.14 High speed chicane test: wheels slip angles. 39
2.15 Mixed path test: forward speed and steering angle. 41
2.16 Road used for the trial. 41
2.17 Vertical road profile. 42
2.18 Velocity reference profile. 42
2.19 Steering angle and forward speed signals. 43

2.20 Steering angle and forward speed signals in presence of uneven pavement. 44

3.1 Covector Mapping Theorem. 50
3.2 Admissible region. 52
3.3 Obstacle Avoidance. 53
3.4 Optimal results for the original problem: (a) Optimal Control,(b) Optimal State Trajectory, (c) Optimal Costate Trajectory,(d) Hamiltonian. 56
3.5 Demonstration of Bellman's optimality principle. 57
3.6 State Space and Optimal Trajectory. 58
3.7 Path Constraints. 60
3.8 Problem settings (a) and optimal results for lateral speed (b). 63
3.9 Optimal control input and optimal conditions. 63

4.1 View of the rolling disk with the configuration variables. . . . 68

5.1 Connecting contour. 77
5.2 Following of a circular path. 80
5.3 Following of a parabolic path. 81
5.4 Following of a chicane. 82

B.1 Reference scheme. 96
B.2 Centerline distances. 97
B.3 Reference couple calculation. 99
B.4 Optimal Controller. 101
B.5 ADAMS suspension analysis. 102
B.6 Gear change:clutch and accelerator inhibition signals. 104
B.7 Gearshift controller. 106
B.8 Clutch and throttle inhibitor signal generator. 107
B.9 RPM control signals. 108
B.10 Gearshift without RPM controller. 109
B.11 RPM controller Simulink scheme. 109

Introduction

Motivations of the work

This thesis work is about the application of optimal control techniques to mechanical systems, in order to achieve a path following.

In the first part of the work the problem of controlling a ground vehicle along a preassigned trajectory is addressed. This work, completed in collaboration with MSC.Software, is motivated by practical needs.

In the manufacturing field the competition on a global market requires to minimize the research and development cost for new products. Its development process can be divided in preliminary research and design, prototype creation, prototype test and revision. This process usually requires the production of a big number of intermediate pieces, that are a great source of time and money consumption. Some tools have been created to speed up and improve this process through the use of computers; this development technique goes under the name of "Virtual Prototype Development" (VPD).

In ground vehicle designing, the basic idea is to build a mathematical model (the virtual prototype) that captures significant aspects of the physical dynamics, allowing various performance analysis and aiding in design tradeoffs. Indeed with such a model, it can be possible to set and run an indefinite number of inexpensive and quick experiments (when compared with real ones) to improve safety and maneuverability of the design. A mathematical model is also useful to predict how a design variation or a new component might affect the stability and performance of the vehicle.

Beside this class of experiments, which can be classified as "statical tests", there are some others that should be performed to analyze the dynamical behavior of the vehicle, such to follow a path of prescribed geometry following a velocity specification. Open loop simulations are sufficient for and commonly used in handling analysis of automobiles, while are not suitable for

high demanding maneuvers. These trials are commonly performed by a real driver on a physical prototype, with consequent waste of money and time, and with the major drawback of the lack of test repeatability.

To properly simulate these complex maneuvers, feedback should be employed as part of a virtual rider to provide the steering and throttle/brake inputs necessary to accomplish the task. The development of such a driver, which is in fact an interesting application of control algorithms, is the main aim of this work.

The control scheme is based on Model Predictive Control techniques (MPC). Model Predictive Control is widely adopted in industry as an effective means to deal with large multivariable constrained control problems. The main idea of MPC is to choose the control action by solving on line an optimal control problem. This aims at minimizing a performance criterion over a future horizon.

The choice of a receding horizon strategy was suggested by observing the human driving activity. While driving, people know, through their sight, a portion of the track ahead and use this information to judge how to act, how much to steer the vehicle in order to properly accomplish the desired maneuver. Moreover, consciously or not, the judgment is hardly based on the mental model of the dynamics of the vehicle. The more "accurate" it is, the best the maneuver is performed. So people do not look on the road just below them, as a causal strategy would do. Therefore, appealing to a noncausal control law appears to be the natural choice for the problem and that was confirmed during the experiments made.

In previous works, (see [22]) the control action is computed using desired speed and curvature at current time. Therefore, it is based only on the information related to the vehicle current position on the path and that is why its applicability is restricted to a certain set of possible maneuvers, there working nicely.

As part of the control scheme development, there is the requirement of obtaining a feasible trajectory on the plane and a velocity profile to be followed. To address this problem, firstly solved through the use of the proper tool of MSC.ADAMS, the simulation environment, the "minimum lap time" problem has been studied and solved through the use of pseudospectral methods, which has given some brilliant preliminary results.

The second part of the work is about controlling a class of underactuated mechanical systems to develop a general control scheme that exploits symmetries and other common characteristics.

Underactuated systems are mechanical control systems with fewer controls than the number of configuration variables. Control of underactuated systems is currently an active field of research due to their broad applications in Robotics, Aerospace Vehicles, and Marine Vehicles. The examples of underactuated systems include flexible-link robots, mobile robots, walking robots, robots on mobile platforms, cars, locomotive systems, snake-type and swimming robots, acrobatic robots, aircraft, spacecraft, helicopters, satellites, surface vessels, and underwater vehicles. Based on recent surveys, control of general underactuated systems is a major open problem ([35] and [36]).

Almost all real-life mechanical systems possess kinetic symmetry properties, i.e. their kinetic energy does not depend on a subset of configuration variables called external variables. In this work, we exploit such symmetry properties as a means of reducing the complexity of control design for underactuated systems.

We provide as example the rolling disk system, which is well known and studied in literature. Differently from the previous work we consider only a single control input for the systems and states the problem of path following.

Main contributions

As regards the study of nonlinear control systems, the contributions are twofold. First, we develop a strategy that combines longitudinal and lateral control for vehicle guidance, which is more similar to human driving than the previous ones, which decouple the two dynamics. Second, we provide a preliminary characterization of the optimal solution for the minimum lap time problem, that takes in account the vehicle dynamics and not only its steady-state behavior.

As regards the control of underactuated systems, we provide a new setting for the "rolling disk" problem and develop a control scheme for trajectory tracking purpose, that allows the system to follow a preassigned path with bounded error. This control strategy can be easily applied to a wider class of systems, that present planar symmetries and nonholonomic constraints.

Thesis outline

In Chapter 1 we introduce the mathematical model of the vehicle that will be used in the next chapters for control purpose. The main hypotheses made to obtain a computationally tractable model with a good representation of the physical system are discussed.

In Chapter 2 we provide a new strategy to solve the trajectory tracking problem. We start providing a general definition of the problem addressed and the solution here developed. The experimental results are then presented.

Chapter 3 is a review of pseudospectral methods for the numerical solution of differential equations and their application to the solution of control problems. The application of these methods to the solution to the "minimum lap time" problem is then presented.

In Chapter 4 the problem of controlling a rolling disk along a trajectory on the plane is addressed; this system has been chosen to represent a more general system, in the class of underactuated mechanical systems that present symmetries on the plane. The model is derived using a Lagrangian approach.

In Chapter 5 the developed control strategy is detailed and the results of its application are presented, along with future improvements.

Chapter 1

Vehicle Models

The first step to solve the control problem previously stated is to find a mathematical model that presents the main characteristics of the real vehicle, while being computationally tractable. Next sections are dedicated to explain the developed model and its salient points.

1.1 Vehicle Analysis

A ground vehicle typically consists of the main body (chassis, cabin, engine, transmission) linked to four wheels via the suspension system. Except from translation and yawing motion during travel, the suspended body performs pitching, rolling and vertical translation motions. The vehicle interacts with the environment through tire friction forces, which allow the vehicle to accelerate, decelerate and steer, as well as aerodynamic drag and lift forces generated due to relative motion of the body and the atmosphere. Steering of the vehicle is generated typically by the two front wheels, although it is not unusual for steering to be generated by the rear or all four wheels. The power is transmitted from the engine to the wheels through the transmission system.

To control the vehicle the driver acts typically on:

- Steering angle;
- Throttle;
- Brakes;

- Gear;
- Clutch.

These parameters will be the ones considered as control inputs in the control algorithm later described.

1.2 Coordinate System

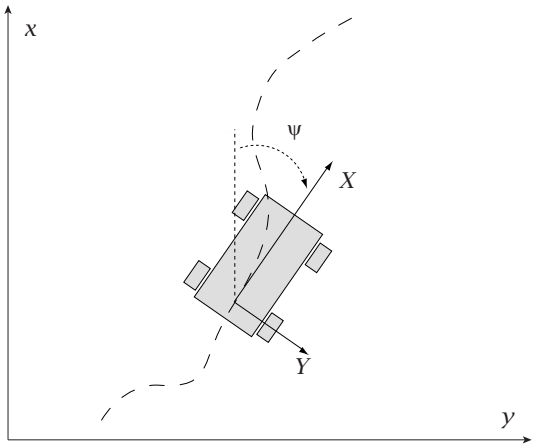


Figure 1.1: Coordinate system.

It is necessary to fix a reference frame in order to describe the system. We consider the frame $\Sigma_1 = (X, Y)$ attached to the vehicle, further called *body frame*, which origin is situated on the contact point of the rear wheel, and which orientation is consistent with the Society of Automative Engineers (SAE) convention[1].

To analyze the vehicle motion, it is necessary to consider a second reference frame $\Sigma_2 = (x, y)$, inertial and oriented coherently to the first one and z axis pointing down; in this way clockwise rotation are positive. This frame will be called *world frame*.

[1]SAE convention fixes a frame with the x axis in the direction of motion, the z axis pointing downward and the y one accordingly oriented.

Vehicle position can be unambiguously[2] determined by the triple (x, y, ψ), where x and y are the planar coordinates of the rear wheel contact point, while ψ is the yaw angle, that is the angle between the world frame x axis and the body frame x axis, which represents the vehicle direction. These coordinates can be then chosen as *generalized coordinates* for system position description.

From the previous quantities, we can straightforwardly obtain the *generalized velocity* $\left(\dot{x}, \dot{y}, \dot{\psi}\right)$.

Calling V_T e $V_\perp$ the longitudinal and lateral vehicle velocities, that are the velocities along the X and Y axes of Σ_1, the following equivalence fits:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} V_T \\ V_\perp \end{bmatrix} \triangleq R_\psi \begin{bmatrix} V_T \\ V_\perp \end{bmatrix}$$

R_ψ is a rotational matrix, which is always invertible, therefore the triple $(V_T, V_\perp, \psi)$ is a generalized velocities set equivalent to the first one.

1.2.1 Motion Equation

To obtain a mathematical handleable model for the vehicle, some assumptions should be made. In the following list the main ones are summarized:

- We consider no rear wheel steering. For further simplification we can assume equal steering angles for the front left and front right wheels.
- We assume that the center of gravity (C.G.) is located on the longitudinal vehicle axis. The longitudinal position of the C.G. is a parameter of our model. The static vertical load though is equally distributed to the left and right wheels.
- The rolling motion of the suspended mass is neglected. In addition we assume that friction forces and velocities appearing on wheels of the same axle are identical. Taking into consideration all the assumptions above, the model is now equivalent to a 2-wheel vehicle, classically called *Bicycle Model* or *1-Track Model*, as in figure 1.2, with one front and one rear wheel on the longitudinal x body axis of the vehicle. The friction forces of each wheel of the half-car model is equal to the sum of friction forces of the wheels of the same axle.

[2]This happens because we neglect roll and pitch dynamics,thanks to the non deformable suspensions hypothesis.

- The mass of the wheels is neglected. The vehicle is modelled as a single rigid body. However, equations to describe the rotation of each wheel are necessary in order to compute the angular speed of each wheel. This allows the computation of the relative velocity of each wheel with the ground and thus the friction forces generated by each tire by use of a tire friction model.
- The aligning moment of the tires is neglected for simplicity as it is common in the literature [[40]], [[11]], [[10]] for trajectory following and optimization applications.
- Aerodynamic lift is neglected.
- Engine, transmission and brakes dynamics are neglected.
- We assume that the longitudinal control of the vehicle is given by two independent torques on each wheel. However, the controls in a real car are one acceleration pedal and one brake pedal. Thus the input torques of each wheel are coupled. On the other hand, several race driving techniques, such as "hand-brake cornering" and "left foot braking" allow some independence on the longitudinal control of each wheel.

Motion equation can be obtained through the Lagrangian approach[3]; the resultant model is:

$$\begin{bmatrix} \dot{v}_T \\ \dot{v}_\perp \\ \ddot{\psi} \end{bmatrix} = M^{-1} \{\tau - C\}$$

where:

- M is the mass matrix:

$$M = \begin{pmatrix} m & 0 & 0 \\ 0 & m & mb \\ 0 & mb & mb^2 + I_{zz} \end{pmatrix}$$

- τ is the generalized forces vector:

$$\tau = \begin{pmatrix} F_\top^f \cos\delta - F_\perp^f \sin\delta + F_\top^r - F_A \\ F_\top^f \sin\delta + F_\perp^f \cos\delta + F_\perp^r \\ p(F_\top^f \sin\delta + F_\perp^f \cos\delta) \end{pmatrix}$$

[3]for further details see appenxix A.

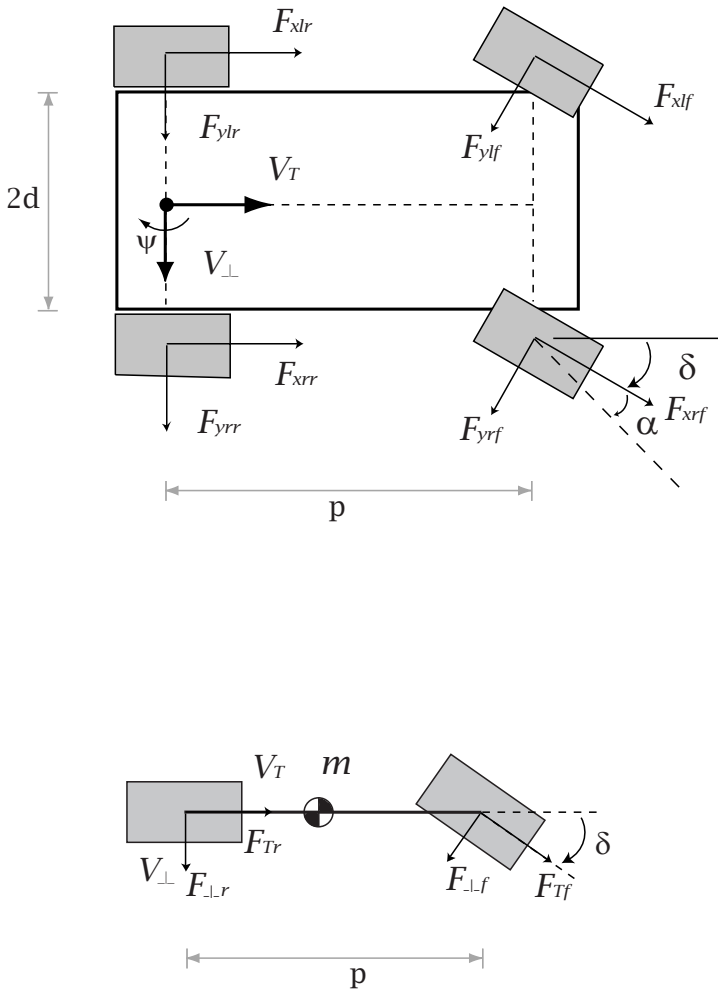


Figure 1.2: (a) 2 Track Model, (b) Bicycle Model.

- C is the Coriolis vector:

$$C = \begin{pmatrix} -mv_{\perp}\dot{\psi} - mb\dot{\psi}^2 \\ mv_{\top}\dot{\psi} \\ mbv_{\top}\dot{\psi} \end{pmatrix}$$

- m is the vehicle mass;
- p is the wheelbase, that is the distance between the front and the rear contact point;
- b is the distance between the rear contact point and the center of mass;
- I_{zz} is the vehicle inertia with respect to the vertical axis;
- $F_{\top}^{r}$ and $F_{\top}^{f}$ are the longitudinal forces:

$$\begin{aligned} F_{\top}^{r} &= f(\alpha_r, \kappa_r, N_r) \\ F_{\top}^{f} &= f(\alpha_f, \kappa_f, N_f) \end{aligned}$$

- $F_{\perp}^{r}$ and $F_{\perp}^{f}$ are the lateral forces[4]:

$$\begin{aligned} F_{\perp}^{r} &= f(\alpha_r, \kappa_r, N_r) \\ F_{\perp}^{f} &= f(\alpha_f, \kappa_f, N_f) \end{aligned}$$

- N_r and N_f are the normal forces;
- F_A is the aerodynamic drag, given by $F_A = \frac{1}{2}\rho C_D A v_T^2$, where ρ is the air density, C_D is the drag coefficient and A is the drag area.

1.3 Tire Model

The forces and torques developed by the pneumatic tire affect the vehicle in a variety of ways. The tires support the vehicle weight, and any other vertical forces developed (for example due to aerodynamic effects or road banking). The interaction between the tires and the road supply the tractive, braking and cornering forces for maneuvering. The tires also supply the forces used for controlling and stabilizing the vehicle and for resisting external disturbances from road and wind.

[4]Longitudinal and lateral forces are generated by the interaction of the pneumatic tires with the road.

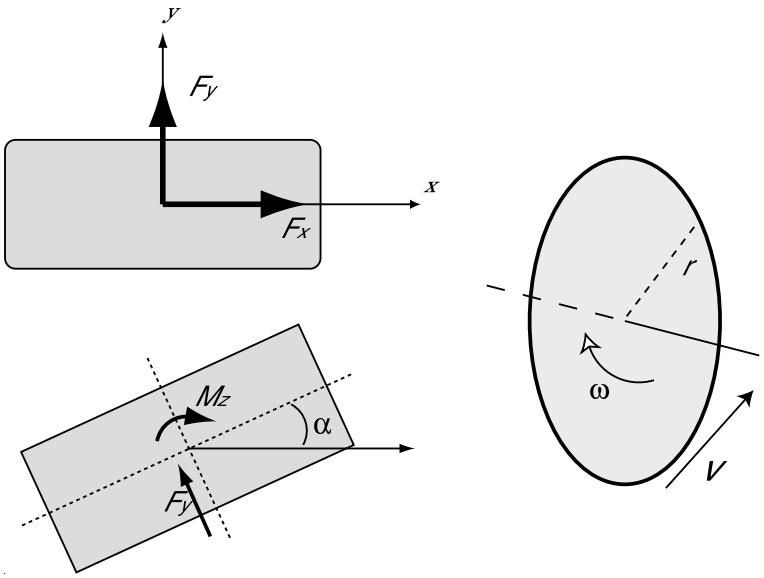


Figure 1.3: Tire model.

These forces are generated as consequence of a wheel sliding, according to the model developed by H.B. Pacejka in [14] and [32]. When the vehicle is moving forward, the slip has two components, that can be defined as follow:

- α_r and α_f are the lateral slip;

$$\begin{aligned} \alpha_r &= atan(\tfrac{v_\perp}{v_\top}) \\ \alpha_f &= atan(\tfrac{v_\perp + p\dot{\psi}}{v_\top}) - \delta \end{aligned}$$

- κ_r and κ_f are the longitudinal slip;

$$\begin{aligned} \kappa_r &= \tfrac{-\dot{\theta}_r R_r - v_\top}{v_\top} \\ \kappa_f &= \tfrac{-\dot{\theta}_f R_f - v_\top \cos\delta - v_\perp \sin\delta}{v_\top \cos\delta + v_\perp \sin\delta} \end{aligned}$$

where:

κ : tire *longitudinal slip* [%];

α : tire *slip angle* [rad];

v_T :vehicle longitudinal velocity [m/s];

$v_\perp$: vehicle lateral velocity $[m/s]$;

$\omega_i = \dot{\theta}_i, i = R, F$: tire angular velocity $[rad/s]$;

$R_i, i = R, F$: wheel radius $[m]$;

δ : steering angle $[rad]$;

$\dot{\psi}$: vehicle yaw rate $[rad/s]$;

p : vehicle wheelbase $[m]$.

Longitudinal and lateral forces can be obtained, knowing the slip values, using the following equations, which are known as ***Pacejka Magic Formula***:

$$F_x = D_1 \sin\left(C_1 \arctan\left(B_1\kappa' - E_1\left(B_1\kappa' - \arctan\left(B_1\kappa'\right)\right)\right)\right) + S_{v1} \quad (1.1)$$
$$F_y = D_2 \sin\left(C_2 \arctan\left(B_2\alpha' - E_2\left(B_2\alpha' - \arctan\left(B_2\alpha'\right)\right)\right)\right) + S_{v2} \quad (1.2)$$

The coefficients $B_i, C_i, D_i, E_i, S_{hi}, S_{vi}$ determine the curve shape and are experimentally determined through measurement on the real tire. The slip there used are given by: $\kappa' = \kappa + S_{h1}$, $\alpha' = \alpha + S_{h2}$ and $X' = \alpha + S_{h3}$. See [14] and [32] for further details.

Figure 1.4 shows the forces as a function of the tire slip. The relation between slip and generated force is clearly non-linear; the force presents a peak for a particular value κ_M of the longitudinal slip (or similarly for a value α_M of the lateral slip angle) and then it decreases. We should take in account of this trend in the design of the control law for the system, because exceeding the peak values determines a reduction of the generated force while increasing the wheel angular velocity (for longitudinal force) or the steering angle (for lateral one); in this case increasing the engine speed(i.e. accelerating) leads to a reduction of the tractive force, while increasing the steering command decreases the curvature value of the trajectory.

1.3.1 Combined Pacejka Formula

The F_x and F_y values previously computed can be reached only if the vehicle is in pure accelerating maneuver or in pure cornering one, when only one component of the wheel sliding is present. Most of the time both the longitudinal slip ratio and the lateral slip angle are non-zero; it is then necessary

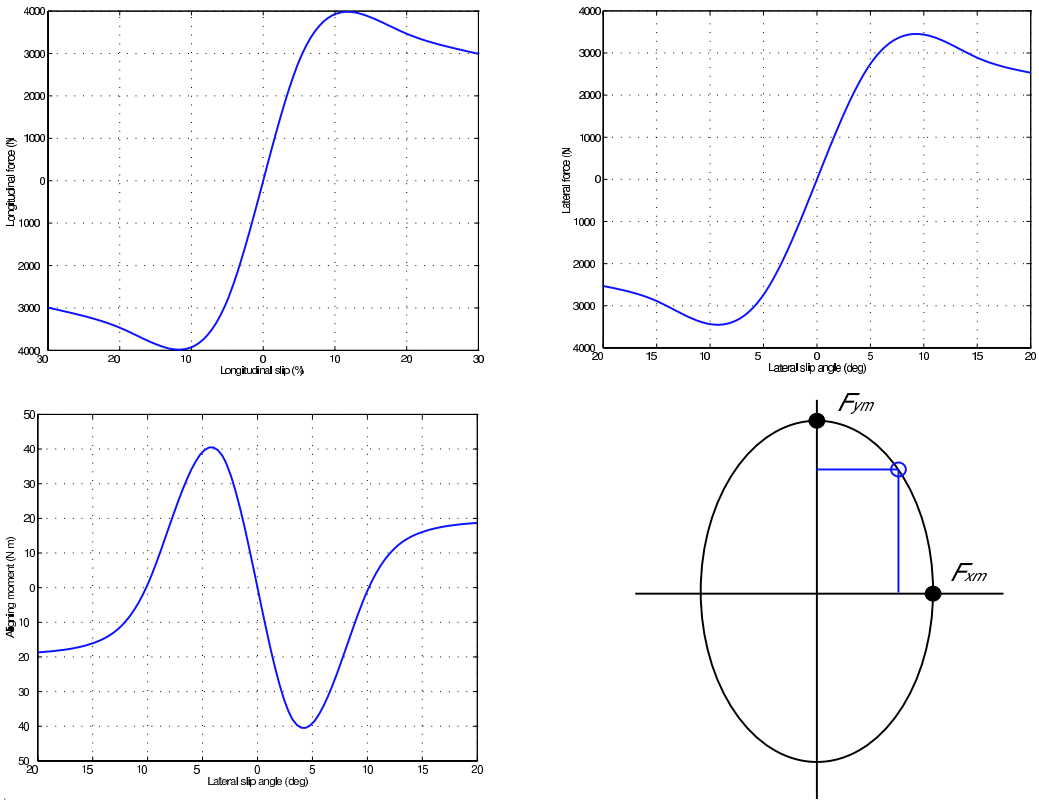


Figure 1.4: Relation between slip and generated forces.

to further investigate what happens when both the tractive and cornering forces are developed.

The previous model can be enriched through the *tractive ellipse* idea (See figure 1.4).

The highest longitudinal and lateral forces that can be separately computed through the Pacejka Magic Formula determines the ellipse length and width. Chosen a point on the ellipse contour, its distance from the origin represents the overall force that can be generated by the tire, while its projections on the axes are the greatest longitudinal and lateral forces. Their expressions can be obtained from the previous ones (i.e. 1.1 1.2), introducing opportune scaling factors G_x e G_y :

$$\begin{aligned} F_x &= F_{x0} G_{x\alpha}(\alpha, \kappa, F_z) \\ F_y &= F_{y0} G_{y\kappa}(\alpha, \kappa, \gamma, F_z) + S_{Vy} \end{aligned}$$

The analytical expression of the functions G_x e G_y can be found in [29].

1.3.2 Linear Approximation

For small values of longitudinal and lateral slip, the function determined above can be approximated with a linear one, with a direct dependence of the force from the slip value. The static slip coefficient that realizes this function is calculated as the slope of the function graph at the origin. This approximation, that will be extensively used in the following chapter, will be as much accurate as the control will be able to maintain the vehicle in the region with small slip values.

The expression of the forces in this hypothesis will then be:

- Longitudinal forces:

$$\begin{aligned} F_T^r &= K_\kappa^r \kappa_r N_r; \\ F_T^f &= K_\kappa^f \kappa_f N_f; \end{aligned}$$

- Lateral forces:

$$\begin{aligned} F_\perp^r &= -K_\alpha^r \alpha_r N_r \\ F_\perp^f &= -K_\alpha^f \alpha_f N_f \end{aligned}$$

1.4 Steady-State Handling Characteristics

The key step to understand the vehicle directional behavior is to analyze the vehicle steady-state characteristics. The SAE definition of steady state is:

"Steady-state exists when periodic (or constant) vehicle responses to periodic (or constant) control and/or disturbances inputs do not change over an arbitrary long time. The motion responses in steady-state are referred to as steady-state responses".

Generally steady-state is said to be an equilibrium condition. In some physical terms, assuming a vehicle in an equilibrium condition in v_y and ω means that the vehicle is negotiating a turn with constant radius and with a constant sideslip angle β, since v_y can be different from zero.

We neglect longitudinal slip in the following section, because only the vehicle cornering behavior is here considered.

