

POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering

Master of Science Thesis

**SITO Control Approach to
Autonomous Parking**



Supervisor:
Prof. Diego Regruto

Candidate:
Giuseppe Petralia

Academic Year 2018/2019

To my family,
the best part of me.

To my grandmother Maria,
the angel that protects me from above.

And to me,
that I never gave up.

“There is a driving force more powerful than steam, electricity and nuclear power: the will.”

(A. Einstein)

“If you can dream it, you can do it”

(E. Ferrari)

INDEX

INTRODUCTION	6
AUTONOMOUS DRIVING	8
1. HISTORY OF AUTONOMOUS DRIVING.....	9
1.1 AUTONOMOUS DRIVING AND VEHICLES	11
1.1.1 AUTONOMOUS VEHICLES CAPABILITIES.....	14
1.1.1.1 PERCEPTION	14
1.1.2 LOCALIZATION	19
1.1.3 PLANNING	20
1.1.3.1 AUTONOMOUS VEHICLE PLANNING SYSTEMS	20
1.1.3.2 MISSION PLANNING	21
1.1.3.3 BEHAVIOURAL PLANNING.....	21
1.1.3.4 MOTION PLANNING.....	22
1.1.3.5 COMBINATORIAL PLANNING	22
1.1.3.6 SAMPLING-BASED PLANNING.....	23
1.1.3.7 DECISION MAKING FOR OBSTACLE AVOIDANCE.....	24
1.1.3.8 PLANNING IN SPACE-TIME	25
1.1.3.9 PLANNING SUBJECT TO DIFFERENTIAL CONSTRAINTS.....	25
1.1.3.10 INCREMENTAL PLANNING AND REPLANNING.....	27
AUTONOMOUS PARKING	29
2. AUTONOMOUS PARKING EVOLUTION	30
2.1 DIRECT TRAJECTORY PLANNING: HUMAN-LIKE PARKING.....	32
2.2 INTELLEAGENT AUTONOMOUS PARKING CONTROL SYSTEM	38
2.2 FAST PARALLEL PARKING USING GOMPERTZ CURVES	41
LATERAL DYNAMICS PROBLEM	49
3. LATERAL DYNAMICS ANALYSYS.....	50
3.1 YAW STABILITY CONTROL SYSTEMS	50
3.2 KINEMATIC MODEL OF LATERAL VEHICLE MOTION	51
LINEARIZATION OF NONLINEAR SYSTEMS	60
4. FEEDBACK LINEARIZATION	61
4.1 CANONICAL FORM OF FEEDBACK LINEARIZATION	61
4.2 INPUT-STATE LINEARIZATION.....	62
4.4 THE INTERNAL DYNAMICS OF LINEAR SYSTEM	64
4.5 THE ZERO-DYNAMICS	65

4.6 INPUT-STATE LINEARIZATION OF SISO SYSTEMS.....	66
SYSTEM CONTROL DESIGN	69
5. TIME-STATE CONTROL FORM.....	70
5.1 MATHEMATICAL ANALYSIS AND SIMULINK TSC DESIGN	72
5.2 LINEAR PLANT MODEL AND CONTROL DESIGN	76
5.2.1 SINGLE INPUT TWO OUTPUT (SITO) SYSTEMS.....	80
CONCLUSIONS	87
LIST OF FIGURES	88
BIBLIOGRAPHY	91

INTRODUCTION

Since the production of the first vehicles, the companies have always pushed themselves to overcoming every limit and to the continuous research of perfection in terms of aesthetics, safety, performance, drivability. In recent decades, however, with the advent of the new generations with different mentalities and needs than those of the past, new objectives and other frontiers have been added to overcome: to ensure that it is as autonomous as possible and that it can gradually free the driver from the load driving, eliminating some of those daily sources of stress, tension, fatigue while achieving a smoother and free-error driving experience.

Within this context, the goal of the thesis is to preliminary analyse the autonomous driving and vehicles and the related problems and then to face with the realization of one the main studied ADAS (Advanced Driver Assistance Systems).

The first chapter highlights the evolution of the autonomous vehicles over the years. In particular, it is explained the reasons that made the car-maker and vendors think about the development of something with strong potential that could change drastically the next future of the transport system and framework. The focus analysis of this section is to understand how the autonomy concept within this context can put considerable contribution by upsetting the human life. For this aim, evaluation of all pros and cons to 360 degrees was made taking into account the autonomy levels reached and to be achieved in the coming years. The last, but the main part of this first chapter is dedicated to a detailed analysis of the different working principle of an autonomous vehicle in terms of electronic perception of each elements of the surrounding environment and the different classes of planning.

Among the various autonomous driving systems that are taking hold in recent years, one of the most important is that related to parking, the so-called park assist that allows the driver to make a perfect parking trajectory in terms of safety and efficiency. Particularly, in the second chapter it is presented an historical excursus on how this technology has evolved starting from a purely sensor notice approach to a recent studied autonomous parking with a multiple level control process.

For this reason, several literature approaches to this problem are listed and explained within this part. Obviously, for each researcher contributions the mathematical aspects are pointed out with a particular attention to the vehicle dynamics.

The main point of this thesis work was the autonomous parking realization in the MATLAB/Simulink environment by means a 2-degrees of freedom multivariable feedback

control system based on control of two reference variable: lateral position in terms of trajectory to follow, a typical parallel parking S-trajectory, and yaw angle of the vehicle both generated by an appropriate external control, the Time State control discussed in the fifth chapter. The aim of this double particular control is to realize a perfect parallel parking that is one the three noted types of parking situations.

To obtain the desired results, a linear bicycle model is considered after a linearization of a starting nonlinear model whose equations are derived in the third chapter. There exist a lot of linearization methods, but the one discussed in the fourth chapter is the feedback linearization which allows to have a more treatable model.

When one thinks about a parking situation, it important to consider the problem of the lateral dynamics, subject of deep study within the thesis. Fundamentally, the control of the yaw stability of a vehicle is treated and one derives that the bicycle model can well-approximates the vehicle behaviour and dynamics under several assumptions.

The final linear bicycle model represents the plant of the system as a SITO system, a single output (steering angle) and the two-output system to be controlled as it is possible to see in the fifth and last chapter.

The overall final control is realized thanks to a combination of a TISO Control derived from the Freudenberg and Middleton studies and several feedforward filters mathematically calculated and design based on the matrix assumptions which ties the input and the output of the entire system.

The base idea applied to obtain the optimal following of the refence inputs is to impose in a cross way the sensitivity and complementary sensitivity behaviours to the transfer functions involved within the system taking into account the stability of the same.

Finally, a series of consideration and remarks are made on the simulations which show the obtained good and optimal results of the control scheme by means the Simulink scope graph. The simulations are based on a comparison of the reference inputs of yaw angle and lateral trajectory and the same derived output by also analysing the final steering angle.

CHAPTER 1

AUTONOMOUS DRIVING

AUTONOMOUS DRIVING

1. HISTORY OF AUTONOMOUS DRIVING

The story of the driverless car begins in the USA in the 1920. Driver error was seen as being a prime cause of accidents. First developments were done on the field of aviation and radio engineering giving a perspective to obtain accident-free and self-driving automobiles.

In Paris, Lawrence B. Sperry introduced a gyroscope airplane stabilizer; a pilot assistant climbed out onto the right wing during the flight, while the pilot stood up and raised his hands above his head. The system automatically equilibrated the aircraft, even if it did not fully relieve the pilot of steering wheel. Furthermore, engineers thought that the radio technology was one of the technical requirements needed to be able to create a self-driving car. The new science of radio guidance was engaged with the remote control of moving mechanisms by means of radio waves, a technology developed by the US military which was experimenting with remote-controlled ships and aircraft.

From 1930 to 1950 various appeared in the public, where manipulate the brakes, steering wheel and horn of vehicles driving in front of another one was possible, by using a spherical antenna that received the code.



Figure 1.1: Remote-controlled vehicle

In the 1950s, the idea of remote-controlled automobiles was abandoned

introducing a guide wire vision concept and in 1958 General Motor's Center in Michigan completed a test route of one mile.

Engineers used a Chevrolet fitting in the front area of the car two electronic sensors that



Figure 1.2: General Motors Prototype

followed a laid in the road adjusting a steering wheel. In the same year GM made a prototype car which had no steering wheel, by a central console with a uni-control joystick that unified accelerating, braking and steering functions. In the 1970s, the guide wire concept failed and thanks to the rise of microelectronics USA and Japan made progress in attempting to provide

cars with sight. The mechanical laboratory of Tsubuka presented the first visually guided

autonomous vehicle that could record and on-board process pictures of lateral guide rails on the via two cameras, with the car able to move with a speed of 10 Km/h. The rise of microelectronics led to an increasing use of electronics in vehicle technology and the launch of the first on-board computers in the series 7 BMW. The era of active driver-assistance systems that directly intervene in the driving process began with the introduction of ABS in 1978.



Figure 1.3: Tsubuka autonomous vehicle

In the 1980s, the research on autonomous vehicles became a serious research topic for academic and industrial research in many countries with the new concept of vision—based autonomous driving. At first, the industry had expressed its preference for lateral guidance of cars using electromagnetic fields generated by cables in the road, but then the Ernst Dickmanns team from University of Munich successfully convinced the industry to privilege the concept of machine vision that would allow the detection of obstacles and avoid additional costs in infrastructure.