Motion equation will then be:

$$\begin{cases} m\dot{v_x} = mv_y\omega + T - F_f \sin\delta \\ m\dot{v_y} = -mv_x\omega + F_r + F_f \cos\delta \\ I_z\dot{\omega} = aF_f \cos\delta - bF_r \end{cases}$$

We define $\beta = arctan(\frac{v_y}{v_x})$ and $v = \sqrt{(v_x^2 + v_y^2)}$, that leads to the following relation:

$$\begin{cases} v_x = v\cos\beta \\ v_y = v\sin\beta \end{cases}$$

We can then rewrite motion equation as a function of $\bar{\xi} = [v, \beta, \omega]'$ obtaining:

$$\begin{cases} \dot{v} = \frac{T\cos\beta}{m} + \frac{F_r\sin\beta}{m} + \frac{F_f\sin(\beta-\delta)}{m} \\ \dot{\beta} = \frac{F_r\cos\beta + F_f\cos(\beta-\delta) - T\sin\beta}{mv} - \omega \\ \dot{\omega} = \frac{aF_f\cos\delta - bF_r}{I_z} \end{cases}$$

As a second hypothesis we consider $v = constant \Rightarrow \dot{v} \equiv 0$. We can then obtain an explicit form for T:

$$T = -\frac{F_r\sin\beta + F_f\sin(\beta-\delta)}{\cos\beta}$$

The reduced system has $x = [\beta\ \omega]'$ as the state vector and $u = \delta$ as the control input, and the equation of motion becomes:

$$\begin{cases} \dot{\beta} = \frac{F_r + F_f \cos\delta}{mv\cos\beta} - \omega \\ \dot{\omega} = \frac{aF_f \cos\delta - bF_r}{I_z} \end{cases}$$

We want to study the system equilibria as a function of lateral acceleration $a_\perp = v\omega$. Due to the second hypothesis made (i.e. $v = constant$), it is equivalent to analyze the vehicle behavior varying ω.

The constraint $\omega - \bar{\omega} = 0$ is then considered.

After fixing a value for ω, it is possible to obtain the expression of the rear wheel force as a function of the front one:

$$\dot{\omega} = 0 \Rightarrow F_r = \frac{a}{b} F_f \cos\delta$$

There is only one equation left:

$$\dot{\beta} = \frac{(a+b)F_f \cos\delta}{mbv\cos\beta} - \omega = \tilde{f}(\beta, \delta)$$

And the equilibria analysis corresponds to find the solutions (β_e, δ_e) that satisfy $\tilde{f}(\beta, \delta) = 0$.

Linear Steady-State Handling Characteristics

Firstly we consider the linearization of the model around the operating condition $v_y = 0, \omega = 0$ and $\delta = 0$; the linearized model is

$$\begin{bmatrix} \dot{\omega} \\ \dot{\beta} \end{bmatrix} = A \begin{bmatrix} \omega \\ \beta \end{bmatrix} + B\delta$$

where:

$$A = \begin{bmatrix} a11 & a12 \\ a21 & a22 \end{bmatrix}$$
$$B = \begin{bmatrix} b1 \\ b2 \end{bmatrix}$$

The path curvature ρ corresponding to a given equilibrium condition (in ω and v_y) is

$$\sigma = \frac{1}{R} = \frac{\omega}{V} \simeq \frac{\omega}{v_x}$$

Inserting this expression in the motion equation above and introducing the stability factor K or the understeer gradient K_{us}

$$K = \frac{m}{p^2}\left(\frac{b}{K_f} - \frac{p-b}{K_r}\right)$$
$$K_{us} = \frac{mg}{p}\left(\frac{b}{K_f} - \frac{p-b}{K_r}\right) = Kgp$$

we obtain an expression for path curvature as a function of vehicle parameters:

$$\rho = \frac{1}{p}\frac{1}{1+KV^2}\delta$$

At low speed, the curvature value is $\rho = \delta/p$ and coincides with the kinematic condition. The term $1 + KV^2$ appears as a corrective factor of the kinematic relation versus longitudinal speed. The relationship between steering angles, turning radius and lateral acceleration can be recast as follows:

$$\delta = \frac{p}{R} + K_{us}\frac{\omega V}{g}$$

Depending on the sign of K_{us} the vehicle can be defined:

neutral when $K_{us} = 0$. In this case the steering angle δ required to negotiate a given curve is independent from speed. For a two wheel neutral steer vehicle, when it is accelerated in a constant radius turn, the driver should maintain the same steering wheel command. The neutral steer behavior can be also expressed with respect to lateral external force disturbances. When a neutral steer car moving along a straight path is subjected to a side force acting on the center of gravity, the vehicle follows a straight path as in figure 1.5. Equal slip angles $\alpha_r$and α_r are developed at the front and rear tyres.

understeer when $K_{us} > 0$. In this case the curvature radius tends to decrease as the speed tends to increase and the driver should increase the steering wheel angle in order to maintain the same turning radius.When a side force acts on the center of gravity of an understeer vehicle originally moving on a straight path, a yaw motion is initiated and the vehicle turns away from the side force, as shown in figure 1.5. For an understeer vehicle, a characteristic speed $\bar{V}$ can be defined. It is equal to the speed at which the steer angle required to negotiate a turn is twice of the one required by a neutral steer vehicle. The characteristic speed is

$$\bar{V} = \sqrt{\frac{gp}{K_{us}}} = \sqrt{\frac{1}{K}}$$

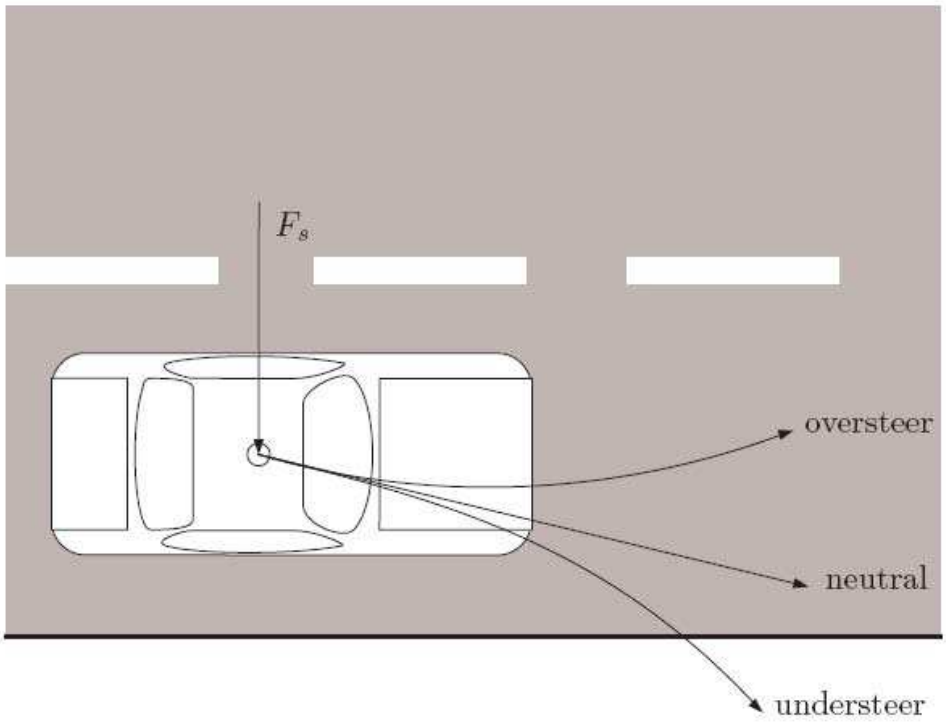


Figure 1.5: Directional responses of neutral steer, under and over steer vehicle to a lateral force disturbance F.

In this case the front sideslip angle α_f is greater than the rear one α_r.

oversteer when $K_{us} < 0$. In this case the steering angle required for negotiating a turn decreases as the longitudinal speed increases. For the oversteer vehicle a critical speed can be defined as

$$\bar{V} = \sqrt{\frac{gp}{-K_{us}}} = \sqrt{\frac{1}{-K}}$$

This critical speed represents the speed above which an oversteer vehicle has a directional instability. When a side force acts at the center of gravity of an oversteer vehicle which is moving along a rectilinear path, a yaw motion tends to turn the vehicle into the side force. In this case the rear tire sideslip angle α_r becomes greater than the front one α_f .

Nonlinear Steady-State Handling Characteristics

The characteristics previously pointed out hold until the linearized model is valid, i.e. for small turning radius or for low lateral acceleration. In this case, linearized tyre lateral forces can be assumed proportional to the tire sideslip angles α_r and α_f. At medium and high lateral accelerations, this assumption is not valid and these forces cannot be linearly modeled. In this case the nonlinear relationship between the sideslip angles and the lateral forces, modeled by the Magic Formula, is taken into account.

The model, when the steering angle is zero, has three equilibrium points. The stable one is $v_x = V$, $v_y = 0$, $\omega = 0$ which corresponds to the longitudinal uniform motion with constant speed. The two additional equilibrium points are unstable. Increasing the front steering angle, the stable equilibrium point tends to an unstable equilibrium point. For a certain value of the front steering angle, the stable equilibrium point coincides with the unstable one. This is the so-called *saddle-node bifurcation*. Hence, for front steering angles greater than the critical value ($\delta = 0.0285[rad]$), the vehicle has only an unstable equilibrium point and is intrinsically unstable. Plots of the yaw rate steady-state and lateral values versus steering angle are shown in figures 1.6 and 1.7.

In the previous section, the definitions of neutral steer, understeer and oversteer vehicle were given for a vehicle with linearized tires. Now, the under-over steering concept is not related with the car but with the specific equilibrium condition. In other words, the same vehicle can be underteered or oversteered depending on the longitudinal speed and the steering angle.

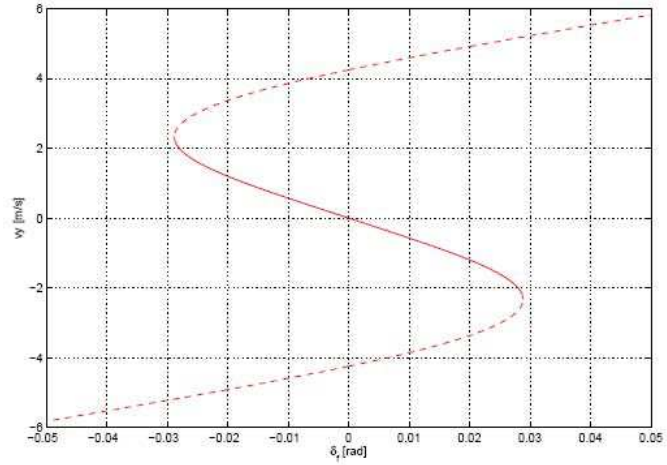


Figure 1.6: Relation between steering angle and lateral speed.

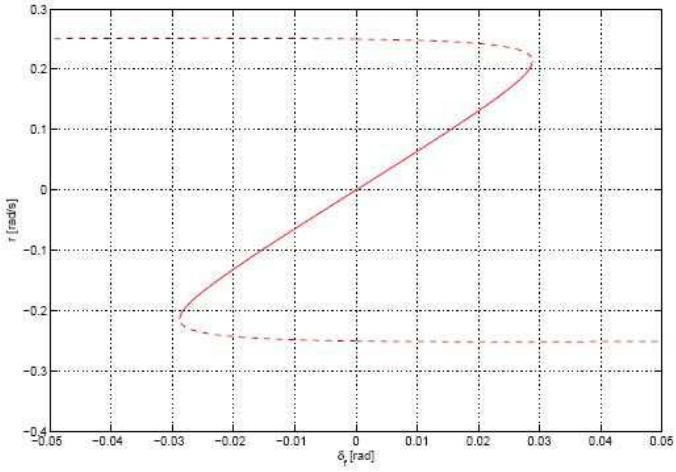


Figure 1.7: Relation between steering angle and yaw rate ω.

According to [26], the vehicle behavior can be defined as neutral when it satisfies the following relation

$$\frac{d}{dx}\left(\delta - \frac{p}{R}\right) = 0$$

where x is the parameter which control the considered manoeuvre. In fact, a vehicle can be considered neutral if the variation of δ required for the considered manoeuvre is equal to the required variation of the term p/R (also called Ackermann angle). Similarly, a vehicle can be defined understeering if such derivative has positive sign, oversteering otherwise.

Chapter 2

Optimal Controller

Following a given trajectory in the plane and a speed profile is the control objective of this work.

In order to control the vehicle, we can use only the steering couple and the accelerator/brake subsystem as input (clutch and gearshift effects will be neglected by now). The first one acts on the lateral dynamics, while the other two on the longitudinal dynamics, allowing speed changes.

Most of the classical solutions decouple the problem of controlling the vehicle in two separate sub-problems; the first one is about trajectory tracking, while the other is about speed profile tracking.

On the other side our control strategy is about coupling the two dynamics in order to obtain a driver behavior more similar to the human one. As every driver knows, human driving strategy is unique and use both the inputs to obtain the result. On the next sections there will be the description of our control strategy.

2.1 Lateral Control

One of the objective of this work is to make the vehicle following a preassigned trajectory on the plane. This problem is classically classified as "*path following problem*". The common characteristic of this class of problems is the presence of a trajectory given as a sequence of points in the plane without any time dependence (this is the difference from tracking problems, where the time dependence is explicit).

2.1.1 Path Following

Lateral control has to follow the preassigned trajectory in robust manner and real-time. The implemented strategy is based on the choice, at each step, of a "connecting contour" that allows the vehicle to reach the reference trajectory.

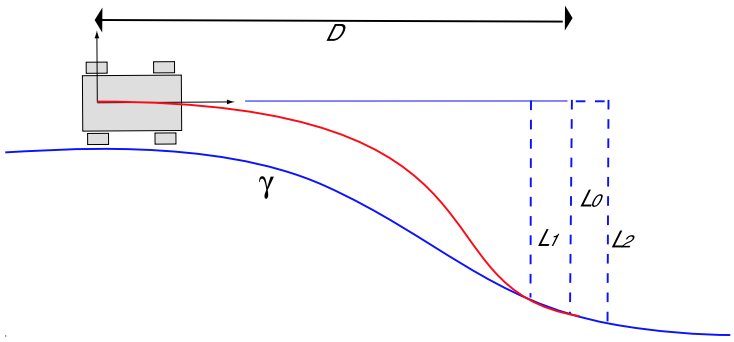


Figure 2.1: Vehicle distances from reference.

In figure 2.1 γ is the reference trajectory defined in an inertial frame $\Sigma_1 = (X, Y)$. This trajectory is a function of the curved abscissa s:

$$\gamma = \{X(s), Y(s), s \in [s_0, s_1] \subseteq \mathbb{R}\}.$$

Let γ be a continuous function, differentiable and with continuous first derivative: $\gamma \in C^1(\mathbb{R}^2)$.

Let $\Sigma_2 = (x, y)$ be a frame attached on the vehicle (body frame). In this frame we can locally represent the curve γ as a function dependent from only the first coordinate x and the time: $y = \gamma(x, t)$, with $x \in [0, D]$, being D the distance (which is a project parameter) where theoretically the contour is connected to the reference path. This connection will be found imposing the equality of the first derivative of the trajectory and the connecting contour at $y(D)$.

2.1.2 Connecting Contour Computation

The connecting contour has to be a feasible trajectory for the vehicle, so it has to satisfy some boundary conditions. Constraints are imposed at the initial condition ($x = 0$) and the final condition $x = D$ and can be expressed as:

$$\begin{bmatrix} \gamma_c(0,t) \\ \frac{\partial \gamma_c}{\partial t}(0,t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{2.1}$$

$$\begin{bmatrix} \gamma_c(D,t) \\ \frac{\partial \gamma_c}{\partial t}(D,t) \end{bmatrix} = \begin{bmatrix} \gamma(D,t) \\ \frac{\partial \gamma}{\partial t}(D,t) \end{bmatrix}. \tag{2.2}$$

Equation 2.1 is equivalent to impose that the connecting contour starts exactly in the origin of the body frame and it's tangent to the x^1 axis, while equation 2.2 states that the connecting contour connects to the curve at distance D, both curve having the same derivative in $x = D$. In this way the connection is granted to be smooth.

The class of polynomial functions has be chosen to represent the connecting contour; this class has the global curvature minimization property. Equations 2.1 and 2.2 define 4 independent constraints, so a cubic curve will suffice:

$$\gamma_c(x,t) = a_3(t)x^3 + a_2(t)x^2 + a_1(t)x + a_0.$$

The parameters a_i are found with a straightforward calculation; equation 2.1 states that $a_0 = 0$ and $a_1 = 0$, while the others parameters are found solving the following system:

$$\begin{bmatrix} D^2 & D^3 \\ 2D & 3D^2 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \gamma(D,t) \\ \frac{\partial \gamma}{\partial t}(D,t) \end{bmatrix}. \tag{2.3}$$

2.2 Longitudinal Control

One of the fundamental objective of the controller is to follow a preassigned velocity profile. To obtain this result we can only act on throttle and brake commands. The gearshift can be considered apart, because its only purpose is to allow the engine to work in the right regimen.

The implemented algorithm determines the right vehicle acceleration in order to follow the speed profile at each step; we then have to determine the vehicle inputs, that are related to this value in a non-linear manner through the engine.

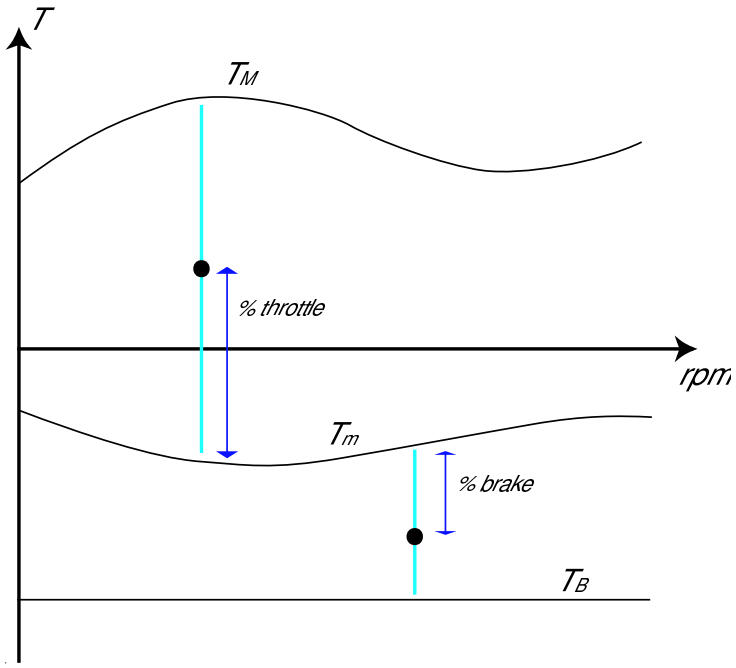


Figure 2.2: Example of relation between the engine speed, the maximum traction couple T_M and the maximum braking couple T_m.

Acceleration Couple to Throttle/Brake Conversion

The engine model is characterized by two parameters T_M and T_m, which values depend on its speed, that are the maximum traction couple and the maximum braking couple; in figure 2.2 there is an example of the relation between these quantities and the engine revolution speed.

Let T be the required engine couple (which is directly related to the whole vehicle acceleration); if its value is between T_M and T_m the engine will be able to give sufficient traction to the vehicle, then the right throttle value is the percentage of the required couple respect to the maximum value.

$$throttle = \frac{T - T_m\left(rpm\right)}{T_M\left(rpm\right) - T_m\left(rpm\right)} \times 100 \qquad \text{if} \quad T_m\left(rpm\right) \leq T \leq T_M\left(rpm\right).$$

If the required couple is greater than the maximum value, then it is

[1]The x axis of Σ_2 is coincident to the forward velocity direction of the vehicle, in the non-slipping hypothesis.

saturated and the maximum throttle value is applied. When the couple is lower than the minimum value T_m it is necessary to act on the braking system to ensure the velocity profile following.

The vehicle is also characterized by a further quantity T_B, which represents the maximum value of the braking couple. This value is supposed to be constant with respect to the engine and vehicle speeds. The braking couple applied is then:

$$brake = \frac{T - T_m\left(rpm\right)}{T_{dist}} \times 100 \qquad \text{if} \quad T_B \leq T \leq T_m\left(rpm\right),$$

while when the required couple is smaller, it will be saturated and the maximum braking couple will be applied.

In the end the whole conversion scheme is:

$$\begin{bmatrix} throttle \\ brake \end{bmatrix} = \begin{cases} \begin{bmatrix} 100 \\ 0 \end{bmatrix} & \text{if} \quad T \geq T_M\left(rpm\right); \\ \begin{bmatrix} \frac{T - T_m(rpm)}{T_M(rpm) - T_m(rpm)} \times 100 \\ 0 \end{bmatrix} & \text{if} \quad T_m\left(rpm\right) \leq T < T_M\left(rpm\right); \\ \begin{bmatrix} 0 \\ \frac{T - T_m(rpm)}{T_{dist}} \times 100 \end{bmatrix} & \text{if} \quad T_B \leq T < T_m\left(rpm\right); \\ \begin{bmatrix} 0 \\ 100 \end{bmatrix} & \text{if} \quad T < T_B. \end{cases}$$

2.3 The Controller

The developed controller is based on Model Predictive Control (MPC). This control algorithm solves a finite horizon optimization problem in line. A receding horizon approach is used to find the solution, which can be summarized as:

- an open loop control problem is solved at $t = k$ for the current state vector $x(k)$;
- the first step of the optimal sequence computed is applied to the system;

- the procedure is iterated at $t = k + 1$ for $x(k + 1)$.

This solution is then converted in a closed loop strategy taking in account the current value of the state vector (or an estimated one, obtained through an observer, if it's not directly measurable).

Let us consider the following system:

$$\begin{cases} x(l+1) = f\left(x(l), u(l)\right) \\ x(k) = x_k \end{cases}.$$

MPC control for (x, k) is obtained solving an optimization problem:

$$\mathcal{P}_N(x): \qquad V_N^o(x) = \min_{U \in \mathcal{U}_N} V_N(x, U),$$

where:

$$U = \{u(k), u(k+1), \ldots, u(k+N-1)\}$$

$$V_N(x, U) = \sum_{l=k}^{k+N-1} L(x(l), u(l)) + F(x(k+N))$$

and $\mathcal{U}_N$is the set of input U that satisfy the constraints in $[k, k + N - 1]$

$$u(l) \in \mathbb{U} \qquad l = k, \ldots, k+N-1)$$

$$x(l) \in \mathbb{X} \qquad l = k, \ldots, k+N)$$

as well as the final state constraint:

$$x(k+N) \in Ws$$

where $\mathbb{U} \subset \mathbb{R}^m$ is a closed and compact set, $\mathbb{X} \subset \mathbb{R}^n$ is a closed and convex set and W is chosen in order to assure stability.

In this case the cost function is time-invariant, therefore we can consider, without loss of generality, $k = 0$ in the open loop problem formulation; applying standard resolution methods we can obtain the input sequence that minimizes the chosen cost index. This sequence can be written as:

$$U_N^o = \{u(0), u(1), \ldots, u(N-1)\}$$

We apply only the first element of the sequence above:

$$u = u_x^o(0) \tag{2.4}$$

At the following step the finite horizon optimal solution is recalculated, taking in account the new state values; this procedure is then iterated in order to generate the control.

The combined control here studied let us transform the two separate problems of path and speed following in a motion planning problem.

2.3.1 Model Linearization

The first step in the developed control strategy requires to generate a trajectory for state and input vectors, in order to follow speed and curvature reference.

A simplified vehicle model is used in this step in order to easily calculate the trajectory. This model is a non-holonomic one[2] (we impose that lateral velocity is identically zero).

Starting from the required forward speed V and the curvature σ previously calculated[3], the trajectory is obtained solving an inverse dynamic problem, whose solution is:

$$\begin{cases} v_T = V \\ V_\perp \equiv 0 \\ \dot{\phi} = \sigma V \\ \omega_r = -\frac{V}{R_r} \\ \omega_f = -\frac{V}{R_f} \\ \delta = \arctan(\sigma p) \end{cases} .$$

This trajectory in general is not a feasible one, because it's generated from a simpler model, but it is a good approximation.

The whole model is then linearized around this reference trajectory, obtaining a linear model for the error:

$$\dot{x} = f(x, u, t) \Rightarrow \dot{\tilde{x}} = A(t)\tilde{x} + B(t)\tilde{u}$$

where

$$\begin{aligned} \tilde{x} &= x - x_{ref} \\ \tilde{u} &= u - u_{ref} \\ A &= \left[\frac{\partial f}{\partial x}\right]\Big|_{\substack{x=x_{ref} \\ u=u_{ref}}} \\ B &= \left[\frac{\partial f}{\partial u}\right]\Big|_{\substack{x=x_{ref} \\ u=u_{ref}}} \end{aligned}$$

[2] See chapter 4 for further details on non-holonomic systems.

[3] See section 2.1.

2.3.2 Optimal Control

In the previous step a linear error model is derived. The control objective is to set at zero that error, which represents how far the real system is from the desired trajectory previously generated.

To do that a regulation problem is stated and solved applying optimal control techniques. As the error model is linear, the problem belongs to the well known LQR class of problems, once a quadratic cost function is chosen.

The problem is then stated as:

$$\min_{u\in U} J = \int_{t_0}^{t_f} \frac{1}{2}\left[\tilde{x}^T(t)Q(t)\tilde{x}(t) + \tilde{u}^T R(t)\tilde{u}\right] dt + \frac{1}{2}\tilde{x}^T(t_f)Q_f(t_f)\tilde{x}(t_f)$$

where Q and Q_f are diagonal and positive semi-definite matrix, while R is diagonal and positive definite. These matrices represent how internal states and control inputs are weighted in the total cost, so changing them has the consequence of changing how each component influences the solution.

In the absence of control constraints it has been shown[4] that the requested solution $\tilde{u}^*$ is a linear, time-variant state feedback:

$$\tilde{u}^* = K(t)\tilde{x}(t)$$

where

$$K(t) = -R^{-1}B^T(t)P(t)$$

and $P(t)$ is the solution of the Riccati matrix differential equation:

$$\begin{cases} -\dot{P}(t) = P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + Q(t) \\ P(t_f) = Q_f \end{cases}.$$

The optimal solution here obtained should then be added to the feed-forward action $\boldsymbol{u_{ref}}$ previously calculated to obtain the real control input signal, which is then recalculated at each simulation step.

2.4 Controller Implementation

In order to implement the algorithm described above, it's necessary to introduce some changes to deal with our simulation tools constraints, mostly due to numerical approximations. See appendix B for further details.

[4] See [8] for further details.

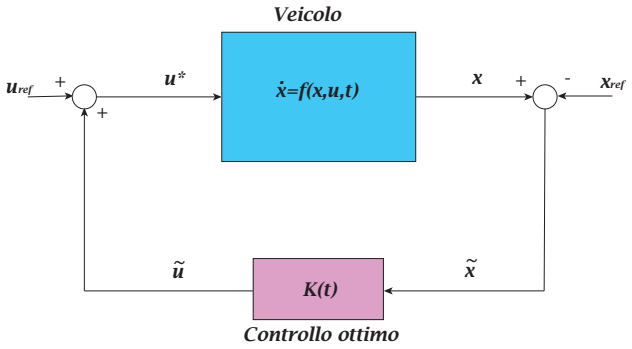


Figure 2.3: Control scheme.

2.5 Simulation Results

In the following section we present the results of our simulations, obtained for various guidance configuration. This choice allows to underline our controller characteristics.

2.5.1 Vehicle Characteristics

All the simulation tests are obtained using the standard virtual prototype vehicle included in MSC.Adams 12.0, the dynamic simulator. This vehicle has the following technical characteristics:

- the engine is 5000 cc; it can provide 485 Nm couple while operating at 5500 rpm;
- the braking system can supply a maximum couple of 696 Nm for front wheels and 416 Nm for rear wheels;
- maximum steering angle is 720 degrees in both verses;
- the chassis has good aerodynamic properties.