Figure 1.4: Dickmanns team vehicle

This team, in 1994, developed a vehicle able to drive more than 1000 Km autonomously on three lane highways around Paris with a speed of up to 130 Km/h. The system was based on real time evaluation of image sequences caught by four cameras. Steering, throttle and brakes were controlled automatically through computer commands, demonstrating the

capability of deriving autonomously decisions for lane changing and passing.

In 1995, members of NevLab in the USA a partially autonomous vehicle presented that drove from Pittsburgh to San Diego; they also used a vision-based approach where steering was based on camera images of the road, but human had to control brakes and acceleration, and the automated longitudinal and lateral control of the car was based only on video image processing from the front hemisphere.

In last years, several business entities pushed the limits of this reality on urban roads achieving tremendous progress. In particular, Google is the most experienced in this field with its over 2 million miles of autonomous vehicles test. Uber that is a transportation

Autonomous driving can definitely be scary to some, but it's hard to deny the benefits in terms of additional safety, performance improvement, greater accessibility and increased productivity. Furthermore, they surely have a positive impact on the environment thanks to the capability to alleviate the road congestion and then improve the road efficiency. It would also greatly improve the mobility of elderly and disabled people.

The fundamental aspect introduced by the autonomous vehicles is the possibility to free people from the driving task making comfortable, safe and effective the control of driving situations and the related load by eliminating human error in different situations.



Figure 1.6: Free driver from driving tasks

Human factors and interactions have been recognized as the most important problems in automotive controls even if they will be included in future controls because the first purpose will remain assist drivers without upsetting human drivers on the road. To achieve this objective, in summary, it is important for autonomous vehicles have human-acceptable driving performance, but the main problems are related to the assurance of reliability linked to a series of obstacles and issues which will not be overcome in the near future.

Another important aspect to be considered is that many operating environments are not static, but continuously changing and thus not known a priori. In an urban environment, the vehicle must constantly adapt itself to new perceived changes in the environment and be able to react considering several uncertainties related to localization accuracy, sensor precision and control policy execution. In application, perhaps the largest uncertainty source is the surrounding obstacles' movements.

When one speaks about autonomous vehicles it is important to consider that there are 6 different levels (if Level 0 is considered) of driving automation because helps to understand where one stands with this rapidly advancing technology:

- *Level 0* – No Automation

At this level of autonomy, the driver manages all operating tasks like steering, braking, accelerating or slowing down, and so forth.

- *Level 1* – Driver Assistance

At this level, the vehicle can assist the driver with some functions, but the driver still handles all accelerating, braking, and monitoring of the surrounding environment.

- **Level 2 – Partial Automation**
At this level most companies are currently developing vehicles, where they can assist with steering or acceleration functions and allow the driver to disengage from some of their tasks. However, the driver needs to be always ready to take control of the vehicle and it still responsible for most safety-critical functions and all monitoring of the environment.
- **Level 3 – Conditional Automation**
The biggest progress from Level 2 to Levels 3, the vehicle itself controls all monitoring of the environment (using sensors like LIDAR). The driver’s attention is still important at this level but can disengage from “safety critical” functions like braking and leave it to the technology when conditions are safe.
- **Level 4 – High Automation**
At Level 4, the autonomous driving system would first notify the driver when conditions are safe, and only then does the driver switch the vehicle into this mode. The only problem is that it is not able to distinguish between more dynamic driving situations like traffic jams or a merge onto the highway.
- **Level 5 – Complete Automation**
The Level 5 of autonomy requires absolutely no human attention in terms of no need for pedals, brakes, or a steering wheel, as the autonomous vehicle system controls all critical tasks, monitoring of the environment and identification of unique driving conditions like traffic jams. [1], [2], [3]

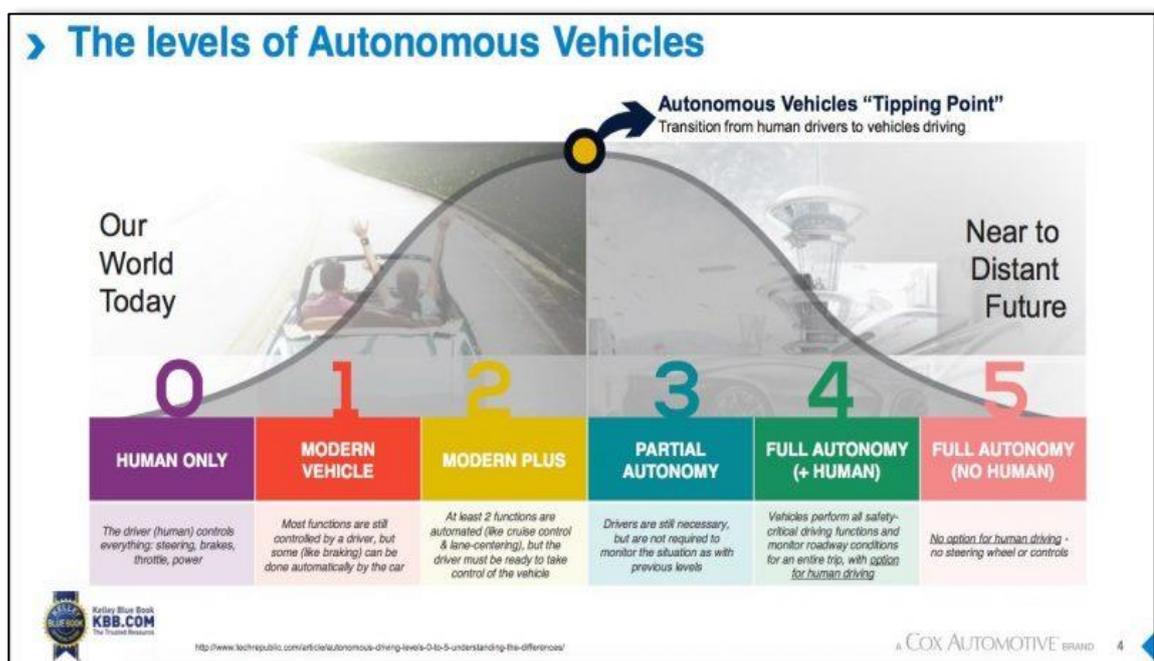


Figure 1.7: Autonomous vehicles levels

1.1.1 AUTONOMOUS VEHICLES CAPABILITIES

The main capabilities of an autonomous vehicles software system can be categorized into three categories:

- perception
- planning
- control

Also, the communications between two vehicles, Vehicle-to-Vehicle (V2V) can be exploited to have improvement in the perception or planning areas also through the vehicle cooperation.

1.1.1.1 PERCEPTION

Perception is the interactive capability for an autonomous system to gather information from the external environment and discern crucial knowledge including velocities, obstacles' locations, road signs and marking detection, free available driving areas and every type of contextual understanding of the environment categorizing data starting from their semantic meaning.

In practise the environment perception task can be actuated with different approaches which space from using LIDARs, cameras or a fusion between these two devices to using ultrasonic sensors and short/long-range radars.



Figure 1.8: Perception of the environment

Two important aspect has to be considered in the perception phase for the vision system:

- road detection
- on-road object detection

1.1.1.2 ROAD DETECTION

The road detection includes two categories:

- road surface detection
- lane line marking detection

The road surface detection is important to update the autonomous vehicle about free space locations and where it can drive with safety avoiding collision and accidents.

There are three type of road surface detection approaches divided in three categories:

- feature/cue-based detection
- feature/cue-based learning
- deep learning

In the feature/cue-based detection approaches, patches or feature points are identified in the original image based on some standard features. Then segmentation algorithm is applied to identify road surfaces.

In the feature/cue-based learning approaches, pixels or image patches are analysed in order to extract a set of featured to classify with a road or non-road label.

Deep learning approach has better performance than the other two and its framework has gained popularity in recent years, especially with the development of suitable processors and implementations.

In spite of the deep learning approach is provided of excellent performance, it has not negligible drawbacks: memory requirement for huge computation, long process time and non-traceable process.

The lane line marking detection is needed to detect the road lane line markings and estimate the vehicle pose with respect to the identified lines.

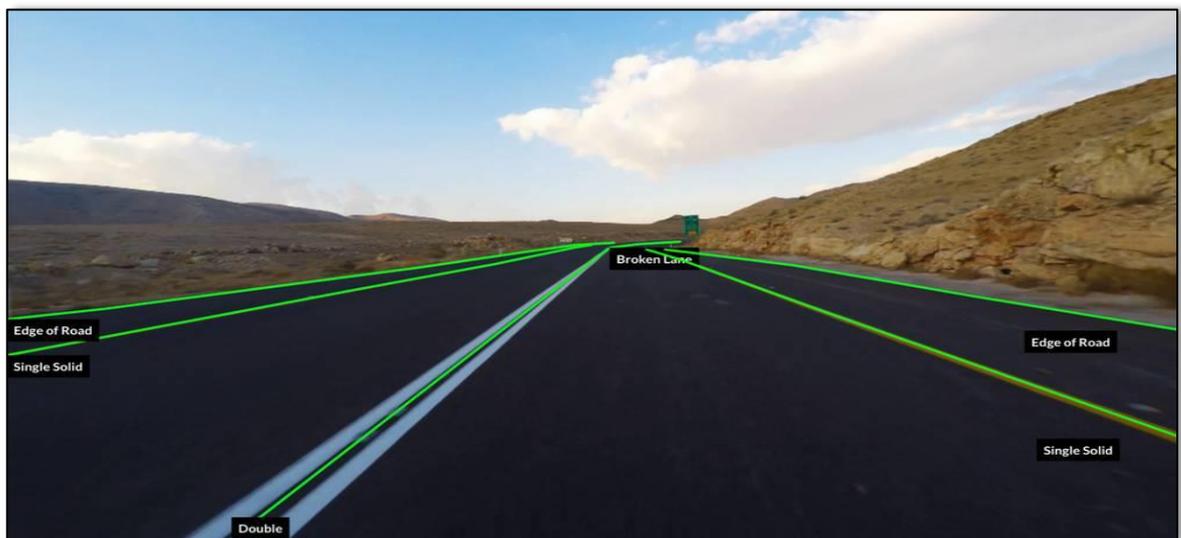


Figure 1.9: Lane line marking detection

The information obtained from this type of detection is useful for the vehicle control systems even if it has remained as a challenging problem due to the fact that it has to cope with a range of uncertainties related to road singularities and traffic road reality, which may include variation of lighting conditions, consumed lane markings, and important markings such as warning text, zebra crossings and directional arrows.

The lane line detection algorithms are generally 3-steps algorithms:

1. lane line feature extraction, to identify pixels of each lane line marking through colour and edge detection and eliminate the non-lane marking pixels basing on the fact that the lane markings have high contrast with respect to road pavement;
2. fitting the pixels into different high-level representations of the lane to obtain a model (for example straight lines, zigzag line, parabolas, and hyperbolas);
3. estimating the vehicle pose based on the extracted model on the previous step.

It may exist a fourth step before the estimation of the vehicle pose to impose temporal continuity, improve estimation accuracy and prevent detection failures.

Most approaches in the literature are based on the observations that lane markings have large contrast compared to road pavement.

Finally, in the lane-level localization the vehicle lateral position and moving orientation are estimated based on the lane line model.

1.1.1.3 ON-ROAD OBJECT DETECTION

This type of detection mainly covers vehicle and pedestrian object classes and in particular it is based on deep learning methods.

Despite the importance of this detection, it's not enough for the autonomous vehicle application because the methods are not robust due to different appearances, shapes, sizes and types of objects.

1.1.1.4 LIDAR

LIDAR (that stands for Laser Imaging Detection and Ranging or Light Detection and Ranging) is a remote sensing technique that allows to determine the distance of an object or a surface using millions of light pulses per second sent in a well-designed pattern. For most of the autonomous vehicles which are research object LIDARs are the basis for object detection, even if the cost of 3D LIDARs can be prohibitive to many applications.

In particular, with its rotating axis, LIDAR creates a dynamic 3D map of the environment.

The source of a LIDAR system is a laser, which is a coherent beam of light with precise wavelength, sent to the system or the object to be observed and reflected back as sparse 3D points representing an object's surface location. The problem is that the reconstruction is never perfect because generally there are missing points returned by the LIDAR and patterns result unorganized.

There are three representations of the points generally used:

- point clouds
- features
- grids

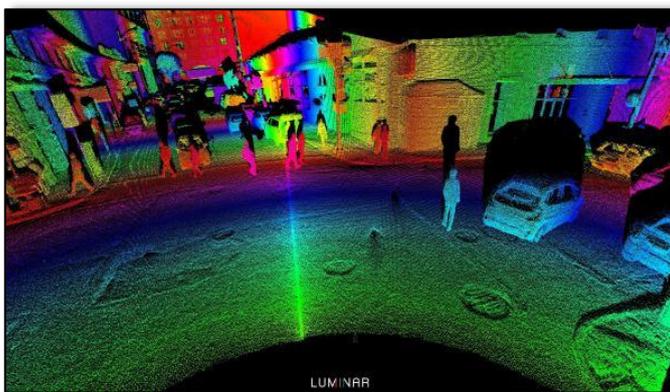


Figure 1.10: LIDAR detection

Point cloud-based approaches provide a great environment representation using raw sensor data.

This approach is useful for further processing but involves an increase of processing time and a reduction of the memory efficiency.

Feature based approaches represent the environment thanks to the parametric features (lines and surfaces) extracted out of the point cloud. Even if this approach is too abstract, it's the most memory-efficient and accurate.

Grid based approaches discretize the space creating small grids full of information from the point cloud in order to establish a neighborhood point.



Figure 1.11: LIDAR

Generally, two procedures are needed to receive 3D point cloud information:

1. segmentation
2. classification

And non-mandatory third one, which is the time integration, that improve the consistency and the accuracy of the two previous steps.

Segmentation refers to the important clustering process that gathers the points into multiple homogeneous groups.

Classification is the process that recognizes the class and type of segmented clusters (pedestrian, road surface, bike, car, etc.).

The algorithm for the segmentation can be part of five categories:

- edge based method
- region based method
- model based method
- attribute based method
- graph based method

Edge based methods are generally noise susceptible and they are adopted in tasks in which the considered object has artificial edge features (for example curb survey). For this reason this approach is not suitable for nature scene detection.

Region based methods through certain criteria (surface normal, Euclidean distance, etc.) use particular region growing mechanism to cluster neighborhood points.

Model based methods are normally designed to segment the ground plane. They exploit standard models in mathematic form like plane, cone, sphere, cylinder in order to fit the points.

Attribute based methods are a 2-step approach in which for first the attribute is computed for each point and later these points are clustered depending on the associated attributes.

Graph based methods insert the point cloud into a graph structure in which each point is a vertex/node and the connections between near points are graph edges. This method is very effective in image semantic segmentation.

After the segmentation process each cluster, that contains information from spatial relationship to LIDAR intensity of the points, need to be assigned to an object category.

Generally, in order to make efficient the entire perception process a sensor fusion technique is applied in order to exploit at maximum the advantages of each sensor.

A fusion between a LIDAR and a camera can be convenient because advantages of one are

able to provide for disadvantages of the other one. In particular, in the autonomous vehicle environment perception, LIDAR is generally able to produce 3D information even if low objects' appearances data despite its performance is not affected by the illumination of the environment. On the other hand, camera provides much more detailed objects' appearance information, but it is not able to extract 3D information and not tolerate different illumination conditions. LIDAR and camera fusion are necessary to obtain the best perception result and it can be divided into two main categories based on their fusion process locations considering a fusion at feature level and a fusion at decision level.

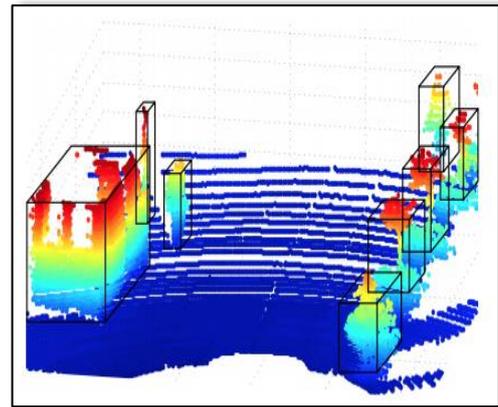


Figure 1.12: LIDAR Image

1.1.2 LOCALIZATION

Localization is the fundamental ability to enable an autonomous system determining the pose (position and orientation) with respect to the environment.

Due to the fact that determining the pose is generally difficult the localization problem is considered as a pose estimation problem divided in two sub-problems:

- pose fixing problem, in which an algebraic equation describes the measurement related to the vehicle pose, so to predict the measurement given a pose;
- dead reckoning problem refers to the computation of a current position using a previously position considering a set of differential equations to be integrated.

Localization of a vehicle is generally made up of a fusion between a satellite-based systems and inertial navigation systems. GPS (global position systems) and GLONASS (Global Navigation Satellite System) are the most used satellite-based systems. can provide a regular fix on the global position of the vehicle. Their accuracy can vary from a few of tens of meters to a few millimetres depending on the signal strength, and the quality of the equipment used. Inertial navigation systems, which use accelerometer, gyroscope, and signal processing techniques to estimate the attitude of the vehicle, do not require external infrastructure. However, without the addition of other sensors, the initiation of inertial navigation system can be difficult, and the error grows in unbounded fashion over time.

GPS in localization requires reliable service signals from external satellites and high-precision sensors.

In recent years, map aided localization algorithms, like Simultaneous Localization and Mapping (SLAM), have seen a remarkable advancement through local features it was possible to achieve highly precise localization.

The SLAM goal is to create a map and use it simultaneously as it is built. SLAM algorithms use statistical modelling to exploit old features observed by system sensors to estimate its position in the map and identify new features even if the absolute position is almost indefinable.

Bayesian filtering and smoothing are the main approaches used for solving the SLAM problem formulated as an optimization problem to minimize the error.

1.1.3 PLANNING

Planning refers to the process of making focused decisions in order to achieve the system's higher order goals, generally to move the vehicle from a starting location to a target location avoiding obstacles and in the most optimal way as possible.

1.1.3.1 AUTONOMOUS VEHICLE PLANNING SYSTEMS

The early stages of self-driving vehicles (SDV) were practically semi-autonomous in nature and limited to functions and performance bases such as lane following or adaptive cruise control. Wider capabilities were remarkably visible in the DARPA Urban Challenge (DUC) or DARPA Grand Challenge (DGC) organized in 2007. The DARPA Grand Challenge is a competition for driverless vehicles, funded by DARPA, the most important US Defence Department agency for the development of military technologies.