These characteristics put this vehicle between a common car and an high performance one. In order to evaluate our controller performances, we have compared its test response with smartdriver pilot (SD), a MSC.Adams toolkit, which implements a non interagent lateral and longitudinal control.

2.5.2 Straight Path

When a new vehicle model is developed, one of the standard test, used to verify it's reliability and performance, is the acceleration/braking check on a straight path. In this way only the longitudinal dynamics of the vehicle is excited: there are several speed changes that require the use of accelerator and brake, while the path curvature is constantly zero.

The reference speed profile is shown in figure 2.4; in its first part there is a change of speed from 28$[m/s]$ to 50$[m/s]$ in almost 300 meters, then the speed is abruptly lowered to 20$[m/s]$ in less than 150 meters, and then it's taken constant for other 100 meters; after that there is an another speed reduction to 10$[m/s]$ in 100 meters, and then in the last part of the curve the velocity is constant.

In figure 2.5 there are plotted the results of this simulation. Magenta lines are for the reference values, blue lines are for the old controller, represented by the smartdriver toolkit and red lines are for our controller (further indicated with "new controller" term).

The controller is able to follow the velocity profile more accurately than the commercial toolkit, especially when there is a reference speed change; here our controller presents an error which is an order of magnitude smaller than the old one.

The drawback is the increased noise in the steering control signal when a change of speed occurs; this is due to the coupling of longitudinal and lateral control actions. The requested steering angles are very small though, so the vehicle dynamics is not substantially influenced by that, as it can be noticed in figure 2.6, where the steering angle and the vehicle lateral acceleration are compared.

2.5.3 Chicane

Another fundamental trial to test the algorithm performances is following a single path at different speed. We have chosen a chicane, which is composed

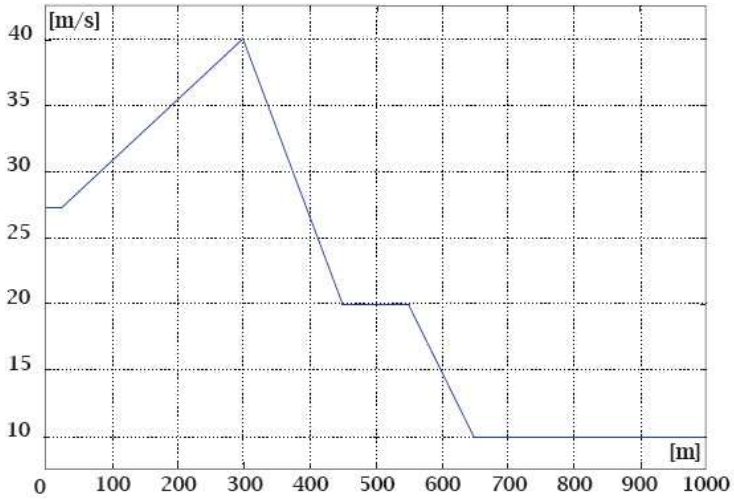


Figure 2.4: Straight test: speed reference.

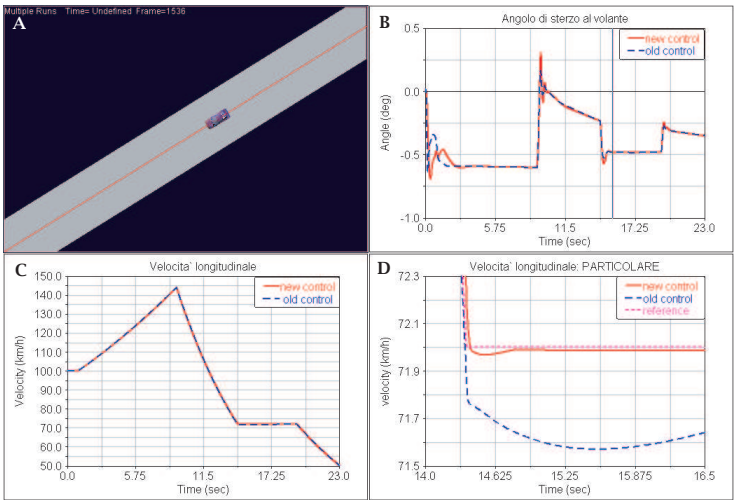


Figure 2.5: Straight test: speed and steering angle.

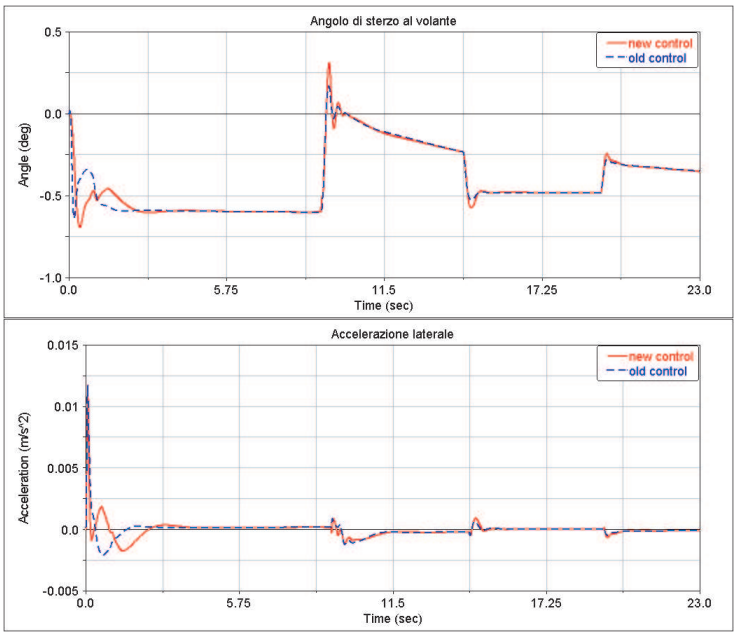


Figure 2.6: Straight test: lateral acceleration.

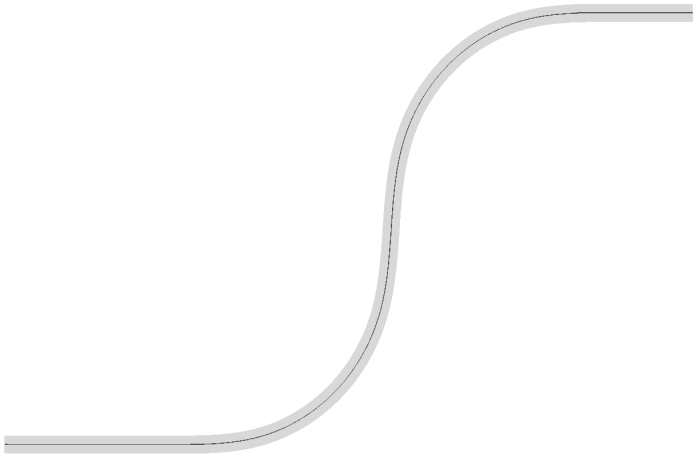


Figure 2.7: Chicane test: path reference.

by two 90 degrees curves with a straight path in between. Here both the longitudinal and lateral control is required to follow properly the assigned path. To check the controller behavior and it's limit 3 different speed values are chosen:

- standard guidance (15$[m/s]$ which corresponds to 54$[km/h]$)
- guidance near the limit of model validity(25$[m/s]$ which corresponds to 90$[km/h]$)
- guidance beyond the limit (30$[m/s]$ which corresponds to 108$[km/h]$)

Low Speed Chicane

When the requested speed is low the hypotheses of model validity are satisfied and the controller gives good results as expected. It's ability in following the reference trajectory is higher than the old controller one, as you can see in figure 2.8. The steering angle control signal is quite nervous, but it's vibration barely influence the vehicle dynamics, as it can be seen in figure 2.9d) where lateral velocity is displaced.

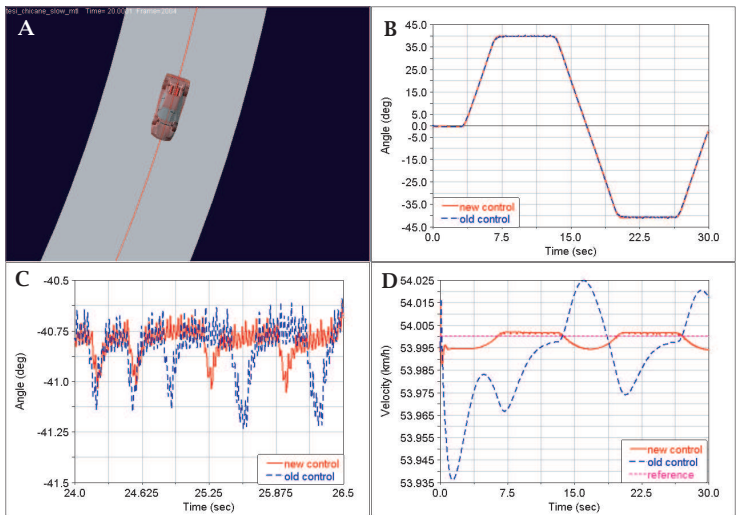


Figure 2.8: Low speed chicane test: speed and steering angle.

Medium Speed Chicane

The second trial is done with a constant reference speed at 90$[km/h]$. As it can be noticed in figure 2.10 the control objective is reached in a satisfactory manner. This is because the system lies in the range of validity of the model used to control it (see figure 2.11).

A critical part in the control algorithm is the choice of state and control weights in the cost functional. A very accurate analysis is then required to obtain an effective controller.

The problem here faced is about a reference speed and curvature trajectory pursuit; for this reason we have decided to tune the weight parameter of longitudinal velocity and yaw rate in the state matrix Q, while setting to zero the others[5]. In the matrix R the weight parameter of the steering angle is chosen quite big, to avoid as much as possible control signal vibrations, while the one of the throttle/brake couple is set to a small value, to let the controller change easily their values. In figure 2.12 the difference between a well-tuned optimal controller and a randomly tuned one is depicted.

High Speed Chicane

The last trial is done to test the controller limits. An high speed is selected,which leads the vehicle slip angles to assume values out of the range of validity of the controller. In this case the controller can't control the vehicle in a satisfactory way.

2.5.4 Mixed Path Test

The global controller behavior can be tested in a course composed by straight tracks and differently shaped bends. This course require intense longitudinal and lateral control, so it's useful to highlight the consequences of their coupling.

As you can see in figure 2.15, where the course, the speed profile and the steering angle are shown, the controller here developed globally overcomes the old one performances.

[5]To avoid numerical instability these parameters are actually set to a very small value though.

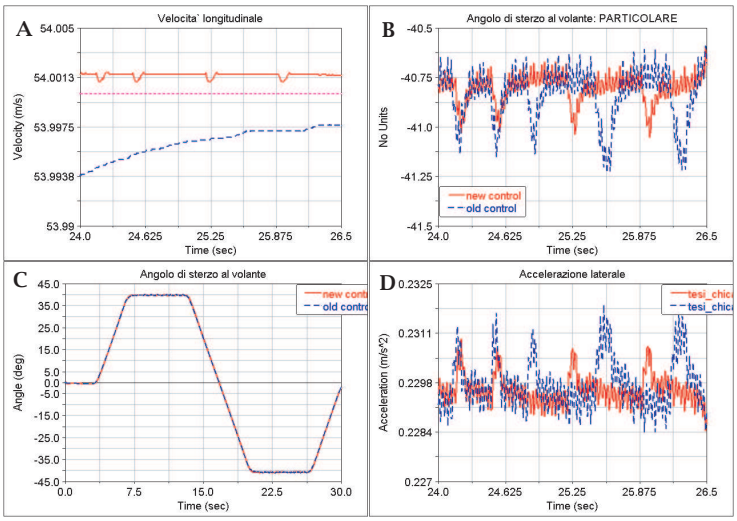


Figure 2.9: Low speed chicane test: lateral acceleration.

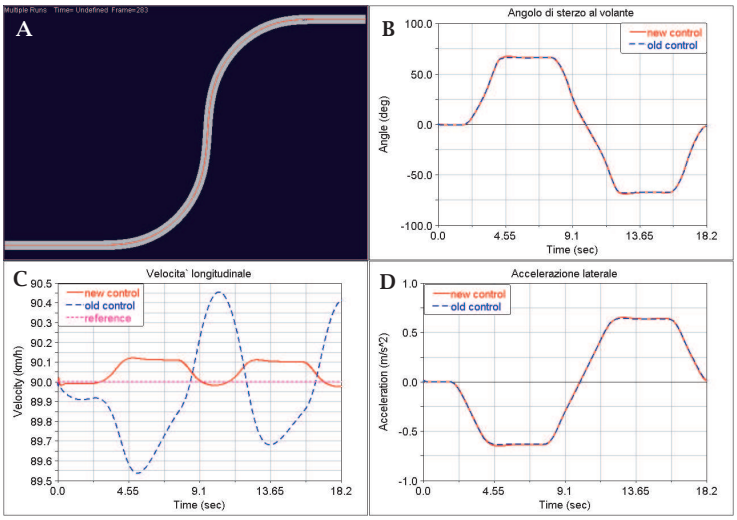


Figure 2.10: Medium speed chicane test: speed and steering angle.

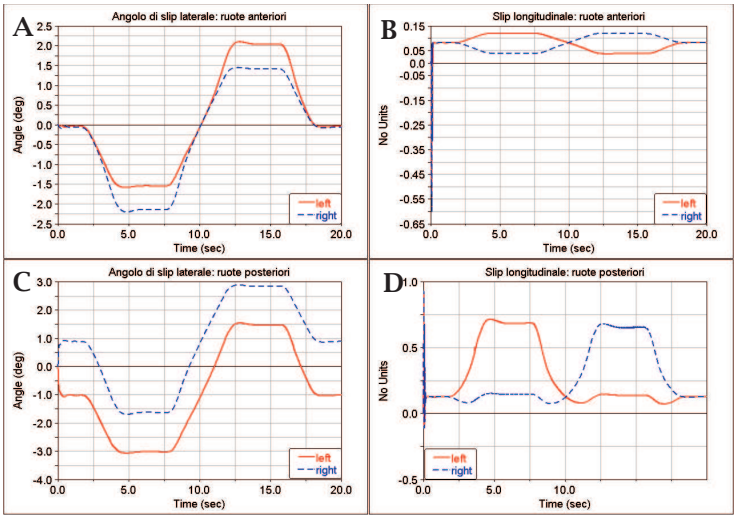


Figure 2.11: Medium speed chicane test: wheels slip angles.

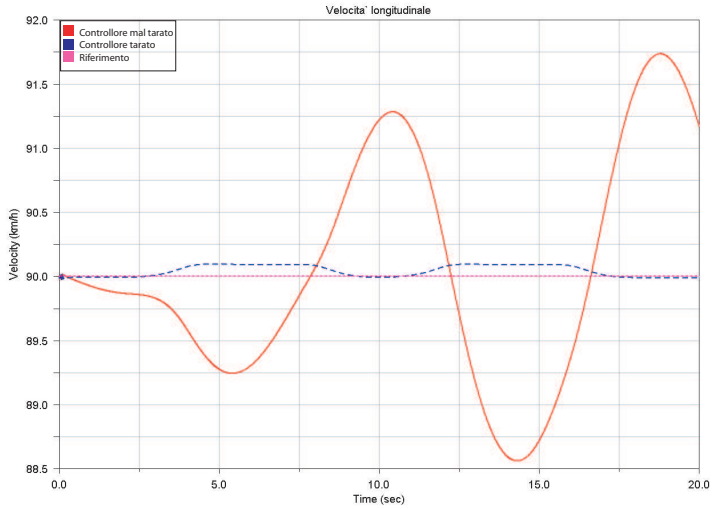


Figure 2.12: Medium speed chicane test: Differences between a well tuned controller and a worse tuned one.

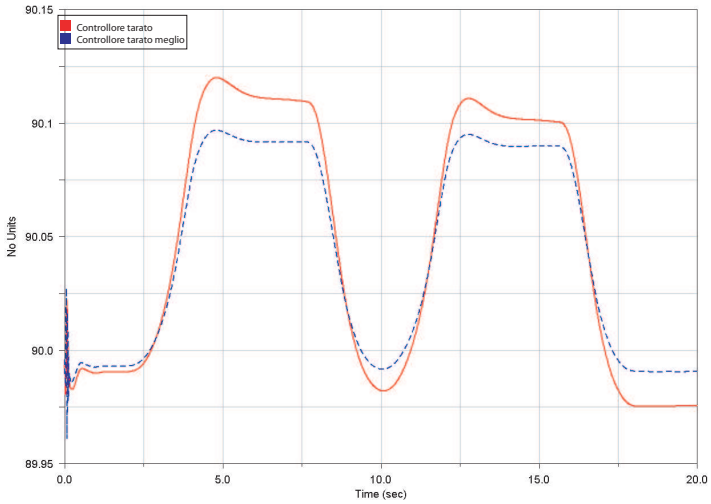


Figure 2.13: Medium speed chicane test: Differences between two differently tuned controllers.

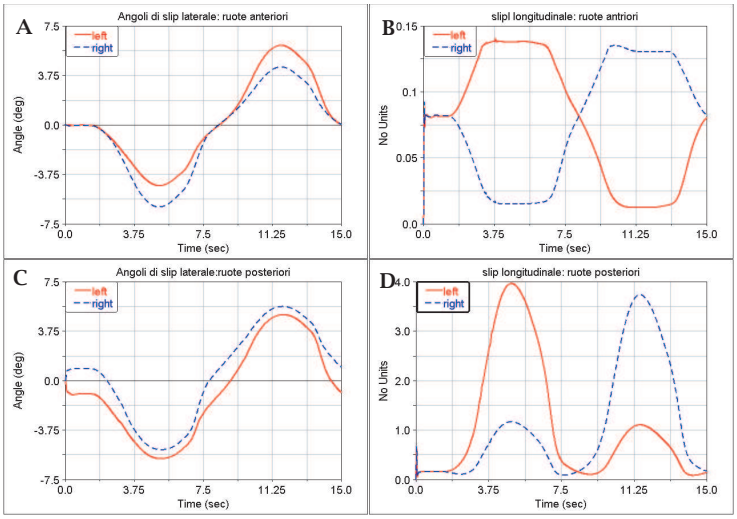


Figure 2.14: High speed chicane test: wheels slip angles.

2.5.5 Path with Vertical Dynamics

An other interesting trial, to check control ability and robustness of the algorithm, is on a course with an uneven surface, in order to see how the control system reacts when a non-modeled vertical dynamics is excited.

The chosen course is shown in figure 2.16. It is composed by:

an initial 160[m] long straight path, with a 1.2[m] height ramp after 60[m];

a left bend with a curvature radius of 100[m];

another straight segment 160[m] long, with banking in the first part (the lateral inclination is 10 left and 20 right);

a right bend with a curvature radius of 50[m]; at the end of the bend the pavement is uneven, presenting a periodic height change, of period 10[m] and amplitude 10[cm];

a straight stretch 100[m] long, where the uneven periodic path lasts for 40[m]; after that there is a change on period and amplitude of the surface oscillations, that become of period2[m] and amplitude 6[cm]

a left bend with a 40[m] radius of curvature; in the middle of the bend there is a ramp with a height of 1.5[m].

The vertical profile of the road is shown in figure 2.17, while the velocity profile is depicted in figure 2.18; both of them are graphed in function of the curved abscissa.

In following figures there are the results of the simulation; you can notice how the controller does a good job as long as the lateral velocity and wheels slip angles are sufficiently small, but isn't able to adequately answer to abrupt changes of simulation conditions, that move the system far from the model validity range.

After 400[m] there is a sharp braking action, that excites the vehicle pitch dynamics, which moves most of the vehicle weight on the front axle; this is potentially a great instability cause, because the vehicle enters the subsequent bend with an unloaded rear axle, which implies that the wheels can't generate a great longitudinal force and tend to slide sideways.

In figure 2.19 the steering input signal (**c**) and the vehicle longitudinal speed (**a**) are depicted. It is interesting to notice how the steering command

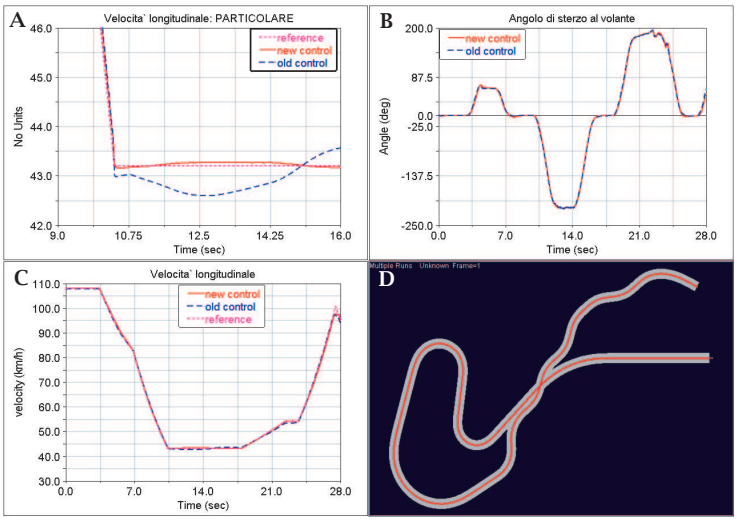


Figure 2.15: Mixed path test: forward speed and steering angle.

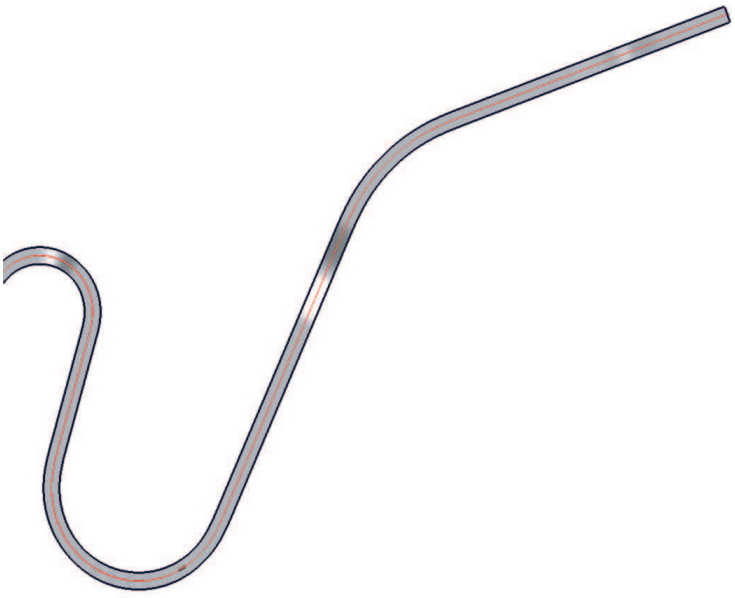


Figure 2.16: Road used for the trial.

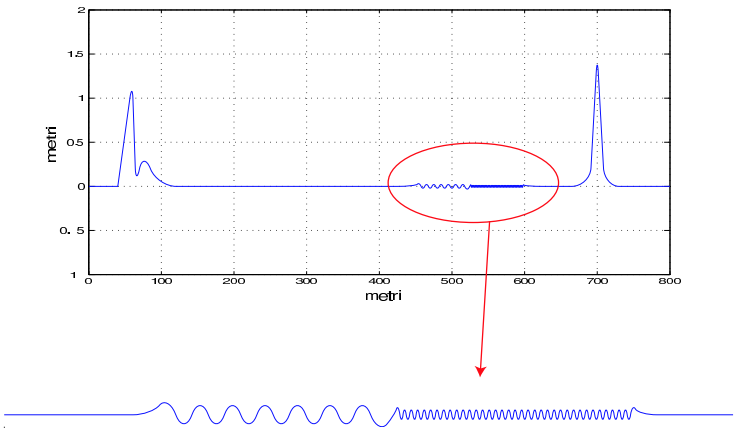


Figure 2.17: Vertical road profile.

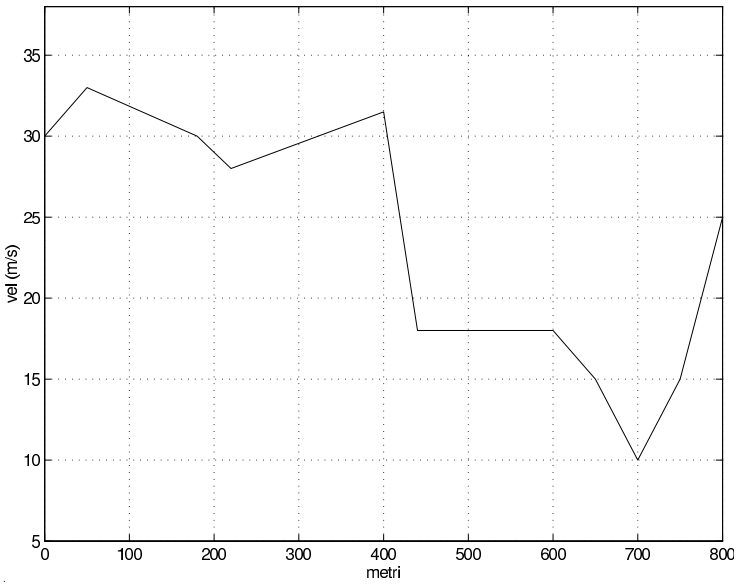


Figure 2.18: Velocity reference profile.

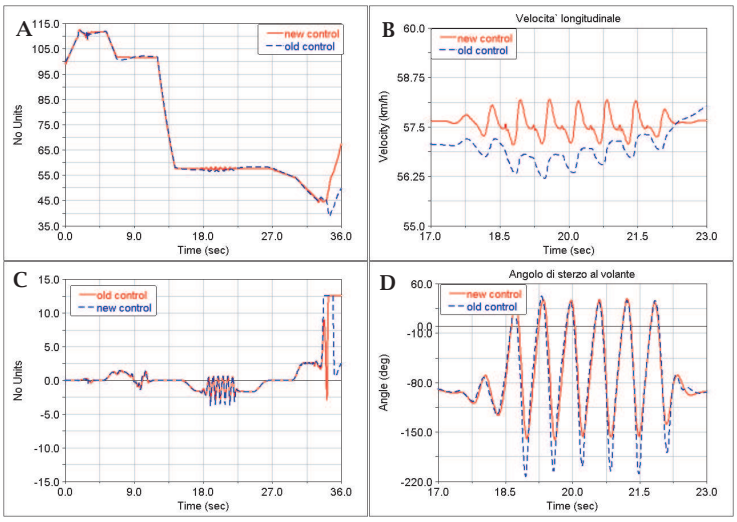


Figure 2.19: Steering angle and forward speed signals.

is non-zero while the car is flying after the ramp (see figure 2.20 **b** and **d**), even if it is not useful; this is because the controller doesn't have any vertical dynamics model inside that can handle this situation. Nevertheless the connecting contour strategy is sufficiently robust to deal with jumps and uneven surfaces, as long as the preview distance is wide enough[6].

In figure 2.20 the steering angle applied as control and the consequent vehicle velocity, relative to the part of the track where the jump happens and the one with banking, are shown. You can easily notice how the stress produced by the road is directly transmitted to the vehicle yaw angle, and how the controller tends to compensate each solicitation; also the vehicle speed is affected by the road stress, so the controller compensates also these variations [7].