Figure 1.13: DARPA CHALLENGE

In the 2007 edition of the competition it was demonstrated the feasibility of self-driving and it is clearer that a SDV can manage a large range of urban driving environment with a complete planning framework, even if the performance was still different from the quality of human drivers. Many competitors have exploited a similar three level hierarchical planning framework composed by a mission planner (or route planner), a behavioural planner (or decision maker) and a motion planner (or local planning), while others trust in a different strategy using a two-level planner with a motion planner and a navigator that provide for the functions of both the mission planner and behavioural planner.

Each planner performs different objectives: the mission planner takes care of the high-level task to achieve, such as which roads the vehicle should be taken; the behavioural planner has to follow rules and restriction, makes in the proper way generating local task, such as change lanes, overtake it, etc. Also, the motion planner has to achieve local objectives, typically reach a target region without obstacle collision by generating suitable actions and proper paths.

However, recent works continue to have this three-hierarchical planning framework.

1.1.3.2 MISSION PLANNING

A graph network reflecting road and path network connectivity perform the mission planning. In the DUC, the competition organizers manually generate a series of prior information given as Route Network Definition File (RNDF) which represents road segments through a graph of nodes and edges that includes information such as lane widths, stop sign and parking locations. This type of information can be generated through automated processes with sensing infrastructure or from direct deduction of vehicle motions. Independently from the method, manual or automated, the path searching problem is linked to a cost of traversing a road segment and subsequent graph search algorithms.

1.1.3.3 BEHAVIOURAL PLANNING

The behavioural planner is important for making decisions on-board through Finite State Machines (FSMs) of different complexity which ensure the interaction of vehicle with other driving agents, the relative response and the respect of stipulated road rules during the increase of the progress along the route prescribed by the mission planner.

This type of Finite State Machines is manually designed for a limited number of specific situations and it can happen that the vehicle is in a situation not explicitly accounted for in the FSM structure, for example in a livelock or in a deadlock state due to a lack of sufficient deadlock protections.

Two terms involved in this phase planning are coined in order to categorize check functions which control logical conditions occurred for certain state transition: (1) precedence observers with the aim to check whether the rules related to the vehicle's current location allow to progress it; (2) clearance observers are needed in order to ensure and guarantee safe clearance to the other traffic participants. In particular, they check the time collision which the shortest time within which a detected obstacle enters in a certain interest region.

1.1.3.4 MOTION PLANNING

Motion planning is a very large research field which space from an application to another one: mobile robots, medicine, security and emergency situation, transportation and agriculture. For example, motion planning applied in the mobile robots application refers to the achievement of a specified goal after a decision process of an actions sequence, typically avoiding collisions with obstacles.

For autonomous transportation, motion planning layer is those responsible for executing the current motion target issued from the behaviours layer. In particular, the motion planner creates a path for the desired goal, then tracks this path by generating a set of candidate trajectories that follow the path and selecting from this set the best trajectory according to an evaluation function. There are different types of evaluation function depending on the context, but it's a choice which includes consideration of static and dynamic obstacles, curbs, speed, curvature, and deviation from the path. The selected trajectory can then be directly executed by the vehicle.

Generally, different motion planners are evaluated and compared one each other in terms of computational efficiency and completeness. Computational efficiency refers to the execution time and to the scalability related to the configuration space dimension. Completeness is related to an algorithm which in a finite time is considered complete. Moreover, this algorithm always returns a solution when one exists and report in the contrary case.

Considering that motion planning problem has a huge computational complexity the challenge became transforming the continuous space model into a discrete model. There are two methods for approaching to this transformation:

- combinatorial planning, which perfectly represents the machines original problem starting from a discrete representation;
- sampling-based planning, which apply a discrete sample searching from the configuration space through a collision module.

1.1.3.5 COMBINATORIAL PLANNING

Combinatorial planners have the function to find a complete solution starting from a built discrete representation of the original problem. The combinatorial methods are limited in application because the computational load increases with the configuration space dimension and with the number of obstacles. For this reason, the sampling-based algorithms are more used than the combinatorial ones.

1.1.3.6 SAMPLING-BASED PLANNING

Sampling-based methods are mainly applied over continuous space and based on a random sampling of them, and the generation of tree or roadmap which represent a feasible trajectory graph. The feasibility is guaranteed and verified thanks to a collision checking of nodes and edges which connect these nodes; in this checking some discretization can occur. For these reasons the sampling-based algorithms are considered as the most popular algorithms for their capability to have probabilistic completeness, in other words which guarantee enough checking time for infinite samples (if a solution exists, the probability to find it converges to one). A good coverage and connection of all obstacle-free spaces should be ideally provided by the generated roadmaps and its paths used to obtain solution of the original motion planning problem.

The most influential sampling-based algorithms primarily differ one each other in generation of a search tree. Two of the most know of this type algorithms are:

- Probabilistic Road Maps (PRM);
- Rapidly-exploring Random Trees (RRT).

PRM is a multi-query method which is effective in planning in high-dimension spaces and in particular able to generate and maintains multiple graphs simultaneously. On the contrary,

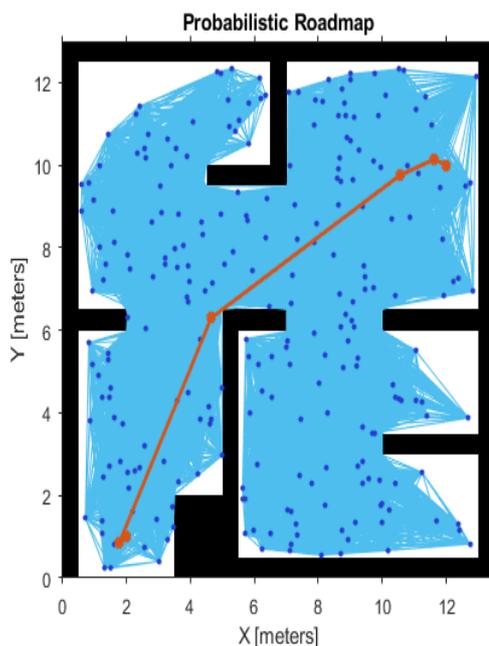


Figure 1.14: Probabilistic Road Maps (PRM)

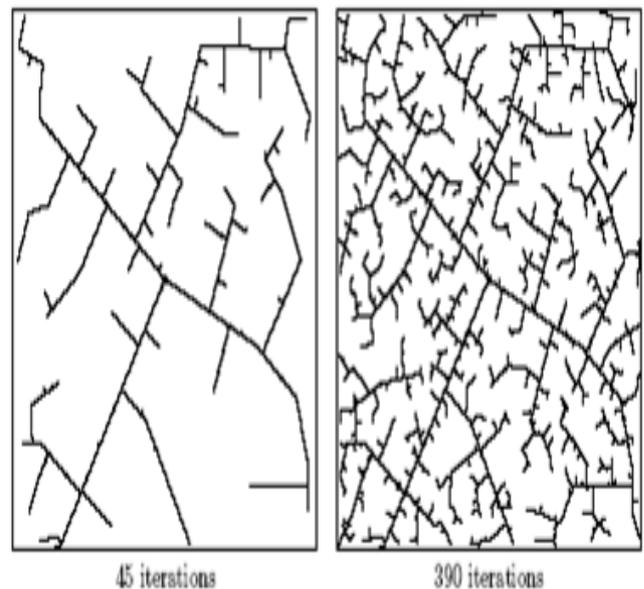


Figure 1.15: Rapidly-exploring Random Trees (RRT)

RRT method rapidly expand a single graph where the map is not well known a priori due to the presence of dynamic obstacles and limited sensor coverage concentrated around the robot's current location.

The quality of the returned solutions is important in many applications, so it must be considered together with completeness guarantees and efficiency in finding that solution. In particular, in many cases it happens that a solution can be found quickly, but for a longer period of time the algorithms continue to run in order to find better solutions based on some heuristics. In the last years, starting from a research of lower cost solutions works, a complete evaluation based on completeness, computational complexity, and optimality of many popular planners was presented with a consecutive proposal of different sampling-based planners and variants of PRM and RRT. From several studies it was highlighted that the popular PRM and RRT algorithms are asymptotically sub-optimal, thus PRM* and RRT* are proposed as asymptotically optimal variants of the first ones then other two variants such as Fast Marching Trees (FMT*) and Stable Sparse Trees (SST*) are suggested to improve the speed with respect to RRT*.