[6]Choosing a preview distance D is equivalent to generate a control that leads the vehicle to be on the reference trajectory before the end of that interval. If D is too small, an abrupt change of the vehicle position or yaw angle, due to the pavement roughness, will generate a wide control action, which could easily conduct to wheels saturation, with consequent vehicle control loss. On the other side a value too big let the vehicle stay far from the requested path.

[7]These speed variations are mostly due to the vehicle pitch dynamics. The speed sensor is indeed attached to the vehicle chassis, therefore a pitch angle different from zero affects the longitudinal velocity measure.

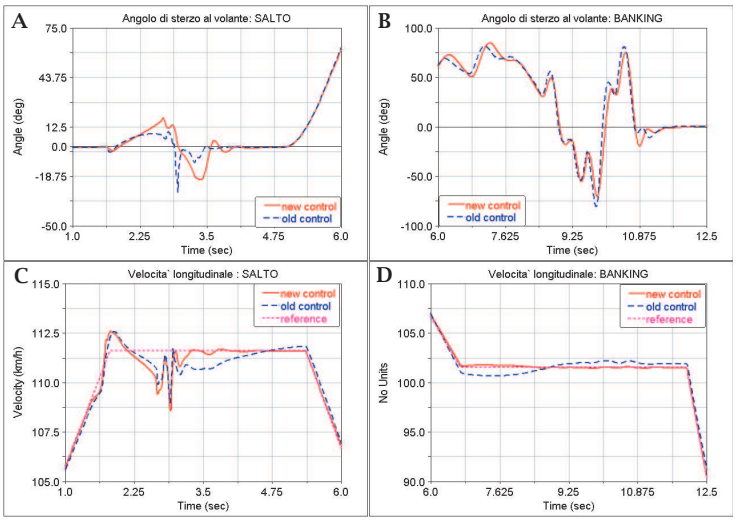


Figure 2.20: Steering angle and forward speed signals in presence of uneven pavement.

2.6 Further Developments

The control here developed aims to follow a preassigned reference trajectory for forward speed and path curvature. Through the analysis of its behavior in several trials it can be seen how the controller here developed reaches the control objective in a more effective way than the previous one. There are some improvement that can be made to increase its effectiveness though.

Currently an optimization problem is solved at each simulation step, and its solution is applied for a single step. This approach causes a lot of computational work, which can be reduced, obtaining a faster and computationally cheaper algorithm, by making the number of steps where the optimal control is applied dependent on the execution conditions; if the vehicle after a step is still close to the initial condition then the optimal control recalculated will be very similar to the one already found, so the smartest choice is continuing to apply the old one as long as the system conditions aren't changed substantially, when the control needs to be recalculated. In this way the control could deal with "calm" situation, for example a straight path with a constant velocity profile, where the control is recalculated once in a while, and with more demanding ones, for example a sharp bend or an abrupt speed change, where the control needs to be calculated more often to obtain satisfactory performances.

The controller gives excellent results when the system is in the range of validity of the hypotheses made for its synthesis, while its performance rapidly decays when the tires slip angles and the vehicle lateral velocity grow too quickly and the model goes beyond its limits. To improve its ability of handling these situation two main changes should be implemented. The first one is about the internal vehicle model used; switching from the bicycle model to a two-track would allow the system to take in account the vehicle pitch and roll dynamics, which can't be neglected when the vehicle is subject to abrupt direction changes or sharp speed variations.

The second change that could improve the controller performances is the enrichment of the model used to generate the reference trajectories for state and input variables, because the non-holonomy requirement is not a so good approximation of vehicle condition in high demanding manoeuver, when the its lateral sliding can't be neglected.

In the vehicle prototyping process it is required not only to verify the vehicle behavior in extreme guidance conditions, but also its response while driving in normal conditions, to study its handling and maneuverability. A model for the human driver is therefore essential.

In particular a real driver can only rely upon their visual feedback, which gives them the vehicle position with respect to the road and its forward speed, and other data that can be derived from the other senses, which are generically called proprioception. Data harvest and analysis are not immediate, but are subjected to physical transmission delays. Furthermore the control action is not continuously generated, but it's calculated only when it is needed.

The controller can be easily modified to simulate the human driver, adding suitable delay blocks, tuned on the human frequencies, and modifying the control calculation frequency moving from a fixed scheme to one in which its activation is tied to road conditions, as explained above. Tuning those parameters allows the controller to represent different driver categories, from the inexperienced one to the professional driver.

Chapter 3

Pseudospectral Methods

In the previous chapter a control problem for a mechanical system has been stated and solved through application of optimization techniques. The theoretical solution has then been implemented to obtain a real solution and the problems encountered during this shift are summarized in appendix B.

One fundamental aspect in optimal control solving is the choice of the method that effectively calculates the optimal solution. In the next sections a brief review of these methods is presented, followed by an introduction to the Pseudospectral Methods, that have been proved to be a great tool for the numerical solution of optimization problems. Some examples of their application are finally presented. The first one is a simple modellization of a wheeled robot moving in a plane with obstacles, while the second one is about the time minimization for a vehicle maneuver.

3.1 Review of Solution Methods

The analytical solution of a state and control constrained nonlinear optimal control problem requires a big effort to be found. The main difficulty arises in seeking a closed-form solution to the Hamilton-Jacobi equations, or in solving the canonical Hamiltonian equations resulting from an application of the Minimum Principle.

Over past decades, many computational methods have been developed for solving nonlinear optimal control problems. For instance, in [5] and [6], various numerical methods such as neighboring extremal methods, gradient methods and quasi-linearization methods are discussed in detail, along with

extensive numerical results. In [9], the feasibility and convergence of a modified Euler discretization method is proved, while a unified approach based on a piecewise constant approximation of the control is proposed in [23].

Numerical methods for solving nonlinear optimal control problems are classically divided in two categories: direct methods and indirect methods (see [1] for further details).

Many numerical methods developed in the past were based on finding solutions to satisfy a set of necessary optimality conditions resulting from Pontryagin's Maximum Principle. These methods are collectively called indirect methods. There are many successful implementations of indirect methods including launch vehicle trajectory design and low-thrust orbit transfer. These methods, although presenting some nice properties, suffer from many drawbacks. The boundary value problem resulting from the necessary conditions is extremely sensitive to initial guesses. In addition, these necessary conditions must be explicitly derived, which is not straightforward for complicated problems and requires a thorough knowledge of optimal control theory (see [1] for more details).

To overcome this difficulty, an alternative approach based on discrete approximations has been developed. The essential idea of this method is to discretize the optimal control problem and solve the resulting large-scale finite-dimensional optimization problem. These types of methods are known as direct methods. At a first sight these methods seem to be simpler and easier to be applied, but they present a wide range of deeply theoretical issues that lie at the intersection of approximation theory, control theory and optimization. Regardless, a wide variety of industrial-strength optimal control problems have been solved by this approach.

Theoretical questions regarding the existence of a solution and convergence of the approximations are still open for a wide range of these methods. Indeed, while the theory for Eulerian methods is well established and proved (see for example [13] and [33]) corresponding results for higher-order methods are not only absent, but often counterintuitive. For example, Hager [27] has shown that a "convergent" Runge-Kutta method does not converge to the continuous optimal solution despite the fact that it satisfies the standard conditions in the Butcher tableau. On the other hand, Betts [2] shows that a nonconvergent Runge-Kutta method converges for optimal control problems. Thus, it is not surprising that even for Eulerian methods, significant restrictions and assumptions are necessary for proofs of convergence, particularly for state-constrained problems. While Eulerian methods are widely studied and useful for a theoretical understanding of discrete approximations, they

are not suitable for solving industrial strength problems; one of the main reasons for their limitation is that they generate a much larger-scale optimization problem than a higher order scheme like a Runge-Kutta method.

Another group of discretization methods is the direct pseudospectral (PS) methods. PS methods were largely developed in the 1970s for solving partial differential equations arising in fluid dynamics [7], and quickly became "one of the big three technologies for the numerical solution of PDEs" [41]. During the last decade PS methods were introduced for solving optimal control problems(see for example [15] and [16]; one of the main reasons of their success is that they demonstrably offer an exponential convergence rate for the approximation of analytic functions while providing Eulerian-like simplicity [41]. Thus, for a given error bound, PS methods generate a significantly smaller scale optimization problem when compared to other methods on problems with highly smooth solutions.

3.2 Pseudospectral Methods

In this section we'll give a brief introduction on Pseudospectral methods and their properties; for further details see [24] and [38].

Numerical solution of a optimal control problem requires the approximation of three types of mathematical objects: the integration in the cost function, the differential equation of the control system, and the state-control constraints.

In PS methods, the continuous functions are approximated at a set of carefully selected quadrature nodes. These nodes and the chosen interpolating polynomials are the object that distinguish one method from the others. Typically, quadrature nodes are Gauss points, Gauss-Radau points or Gauss-Lobatto points and are determined by the corresponding orthogonal polynomial basis used for the approximation. Legendre and Chebyshev polynomials are commonly used.

Integration is approximated by quadrature rules, which provide the best numerical integration result. Gauss quadrature integration, for example, achieves zero error for any polynomial integrand of degree less than or equal to 2N+1 with just N nodes. Because a PS method enforces the system at the selected nodes, the state-control constraints can be discretized straightforwardly.

The continuous optimal control problem is then discretized into a se-

quence of finite dimensional optimization problems that are solved by spectral algorithms. One fundamental property ensured by these methods is the commutation between discretization and dualization. This property is enunciated as the *Covector Mapping Theorem* illustrated in figure 3.1 (see [25] for the proof of the theorem). In this Figure, the continuous optimal control

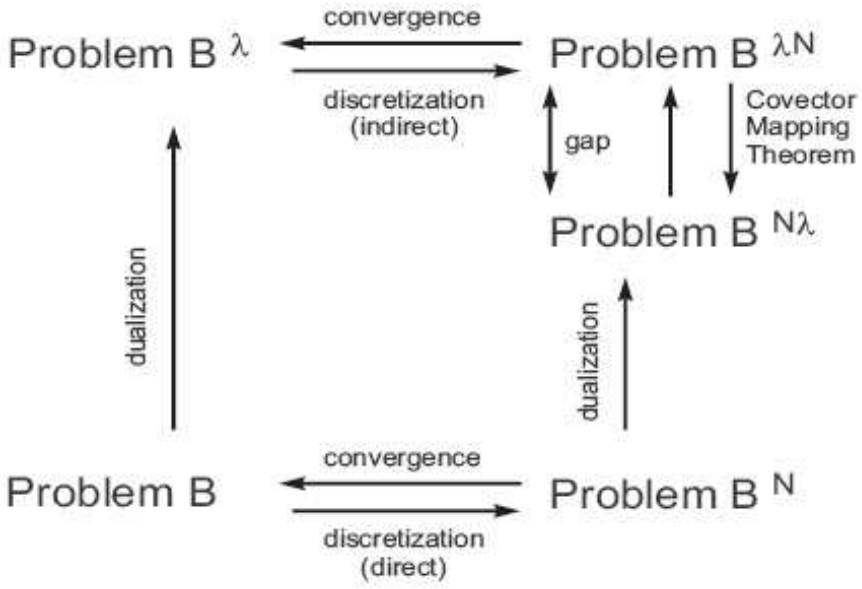


Figure 3.1: Covector Mapping Theorem.

problem is denoted as Problem B and the discretized finite dimensional optimization problem is denoted as Problem B^N. Associated with the original problem, a set of necessary conditions (Problem B^λ) can be obtained from an application of Pontryagin's Minimum Principle. The discretized optimization problem also admits a set of necessary conditions called Karush-Kuhn-Tucker (KKT) conditions (Problem $B^{\lambda N}$). The theorem states that an optimal candidate solution for the discretized problem (B^N) must automatically satisfy the discretized necessary conditions ($B^{\lambda N}$).

Its practical advantage is that nonlinear optimal control problems can be solved efficiently and accurately without developing the necessary conditions. On the other hand, the optimality of the solution can be checked by using the numerical approximations of the covectors obtained from the theorem. Since these solutions can presently be obtained in a matter of seconds, it appears that the Pseudospectral methods can be used for optimal feedback

control in the context of a nonlinear model framework.

In the rest of the chapter, the Legendre Pseudospectral Method is applied to solve the problems presented. This method is based on interpolating functions on Legendre-Gauss-Lobatto (LGL) quadrature nodes.

It has been proved that the interpolating polynomial of any smooth function C^∞ at Legendre-Gauss-Lobatto nodes converges in L^2 sense at the so-called spectral rate, i.e., faster than any polynomial rate [41].

3.2.1 Toy Problem

As example of application of PS methods the problem of a wheeled robot that can move in an horizontal plane with box constraints is presented. The objective is to move the robot from one point to another avoiding the obstacles in the plane, that are randomly positioned.

This problem has many applications in both civilian and military fields; just to cite some examples of real application we mention RoboCup Rescue world competition (see [37]), in which a wheeled robot is programmed to reach an objective position moving through an enviroment with several obstacles, and where the final goal is to build an autonomous agent for search and rescue the hostile enviroment; another example is DARPA Grand Challenge (see [12]), an international competition where an autonomous vehicle is set up to run while avoiding the obstacles along the path.

Mathematical formulation of the problem

The model provided is a non-holonomic car (no sliding on the surface is allowed); the velocity and the steering angle are the controls, while the states are the position of the origin of the body frame (which is positioned in the center of the rear axle of the car,with the x axis pointing along the direction of motion) and the yaw angle, which is the angle between the x axis of the reference frame and the body frame.

The objective choose for this problem is the minimization of the time needed to reach the end point position.

The problem formulation is scaled, so the robot starts in $(0, 0)$ and ends in $(1, 1)$; we suppose that the obstacles are randomly disposed in the plane but the position of every one is known. The optimality problem is the following:

$$\begin{cases} \textbf{minimize} & J\left[x(t), u(t), t\right] = t_f \\ & \dot{x}(t) = v(t)\cos\psi(t) \\ & \dot{y}(t) = v(t)\sin\psi(t) \\ & \dot{\psi}(t) = \frac{v(t)\tan\psi(t)}{p} \\ \text{subject to} & (t_0, x_0, y_0, \psi_0) = (0,0,0,0) \\ & (x_f, y_f) = (1,1) \\ & \mathbf{u} = [v,\psi]' \in \mathbb{U} = \{(v,\psi) \mid 0 \leq v \leq v_{max}\} \\ & \mathbf{x}\ in\ \mathbb{X} \end{cases}$$

Where $\mathbb{X}$ is the region represented in figure 3.2.

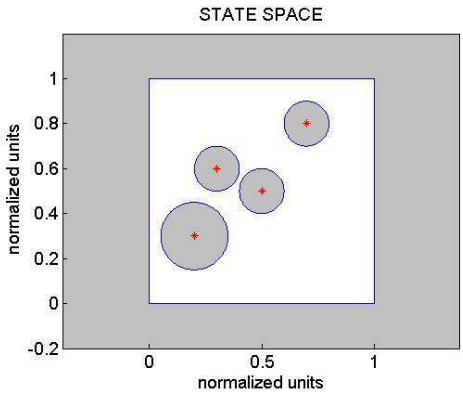


Figure 3.2: Admissible region.

To avoid the obstacles in the plane xy we have to take in account the shape of the car, so for each obstacle and for every time we have to verify that each point of the car is far enough from the center of the obstacle:

$$\forall\,(x_o, y_o, r_o) \qquad \min d\left((x_o, y_o), (x_{car}, y_{car})\right) \geq r_o$$

In Figure 3.3 that distance is represented by the line D_2

In order not to complicated the problem we can consider simply the worst case, so instead of consider all the points of the car shape we just consider the distance from the obstacle center and the origin of the body frame, then subtract to it the maximum distance from that point to the car shape, to guarantee that all the point of the car shape are more far than the radius to the center point of the obstacle; in practice we can simply artificially augment the obstacle radius to $r_o + D_{max}$.

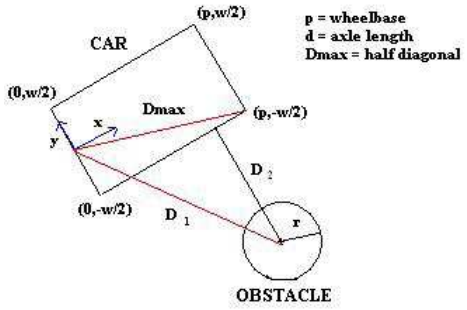


Figure 3.3: Obstacle Avoidance.

To simplify further the problem formulation we'll ignore the physical dimensions of the car and we'll consider the robot as a mass point. For this simplified model the admissible region $\mathbb{X}$ is:

$$\mathbb{X} = \left\{ \begin{array}{c} (x, y, \psi) \in \mathbb{R}^2 \times \mathbb{S}^1 \quad | \quad \forall \; O = ((x_o, y_o), r_o) \\ \\ \sqrt{(x - x_o)^2 + (y - y_o)^2} \geq r_o \wedge 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \end{array} \right\}$$

To speed up the computation we can substitute the condition on the distance between the car and the center of each obstacle with $(x - x_o)^2 + (y - y_o)^2 \geq {r_o}^2$ which is equivalent to the square condition; in this way we also avoid problems for $(x, y) = (0, 0)$ in which the derivative of the Hamiltonian with the first version of the constraint is not bounded and so the Minimum Principle cannot be applied.

Derivation of necessary conditions

The problem formulated presents path constrained, so to obtain the necessary first order conditions always valid we have not to consider simply the Hamiltonian, but the Lagrangian of the Hamiltonian, also called augmented Hamiltonian, as explained in [6] and [28]:

$$\bar{H}\left[\mathbf{x}(),\mathbf{y}(),\lambda(),\mu,\nu,t\right] =$$

$$\lambda_x v\cos\psi + \lambda_y v\sin\psi + \lambda_\psi \tfrac{v\tan\delta}{p} + \mu_v v + \mu_x x +$$

$$+\mu_y y \textstyle\sum_{k=1}^{NUM\_OBS} \mu_k \left[\left((x-x_o)^2 + (y-y_o)^2\right) - {r_o}^2\right].$$

The necessary conditions for the problem are:

- State Equation

$$\dot{\mathbf{x}} = \tfrac{\partial \bar{H}}{\partial \lambda} = f\left(\mathbf{x},\mathbf{u},t\right) = \begin{bmatrix} v\cos\delta \\ v\sin\delta \\ \frac{v\tan\delta}{p} \end{bmatrix}$$

- Adjoint Equation

$$-\dot{\lambda} = \tfrac{\partial \bar{H}}{\partial \mathbf{x}} = \begin{bmatrix} \mu_x + \sum_{k=1}^{NUM\_OBS} 2\mu_k(x-x_o) \\ \mu_y + \sum_{k=1}^{NUM\_OBS} 2\mu_k(y-y_o) \\ -\lambda_x\sin\psi + \lambda_y\cos\psi \end{bmatrix}$$

$$KKT$$

$$\mu_x \begin{cases} \leq 0 \;\; if \;\; x = 0 \\ = 0 \;\; if \;\; 0 < x < 1 \\ \geq 0 \;\; if \;\; x = 1 \end{cases}$$

$$\mu_y \begin{cases} \leq 0 \;\; if \;\; y = 0 \\ = 0 \;\; if \;\; 0 < y < 1 \\ \geq 0 \;\; if \;\; y = 1 \end{cases}$$

$$\mu_i \begin{cases} \leq 0 \;\; if \;\; \left((x-x_o)^2 + (y-y_o)^2\right) = {r_o}^2 \\ = 0 \;\; if \;\; \left((x-x_o)^2 + (y-y_o)^2\right) > {r_o}^2 \end{cases}$$

When the constraints are not active the conditions on λ are computationally verifiable and became: $\begin{cases} \lambda_x = 0 \\ \lambda_y = 0 \end{cases}$

- Boundary Conditions
  $(t_0, x_0, y_0, \psi_0) = (0, 0, 0, 0)$
  $(x_f, y_f) = (1, 1)$

- Transversality Condition
  $\lambda_{\psi_f} = 0$

- Hamiltonian Value Condition
  $\lambda_{xf} v_f \cos\psi_f + \lambda_{yf} v_f sin\psi_f = -1$

- Hamiltonian Evolution Equation
  $\dot{H} = \frac{\partial H}{\partial t} = 0 \Rightarrow H = constant$

- HMC

$$\frac{\partial \bar{H}}{\partial u} = 0 \Rightarrow \begin{cases} \frac{\partial \bar{H}}{\partial v} = -\lambda_x v \sin\psi + \lambda_y v \cos\psi + \lambda_\psi \frac{v \tan\delta}{p} = 0 \\ \frac{\partial \bar{H}}{\partial \psi} = \frac{\lambda_{\psi v}}{p \cos\delta^2} = 0 \end{cases}$$

$$\mu_v \begin{cases} \leq 0 \;\; if \;\; v = 0 \\ = 0 \;\; if \;\; 0 < v < V_{max} \\ \geq 0 \;\; if \;\; x = V_{max} \end{cases} \qquad KKT$$

The computationally verifiable conditions are boxed.

Simulation Results

The numerical solution of the problem is here presented. The results have been obtained through the use of DIDO, a tomlab tool that implements the Legendre Pseudospectal Method (see [39] for further details about the program and its characteristics).

Each simulation has been made applying a bootstrapping procedure, running DIDO with a few nodes to obtain a better guess and then re-running the program to obtain the results; this is done to speed up the procedure.

As said before the problem formulation is about a scaled problem, so all the parameters are properly chosen (if the length unit is meter, the velocity is meter/second and so on). The following table summarize those parameters.

wheelbase	axle length	obstacles		
		x_{center}	y_{center}	radius
0.1	0.04	0.5	0.5	0.1

The optimal solution computed by DIDO is shown in figure 3.4

The optimal cost computed by DIDO is:

$$\mathbf{J}^* = 0.143$$

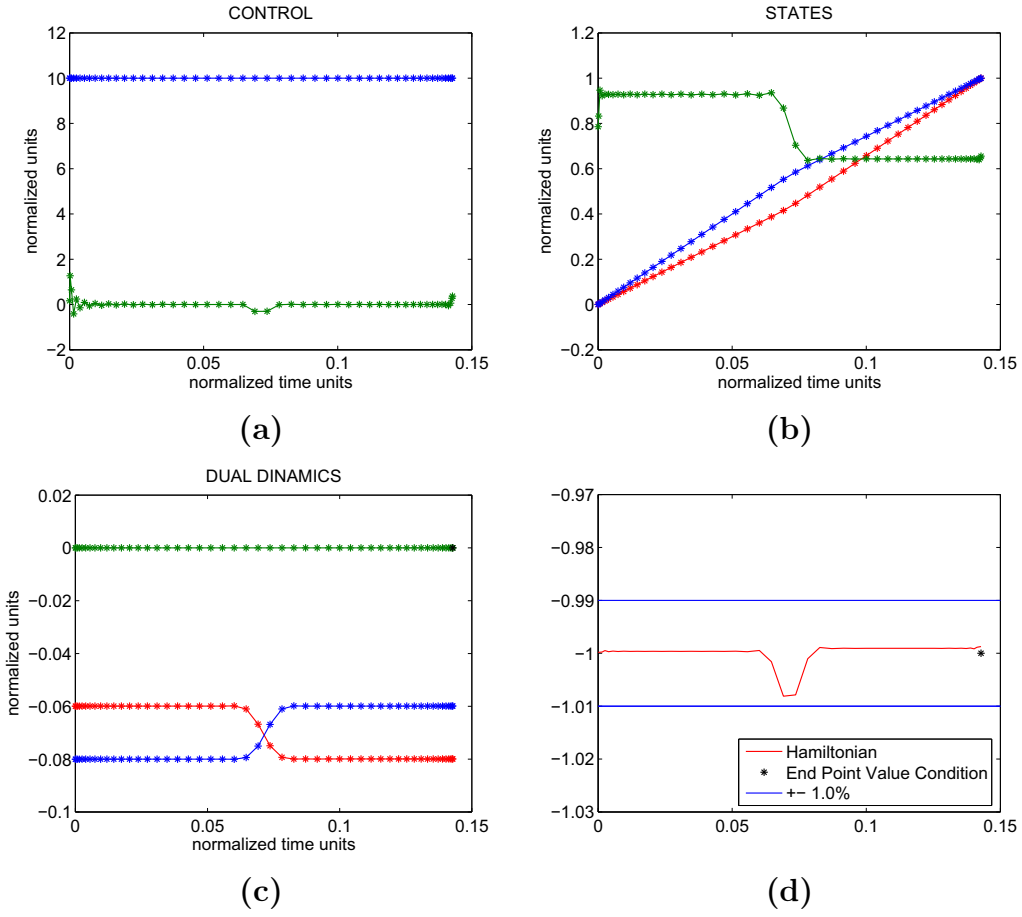


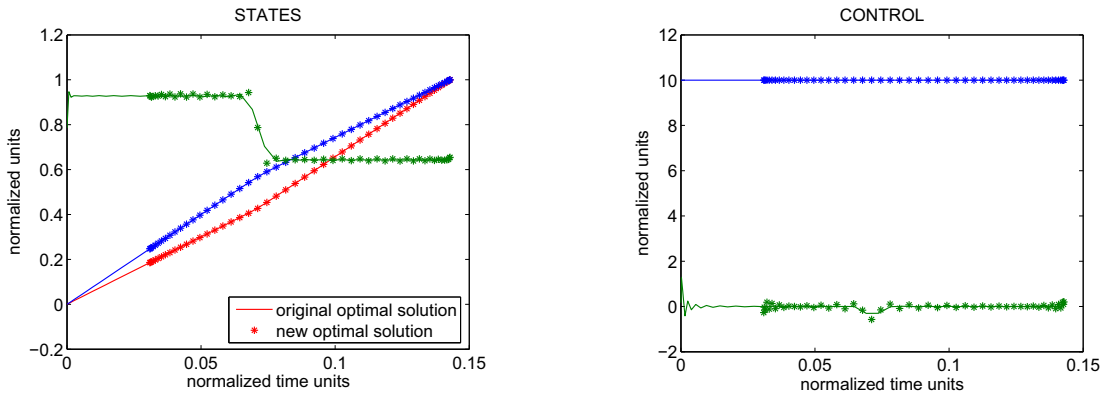
Figure 3.4: Optimal results for the original problem: (a) Optimal Control,(b) Optimal State Trajectory, (c) Optimal Costate Trajectory,(d) Hamiltonian.

Demonstration of computational optimality via Bellman's Principle

In order to prove Bellman's Principle of Optimality we start DIDO simulation from a point lying in the optimal state trajectory.

$$\begin{cases} x_0 = 0.186; \\ y_0 = 0.247; \\ \psi_0 = 0.929; \\ t_0 = 0.031; \end{cases} \qquad \mathbf{J} = 0.143 = \mathbf{J}^*$$

As expected in this way we obtain the same value for the cost function and the same trajectory for states and controls.



(a) **(b)**

Figure 3.5: Demonstration of Bellman's optimality principle.

Here we choose a point not lying in the optimal trajectory:

$$\begin{cases} x_0 = 0.600; \\ y_0 = 0.200; \\ \psi_0 = 0.331; \\ t_0 = 0.063; \end{cases}$$

For this initial condition the optimal cost is

$$\mathbf{J} = 0.153$$

which is higher than the original optimal cost J^* as expected.

In Figure 3.6 the space state with the optimal state trajectory is shown.