1.1.3.7 DECISION MAKING FOR OBSTACLE AVOIDANCE

One of the typical approaches taken by several DARPA Urban Challenge vehicles was to control specific regions labelled as “critical zones” and potentially exposed to obstacle or connect zones at intersection checking the trajectories of all nearby vehicles in order to determine a “time to collision”. Typically, if there was an imminent collision, the vehicle slows down or stop as a consequence, which was an acceptable behaviour in this situation, but too much conservative in other circumstances. There were cases in which the vehicles needed to adopt “defensive driving” manoeuvre to avoid a dangerous situation. From the

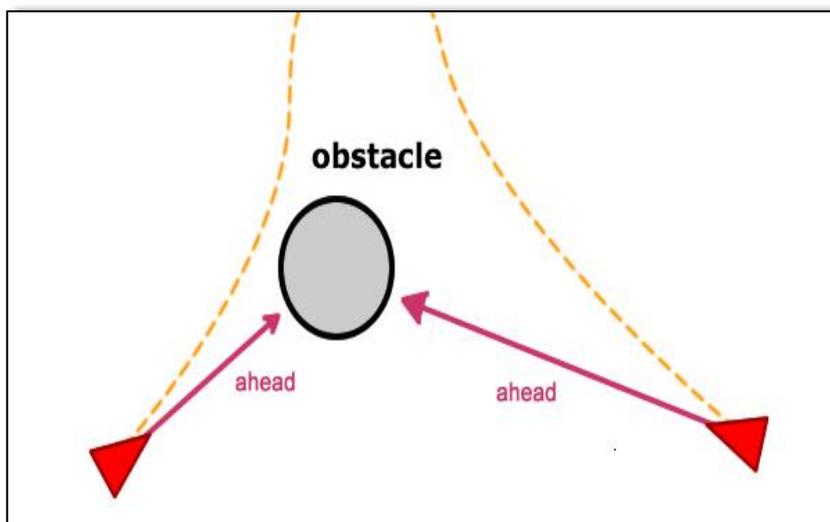


Figure 1.16: Decision making for obstacle avoidance

advantages point of view, these approaches had a computational simplicity since they planned to neglect the time dimension in a low dimensional space. Despite, recent works did not leave the practise of behavioural level decision making for obstacle avoidance, especially in complicated manoeuvres such as lane changing.

1.1.3.8 PLANNING IN SPACE-TIME

It is necessary to include time as a dimension in the configuration space in order to consider in a better way the obstacle movement, but this inclusion increases the problem complexity. If, on the one hand, instantaneous position and velocity of obstacles may be detected, on the other, it is yet difficult to predict future obstacle trajectories.

Previous approaches have used simple assumptions in predicting obstacle movement, such as constant velocity trajectory with errors linked to a rapid iterative re-planning.

Starting from a situation in which it is possible to observe the instantaneous position and velocity of obstacles, it follows that future obstacle trajectories can be predicted under the common assumption of deterministic constant velocity which involves continuous correction or verification through new observations.

Another possible method is to suppose a bounded velocity on obstacles which are represented as conical volumes in space-time with a reduction of updating and re-planning. Other type of on obstacles' assumptions can be applied, such as static assumption and assumption related to constant or bounded velocity and bounded acceleration, each of which produce a limited volume of a different shape in space-time. A more cautious approach would be to hypothesize a large area with the possible presence of obstacles, where the space bounds of the obstacles grow over time based on the limitations of obstacle velocity or acceleration firstly assumed. Obviously, an assumption to avoid would be the one in which the uncertainty related to the prediction of an obstacle's trajectory in the case of obstacle bounded which does not grow over time is ignored. A possible solution can be the one in which a direct plan in the control space is done avoiding specific type of control actions which are predicted to lead to collision.

1.1.3.9 PLANNING SUBJECT TO DIFFERENTIAL CONSTRAINTS

Motion planning is definitively a high-level control problem. Essentially, in order to obtain simplicity or a computation reduction, the control limitations can be ignored in the different levels of motion planner, but this can lead to dangerous operations related to inefficiencies of trajectory and high control errors caused by the poor accounting on the constraints of the system movement. One of the main problems which worries the collision control during the planning phase is related to the discrepancies between the planned trajectory and the one that is essentially performed, and this problem can represent a risk. The trajectories that can be meticulously followed and with longer path length may tend to have shorter execution times

than those that are more difficult to follow, but with shorter path length. Paths can be directly generated from sampling adequate controls, even if the paths won't be optimized through tree rewiring and popular asymptotically optimal planners, such as RRT* require sampling from the configuration space. A possible challenging matter can be the incorporation of differential constraints into state-sampling planners which requires a steering function able to generate and draw an optimal path between two given states which are submitted to the control constraints (if this path exists). Furthermore, querying methods are necessary to tell whether a sampled state is reachable from a potential parent state.

The evolution of time is one of the most important differential constraints in a system, where in general this time t increase at a constant rate $\dot{t} = 1$ as imposition. Independently on the fact that the time is explicitly included as a state parameter, other state parameters will generally have differential constraints with respect to time, such as velocity and/or acceleration limits. Differential constraints are applied to generate velocity profiles and can be solved in two ways: along the geometric path chosen in a decoupled way, or simultaneously solving the geometric path on each connection in the tree in a direct integrated way. The management of the decoupled differential constraint can cause in very inefficient trajectories or not to find a trajectory due to decoupling. On the other hand, the differential constraint managed in a direct integrated manner can lead to improvements, but it's computationally more complex. Common limitations may be related to the radius of rotation that have often been resolved through Dubins curves or Reeds-Shepp curves, which have been shown to ensure a shorter distance given a minimum turning radius.

An efficient state sampling can be made more by limiting the sampled states to only those from within a set of states known to be reachable from the initial condition given the system's kinodynamic constraints applied to an obstacle free environment. Similarly, it is only convenient to check for connectivity between neighbouring states when they are part of each other's reachable sets. Checking any states that are nearby according to Euclidean distance metric but not reachable in a short period of time given kinodynamic constraints cause a waste of computational effort. A possible solution can be adding Reachability Guidance (RG) to state sampling and Nearest Neighbor (NN) searching which provide important efficiency boosts to planning speed, especially for particular condition systems. These systems can be those where motion is highly constrained or the motion checking cost is high. In different recent works it was highlighted how the RG can be incorporated in the motion planning through analytical approaches. To handle differential constraints, it is possible to exploit an asymptotically optimal sampling-based algorithm, Goal-Rooted Feedback Motion

Tree (GR-FMT), restricted in application of controllable linear systems with linear constraints. Furthermore, an important analytical method was presented in order to solving a two-point boundary value problem subject to kinodynamic constraints. This method was limited to systems with linear dynamics, but it could be used for finding optimal state-to-state connections and NN searching.

Another important used approach was the machine learning approach which had the aim to verify whether a state was reachable from a given base state, even though this method required the application of a Support Vector Machine SVM classifier over a 36-feature set for the Dubins car model which could be extremely expensive from the computation point of view. During the recent years there were also relatively few planning effective methods for solving over a configuration space having an appended time. Every job has taken a different path: some explored control sampling approaches providing model simplifications to handle the differential constraints in an online manner, others have performed planning with discrete, time-bounded lattice structure based on motion primitives, or a grid cell decomposition of the state space.

1.1.3.10 INCREMENTAL PLANNING AND REPLANNING

The most common challenges in the autonomous vehicle planning are mainly related to the limited perception range and the dynamic nature of operating environments. Typically, the sensing range is limited not only by sensor specifications, but also reduced for the presences of obstacles which obstructs the view.

It often happens that the system will not be able to perceive the entire path from a starting location to goal location at any one specific instant of time. Thus, for this reason there is the need to generate incremental plans in order to follow trajectories which allow to forward progress towards the final goal location.

One key aspect to consider is that the system performs its planned trajectory, but other mobile agents who have their own objectives can move unexpectedly. Therefore, the environment changes continuously and those trajectories that in a prior time instant are considered safe, in a subsequent time instant may no longer be so. For this reason, it is necessary to apply a substitution in order to regulate the dynamic changes of the environment. This incremental planning mechanism requires a means to generate incremental sub-targets, or alternatively to choose the best trajectory between a set of possible trajectories based on some heuristics.

At least a new plan must be generated with the same frequency of new sub-goal definitions. Given the different planning situation can happens in some cases that no sub-goals were defined or there wasn't a predefined path and thus, the best choice are the trajectories selected based on a combined weighted heuristic of trajectory execution time and distance to goal from the end trajectory state. Bouraine et al. have applied a constant rate replanning timer in which each current solution plan was executed concurrently with the generation of subsequent plan, and each newly planned trajectory would be rooted from an anticipated committed pose given the previous committed solution trajectory.

Replanning in iterative way to generate new solution trajectories represents a potential opportunity to transfer knowledge from previous planning iterations to subsequent ones. If prior planning information is well utilized while a new plan could start from scratch, better solutions may be found faster. In other works, it is suggested that redoing collision-checks over the entire planning tree, as in Dynamic RRT (DRRT), in which the tree structure was utilized to trim child “branches” once a parent state was found to be no longer valid. Recently, it was presented a replanning variant of RRT*, RRTX, which trims the previous planning iteration’s planning tree, but efficiently reconnects disconnected branches to other parts of the tree maintaining the rewiring principal of RRT* responsible for asymptotic optimality.

Safety mechanisms are an aspect that should also be carefully designed considering that a finite time for computation are required for each planning cycle and the environment may change during that time. The problem of obstacle presence has as response passive safety mechanisms prescribed by several works, where passive safety refers to the ability to avoid collision while the system is moving. In general, velocity planning was decoupled from the path planning, and a particular approach called “Dynamic Virtual Bumper” would prescribe reduced speed based on the proximity of the nearest obstacle as measured by a weighted longitudinal and lateral offset from the desired path. In particular, moving obstacles were treated as enlarged static obstacles with the assumption that they occupied the area traced by their current constant velocity trajectory over a short time frame in addition to their current spatial location. [5]

CHAPTER 2

AUTONOMOUS PARKING

AUTONOMOUS PARKING

2. AUTONOMOUS PARKING EVOLUTION

The first approach to autonomous parking was in the early 2000s when a parking assistance technology was implemented by using sensors or cameras helping the driver to control the parking manoeuvre.

The main parking sensor used was the ultrasonic sensor that can detect objects through the use of high frequency sound waves. The sensor is installed on the rear part of vehicle; therefore, it can help the driver to detect a wall or another vehicle during parking. A receiver detects these waves and calculates the distance from the object to vehicle, in the case when the object is too close to the vehicle, the driver is warned via a continuous beep noise which becomes more rapid the closer the car is from the object.