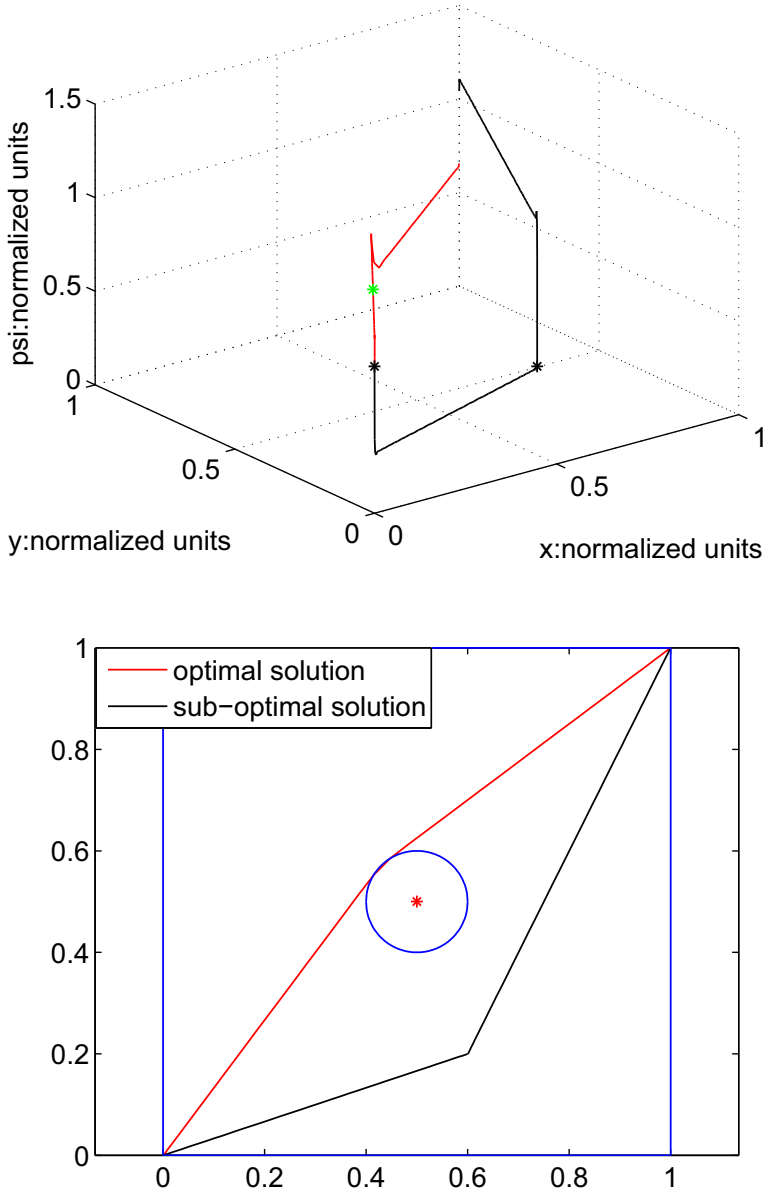


Figure 3.6: State Space and Optimal Trajectory.

3.3 Minimum Lap Time

In this section the problem of minimum time maneuvering of a vehicle is analyzed as a problem in optimal control.

The system dynamics is derived in chapter 1. In order to formulate an optimal control problem, it is necessary to define the cost function to be optimized and specify all state/control constraints and boundary conditions.

Here we present preliminary optimization results, for two different paths: a straight path and a $90[deg]$ corner, that the vehicle has to negotiate in minimum time. The constraint in each case is on the position of the vehicle center of mass.

The cost function for the minimum travel time problem is given by:

$$J = \int_0^{t_f} dt = t_f$$

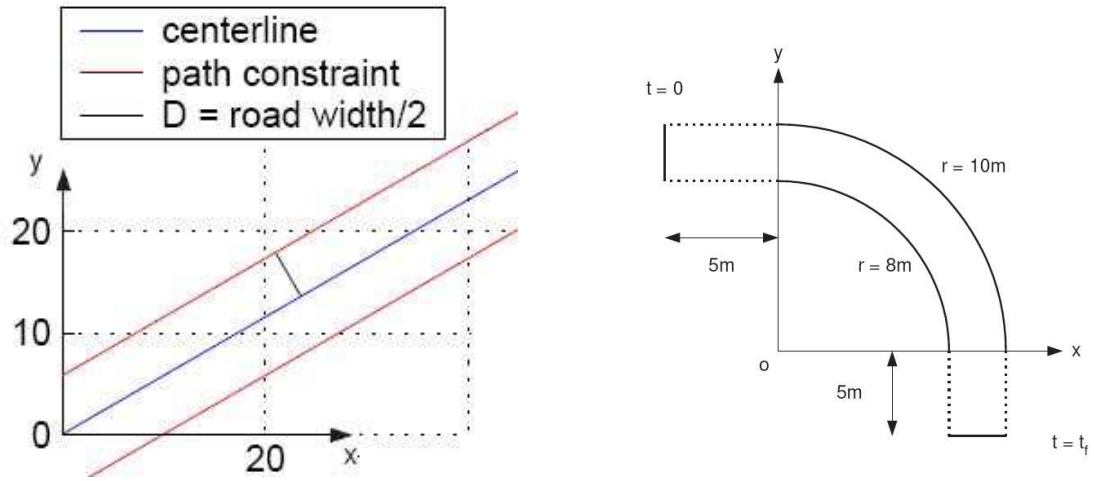
The state constraint for both the path considered are shown in figure 3.7 and can be mathematically described as follows:

- Straight Path (a)
$$120 \leq 20y - 13x \leq 120$$
- Curved Path (b)
$$8 \leq \sqrt{x^2 + y^2} \leq 10 \text{ when } x \geq 0 \text{ and } y \geq 0$$

In the curved path we have added two straight segments of $5[m]$ each before and after the turn so that the car enters the corner after traveling straight and exits the corner in a posture that will lead to straight travel again. The road boundaries of these straight segments are not included in the state constraint. However, the boundary conditions, take into account the width of the road as well.

The initial and final position of the car is within the width of the road, while the longitudinal velocities at the initial and final time and the final time t_f are left free. These conditions guarantee that the vehicle is in the proper direction before and after the corner.

The control inputs are the steering angle δ and the accelerating/braking torques τ_R and τ_F of each axle. The control constraints are



(a) **(b)**

Figure 3.7: Path Constraints.

$$
\begin{aligned}
-0.7 \leq \delta \leq \ \ 0.7 \quad [rad] \\
-1000 \leq \tau_i \leq 1000 \ [Nm],\ i = R, F
\end{aligned}
$$

The constraints on the steering angle is due to physical limitation of the steering mechanism while the one on the accelerating/braking torques reflects the limited power that may be provided by the engine and braking systems.

The optimal necessary condition that are computationally verifiable are:

- Adjoint Equation
  $\dot{\lambda_x} = \frac{\partial H}{\partial x} = 0 \Rightarrow \lambda_x = constant$
  $\dot{\lambda_x} = \frac{\partial H}{\partial x} = 0 \Rightarrow \lambda_x = constant$
- Hamiltonian Value Condition
  $H[t_f] = -1$

- Hamiltonian Evolution Equation
  $\dot{H} = \frac{\partial H}{\partial t} = 0 \Rightarrow H = constant$

Straight Path

In this first case we consider a straight line as given path and (x_f, y_f, ψ_f) as final condition. The choice of this path is made because the analytical solution can be easily computed, therefore a prompt verification of the optimality the solution obtained through the use of PS methods is possible.

In figure 3.9 we can see the optimal condition for the first 2 adjoint variables λ_x and λ_y and the evolution of the Hamiltonian. Those values can be considered as constants, the vibrations are due to the low accuracy of DIDO in determining the dual variables. The control presents the bang-bang behavior predicted.

Curved Path

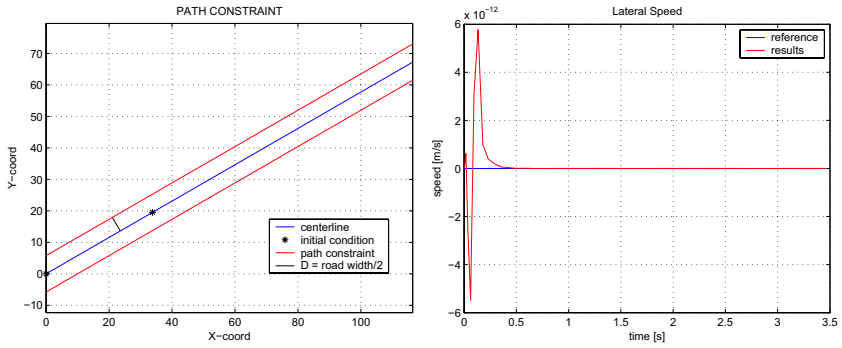
In this second test we consider a curved path of constant radius, preceded and followed by a straight line for numerical stability purpose. The results can not be straightforwardly derived analytically as before, but still some consideration can be made. In particular we can observe how the predicted trajectory is tangent to the inner curb of the road in the middle of the path, while the torque input is switching from braking to full acceleration, showing a well known behavior observed in very trained professional drivers.

3.4 Conclusions

Properties of Pseudospectral methods have been investigated and their effectiveness has been proved through their application to the numerical solution of two optimization problem for mechanical systems.

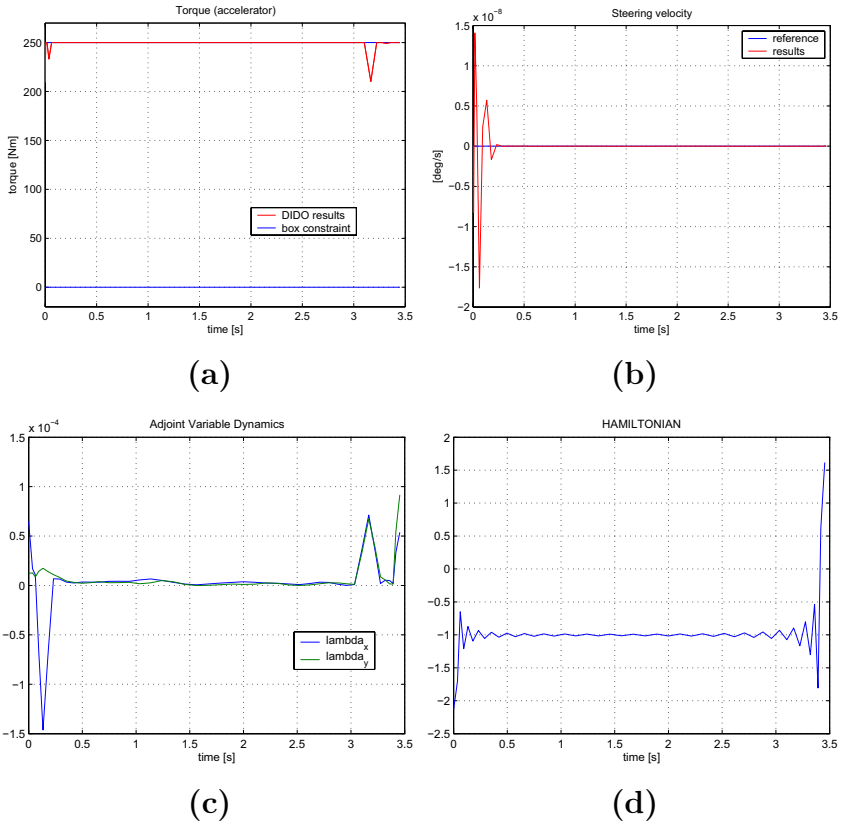
In particular we formulated the problem of trajectory optimization for ground wheeled vehicles as a optimal control problem, in accordance to the current trends in the literature. The problem objective is taking advantage of the full acceleration capacity of the vehicle, minimizing time of travel. The preliminary optimization results indicate that the framework here developed is a good choice to study the problem, not only for simple maneuvers, but also for more complex ones. As side effect the solution of time minimization problem contains the trajectory that has to be followed in order to achieve the results, trajectory that can be used, along with the longitudinal velocity profile, as reference for the optimal vehicle controller previously discussed (see chapter 2).

Therefore numerical optimization schemes provide a powerful tool for off-line analysis of vehicle dynamics, limit operation and trajectory optimization. It allows the study of several vehicle parameters in limit operation which would be extremely difficult using straightforward simulation.



(a) (b)

Figure 3.8: Problem settings (a) and optimal results for lateral speed (b).



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

Figure 3.9: Optimal control input and optimal conditions.

Chapter 4

Rolling Disk

In the following chapters we address the problem of controlling a disk, rolling on a horizontal plane, using only the throttle as control input. We want the disk to follow an assigned path in the plane. The problem is made hard by the high order of underactuation and by the instability of the system. Our controller is based upon an internal manifold and a receding horizon technique. The system tracks a lean angle trajectory by using a backstepping control technique. The lean angle reference trajectory is generated at each instant through a receding horizon algorithm and it is such that, if followed, the system tracks the assigned path with a bounded error.

4.1 Introduction

The control of a rolling disk by throttle only represents a case study for a large class of mechanical systems. It is a high-order underactuated system [31] subject to nonholonomic constraints (pure rolling and no transversal slip). It is also a system with symmetry with respect to the horizontal plane coordinates, in fact its Lagrangian and constraints are invariant with respect to rotations and translations in the plane. Finally it has an unstable internal dynamics (the lean or side inclination angle dynamics).

A mechanical system is called "underactuated" when it presents fewer control inputs than the number of configuration variables. This property is due to various reasons, that can be divided in four main groups:

- dynamics of the system (e.g. spacecraft and underwater vehicles);

- design choices, for reduction of the cost or some practical purposes (e.g. satellites with two thrusters);
- actuator failure (e.g. in a surface vessel or aircraft);
- artificial imposition, to create complex low-order nonlinear systems for the purpose of gaining insight in control of high-order underactuated systems (e.g. the Beam-and-Ball system and the rolling disk).

Almost all mechanical systems possess kinetic symmetry properties. It means that their kinetic energy does not depend on a subset of configuration variables called external variables. Such symmetry properties can be exploited as a means of reducing the complexity of control design.

Many underactuated mechanical systems are subject to nonholonomic constraints. In classical mechanics, a linear constraints is a function $F(q)\dot{q} = 0$, where q denotes the generalized coordinates vector, and is called non-integrable if it can not be written as the time-derivative of some function of the generalized coordinates, i.e. $f(q) = 0$, and thus can not be solved by integration. Nonholonomic constraints are non-integrable ones, and are therefore an essential part of the system dynamics.

Nonholonomic constraints most commonly arise in mechanical systems where constraints are imposed on the motion, for example, underactuated vehicles and underactuated robot manipulators. In addition to classical formulations, nonholonomic constraints can arise in other ways. If the motion of a mechanical system exhibits certain symmetry properties, there exist conserved quantities. If these conserved quantities, for example the angular momentum, are non-integrable, this may be interpreted as a nonholonomic constraint. It should be noted that, in classical mechanics, conserved quantities are not regarded as constraints on a system. In the control community, however, it has been commonly accepted to regard these conserved quantities as constraints that are imposed on the system. Examples of such systems include multi-body spacecraft and underactuated symmetric rigid spacecraft. Nonholonomic constraints also arise as a result of imposing design constraints on the allowable motions of the mechanical system. Examples of such systems include the case of kinematically redundant manipulators and underactuated manipulators. An introduction to nonholonomic control systems is given by Murray et al. in [30].

The control of those systems is currently an active field of research due to their broad applications and is a major open problem. It can be shown that, in general, first-order nonholonomic systems can not be stabilized by

any smooth time-invariant static state-feedback [4].

To our knowledge, the problem of controlling a rolling disk by throttle only has never been investigated so far. Control of a rolling disk has been studied only in the case of two control inputs, throttle and tilt moments, by Frangos and Yavin [20], [17]. They studied the path controllability of the disk and found a control action to track a given trajectory in the horizontal plane. In our work we use a receding horizon technique in order to track a desired path in the plane with a bounded error. At each instant we find a feasible trajectory for the disk which originates from the current position and merges to the path at the time horizon. We call this trajectory "connecting contour", see [18], [19]. Hence we use an internal manifold based control technique, inspired by Getz and Hedrick [22], [21], to find an equilibrium lean angle trajectory to track, in order to follow the connecting contour without the disk falling down.

4.2 Dynamical Model

The model we consider is a falling rolling disk subject to nonholonomic constraints. The disk, represented in figure 4.1, can roll in the horizontal plane without slipping and is subject to gravity acceleration. The configuration space for the disk is $Q = \mathbb{R}^2 \times S^3$ and is parameterized by the generalized coordinates $q = (x, y, \phi, \vartheta, \psi)$, where (x, y) represent the position of the disk in the inertial frame, while ϕ, ϑ and ψ represent the angle between the plane of the disk and the vertical axis, the "self-rotation" angle of the disk and the heading or yaw angle of the disk, respectively. The inertial frame is oriented according to the usual convention adopted in the automotive and aerospace fields, i.e. the x-axis is directed along the positive direction of motion of the vehicle (ψ=0), the z-axis points downward and the y-axis follows. The dynamics of the system can be written using a Lagrangian formalism. The Lagrangian for the unconstrained system is

$$\begin{aligned} L = {} & \tfrac{1}{2}m[(\dot{x} - \dot{\psi}R\sin\phi\cos\psi - R\dot{\phi}\cos\phi\sin\psi)^2 + \\ & +(\dot{y} - R\dot{\psi}\sin\phi\sin\psi + R\dot{\phi}\cos\phi\cos\psi)^2] + \\ & +\tfrac{1}{2}mR^2\dot{\phi}^2\sin^2\phi + \tfrac{1}{2}J(\dot{\phi}^2 + \dot{\psi}^2\cos^2\phi) + \\ & +\tfrac{1}{2}I(\dot{\vartheta} + \dot{\psi}\sin\phi)^2 - mgR\cos\phi \end{aligned} \tag{4.1}$$

where

- m is the mass of the disk;
- I is the moment of inertia about the axis perpendicular to the plane of the disk;
- J is the moment of inertia about an axis in the plane of the disk;
- R is the disk radius.

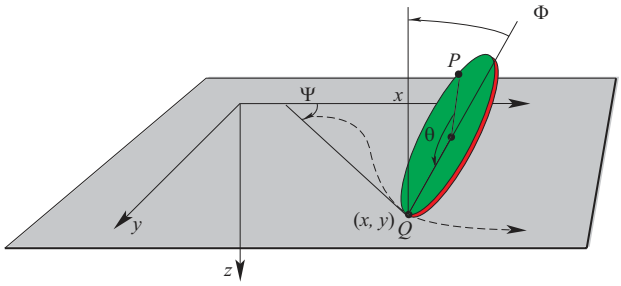


Figure 4.1: View of the rolling disk with the configuration variables.

The following constraints are imposed in order to avoid the wheel to slip laterally and longitudinally:

$$\begin{aligned} \dot{x} &= -R\dot{\vartheta}\cos\psi \\ \dot{y} &= -R\dot{\vartheta}\sin\psi \end{aligned} \tag{4.2}$$

The set of generalized coordinates can be divided into m constrained coordinates and $n-m$ unconstrained, $q=(s,r)$, where m is the number of constraints, see [3]. In our model we have $s=(x,y)$ and $r=(\phi,\psi,\vartheta)$. Now the "constrained" Lagrangian L_c is constructed by substituting the constraints (4.2) in (4.1), and the equation of motion can be written in terms of L_c in the following way:

$$\frac{d}{dt}\frac{\partial L_c}{\partial \dot{r}^\alpha} - \frac{\partial L_c}{\partial r^\alpha} + A^a_\alpha \frac{\partial L_c}{\partial s^\alpha} = \frac{\partial L}{\partial s^b} B^b_{\alpha\beta}\dot{r}^{,\beta} \tag{4.3}$$

where

$$A(r,s)[\dot{r}\ \dot{s}]^T = 0$$

is the velocity constraint and

$$B^b_{\alpha\beta} = \frac{\partial A^b_\alpha}{\partial \dot{r}^\beta} - \frac{\partial A^b_\beta}{\partial \dot{r}^\alpha} + A^a_\alpha \frac{\partial A^b_\beta}{\partial s^a} - A^b_\beta \frac{\partial A^b_\alpha}{\partial s^a}.$$

The dynamics of the unconstrained coordinates is thus obtained in terms of the constrained and unconstrained coordinates:

$$M(s,r)\ddot{r} + C(s,r,\dot{r}) + G(s,r) = T. \tag{4.4}$$

This general form can be simplified in our model thanks to the symmetry of the system. For the rolling disk, in fact, the following holds:

$$L(s,r), A(s,r) \text{ are both independent from } s.$$

Taking this fact into account, after straightforward calculations, we obtain the dynamics of the unconstrained coordinates in the form:

$$M(r)\ddot{r} + C(r,\dot{r}) + G(r) = T, \tag{4.5}$$

where

$$M(r) = \begin{bmatrix} mR^2 + J & 0 & 0 \\ 0 & (mR^2 + I)s_\phi^2 + Jc_\phi^2 & (mR^2 + I)s_\phi \\ 0 & (mR^2 + I)s_\phi & (mR^2 + I) \end{bmatrix}$$

$$C(r,\dot{r}) = \begin{bmatrix} -(mR^2 + I)c_\phi \dot{\psi}\dot{\vartheta} - (mR^2 + I - J)s_\phi c_\phi \dot{\psi}^2 \\ Ic_\phi \dot{\psi}\dot{\vartheta} + 2(mR^2 + I - J)s_\phi c_\phi \dot{\psi}\dot{\varphi} \\ (2mR^2 + I)c_\phi \dot{\psi}\dot{\varphi} \end{bmatrix}$$

$$G(r) = \begin{bmatrix} -mgR\sin\phi \\ 0 \\ 0 \end{bmatrix}$$

and the torque vector

$$T = \begin{bmatrix} 0 \\ 0 \\ \tau \end{bmatrix}.$$

See [3] for more details on the calculations of the dynamics.

The system has only one control input with three unconstrained variables, so it is an underactuated reduced system of order two [31].

If the disk starts from the vertical position ($\phi = 0$) with no lean and yaw velocity ($\dot{\phi} = 0$ and $\dot{\psi} = 0$), the three scalar equations defined by (4.5) are completely decoupled and the control input drives only the $\dot{\vartheta}$ dynamics. Hence only a velocity regulation along the initial direction is possible, while the lateral dynamics are not controllable. The problem is thus well posed when we work far from the condition $(\phi, \dot{\phi}, \dot{\psi}) = (0, 0, 0)$.

The lean angle dynamics is not driven directly by τ; instead it is coupled with the dynamics of ψ and ϑ by the centrifugal acceleration term $\dot{\psi}\dot{\vartheta}$. Thus it can be seen as the zero dynamics of the system (w.r.t. the output $\dot{\psi}\dot{\vartheta}$). Being such dynamics unstable, the system is non-minimum phase.

For control purpose, we rewrite the system dynamics in state space form. Since the dynamics is invariant w.r.t. the yaw angle ψ and the self-rotation angle ϑ, they can be neglected and the state space vector results:

$$x = [x_1\ x_2\ x_3\ x_4]^T = [\phi\ \dot{\phi}\ \dot{\psi}\ \dot{\vartheta}]^T.$$

Multiplying both sides of 4.5 by $M(s,r)^{-1}$ (the matrix of mass $M(s,r)$ is certainly nonsingular for $\phi \in$(-$\pi/2$, $\pi/2$)) it follows:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f_{\phi 1}(x_1) + f_{\phi 2}(x_1)x_3x_4 + f_{\phi 3}(x_1){x_3}^2; \\
\dot{x}_3 &= f_{\psi 1}(x_1)x_2x_4 + f_{\psi 2}(x_1)x_2x_3 + f_{\psi 3}(x_1)\tau; \\
\dot{x}_4 &= f_{\vartheta 1}(x_1)x_2x_4 + f_{\vartheta 2}(x_1)x_2x_3 + f_{\vartheta 3}(x_1)\tau.
\end{aligned} \tag{4.6}$$

with

$$f_{\phi 1}(x_1) = mgR\sin x_1;$$

$$f_{\phi 2}(x_1) = \tfrac{mR^2+I}{mR^2+J}\cos x_1;$$

$$f_{\phi 3}(x_1) = \tfrac{mR^2+I-J}{mR^2+J}\sin x_1 \cos x_1;$$

$$
\begin{aligned}
f_{\psi 1}(x_1) &= -\tfrac{I}{J\cos x_1};\\
f_{\psi 2}(x_1) &= -\tfrac{I\sin x_1}{J\cos x_1} + \tfrac{2\sin x_1}{\cos x_1};\\
f_{\psi 3}(x_1) &= -\tfrac{\sin x_1}{J\cos^2 x_1};\\
f_{\vartheta 1}(x_1) &= \tfrac{I\sin x_1}{J\cos x_1};\\
f_{\vartheta 2}(x_1) &= \tfrac{I\sin^2 x_1}{J\cos x_1} - \tfrac{2}{\cos x_1} + \tfrac{I\cos x_1}{mR^2+I};\\
f_{\vartheta 3}(x_1) &= \tfrac{(mR^2+I)\sin^2 x_1+J\cos^2 x_1}{mR^2+I)J\cos^2 x_1}.
\end{aligned}
$$

Chapter 5

Rolling Disk: Control Strategy and Results

In this chapter we present the developed control strategy and the simulation results.

5.1 Lean angle control design

In this section we present a control strategy to track a preassigned "lean angle" trajectory. The leaning control is based on a backstepping control technique, see [34]. In order to apply it, the system has to be in the so called strict feedback form, that is

$$\begin{aligned} \dot{z}_1 &= f_{11}(z_1)z_1 + f_{12}(z_1)z_2 \\ \dot{z}_2 &= f_{21}(z_1, z_2)z_1 + f_{22}(z_1, z_2)z_2 + f_{23}(z_1, z_2)z_3 \\ &... \\ \dot{z}_n &= f_{n1}(z_1, ..., z_n)z_1 + ... + f_{nn}(z_1, ..., z_n)z_n + \\ &\quad + g(z_1, ..., z_n)u. \end{aligned} \tag{5.1}$$

Even if the dynamics of the disk is not in such form, it can be reduced to (5.1) by way of an approximation on the lean angle dynamics and a change of coordinates. First of all let us ignore, for control design purpose only, the term $f_{\phi 3}(x_1){x_3}^2$ of the lean angle dynamics, then let us define the new set of coordinates

$$\begin{aligned} \zeta_1 &= \phi; \\ \zeta_2 &= \dot{\phi}; \\ \zeta_3 &= \dot{\psi}\dot{\vartheta}; \\ \zeta_4 &= \dot{\vartheta}. \end{aligned}$$

The new system dynamics becomes

$$\begin{aligned}
\dot{\zeta}_1 &= \zeta_2;\\
\dot{\zeta}_2 &= f_{\phi 1}(\zeta_1) + f_{\phi 2}(\zeta_1)\zeta_3;\\
\dot{\zeta}_3 &= f_{\psi\vartheta 1}(\zeta_1)\zeta_2\zeta_4^2 + f_{\psi\vartheta 2}(\zeta_1)\tfrac{\zeta_3^2}{\zeta_4^2}\zeta_2 + \\
&\quad + f_{\psi\vartheta 3}(\zeta_1)\zeta_2\zeta_3 + f_{\psi\vartheta 4}(\zeta_1)\tau;\\
\dot{\zeta}_4 &= f_{\vartheta 1}(\zeta_1)\zeta_2\zeta_4 + f_{\vartheta 2}(\zeta_1)\zeta_2\tfrac{\zeta_3}{\zeta_4} + f_{\vartheta 3}(\zeta)\tau.
\end{aligned} \tag{5.2}$$

with

$$\begin{aligned}
f_{\psi\vartheta 1}(\zeta_1) &= -\tfrac{I}{J\cos\zeta_1};\\
f_{\psi\vartheta 2}(\zeta_1) &= \tfrac{I\sin^2\zeta_1}{J\cos\zeta_1} - \tfrac{2}{\cos\zeta_1} + \tfrac{I}{mR^2+I}\cos\zeta_1;\\
f_{\psi\vartheta 3}(\zeta_1) &= \tfrac{2\sin\zeta_1}{\cos\zeta_1};\\
f_{\psi\vartheta 4}(\zeta) &= \tfrac{[(mR^2+I)\sin^2\zeta_1 + J\cos^2\zeta_1]\zeta_3/\zeta_4}{(mR^2+I)J\cos^2\zeta_1} + \\
&\quad - \tfrac{(mR^2+I)\zeta_4\sin\zeta_1}{(mR^2+I)J\cos^2\zeta_1}.
\end{aligned}$$

The subsystem given by the first three equations is in the form (5.1). As the velocity of the path following is not a concern, the dynamics of $\zeta_4{=}\dot{\vartheta}$ can be ignored. Observe that if ζ_3 remains bounded the only case ζ_4 could diverge is when $\dot{\psi}$ goes to zero. But this can happen only when also ϕ and $\dot{\phi}$ go to zero, which violates the assumption of working far from the condition $(\phi, \dot{\phi}, \dot{\psi}) = (0, 0, 0)$.