The first car to feature such sensor was Toyota Prius, released in 2003. But ultrasonic sensors may not be able to detect objects that lie flat on the ground or that are too far or too close to the car; moreover, always ensure that the sensors are clear of all debris and dirt that may interfere with accurate detection.



Figure 2.1: Toyota Prius sensors (2003)

For this reason, video cameras were introduced to car parking technology. At the beginning only a rearview camera was used but then many companies offered surround-view system, providing 360° video coverage and the images could be shown on a split screen in conjunction with forwarding, rear or side views.

Nissan was the first to develop a surround-view camera at the end of 2007 and starts the business of this type of technology.

In the last years, many new cars implement form of parking assist which is more advanced form of parking aid. One of these make the car's onboard sensors active and they begin

scanning for appropriate parking spaces determining whether space is of a reasonable size for parking the car. Another functionality shows the intended reverse course via onboard multifunctional display, the driver need not to take control of steering wheel which allows him to retain full control of the clutch, accelerator and brake.

First version of park assist was introduced in 2003 but it has become widely available in the last years, especially Kia and Ford.



Figure 2.2: Parking vision camera

Furthermore, Tesla, Mercedes and Volvo have also introduced self-parking facilities. The goal now is to get a driverless parking working independently; many car companies have made claims of introducing cars that are driverless, means that they could drop you off at



Figure 2.3: Volvo autonomous parking

desired destination after which they head off to find a suitable parking spot. [6]

The next paragraph treats several different contributions in terms of research and studies about autonomous parking in the last decades.

Each type of approach to the problem has different way to manage the parking situation.

2.1 DIRECT TRAJECTORY PLANNING: HUMAN-LIKE PARKING

The problem related to the parking control remain something to be solved for autonomous vehicles. Generally, the approaches that already exist first design a parking reference trajectory that does not exactly respect vehicle dynamic constraints and second apply specific online negative feedback control to make the vehicle roughly track this reference trajectory.

The main purpose of designing autonomous vehicles is to complete most driving tasks instead of human drivers. These tasks requirements become complicated also for low speed scenario met by every driver in each day: traffic and parking scenario.

The parking scenario control problem for autonomous vehicles can be define adopting a step-by-step strategy that allow to choose the steering action according to the comparison between the current state pose (position/orientation) and final one.

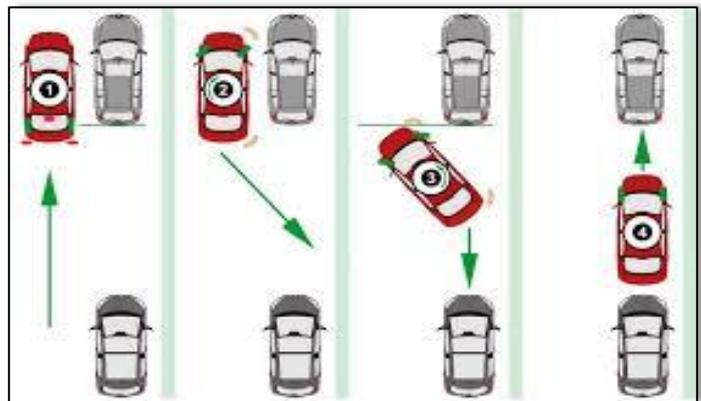


Figure 2.4: Direct trajectory planning

Considering the most existing approaches there are some of these which determine the steering actions using heuristic rules. Then the problem is to prove whether the action is valid or optimal. However, using heuristic rules the choice of the right steering actions is difficult if there are obstacles around the parking.

An indirect trajectory planning method is used to solve this problem by an increasing number of researchers which considers the parking control problem as finding a good trajectory along which an autonomous vehicle can track moving from a given initial pose to a given finale one through a sequence of steering actions.

Obviously, the steering angle for an autonomous vehicle is limited, unlike many mobile robots which have a steering angle variation up to 360 degrees, indeed since the geometry property of trajectory planned is affected by the vehicle dynamics. Moreover, the objective is to find the best compromise between the shortest length and shortest time trajectory among the infinite number of trajectories that create a link between the initial and final state.

There are many indirect trajectory planning methods which solve this problem approaching with a design of a reference parking trajectory that an autonomous vehicle could approximately follow.

Typically, the reference trajectories are particular curves like β -spline curves or polynomial curves with specific geometric properties to obtain a simplification of planning and presentation of the desired trajectories. The gap between the real actual trajectory through the application of the correspondent steering actions and reference parking trajectory will be restricted by means a negative feedback controller to reshape the steering actions and make the vehicle approximately track this reference trajectory. However, it is important to consider that the gap can vary, and it could have deviations along the different part of the trajectory which could cause collision with the obstacles around especially in reverse parking situations.

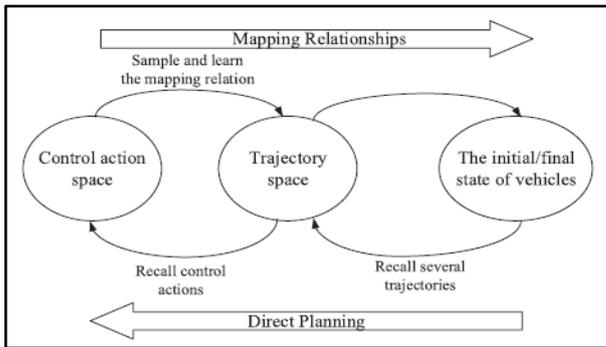


Figure 2.5: Planning scheme

To solve this problem, a possible solution can be the application of the direct trajectory planning method with the idea to number all the valid and possible parking trajectories that a vehicle can make and learn to set up different relationship between any initial/final state

couple and the derived steering actions. After this “numeration”, if an initial/final state pair is given, the autonomous vehicle recalls the desired steering action/parking trajectory. By considering some surveys, it is simply to note that many mature human drivers know in advance the steering actions to apply for parking before beginning to park by just observing the position/orientation of the final parking lot and recalling it in their mind. It is a significant fact which proves that this human drivers aspect is something similar to a direct trajectory planning. The important aspect of this approach is to better establish the relationship between an initial/final state pair and the parking trajectory.

In literature, the direct trajectory planning method can solve the parking problem both in a single-stage style and in a multi-stage one.

In the first case, the solution space dimension increases with time length of the trajectory. It is possible to sample the solution space with a reasonable resolution level, but the computation cost can be huge. For this motive it is needed to consider the trajectory in its importance and the accuracy of the whole trajectory is not to be evaluated. This happens because in moving the vehicle to the desired parking lot, the first half of the trajectory is most important than the second half one. So, it is possible to use a rough resolution sample level for the first half trajectory and a finer resolution level for the second one to reduce the computational cost.

Another well-known feature, based on a series of considerations, to describe the vehicle movements with the direct planning method is that to a dynamic model rather than a kinematic model. First, the general trajectory planning method is designed not only for the parking problem, but it is also extended for other vehicle motion planning/control problems. Second, the complexity of the conventional indirect trajectory planning method that use kinematic model for vehicle movements is greater than direct trajectory planning methods. In a direct planning approach, the design cost of a controller is saved, while in a kinematic approach it is needed to design a controller make and cost increases the calculation complexity. Third, it is very difficult to obtain a very strict gap between the ideal trajectories and the obtained one in indirect trajectory planning methods. Fourth, using the direct planning method, it is possible to not consider vehicle kinematic models with very low speed requirements.

Furthermore, it is possible to assimilate in the same manner both the front steering vehicles parking problem and the full steering vehicles one.

To sample trajectory solution space the bicycle model in a flat surface is adopted to describe the dynamics of a vehicle. In the model of the figure, CG is chosen as the reference point center of gravity of the vehicle body and its (x, y) coordinate represents the position of the vehicle.

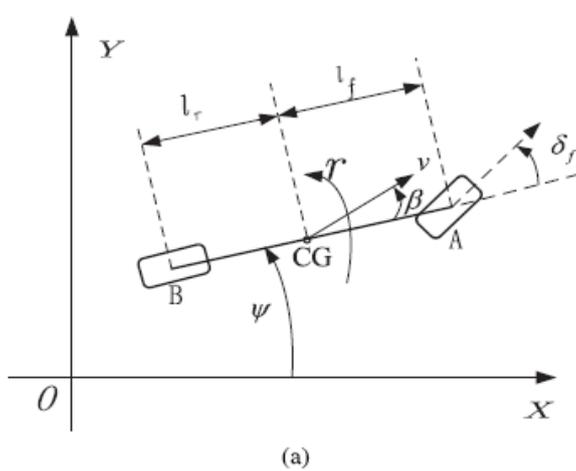


Figure 2.6: Bicycle model - Forward motion

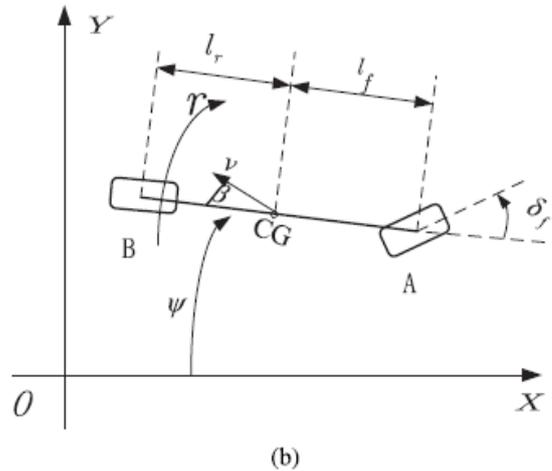


Figure 2.7: Bicycle model - Backward motion

Vehicle velocity v is defined at the center of gravity CG point in which is fixed the origin at the initial time, while the heading angle ψ is the angle from the X -axis to the longitudinal axis of the vehicle body AB and the slide-slip angle β is the angle from the longitudinal axis of the vehicle body AB to the direction of the vehicle velocity.