Now we choose ζ_3 as the virtual control input to follow the lean angle reference. Using dynamic inversion for the lean angle dynamics we have:

$$\begin{aligned}
\overline{\zeta_3} = \tfrac{1}{f_{\phi 2}(\zeta_1)}(&-f_{\phi 1}(\zeta_1) - k_\phi(\zeta_1 - \phi_{ref}) + \\
&-k_{\dot{\phi}}(\zeta_2 - \dot{\phi}_{ref}) + \ddot{\phi}_{ref}).
\end{aligned} \tag{5.3}$$

At this point we can make ζ_3 follow $\overline{\zeta_3}$, so we obtain the following throttle control:

$$\tau = \tfrac{1}{f_{\psi\vartheta 4}(\zeta_1)}\left(f_{\zeta_3}(\zeta) - k_\zeta(\zeta_3 - \overline{\zeta_3}) + \dot{\overline{\zeta_3}}\right) \tag{5.4}$$

where

$$f_{\zeta_3}(\zeta) = f_{\psi\vartheta 1}(\zeta_1)\zeta_2\zeta_4^2 + f_{\psi\vartheta 2}(\zeta_1)\frac{\zeta_3^2}{\zeta_4^2}\zeta_2 + \\ + f_{\psi\vartheta 3}(\zeta_1)\zeta_2\zeta_3;$$

and

$$\dot{\overline{\zeta_3}} = \frac{mR^2+J}{(mR^2+I)\cos\zeta_1}\Big(-\frac{mgR\zeta_2\cos\zeta_1}{mR^2+J} - k_\phi(\zeta_2 - \dot{\phi}_{ref}) + \\ -k_{\dot{\phi}}(\dot{\zeta}_2 - \ddot{\phi}_{ref})\Big) + \phi^{(3)}_{ref} + \\ + \frac{(mR^2+J)\zeta_2\sin\zeta_1}{(mR^2+I)\cos^2\zeta_1}\Big(-\frac{mgR\sin\zeta_1}{mR^2+J} + \\ -k_\phi(\zeta_1 - \phi_{ref}) - k_{\dot{\phi}}(\zeta_2 - \dot{\phi}_{ref}) + \ddot{\phi}_{ref}\Big).$$

The resulting closed-loop system is:

$$\begin{aligned} \dot{\zeta}_1 &= \zeta_2; \\ \dot{\zeta}_2 &= -k_\phi(\zeta_1 - \phi_{ref}) - k_{\dot{\phi}}(\zeta_2 - \dot{\phi}_{ref}) + \ddot{\phi}_{ref} + \\ &\quad + f_{\phi 2}(\zeta_1)\tilde{\zeta}_3; \\ \dot{\zeta}_3 &= -k_\zeta(\zeta_3 - \overline{\zeta_3}) + \dot{\overline{\zeta_3}} \end{aligned} \tag{5.5}$$

where

$$\tilde{\zeta}_3 = \zeta_3 - \overline{\zeta_3}.$$

From the last equation ζ_3 converges to $\overline{\zeta_3}$, that is $\tilde{\zeta}_3$ goes to zero. Since the (ζ_1, ζ_2) subsystem is a zero-GAS (Globally Asymptotically Stable) linear system, it is ISS (Input to State Stable) [34], hence ζ_1 converges to the desired value.

The term $f_{\psi\vartheta 4}(\zeta_1)$ in (5.4) becomes zero when $\zeta_1{=}\phi$ and $\zeta_3{=}\dot{\psi}$ go to zero, thus making our control singular. If this happens, then $\zeta_2{=}\dot{\phi}$ also has to be zero. However in section 2 we showed that the condition $(\phi, \dot{\phi}, \dot{\psi}) = (0, 0, 0)$ is in somehow pathological in that it makes the system uncontrollable. Nevertheless the singularity is avoided under the assumption that the system works far from that condition.

5.2 Receding horizon control

Our objective is to follow a trajectory in the plane that is close, in some sense, to a desired path. Since we handle a control only, we will not be concerned

if the velocity of the path is followed. The lean angle control we designed in the previous section allows to track a lean angle trajectory, but it does not ensure tracking an exact path in the plane. The dynamics of the disk (hence the lean angle dynamics) is, in fact, invariant with respect to x, y and ψ, hence the same lean trajectory could correspond to the same path rotated or translated in the plane. Moreover, in dependence of the rotation velocity $\dot{\vartheta}$, the same lean angle trajectory could correspond to different trajectories of $\dot{\psi}$ and hence to different curvatures of the path. In fact the dynamics of ϕ

$$\ddot{\phi} = f_{\phi 1}(\phi) + f_{\phi 2}(\phi)\dot{\psi}\dot{\vartheta} + f_{\phi 3}(\phi)\dot{\psi}^2 \tag{5.6}$$

is invariant with respect to every $\dot{\psi}$ and $\dot{\theta}$ such that

$$f_{\phi 2}(\phi)\dot{\psi}\dot{\vartheta} + f_{\phi 3}(\phi)\dot{\psi}^2 = a(\phi(t)), \tag{5.7}$$

with $a\left(\phi(t)\right)$ satisfying (5.6). Since

$$\dot{\psi} = -\sigma(s(t))R\dot{\vartheta}(t), \tag{5.8}$$

where

$$s(t) = \int_0^t R\dot{\vartheta}(\tau)d\tau,$$

then different possible paths are compatible with (5.7), (5.8). Our idea is to use a predictive control strategy to converge to the exact path in the manifold of the paths compatible with the same lean angle trajectory. We generate a trajectory in the plane which is feasible for the disk, starting from the current position and orientation and merging to the path at look ahead distance D, figure 5.1. Then we find a ϕ trajectory that, if followed, makes the disk stay close to the generated path and, at this point, we use the previous closed-loop control action.

The control strategy can be summarized as follows:

1. at time t the contour and its first p derivatives are computed at look-ahead distance D;
2. a feasible trajectory which connects at look-ahead distance D to the path is generated. We call it the trajectory connecting contour;
3. the angular velocity trajectory $\dot{\psi}_c(t)$ associated to the connecting contour is computed, assuming knowledge of the velocity of the disk along the path;

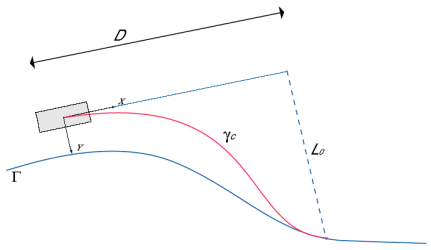


Figure 5.1: Connecting contour.

4. an equilibrium lean angle trajectory compatible with $\dot{\psi}_c(\mathrm{t})$ (and with the velocity of the disk) is generated;

5. the lean angle control action of section 5.1 is applied to track the equilibrium ϕ trajectory;

6. the previous steps are repeated recursively.

So far we can see that this strategy is predictive and closed loop, but we need to specify how to make it causal, since we suppose to know the velocity of the disk in the future (and that velocity depends on the control action which is what we want to compute). Observe, however, that the actual trajectory of the disk is not any single connecting contour but, rather, the envelope of the connecting contours.

Assume that, in the moving frame attached to the disk, the path can be represented locally as:

$$y = \gamma(x, t) \; x \in [0, D]. \tag{5.9}$$

The connecting contour should satisfy two conditions: it should be feasible for the system and it should satisfy some optimality criterion. The simplest criterion is to design a trajectory which intersects the path with the proper derivatives. Therefore the connecting contour $\gamma_c(x, t)$ must satisfy the following boundary conditions:

$$\begin{bmatrix} \gamma_c(0,t) \\ \frac{\partial \gamma_c}{\partial x}(0,t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} \gamma_c(D,t) \\ \frac{\partial \gamma_c}{\partial x}(D,t) \\ \vdots \\ \frac{\partial^{(p)} \gamma_c}{\partial x}(D,t) \end{bmatrix} = \begin{bmatrix} \gamma(D,t) \\ \frac{\partial \gamma}{\partial x}(D,t) \\ \vdots \\ \frac{\partial^{(p)} \gamma}{\partial x}(D,t) \end{bmatrix} \tag{5.10}$$

Among all possible connecting contours, we chose polynomials (which have the advantage of minimizing the overall curvature) and set $p = 2$ for the simulations. If we now suppose to know the velocity of the disk along the connecting contour $\dot{\vartheta}_c(t)$, we can compute the angular velocity $\dot{\psi}_c(t)$ associated to the path as:

$$\dot{\psi}_c = -\sigma_c(s(t)) R \dot{\vartheta}_c(t). \tag{5.11}$$

Having determined the value of $\dot{\psi}_c(t)$, we seek a bounded lean angle trajectory compatible with $\dot{\psi}_c(t)$ and $\dot{\vartheta}_c(t)$. This trajectory is a time varying equilibrium for the lean angle dynamics and is obtained by imposing:

$$\begin{aligned} 0 = \tfrac{1}{mR^2+J}(mgR \sin \phi_c + (mR^2 + I)\dot{\psi}_c \dot{\vartheta}_c \cos \phi_c + \\ +(mR^2 + I - J)\dot{\psi}_c^2) \end{aligned} \tag{5.12}$$

At this point, observe that the receding horizon algorithm involving the control action is applied only at the first time step. Since the lean angle control action is not predictive we only need the value of $\dot{\phi}_c(t)$ at the current time, which we call $\bar{t}$. This implies that we need the value of $\dot{\vartheta}_c(t)$ at $\bar{t}$ only. Hence we can use its current value $\dot{\vartheta}\left(\bar{t}\right)$. This means we suppose to follow the path at the current velocity, at least in the first time interval. In the lean angle control we would also need the value of $\dot{\phi}_c\left(\bar{t}\right)$, $\ddot{\phi}_c\left(\bar{t}\right)$ and $\phi_c^{(3)}\left(\bar{t}\right)$. Since in the control design we made an approximation on the ϕ dynamics and we are not interested in an exact tracking, we can neglect the last two feedforward terms. On the contrary $\dot{\phi}_c\left(\bar{t}\right)$ can be computed by considering the approximated value of $\phi_c\left(\bar{t}\right)$

$$\phi_c\left(\bar{t}\right) = -\arctan\left(\frac{(mR^2+I)\dot{\psi}_c\left(\bar{t}\right)\dot{\vartheta}_c\left(\bar{t}\right)}{mgR}\right) \tag{5.13}$$

as:

$$\dot{\phi}_c\left(\bar{t}\right) = -\frac{mgR(mR^2+I)\left(\ddot{\psi}_c\left(\bar{t}\right)\dot{\vartheta}_c\left(\bar{t}\right)+\dot{\psi}_c\left(\bar{t}\right)\ddot{\vartheta}_c\left(\bar{t}\right)\right)}{(mgR)^2+\left((mR^2+I)\dot{\psi}_c\left(\bar{t}\right)\dot{\vartheta}_c\left(\bar{t}\right)\right)^2} \tag{5.14}$$

where $\ddot{\vartheta}_c\left(\bar{t}\right)$ is known and

$$\ddot{\psi}_c\left(\bar{t}\right) = \tfrac{\partial \sigma_c}{\partial s}|_{s=s\left(\bar{t}\right)} R^2 \dot{\vartheta}_c^2\left(\bar{t}\right) - \sigma_c\left(s\left(\bar{t}\right)\right) R \ddot{\vartheta}_c\left(\bar{t}\right).$$

5.3 Simulation results

In this section we show the simulations results of the controlled disk following a circle, a parabolic path and a chicane. The disk parameters used in the simulations are:

- $m = 10[Kg]$;
- $R = .3[m]$;
- $I = .4[Kg \cdot m^2]$;
- $J = .2[Kg \cdot m^2]$;
- $g = 9.8[m/s^2]$.

Figure 5.2 shows the disk tracking a circle path. The disk starts from the condition $x(0) = .5[m]$, $y(0) = 2.1[m]$, $\phi(0) = 70[deg]$, $\dot{\phi}(0) = 0[deg/s]$, $\psi(0) = 45[deg]$, $\dot{\psi}(0) = 25[deg/s]$, $\dot{\vartheta}(0) = 10\pi/R[rad/sec]$, recovers to the right inclination and converges to the circle. In figure 5.2 (a) the resulting path in the ground plane is shown. Figure 5.2 (b) shows the lean angle trajectory, compared with the equilibrium trajectory, and its derivative. In figure 5.2(c) the yaw trajectory and its derivative are shown and finally the control input and the rotation velocity are depicted in figure 5.2(c).

Figure 5.3 shows the same graphs in the case of a parabolic path. The initial condition is set to $x(0) = .5[m]$, $y(0) = 2.1[m]$, $\phi(0) = 55[deg]$, $\dot{\phi}(0) = 0[deg/s]$, $\psi(0) = -20[deg]$, $\dot{\psi}(0) = 25[deg/s]$, $\dot{\vartheta}(0) = 10\pi/R[rad/sec]$.

Finally figure 5.4 represents the path following of a chicane. In this case figure 5.4 (a) is a 3D image representing the position and orientation of the disk.

5.4 Conclusions and future work

The problem of controlling a rolling disk has been studied. The disk is made to follow an assigned path in the plane using throttle only. We used a receding horizon control strategy to generate, at each instant, a feasible trajectory in the plane starting from the current position of the disk and merging to the path at a look-ahead distance. A bounded equilibrium lean trajectory, compatible with this trajectory, is computed and tracked by means of a back-stepping control strategy. The objective of this work was not only to solve

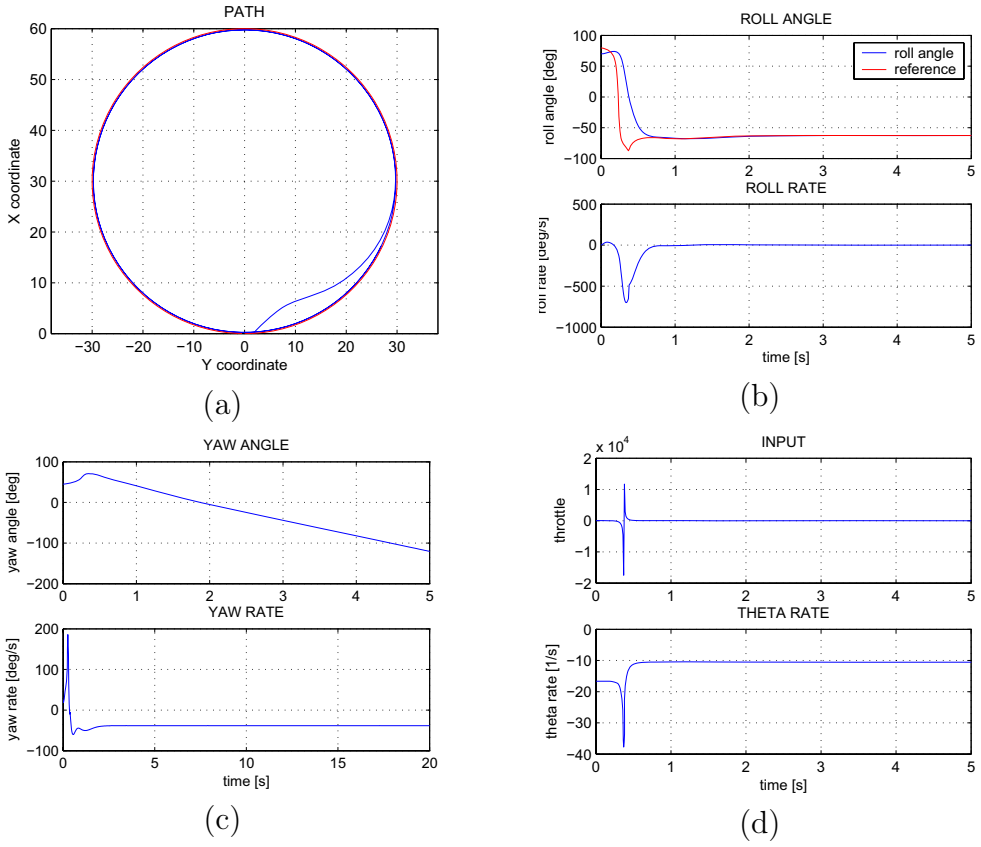


Figure 5.2: Following of a circular path.

the specific problem, but also to introduce a general control technique for controlling a large class of both nonminimum phase and high order underactuated systems.

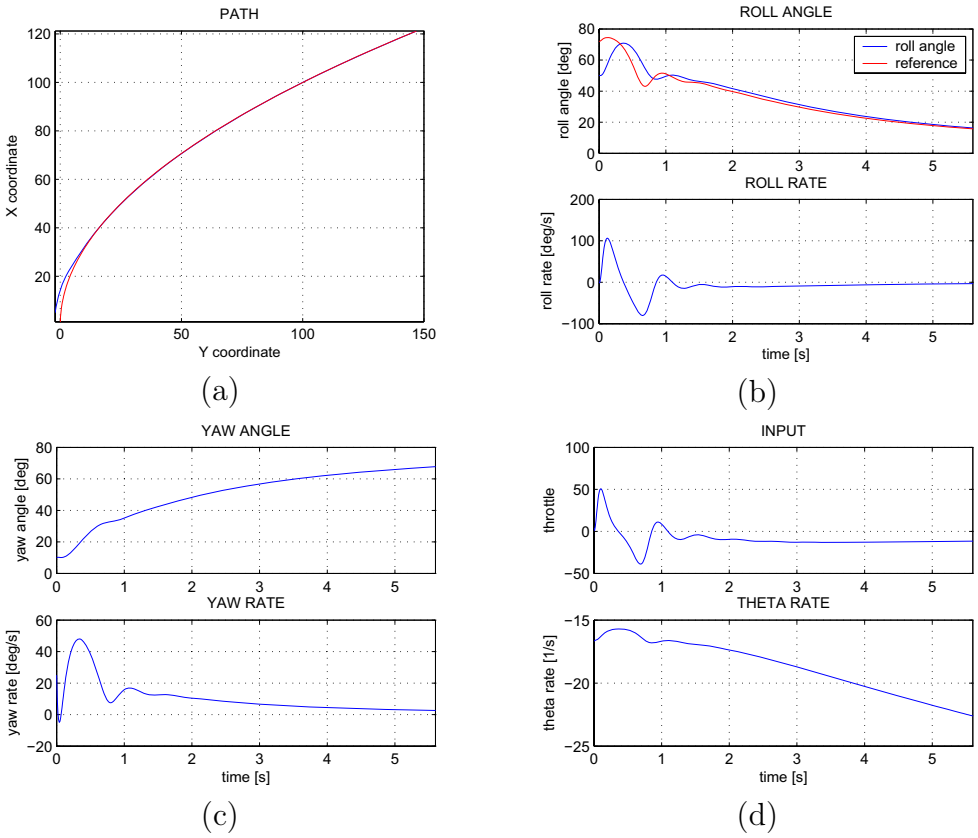


Figure 5.3: Following of a parabolic path.

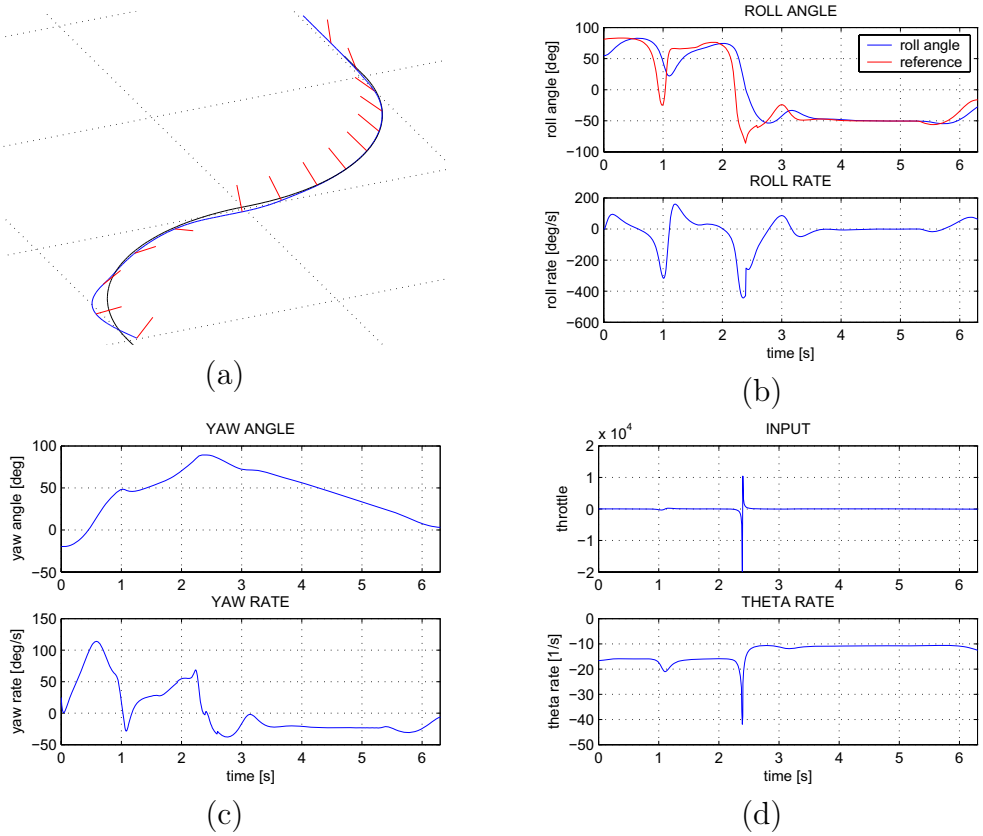


Figure 5.4: Following of a chicane.

Perspectives

In the first part of the dissertation we studied the problem of controlling a ground vehicle along a preassigned trajectory. We developed an effective closed loop control scheme based on Model Predictive Control techniques.

Through the analysis of its behavior in several trials it can be seen how the controller here developed reaches the control objective in a more effective way than the previous one. The controller gives excellent results when the system lays in the admissible set, while its performance rapidly decays when the tires slip angles and the vehicle lateral velocity grow too quickly and the model goes beyond its limits. To improve its performances an enrichment of the vehicle model is required; switching from the bicycle model to a two-track would allow the system to take in account the vehicle pitch and roll dynamics, which can't be neglected when the vehicle is subject to abrupt direction changes or sharp speed variations. This improvement would be useful to simulate a rally driver.In fact rally driving technique is quite different from the common driving and even from other types of car racing, such as F1 racing driving. Rally drivers drive through corners inducing drifting, that is driving at high slip vehicle angles α. In such situations drivers manage to balance the oversteering vehicle by steering the front wheels to the opposite direction of the corner (applying "opposite lock"). Such a technique could be reproduced by a control algorithm that includes roll and pitch dynamics and able to handle high slip angle values.

Another direction of investigation is in the area of the time scheduling of the optimal control. Currently an optimization problem is solved at each simulation step, and its solution is applied for a single step. This approach causes a lot of computational work, which can be reduced, obtaining a faster and computationally cheaper algorithm, by making the number of steps where the optimal control is applied dependent on the execution conditions; if the vehicle after a step is still close to the initial condition then the optimal control recalculated will be very similar to the one already found, so the smartest choice is continuing to apply the old one as long as the system conditions

aren't changed substantially, when the control needs to be recalculated. In this way the control could deal with "calm" situation, for example a straight path with a constant velocity profile, where the control is recalculated once in a while, and with more demanding ones, for example a sharp bend or an abrupt speed change, where the control needs to be calculated more often to obtain satisfactory performances.

In the vehicle prototyping process it is required not only to verify the vehicle behavior in extreme guidance conditions, but also its response while driving in normal conditions, to study its handling and maneuverability. A model for the human driver is therefore essential. In particular a real driver can only rely upon their visual feedback, which gives them the vehicle position with respect to the road and its forward speed, and other data that can be derived from the other senses, which are generically called proprioception. Data harvest and analysis are not immediate, but are subjected to physical transmission delays. Furthermore the control action is not continuously generated, but it's calculated only when it is needed. The controller can be easily modified to simulate the human driver, adding suitable delay blocks, tuned on the human frequencies, and modifying the control calculation frequency moving from a fixed scheme to one in which its activation is tied to road conditions, as explained above. Tuning those parameters allows the controller to represent different driver categories, from the inexperienced one to the professional driver.

We have also introduced pseudospectral methods and their application to the numerical solution of optimal control problem for mechanical systems; their properties have been investigated and their effectiveness proved. In particular we formulated the problem of trajectory optimization for ground wheeled vehicles as a optimal control problem, in accordance to the current trends in the literature. The preliminary optimization results indicate that the framework here developed is a good choice to study the problem, not only for simple maneuvers, but also for more complex ones. As side effect the solution of time minimization problem contains the trajectory that has to be followed in order to achieve the results, trajectory that can be used, along with the longitudinal velocity profile, as reference for the optimal vehicle controller previously developed.

In the second part of the thesis we studied the control of a class of systems called underactuated, to develop a general control scheme that exploits symmetries and other common characteristics.

Underactuated systems are mechanical systems with fewer controls than the number of configuration variables. We provided as example the rolling

disk system. The disk is made to follow an assigned path in the plane using throttle only. We used a receding horizon control strategy to generate, at each instant, a feasible trajectory in the plane starting from the current position of the disk and merging to the path at a look-ahead distance. A bounded equilibrium lean trajectory, compatible with this trajectory, is computed and it is tracked by means of a beckstepping control strategy. This control technique is for controlling a large class of both nonminimum phase and high order underactuated systems. Future directions of research include studying the convergence of the method and improve the control technique to achieve a zero tracking error instead of a bounded one.

Appendix A

One Track Model: Motion Equation

In this chapter it is explained the method that has been used to obtain the motion equation and subsequently the state variable model.

A.1 Lagrangian Motion Equation

A system dynamic description can be obtained through the lagrangian formalism.

The *Lagrangan L* is defined as:

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q)$$

where T is the kinetic energy, V is the potential energy, q is the generalized coordinates vector. The equations of motion are obtained by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i$$

where τ_i is the generalized force conjugated to the generalized velocity $\dot{q}_i$. The kinetic energy of the i-th rigid body can be written as

$$T_p(q, \dot{q}) = \frac{1}{2} m_p v_p(q)^2 + \frac{1}{2} \omega_p(q)^T I_p \omega_p(q)$$

where $\omega_p(q)$ is the body angular velocity given by $R_p^T(q)\dot{R}_p(q)$.

Using a coordinate frame for the euclidian space $\mathbb{R}^3$ with the z-axis pointing down, the potential energy of the p-th rigid body due to gravity can be expressed by

$$V_p(q) = -m_p g z_p(q)$$

Computing the Euler-Lagrange's equations

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i$$

it can be noticed that motion equations assumes the following structure

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau$$

and it turns out that

$$M(q)_{ij} = \sum_p m_p\left(\frac{\partial x_p}{\partial q_i}\frac{\partial x_p}{\partial q_j} + \frac{\partial y_p}{\partial q_i}\frac{\partial y_p}{\partial q_j} + \frac{\partial z_p}{\partial q_i}\frac{\partial z_p}{\partial q_j}\right) + \omega_p^i(q)^T I_p \omega_p^j(q)$$

where $\omega_p^i(q) = R_p^T(q)\dfrac{\partial R_p(q)}{\partial q_i}$ [1].

$C_i(q,\dot{q})$ is the Coriolis vector and it results to be

$$C_i(q,\dot{q}) = \sum_{j,k} \Gamma_{ijk}(q)\cdot \dot{q}_j \dot{q}_k$$

where Γ_{ijk} is made of

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}(q)}{\partial q_k} + \frac{\partial M_{ik}(q)}{\partial q_j} - \frac{\partial M_{kj}(q)}{\partial q_i}\right)$$

For multi-body systems it results:

$$\Gamma_{ijk} = \sum_p m_p\left(\frac{\partial x_p}{\partial q_i}\frac{\partial^2 x_p}{\partial q_i \partial q_k} + \frac{\partial y_p}{\partial q_i}\frac{\partial^2 y_p}{\partial q_i \partial q_k} + \frac{\partial z_p}{\partial q_i}\frac{\partial^2 z_p}{\partial q_i \partial q_k}\right)$$

$$+\frac{1}{2}\omega_p^i(q)^T I_p\left(\frac{\partial \omega_p^j(q)}{\partial q_k} + \frac{\partial \omega_p^k(q)}{\partial q_j}\right)$$

[1] Given $\omega = (\omega_x, \omega_y, \omega_z)^T$, $\hat{\omega}$ indicates the linear operator associated to the product, that is $\hat{\omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$

$$+\frac{1}{2}\omega_p^j(q)^T I_p(\frac{\partial \omega_p^i(q)}{\partial q_k} - \frac{\partial \omega_p^k(q)}{\partial q_i})$$
$$+\frac{1}{2}\omega_p^k(q)^T I_p(\frac{\partial \omega_p^i(q)}{\partial q_j} - \frac{\partial \omega_p^j(q)}{\partial q_i})$$

Finally, the potential forces vector G can be obtained from

$$G_i(q) = \sum_p m_p \frac{\partial z_p}{\partial q_i}$$

A.2 The Model

The model is a single rigid body that can move on the plane. It is described by this set of generalized coordinates:

- ψ is the yaw angle;
- x is the abscissa of the rear wheel contact point;
- y is the ordinate of the rear wheel contact point.

The steering axis is considered vertical with respect to ground, the wheel are thin, in order to consider the contact surface as a point. The z axis of the inertial frame (or world frame) is pointing downward;in this way clockwise rotation are positive.The rotation matrix R which describes the rigid body orientation can be written as [2]

$$R = R_z(\psi) = \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The Lagrangian of the system can be written as

$$L = \frac{1}{2}m(\dot{x}_{CM}^2 + \dot{y}_{CM}^2) + \frac{1}{2}\omega^T I \omega + mgh$$

where:

- m is the vehicle mass;

[2] In the rest of the chapter the following notation will be used, for the sake of brevity: $c_\psi := \cos(\psi)$, $s_\psi := \sin(\psi)$, etc....

- x_{CM} is the center of mass abscissa;
- y_{CM} is the center of mass ordinate;
- ω is the body angular velocity of the rigid body ($\hat{\omega} = R^T \dot{R}$);
- h is the center of mass height;
- p is the vehicle wheelbase (i.e. the distance between the front and the rear wheel contact point);
- b is the distance between the projection on the ground of the center of mass and the rear wheel contact point;
- g is the gravity acceleration.

It is straightforward to compute the position of the center of mass:

$$\begin{cases} x_{CM} = x + bc_\psi \\ y_{CM} = y + bs_\psi \\ z_{CM} = -h \end{cases}$$

The angular velocity ω in the body frame is, as previously noticed, equal to

$$\begin{aligned} \hat{\omega} &= R^T \dot{R} = R^T \tfrac{\partial R}{\partial x}\dot{x} + R^T \tfrac{\partial R}{\partial y}\dot{y} + R^T \tfrac{\partial R}{\partial \psi}\dot{\psi} = \\ &= \hat{\omega}^x \dot{x} + \hat{\omega}^y \dot{y} + \hat{\omega}^\psi \dot{\psi} \end{aligned}$$

From R defitinition it follows that

$$\hat{\omega}^\psi = R^T \tfrac{\partial R}{\partial \psi} = R_x^T(\varphi) R_z^T(\psi) \tfrac{\partial R_z(\psi)}{\partial \psi} R_x(\varphi) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

consequently

$$\omega^x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \omega^y = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \omega^\psi = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

In the following table the derivatives of x_{CM}, y_{CM} and the components of the angular velocity ω are shown

	$\frac{d}{dx}$	$\frac{d}{dy}$	$\frac{d}{d\psi}$
x_{CM}	1	0	$-bs_\psi - hs_\varphi c_\psi$
y_{CM}	0	1	$bc_\psi - hs_\varphi s_\psi$
z_{CM}	0	0	0
ω	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

(A.1)

The mass matrix $M(q)$ can be computed using the data in table (A.1), obtaining:

$$\begin{pmatrix} m & 0 & -mbs_\psi \\ 0 & m & mbc_\psi \\ -mbs_\psi & mbc_\psi & mb^2 + I_{zz} \end{pmatrix}$$

In the same way Coriolis forces vector C and potential forces vector G can be computed

$$C(q, \dot{q}) = \begin{pmatrix} -mbc_\psi \dot{\psi}^2 \\ -mbs_\psi \dot{\psi}^2 \\ 0 \end{pmatrix} \tag{A.2}$$

$$G(q) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{A.3}$$

The potential forces vector G is identically zero for this system, therefore it will be neglected in the rest of the chapter.

A.2.1 Change of coordinates

Analyzing the obtained model structure it can be noticed that the equations don't depend on x, y, $\dot{x}$, $\dot{y}$ variables; the system state vector q can be rearranged as $q = (r, s)^T$ where $r := (x, y)^T$ and $s := (\psi)^T$ obtaining the following form for the motion equations

$$\begin{pmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{pmatrix} \begin{pmatrix} \ddot{r} \\ \ddot{s} \end{pmatrix} + \begin{pmatrix} C_1(s, \dot{s}) \\ C_2(s, \dot{s}) \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{A.4}$$

Analyzing the system dynamics it terms of x,y is not so natural,it is therefore convenient to make a coordinate change, in order to decouple the lateral and longitudinal system dynamics.

We will substitute the acceleration $(\ddot{x}, \ddot{y})$ with $a := (a_\top, a_\perp)$, the longitudinal and lateral accelerations. For this purpose, note that

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = R(\psi) \begin{pmatrix} a_\top \\ a_\perp \end{pmatrix}$$

where the rotation matrix $R(\psi)$ is equal to

$$\begin{pmatrix} c_\psi & -s_\psi \\ s_\psi & c_\psi \end{pmatrix}$$

Applying the following transformation the new motion equation can be obtained

$$\begin{pmatrix} R^T M_{11}(s)R & R^T M_{12}(s) \\ M_{21}(s)R & M_{22}(s) \end{pmatrix} \begin{pmatrix} a \\ \ddot{s} \end{pmatrix} + \begin{pmatrix} R^T C_1(s,\dot{s}) \\ C_2(s,\dot{s}) \end{pmatrix} = \begin{pmatrix} R^T \tau_1 \\ \tau_2 \end{pmatrix} \tag{A.5}$$

Doing the computations in our special case, the mass matrix M and the Coriolis' term C change into these new terms

$$\tilde{M}(q) = \begin{pmatrix} m & 0 & 0 \\ 0 & m & mb \\ 0 & mb & mb^2 + I_{zz} \end{pmatrix}$$

and

$$\tilde{C}(q,\dot{q}) = \begin{pmatrix} -mb\dot{\psi}^2 \\ 0 \\ 0 \end{pmatrix} \tag{A.6}$$

To obtain a differential equation system it is then necessary to relate the accelerations $a_\top$ and $a_\perp$ to the lateral and longitudinal velocities $v_\top$ and $v_\perp$.

Differentiating and reordering the velocity equations

$$\begin{aligned} \dot{x} &= v_\top c_\psi - v_\perp s_\psi \\ \dot{y} &= v_\top s_\psi + v_\perp c_\psi \end{aligned}$$

we obtain

$$\begin{aligned} a_\top &= \dot{v}_\top - v_\perp \dot{\psi} \\ a_\perp &= \dot{v}_\perp + v_\top \dot{\psi} \end{aligned}$$

Therefore the mass matrix for the model results $\hat{M}(q) = \tilde{M}(q)$, the Coriolis' force term is

$$\hat{C}(q,\dot{q}) = \begin{pmatrix} -mv_\perp\dot{\psi} - mb\dot{\psi}^2 \\ mv_\top\dot{\psi} \\ mbv_\top\dot{\psi} \end{pmatrix} \tag{A.7}$$

A.2.2 The Whole Model

To obtain a complete model, the forces generated by the wheels interaction with the ground should be considered; they can be found through the use of a model, for example the Paceijka "magic formula", in the hypotheses of knowledge of the normal forces at the wheel contact point.First of all the drift angle of each wheel is founded, that are given by

$$\begin{aligned} \alpha_f &= \arctan(\tfrac{v_\perp + p\dot{\psi}}{v_\top}) - \delta \\ \alpha_r &= \arctan(\tfrac{v_\perp}{v_\top}) \end{aligned}$$

where δ is the steering angle. The longitudinal slip is given by

$$\begin{aligned} \kappa_f &= \frac{-\dot{\theta}_f R_f - v_\top}{v_\top} \\ \kappa_r &= \frac{-\dot{\theta}_r R_r - v_\top \cos\delta - v_\perp \sin\delta}{v_\top \cos\delta + v_\perp \sin\delta} \end{aligned}$$

in which the simple wheel dynamics is

$$\begin{aligned} I_f \ddot{\theta}_f &= R_f F_\top^f(\kappa_f, \alpha_f, F_n^f) + u^f \\ I_r \ddot{\theta}_r &= R_r F_\top^r(\kappa_r, \alpha_r, F_n^r) + u^r \end{aligned}$$

where θ_f and θ_r are the wheels angle position[3], I_f and I_r represent the wheel inertia with respect to their rotational axis, r_f and r_r are the wheels radius, u^f and u^r are the input couple applied to each wheel (generated by the engine and the braking system) and finally $F_\top^f$ and $F_\top^r$ are the longitudinal wheels forces which are a function of the drift angle α, of the longitudinal slip κ and the normal force F_n. Considering also the lateral forces generated at the wheels the whole model is obtained:

$$\hat{M}(q) \begin{pmatrix} \dot{v}_\top \\ \dot{v}_\perp \\ \ddot{\psi} \end{pmatrix} + \hat{C}(q, \dot{q}) + \hat{G}(q) = \begin{pmatrix} F_\top^f c_\delta - F_\perp^f s_\delta + F_\top^r - F_A \\ F_\top^f s_\delta + F_\perp^f c_\delta + F_\perp^r \\ p(F_\top^f s_\delta + F_\perp^f c_\delta) \end{pmatrix} \tag{A.8}$$

and the term F_A is the aerodynamic drag, given by $F_A = \frac{1}{2}\rho C_D A v_T^2$, where ρ is the air density, C_D is the drag coefficient and A is the drag area.

[3]When the vehicle is moving forward the wheels angular velocity $\dot{\theta}_f$ and $\dot{\theta}_r$ are negative.

Appendix B

Optimal Controller Matlab Implementation

The main topic of this chapter is the coding of the optimal control algorithm developed for vehicle guidance.

For development purpose we have used *MATLAB 6.5*™, the MathWorks computing environment, and the *Simulink*™ toolbox. *MSC.ADAMS 12.0*™, developed by MSC.Software, was instead used to simulate the vehicle dynamics.

B.1 MSC.ADAMS 12.0

MSC.ADAMS is a motion simulation environment for analyzing the complex behavior of mechanical assemblies. It allows the user to easily develop and test virtual prototypes and optimize designs for performance, safety, and comfort, without having to build and test numerous physical prototypes, as well as to analyze the characteristics and behaviors of individual parts of the system.

Next to the core package (Adams/View, Adams/Solver, and Adams/PostProcessor), used to build a solid model of the mechanical system from scratch checks the model and then runs simultaneous equations for kinematic, static, quasi-static, and dynamic simulations, there are a range of modules tailored specifically the automotive industry. For this work the package ADAMS/Car and ADAMS/Controls have been extensively used.

The control system, developed using MathWorks Matlab and Simulink toolbox, can interact with the simulator through a proper interface, which is represented by the block "ADAMS 12.0" in the general scheme shown in Figure B.1, which outputs are the vehicle sensors measures, while the inputs are the actuators commands.

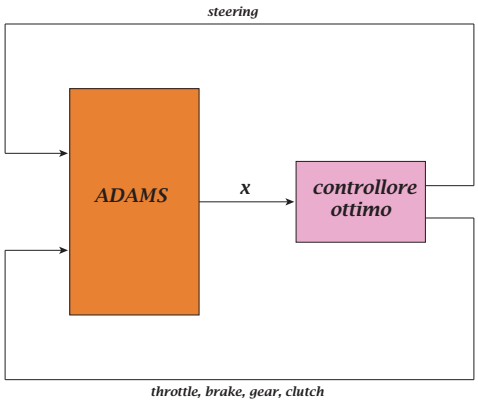


Figure B.1: Reference scheme.

B.2 The Controller

The controller implementation is here detailed.

In the following sections the details of the various parts are explained, in order to show how the requested control action is generated.

B.2.1 Trajectory Generation

The first step is the calculation of state and input references. Every time the control needs to be recalculated, it's necessary to generate the reference trajectory for each state and input variable, to allow the vehicle path following.

First of all a speed and curvature reference is calculated in $[t_0; t_0 + T_{prw}]$, where t_0 is the current time and T_{prw} is the chosen preview time, that is the preview horizon.

The function *reference_profiles* is delegated to calculate the proper acceleration and speed profile starting from the current vehicle position.

Curvature reference is calculated as explained in section 2.1. It's not possible to directly apply what is there described though, as the solution of the equation 2.3 requires the knowledge of the function $\gamma_c(x,t)$ and it's derivatives around $x = D$. That knowledge is beyond this control purpose, as a human driver is not able to know exactly these quantities.

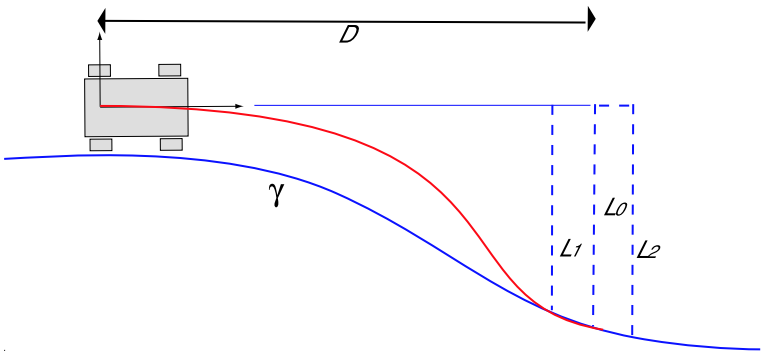


Figure B.2: Centerline distances.

To get round of this drawback, we assume that we can know the distance from the reference path at the points with abscissa D, $D - 0.5m$ and $D + 0.5m$. These distances are the values $\gamma_c(D)$, $\gamma_c(D - 0.5)$, $\gamma_c(D + 0.5)$ and are denoted with L_0, L_1 and L_2 respectively.

The first and second derivatives are then approximated as normalized differences:

$$\dot{\gamma}_c(x) \simeq \frac{\gamma_c(x + \Delta) - \gamma_c(x - \Delta)}{2\Delta}$$

$$\ddot{\gamma}_c(x) \simeq \frac{\dot{\gamma}_c(x + \Delta) - \dot{\gamma}_c(x - \Delta)}{2\Delta} \simeq \frac{\gamma_c(x + 2\Delta) + \gamma_c(x - 2\Delta) - 2\gamma_c(x)}{(2\Delta)^2}.$$

We impose $\Delta = 0.5[m]$ in the first equation and $\Delta = 0.25m$ in the second one, obtaining the following approximations of γ_c and its derivatives in $x = D$:

$$\begin{aligned} \gamma_c(D,t) &\simeq L_0(t) \\ \dot{\gamma}_c(D,t) &\simeq L_2(t) - L_1(t) \quad . \\ \ddot{\gamma}_c(D,t) &\simeq \tfrac{L_2(t)+L_1(t)-2L_0(t)}{0.5^2} \end{aligned} \tag{B.1}$$

In this way we can express the constraint in 2.1.2 as:

$$\begin{bmatrix} \gamma_c(0,t) \\ \frac{\partial \gamma_c}{\partial x}(0,t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{B.2}$$

$$\begin{bmatrix} \gamma_c(D,t) \\ \frac{\partial \gamma_c}{\partial x}(D,t) \end{bmatrix} = \begin{bmatrix} \gamma(D,t) \\ \frac{\partial \gamma}{\partial x}(D,t) \end{bmatrix} = \begin{bmatrix} L_0 \\ L_2 - L_1 \end{bmatrix} \tag{B.3}$$

The connecting contour will be a cubic polynomial of equation:

$$\gamma_c(x,t) = \alpha_3(t)\,x^3 + \alpha_2(t)\,x^2 + \alpha_1(t)\,x + \alpha_0$$

From (B.2) it follows that $\alpha_0 = 0$ e $\alpha_1 = 0$. α_2 and α_3 are then found solving the following linear system:

$$\begin{bmatrix} D^2 & D^3 \\ 2D & 3D^2 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \gamma(D,t) \\ \frac{\partial\gamma}{\partial x}(D,t) \end{bmatrix} = \begin{bmatrix} L_0 \\ L_2 - L_1 \end{bmatrix} \tag{B.4}$$

which gives the searched solution:

$$\begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} = \frac{1}{D^4} \begin{bmatrix} 3D^2 & -D^3 \\ -2D & D^2 \end{bmatrix} \begin{bmatrix} L_0 \\ L_2 - L_1 \end{bmatrix} = \begin{bmatrix} \frac{3L_0}{D^2} - \frac{L_2 - L_1}{D} \\ -\frac{2L_0}{D^3} + \frac{L_2 - L_1}{D^2} \end{bmatrix} \tag{B.5}$$

In this way we can now compute $\sigma(s)$ and switch to the time dependence knowing the vehicle speed. This step is computed by the function *traject.s* (see figure B.4 for more details).

After obtaining $V(t)$ and $\sigma(t)$, we can generate the state and input references, as seen in section 2.3.1; beware that here the used model is the non-holonomic one, simpler than the one actually used for control purpose.

The wheels couple reference is generated separately from the rest; to obtain a reasonable value it is indeed necessary to consider the forces involved in the motion. This quantity is calculated taking into account the system velocity (from which the drag resistant force depends) and the vehicle inertia. The function *traject.mdl* is devoted to this task; it is called from the Matlab function *opt_ control.m* whenever it is necessary (see figure B.3).

B.2.2 Model Linearization

As seen in 2.3, next algorithm step is the model linearization around the trajectory previously generated.

In the vehicle the braking couple is applied to all the wheels while the throttle is applied only to the rear wheels[1] It is then necessary to develop two different models to take in account the vehicle behavior while accelerating or braking. It is also mandatory to model the transition from one model to the other, in order to guarantee a global convenient system representation and control.

The wheels dynamics is modeled as:

$$\begin{aligned} I_f\ddot{\theta}_f &= r_f F_\top^f(\kappa_f, \alpha_f, F_n^f) + u^f \\ I_r\ddot{\theta}_r &= r_r F_\top^r(\kappa_r, \alpha_r, F_n^r) + u^r \end{aligned}$$

[1]We made the hypotheses that the vehicle has rear-wheel drive.

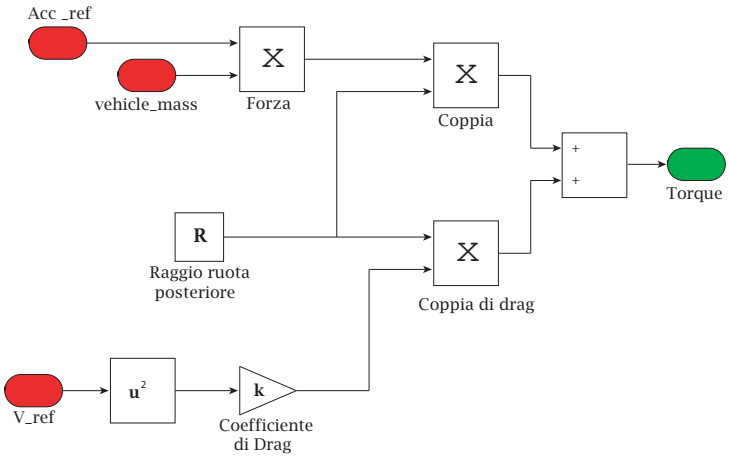


Figure B.3: Reference couple calculation.

In these differential equations u^f and u^r are the input signals applied to the front and the rear axle respectively. These quantities assume different values in the two cases; let τ be the total couple applied to the vehicle, then for the acceleration model we have:

$$\begin{cases} u^r = \tau; \\ u^f = 0; \end{cases}$$

while for the braking case the input signals are:

$$\begin{cases} u^r = (1-a)\tau; \\ u^f = a\tau; \end{cases}$$

where a is the *bias*, that represents how the braking couple is split between front and rear axle (to be more precise it is the part of the whole signal that is applied to the front axle) and it is a braking system characteristic (it means that it is not modifiable once chosen the vehicle model). The model we use for testing purpose presents $a = 0.6$.

The implemented controller should have both models coded inside, choosing the right one to use while running. So at each time t_0 when the control is recalculated the linearization around the reference trajectory for both the models, obtaining two linearized state models for the error as seen in section 2.3.1.

B.2.3 Optimal Control Calculation

After obtaining the linear model for the error, we can calculate the optimal control $\tilde{u}^*$ as seen in 2.3.2. At each step a LQR problem is solved for both the models, obtaining a time-variant feedback matrix for each of them, which represents the matrix error gain to control the system. This operation is done by the function *optimal.c* (see figure B.4 for further details).

B.2.4 Control Application

We have then to determine the current control action, starting from the control calculated above for all the time horizon. This is done by the function *opt_feedback.m*, that calculates the whole control action, as a combination of the feedforward signal and the state feedback. This is generated for both the accelerating and braking models.

As seen before, it is essential to provide a switching mechanism between the two models, to avoid numerical problems and infeasibility. The switch block (see figure B.4) is the one deputed to this operation; at each step it controls the vehicle state and chooses the input signal properly.

In this way the controller output is the right value of steering angle and couple applied to the wheels.

The main part of the controller is a Simulink scheme. This was a forced choice, because this toolbox is the only one that allows Matlab and ADAMS co-simulation.

B.3 Simulator Input Signals

ADAMS dynamic simulator requires as input signals the angle of the steering wheel and the accelerator, brake, clutch and gearshift commands. It is therefore necessary to provide a mechanism that generates these signals starting from wheels couple and steering, which are the controller outputs.

The steering angle is simply calculated with a suitable gain coefficient, which depends on the vehicle characteristics:

$$steering = K_{steer}\delta^*$$

where δ is the steering angle determined by the controller.

This gain depends on steering-suspension subsystem geometry and can be calculated through ADAMS simulator with a standard suspension analysis. Steering angle is linearly changed and wheel response is then analyzed. In figure B.5 the

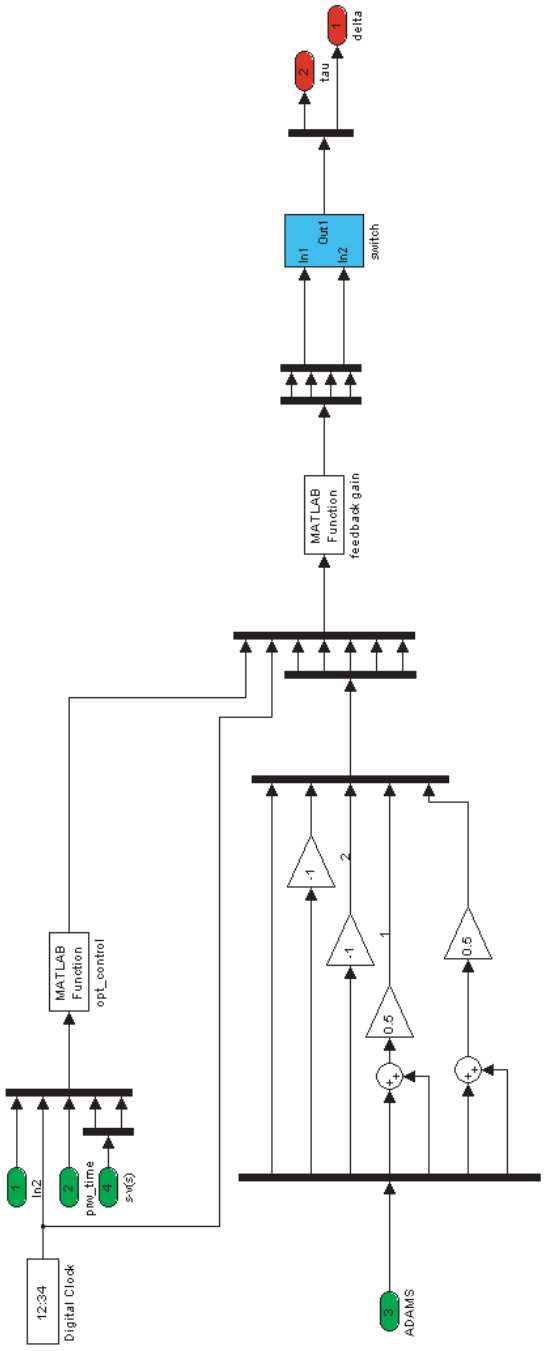


Figure B.4: Optimal Controller.