The state space model of the bicycle model can be written as:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

with coefficients

$$a_{11} = -\frac{c_f + c_r}{mv} a_{12} = -1 - \frac{c_f l_f + c_r l_r}{mv^2}$$

$$a_{21} = -\frac{c_f l_f + c_r l_r}{I} a_{22} = -\frac{c_f l_f^2 + c_r l_r^2}{Iv}$$

$$b_1 = \frac{c_f}{mv} b_2 = \frac{c_f l_f}{I}$$

The direction of the positive X -axis is assumed to point to the head of the vehicle, while v_x and v_y are the projection of the velocity v onto the X, Y axes.

If the vehicle goes forward the equations are the follows:

$$v_x = v \cos(\beta + \psi)$$

$$v_y = v \sin(\beta + \psi)$$

while, if the vehicle goes backward the equations became the follows:

$$v_x = -v \cos(\beta + \psi)$$

$$v_y = v \sin(\beta + \psi)$$

Thanks to these equations it is possible to calculate the position and the orientation of the vehicle during the parking.

Symbol	Meaning	Value
$X - Y$	Coordinate system	
β	Vehicle sideslip angle	
ψ	Heading angle	
r	Yaw rate of the vehicle, $r = \dot{\psi}$	
δ_f	Front steering angle	
v	Vehicle velocity	
δ_{max}	Maximum of the front steering angle	0.6 rad
m	Mass of the vehicle	1500 kg
I	Inertia moment around the vertical axis through CG	$2500 \text{ kg} \cdot \text{m}^2$
l_f	Distance from point A and point CG	1.20 m
l_r	Distance from point B and point CG	1.50 m
c_f	Front tire stiffness coefficients	80000 N/rad
c_r	Rear tire stiffness coefficients	80000 N/rad

Given the value of input vehicle speed and front steering angle within a time range the resulting trajectory of the vehicle is calculable based on the dynamic model that allow to generate a sample mapping relation between the two variables and the trajectory. To obtain a complex and richer mapping relation it can be assumed that the vehicle keeps a low velocity of 1/m/s during the whole parking process, from the starting process to the stopping one (instantaneous parking assumption). If enough samples are obtained, a mapping relation is more simply to organize.

The idea is to discretize the control variable value of δ_f along the time axis and obtain a set of allowable parking trajectories by assigning different steering sequences $\delta_f^N = \delta_f(1), \dots, \delta_f(N)$ for N consequent time segments. The steering angle is chosen from a set of K angle $S_1 = \delta_1, \delta_2, \dots, \delta_K$ and it is kept constant for the entire time segment. Usually, the samples are done to obtain an allowable value range of δ_f written as $[-\delta_{max}, \delta_{max}]$.

Some example figures about this procedure can be more understandable with respect the simply theory:

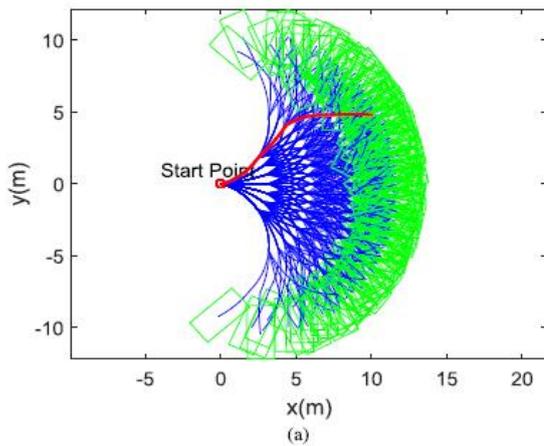


Figure 2.8: Possible trajectories

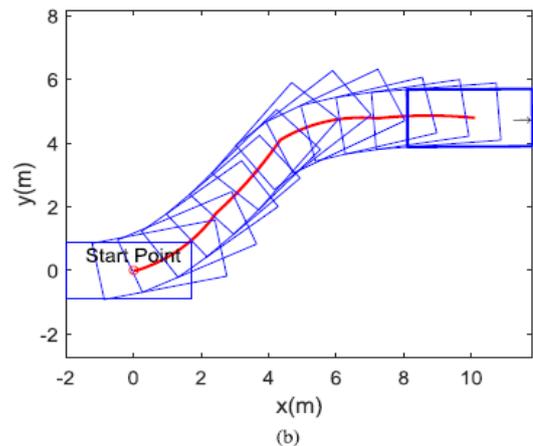


Figure 2.9: Specific trajectory

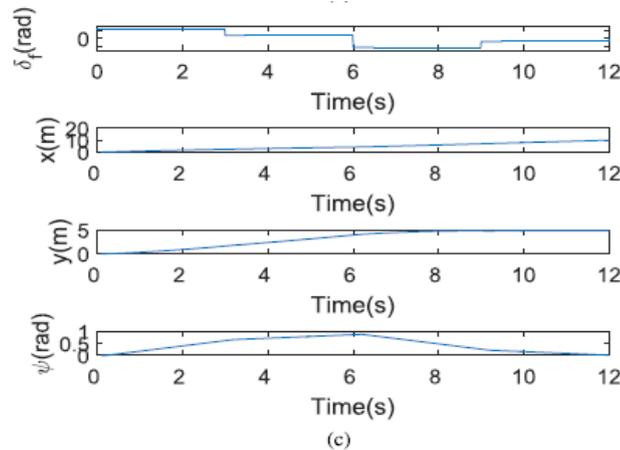


Figure 2.10: Steering angle sequence

in which are represented the set of possible trajectories (a), a specific trajectory (b) and the corresponding steering angle sequence (c).

The Single-Stage direct trajectory planning is based on the whole control action sequence generated at the beginning of the parking process and the difference with respect to the indirect trajectory planning method is that in this last case the trajectory is generated by vehicle dynamics, while in the first case through approximated curves.

All the found trajectories of the SSDP are stored in discrete form as a consecutive series of states thanks to a discretization with a specific chosen time interval. R1 will denote the final obtained trajectory and a threshold is set in order to understand if a given final state can be matched with a stored trajectory by controlling the distance between this state and R1.

Since there exist lots of trajectories, the important trajectory to be stored is the optimal one and to do this a performance index of a trajectory is defined:

$$J = \int_0^{\tau} \left(v(t) + \delta_f^2(t) + c(t) \right) dt + [h(x_{\tau}, y_{\tau}, \psi_{\tau}) - h(x_p, y_p, \psi_p)]^2$$

with τ terminal time, $c(t)$ curvature at time t along the trajectory, $(x_{\tau}, y_{\tau}, \psi_{\tau})$ final state of the vehicle and (x_p, y_p, ψ_p) stop state of the vehicle at the parking lot center.

h is the surface integral of the function $f(x, y)$ with respect to the rectangular region D_t which represents the vehicle at the state $(x_{\tau}, y_{\tau}, \psi_{\tau})$.

$$h(x_{\tau}, y_{\tau}, \psi_{\tau}) = \iint_{D_t} f(x, y) d\sigma$$

A vehicle parked close to the center of the berth corresponds to a small integral.

The hinge point is that the function $f(x, y)$ is defined in a new coordinate system with the origin placed at the center of the parking lot and with the axes parallel to the two vertical edges of the lot. It can be written as:

$$f(x, y) = \begin{cases} kx, & -kx < y \leq kx; \\ -kx, & kx \leq y \leq -kx; \\ y, & -y < kx < y; \\ -y, & y < kx \leq -y; \end{cases}$$

with k ratio between the length and the width of the parking berth.

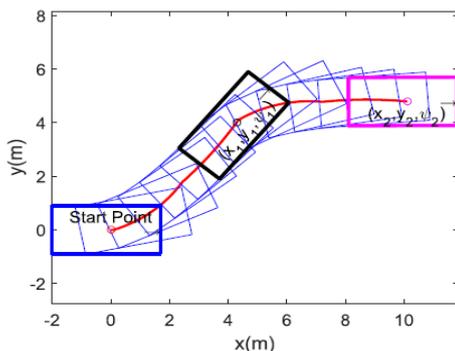


Figure 2.11: From the initial to the finale state

To have an efficient parking to the final state, it is necessary to monitor it and stop when the vehicle reaches the (x_1, y_1, ψ_1) position.

The main difficulty of this parking control is linked to the cost associated to the inquiry and cost of the storage, because the control action space might be huge. [7]

2.2 INTELLEAGENT AUTONOMOUS PARKING CONTROL SYSTEM

A parking trajectory can be planned starting from a mathematical formulation of the problem based on finding the minimum length of the trajectory and minimum number of maneuver space. Parking control of this type can be based on the fuzzy logic evaluating the problem from three aspects:

- detection of the parking berth;
- evaluation of the present position and path generation;
- trajectory correction through the motion.

For the first aspect, the spatial orientation and parking detection control are managed by means a lot of sensors such as lidars, laser scanners, video sensors, ultrasonic sensors or cameras.