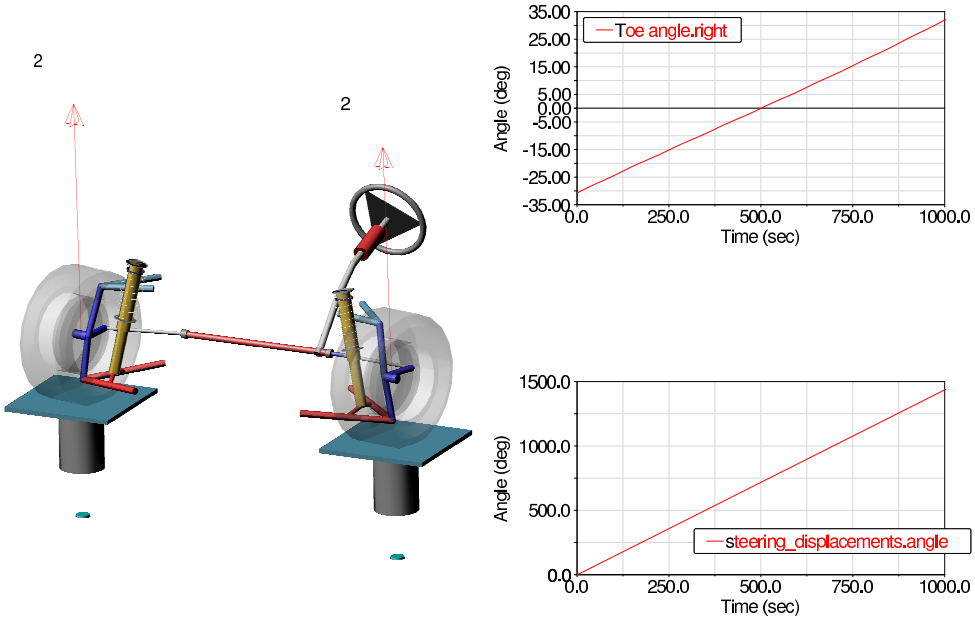


Figure B.5: ADAMS suspension analysis.

tested subsystem, the steering control input (*steering_ displacements.angle*) and front wheels response angle δ_F[2] the (*toe.angle*) are shown.

Generally the relation between these quantities is non-linear. Nonetheless we can well approximate it with a linear one, restricting the wheels angle range to $\left[-\frac{\pi}{4};\frac{\pi}{4}\right]$, which anyway represents common tires range of action.

It is then necessary to add a further block to obtain the right simulator input command; it would be simply a saturation block that restricts the angle in the physical range above founded.

Longitudinal control is more complex, because it requires to generate suitable signals for *throttle, brake, gear and clutch* starting from the couple. This transformation will be the topic of the following sections.

[2]Right and left front wheels have different angles, due to steering mechanism. Here we consider the mean of these two values to have a suitable value for the bicycle model used.

B.3.1 Gearshift Mechanism

Any commercial vehicle is provided with a gearbox, which allows the engine to work in the proper range of speed, varying the transmission ratio between the engine and the tractive axle. This mechanism is directly controlled by the driver, so the controller has to provide a valid gear change function. The vehicle has also a mechanism called clutch, that allows gear changes decoupling engine shaft and axle shaft, that has to be taken in account.

A gear change is composed of the following sequence:

- clutch disengagement;
- accelerator in stand-by, to avoid engine out of revolution speed;
- new gear engagement;
- clutch engagement, to give traction to the vehicle;
- accelerator in action, to restart control action.

These operations must be synchronized, to obtain a smooth change.

Let T_c be the interval time when the controller acts on the clutch; this value represents the time needed by the driver to complete the gear change. In order to guarantee a smooth transition, the clutch signal is modulated with a raised-cosine profile. Let $T_{cl_{rise}}$ and $T_{cl_{fall}}$ be the rising and the falling time of the signal in figure B.6; the clutch command equation is:

$$clutch = \begin{cases} \frac{1}{2}\left[1+\cos\left(\pi+\frac{\pi t}{T_{cl_{rise}}}\right)\right] & \text{if} \quad 0 \le t < T_{cl_{rise}} \\ 1 & \text{if} \quad T_{cl_{rise}} \le t < T_c - T_{cl_{fall}} \\ \frac{1}{2}\left[1+\cos\left(\frac{\pi\left(t-T_c+T_{cl_{fall}}\right)}{T_{cl_{fall}}}\right)\right] & \text{if} \quad T_c - T_{cl_{fall}} \le t < T_c \\ 0 & \text{otherwise} \end{cases} \tag{B.6}$$

While the clutch is disengaged the engine is unloaded, so it can be easily reach a high revolution speed if accelerated. To avoid that, the accelerator signal should be disabled synchronously to the clutch one; to do that this signal is multiplied by

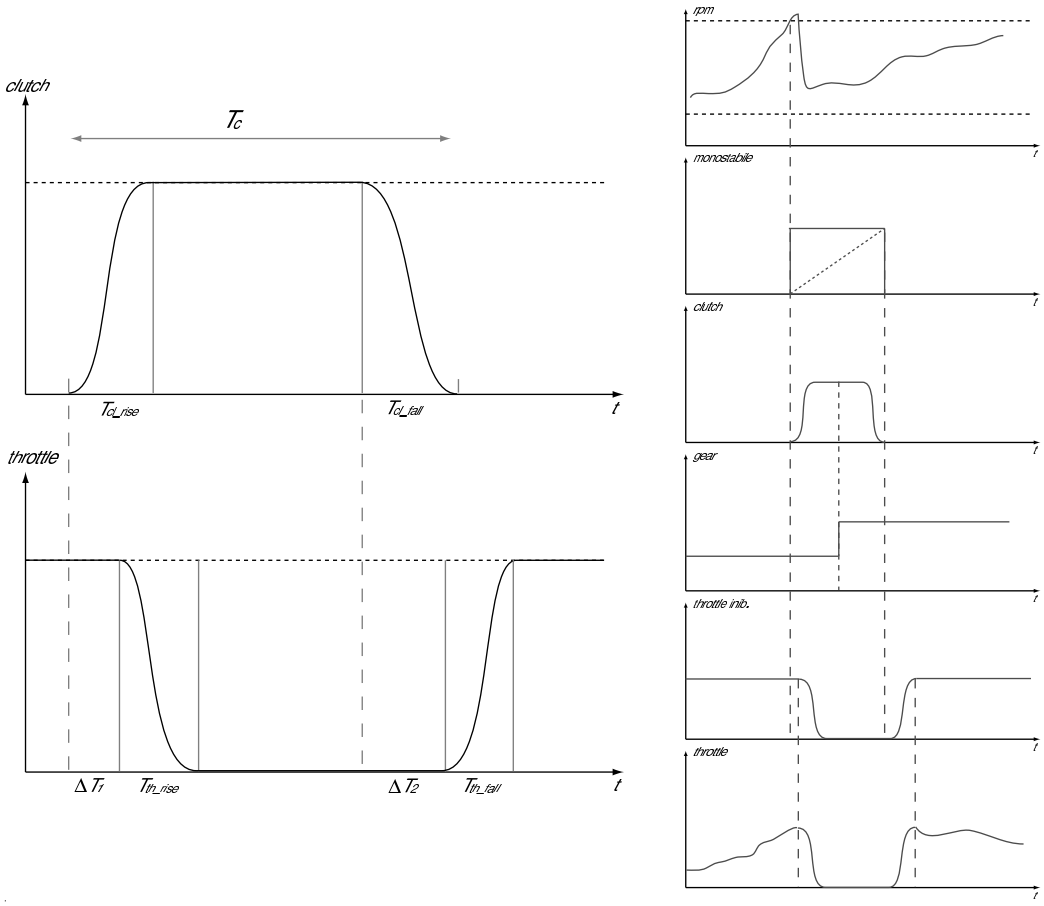


Figure B.6: Gear change:clutch and accelerator inhibition signals.

the following inhibition signal:

$$throttle = \begin{cases} 1 & \text{if} \quad 0 \leq t < \Delta T_1 \\ 1 - \frac{1}{2}\left[1 + \cos\left(\pi + \frac{\pi(t-\Delta T_1)}{T_{th_{rise}}}\right)\right] & \text{if} \quad \Delta T_1 \leq t < T_{th_{rise}} + \Delta T_1 \\ 0 & \text{if} \quad T_{th_{rise}} + \Delta T_1 \leq t \\ & \quad < T_c - T_{cl_{fall}} + \Delta T_2 \\ 1 - \frac{1}{2}\left[1 + \cos\left(\frac{\pi\left(t - T_c + T_{cl_{fall}} - \Delta T_2\right)}{T_{cl_{fall}}}\right)\right] & \text{if} \quad T_c - T_{cl_{fall}} + \Delta T_2 \leq t \\ & \quad < T_c - T_{cl_{fall}} + \Delta T_2 + T_{th_{fall}} \\ 1 & \quad \text{otherwise} \end{cases} \tag{B.7}$$

where ΔT_1 is the delay between the clutch disengagement and the accelerator inhibition and ΔT_2 is the delay between the clutch engagement and the accelerator restoration.

The gear shift upward or downward, is triggered by the exceeding of a thresholds . Let G_U the engine speed that triggers the gear shift upwards, and G_D the one that imposes to shift downward. When the engine exceeds the upper threshold G_U, the gear change starts. Since the engine has a certain inertia and the pilot is still giving gas for a certain period of time, the engine speed will be over the threshold for a while. Moreover any fluctuations could cause additional crossing of the threshold G_U during the gear change maneuver. We need, therefore, to inhibit the starting of a new procedure till the previous one is still on way. For this reason we add a monostable circuit, which generates a signal T_c, which inhibits any other change due to fluctuations in engine rpm. The structure of circuit is quit simple: an integrator is starting at $t = 0$ when the changing gear procedure starts, and continues to integrate up to a threshold. Acting on the threshold, the duration of the inhibition signal can be adjusted; in particular, we can obtain $\Delta t = T_c$, using $f(t) = \int dt$ as integral function and set the threshold to T_c: thus we get a signal duration $\Delta t = T_c$.

In figure B.7 the gear-shift controller is depicted. Its structure is symmetric on the upwards and downwards shift. The inputs are the number of engine revolutions, the time T_c, the lowest and highest gear admissible. In addition, the controller knows the upper and lower thresholds G_U and G_D. When the engine speed exceeds the threshold G_U then the gear shift upward takes place,

The block "Soglia_ sup_ RPM" gives the starting signal if the two following conditions are simultaneously verified:

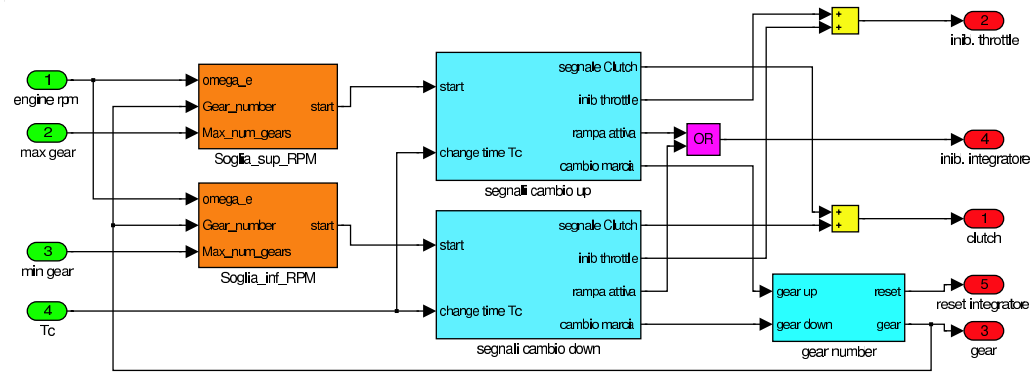


Figure B.7: Gearshift controller.

- the engine speed is higher than threshold G_U
- the current gear is not the highest admissible;

The block "segnali cambio up" takes as input the time T_c and the starting signal, which also starts the monostable circuit, preventing other gear shift commands due to fluctuations of engine revolutions speed

Let t_0 be the moment when the change shift starts. In this moment the monostable circuit is activated and the raised-cosine clutch signal is generated as explained in (B.6). With a delay ΔT_1 the throttle signal inhibition is generated (B.7), stops the accelerating action on the engine.

At $t_0 + T_c/2$ the new gear is mounted. In this moment the clutch is disengaged, and the accelerator is completely released : there is no torque applied to the transmission shaft. Afterwards, according to the profiles set (B.6) and (B.7), friction is engaged again and the accelerator is re-habilitated. In figure B.8 the Simulink scheme is presented.

B.3.2 RPM Controller

During the gear change, when the clutch is disengaged, engine revolutions depends only by its inertia and the throttle command. Stepping on the gas while the engine is unloaded quickly increases its revolution speed; leaving the engine without accelerating command has a consequence the reduction of the speed, due to engine inertia.

To avoid discrepancies between engine revolution speed and transmission shaft speed during the engagement action, the RPM controller has been inserted. It acts on throttle signal while the clutch is disengaged and allows the engine speed to

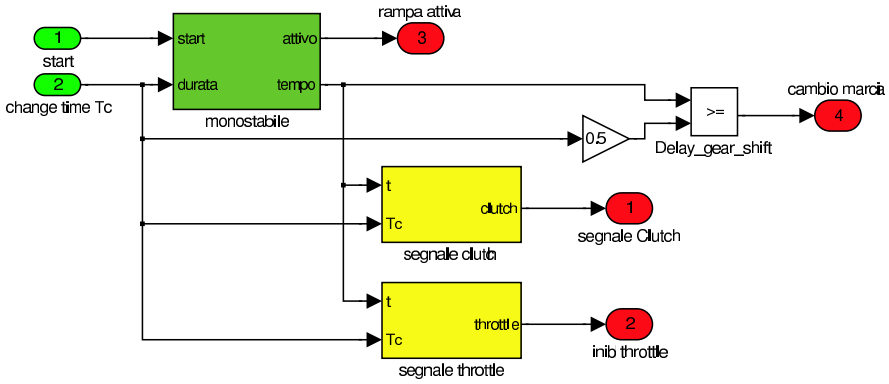


Figure B.8: Clutch and throttle inhibitor signal generator.

be as close as possible to the transmission shaft speed (see figure B.9 for further details).

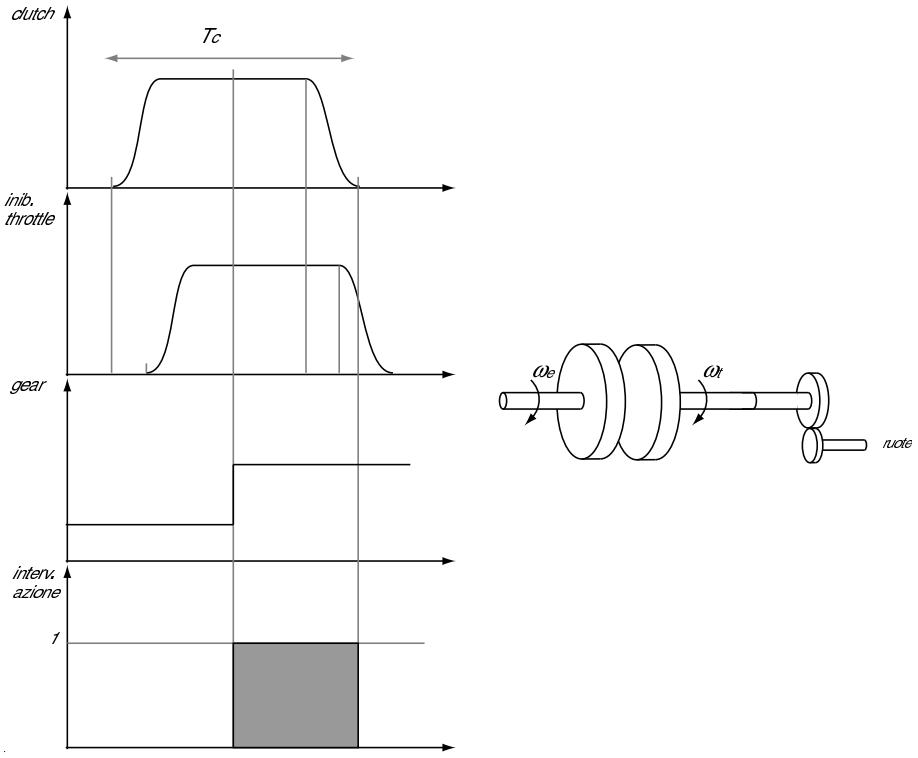


Figure B.9: RPM control signals.

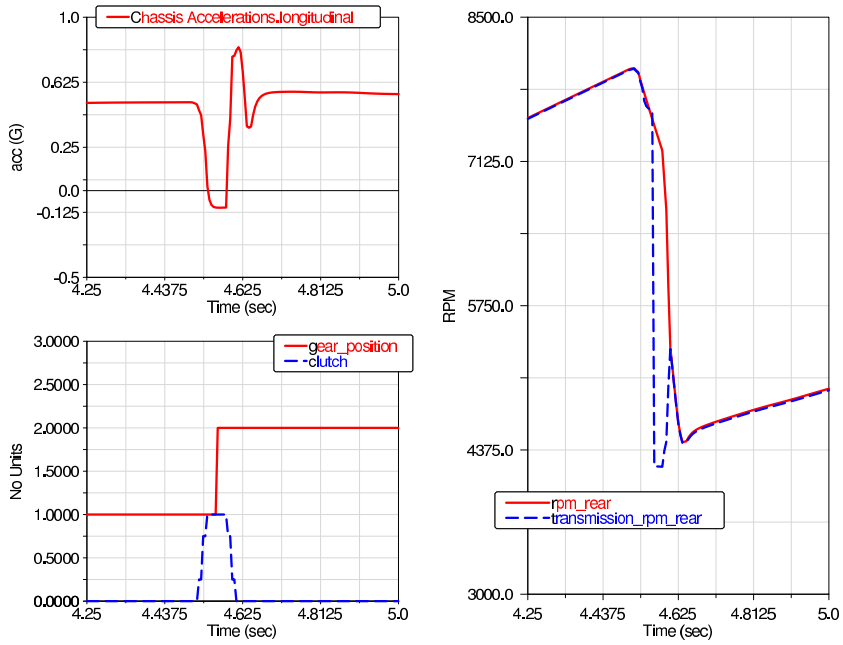


Figure B.10: Gearshift without RPM controller.

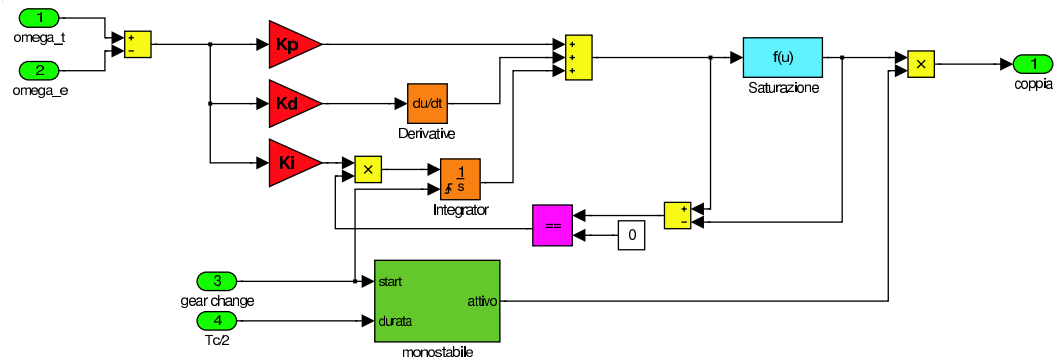


Figure B.11: RPM controller Simulink scheme.

Acknowledgements

My first thanks go to my advisor, Professor Frezza, for giving me the great opportunity of working with him. Thanks for supporting me in any occasion and for giving me the unique opportunity of visiting new exciting places, and collaborating with or just meeting some of the best professors in the world.

Thanks to Professor Kang and to Qi Gong, my office mate at NPS for being so kind and helpful with me.

Thanks to all the professors of the control group at DEI. It had been a great pleasure to be part of this excellent group.

Thanks to Ruggero, Paolo and Luca for starting with me this amazing adventure and sharing courses material and homeworks, along with happiness and distress. Thanks to "the old guys", Ale, Andrea and Ticoz. Thanks to Giuseppe for being so patient with me and being the good example to follow. Thanks to Martina, my first girl-mate in the office, for her friendship and her cakes. Thanks to Lucia, my "preferred roommate" (please, don't anybody tell her that she's the only one I have!...:-) I was looking for someone to rent a house with and I found a very special friend... Thanks to Alberto, for being my most faithful passenger and have been my chauffeur when I was injured. Thanks to Damiano, Mattia and Saverio, my "FUNclub". Thanks to all the guys of the office, Federica, Federico, Francesca, Giulia, Mirco, Simone and Stefano and all the other people that contributed to have a really friendly atmosphere in the lab during these years.

Thanks to all my "ex robocup mates" (il piccolo Al, Koral, Andrea, Luca, Arrigo e Vale), I'm very lucky to have met all you guys!

Back to Monterey, I would like to thank Noemi and Riccardo, Guido, Marcello and Paola for their kindness and for being my family oversea; and then thanks to all the guys I met during my wonderful experience in California. Thanks to Ale, for his support and for being my "Raimondo Vianello" , and, of course, for his friendship, even if almost anybody couldn't believe to that. Thanks to Derek, even if he hasn't checked my grammar as promised! :-p

Thanks to Elisa for her cineforum, the tango lessons and the chats; thanks to Marta, for her great telematic support from Siena.

Thanks to Mara, my “first friend”, for her support and for being my partner in crime. Thanks to Kiara for her patience (I know, I was so annoying while writing this thesis :-) to Sandro for being always so supportive and to all the guys of my speleo group, for all the incredible experiences we shares.

I’d like to give a special thanks to my aunt Diana, for being my best supporter, to my uncle Pietro, who first taught me the love for reading and to be open minded, and to my grandparents (Gigio&Geni, Davide&Elda) for caring for me so much and being the best example ever.

Last but not least, I would like to thank my parents and my brother Enrico for their unconditional support during this wonderful experience...
I couldn’t have done all this without you!!!!

Bibliography

[1] J. T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia, 2001.

[2] J. T. Betts, N. Biehn, and S. L. Campbell. Convergence of nonconvergent irk discretizations of optimal control problems with state inequality constraints. *SIAM J. Sci. Comput.*, 23(6):1981–2007, 2002.

[3] A. M. Bloch. *Nonholonomic Mechanics and Control*. Springer, 2003.

[4] R. W. Brockett. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, pages 181–191, 1983. Birkhauser, Boston.

[5] A. E. Bryson. *Dynamic Optimization*. Addison-Wesley, Reading, MA, 1999.

[6] A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Hemisphere, New York, 1975.

[7] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Method in Fluid Dynamics*. Springer-Verlag, New York, 1988.

[8] Vittore Cossalter. *Cinematica e dinamica della motocicletta*. Edizioni Progetto Padova, 1997.

[9] Jane Cullum. Finite-dimensional approximations of state-constrained continuous optimal control problems. *SIAM Journal on Control*, 10(4):649–670, 1972.

[10] Casanova D., Sharp R.S., and Symonds P. On minimum time optimisation of formula one cars: The influence of vehicle mass. *in Proceedings of AVEC*, August 2000. Ann-Arbor, MI.

[11] Casanova D., Sharp R.S., and Symonds P. Minimum time manoeuvring: The significance of yaw inertia. *Vehicle System Dynamics*, 34:77–115(39), August 2000.

[12] http://www.darpa.mil/grandchallenge.

[13] A. L. Dontchev and W. W. Hager. The euler approximation in state constrained optimal control. *Mathematics of Computation*, 70:173–203, 2000.

[14] H.B. Pacejka E. Bakker and L. Lidner. A new tyre model with application in vehicle dynamics studies. *4th Autotechnologies Conference, Monte Carlo*, 1989. SAE 890087.

[15] G. Elnagar and M. A. Kazemi. Pseudospectral chebyshev optimal control of constrained nonlinear dynamical systems. *Computational Optimization and Applications*, 11, 1998.

[16] F. Fahroo and I. M. Ross. Costate estimation by a legendre pseudospectral method. *Journal of Guidance, Control, and Dynamics*, August 1998.

[17] C. Frangos and Y. Yavin. Tracking control of a rolling disc. *IEEE Transaction on Systems, Man and Cybernetics*, 30(2), April 2000. part B: Cybernetics.

[18] R. Frezza. Path following for air vehicles in coordinated flight. *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*, pages 884–889, 1999.

[19] R. Frezza, G. Picci, and S. Soatto. Non-holonomic model-based predictive output tracking of an unknown three-dimensional trajectory. *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, 4:3731–3735 vol.4, 16-18 Dec 1998.

[20] C. Frangos G. W. Ehlers, Y. Yavin. On the motion of a disc rolling on a horizontal plane: Path controllability and feedback control. *Computer Methods in Applied Mechanics and Engineering*, 137:345–356, November 1996.

[21] N.H. Getz and J.K. Hedrick. An internal equilibrium manifold method of tracking for nonlinear nonminimum phase systems. *American Control Conference, 1995. Proceedings of the*, 3:2241–2245 vol.3, 21-23 Jun 1995.

[22] N.H. Getz and J.E. Marsden. Control for an autonomous bicycle. *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 2:1397–1402, 21-27 May 1995.

[23] C. J. Goh and K. L. Teo. Control parametrization: A unified approach to optimal control problems with general constraints. *Automatica*, 24:3–18, 1988.

[24] Q. Gong, W. Kang, and I. M. Ross. A pseudo-spectral method for the optimal control of constrained feedback linearizable systems. *IEEE Transactions on Automatic Control*, 51(7), July 2006.

[25] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo. Convergence of pseudospectral methods for constrained nonlinear optimal control problems. *in Proc. IASTED International Conference on Intelligent Systems and Control*, pages 209–214, 2004.

[26] M. Guiggiani. *Dinamica del Veicolo*. Citt‡ Studi Edizioni, Torino, 2002.

[27] W. W. Hager. Runge-kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87:247–282, 2000.

[28] R. F. Hartl, S. P. Sethi, and R. G. Vickson. A survey of the maximum principle for optimal control problems with state constraints. *SIAM*, 1995.

[29] MSC Software. *Using ADAMS/Tire*, msc adams 12.0 edition.

[30] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical introduction to Robotic Manipulation*. CRC Press, 1994.

[31] R. Olfati-Saber. *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*. PhD thesis, Massachusetts Institute of Technology, February 2001.

[32] H. B. Pacejka. *Tyre and Vehicle Dynamics*. Elsevier Butterworth-Heinemann, Oxford, 2002.

[33] E. Polak. A historical survey of computations methods in optimal control. *SIAM Review*, 15:553–548, 1973.

[34] P. Kokotovic R. Freeman. *Robust Nonlinear Control Design*. Birkhauser, Boston, 2004.

[35] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, and I. Kolmanovsky. Dynamics and control of a class of underactuated mechanical systems. *IEEE Transactions on Automatic Control*, 44(9):1663–1671, 1999.

[36] M. Reyhanoglu, A. van der Schaft, N.H. Mcclamroch, and I. Kolmanovsky. Dynamics and control of a class of underactuated mechanical systems. *Automatic Control, IEEE Transactions on*, 44(9):1663–1671, Sep 1999.

[37] http://www.robocup.org.

[38] I. M. Ross and F. Fahroo. *Legendre Pseudospectral Approximations of Optimal Control Problems*, volume 295. Springer-Verlag, New York, 2003. Lecture Notes in Control and Information Sciences.

[39] M. Ross. *User's Manual For DIDO (Ver. PR.1): A MATLAB Application Package for Solving Optimal Control Problems*. Naval Postgraduate School, Monterey, CA, tech. rep. 0401.0 edition, February 2004. Available: http://tomlab.biz/docs/DIDOManualPR1.pdf.

[40] Sharp R. S., Casanova D., and Symonds P. A mathematical model for driver steering control, with design, tuning and performance results. *Vehicle System Dynamics*, 33:289–326, 2000.

[41] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, Philadelphia, 2000.