The bicycle model can be most important assumption and variant of the mathematical motion description of a vehicle. The model assimilated as a rigid body has a rotated front wheel (vertical axis) to control the vehicle movement and the trajectory and a fixed and nonrotative rear wheel (x_v center of the rear axle which shows the direction of movement).

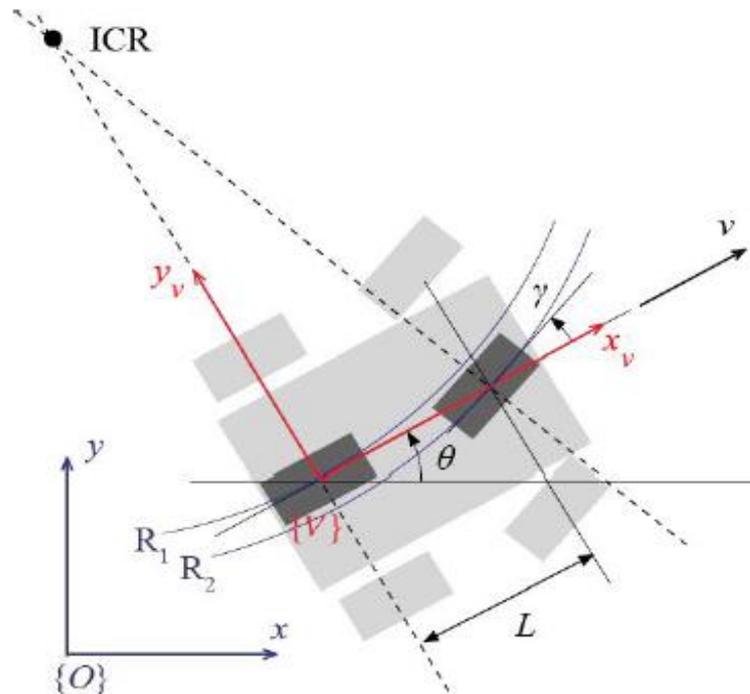


Figure 2.12: Bicycle model representation

The vehicle position is represented by the coordinates $\alpha = x, y, \theta$ in the coordinates system θ where x and y are reference point in the coordinates system V associated with the vehicle and θ the angle between the x and x_v axis. $2L$ is the distance between the rear and front wheels axles, ICR is the instantaneous center of rotation of the vehicle, while y and y_v are the steering angle and the direction of rotation respectively.

The relation and dependence between orientation, rotation angle of the steering wheels, vehicle velocity and vehicle current position coordinates are described by this equation system:

$$\begin{cases} \dot{y} = v \cdot \sin \theta \\ \dot{\theta} = \frac{v}{L} \cdot \tan \gamma \\ \dot{x} = v \cdot \cos \theta \end{cases}$$

through the geometric transformations it is possible to determine the velocity projection of the plant in O coordinates systems on y axis in the coordinates system V .

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0$$

The next step is the path generation which it must be as accurate as possible to minimize the parking area and to perform motion along two arcs without straight line between them through the maximum rotation angle of the steering wheels.

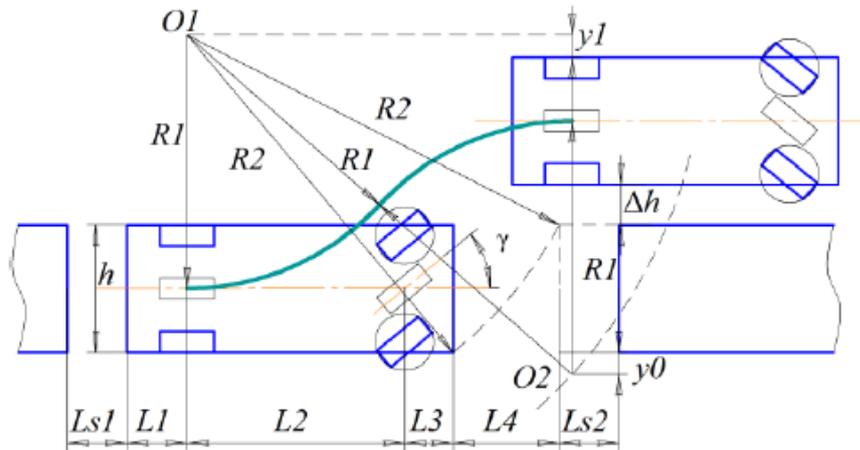


Figure 2.13: Steering wheels and maximum rotation angle

Circle radius can be calculated given a known steering angle γ : $R_1 = L_2 \tan \gamma$. Obviously, the safety condition must be guaranteed by correct maneuvers which avoid hitting front and back car corner. The safety system become an important independent system which estimate risks and prevents its if necessary. So, it can possible define to safety zones L_{S1} and L_{S2} and the minimum parking trajectory length S calculated as: $S = L_{S1} + L_1 + L_2 + L_3 + L_4 + L_{S2}$

Since information about vehicle coordinates and orientation could be reachable in each period of time, the vehicle's attitude relative to the surrounding area could be calculated in each time moment. S' and S'' are sensors ray beams. S' has a slop coefficient equal to $k = \tan \theta$ with θ angle of rotation.

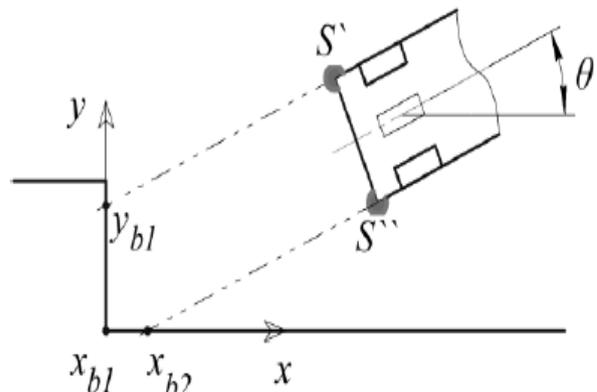


Figure 2.14: Sensors information

The piercing point of a line corresponding to the sensor's ray beam with axes of reference coordinate system could be calculated with this equation:

$$y_{b1} = k \cdot x_{b1} + b$$

which allow to obtain the coordinate of the piercing point x_{b1} and y_{b1} that are the limit points of safety maneuver completion zone.

The sensors coordinate must be converted into reference coordinate system in order to find coefficient b :

$$b = y_p^0 - \tan \theta \cdot x_p^0$$

where x_p^0 and y_p^0 are the distance coordinates of the sensor in reference coordinate system.

For the sensor S' the body axes coordinates are stored in the vector v_p :

$$v_p = \begin{bmatrix} x_{s1} \\ y_{s1} \\ 0 \end{bmatrix}$$

that then converted in reference coordinate system become the following ones:

$$v_p^0 = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{s1} \\ y_{s1} \\ 0 \end{bmatrix} + \begin{bmatrix} x_v^0 \\ y_v^0 \\ 0 \end{bmatrix} = \begin{bmatrix} x_p^0 \\ y_p^0 \\ 0 \end{bmatrix}$$

The minimal safety distance to the nearest obstacle could be calculated in this way:

$$l_{max1} = \sqrt{l_x^2 + l_y^2} = \sqrt{(x_p^0 - x_{b1})^2 + (y_p^0 - y_{b1})^2}$$

this algorithm allows to realize the fuzzy controller which performance are base on information about normalized distance to obstacle calculated as follows:

$$l_{n1} = \frac{l_{s1}}{l_{max1}}$$

where l_{s1} is the distance derived from the sensor S' .

The fuzzy controller has two inputs which are normalized distance values obtained by means ultrasonic sensors, while the output is the normalized vehicle velocity.

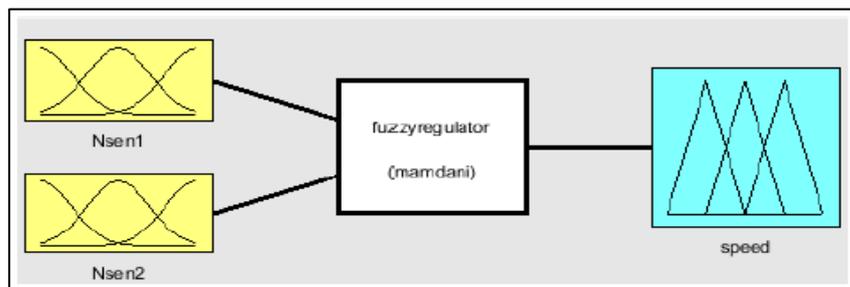


Figure 2.15: Fuzzy controller

The working principle is based on the distance controlling: distance sensor less than minimal safety distance, then velocity is reducing up to emergency stop in proximity of the obstacle.

The obtained result is changing the vehicle velocity according to the information derived from the proximity sensors.

For the distance sensors inputs and outputs, membership functions are presented:

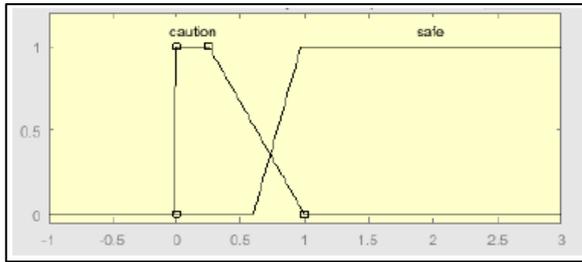


Figure 2.16: Inputs and outputs of controller

- Input $Nsen1$;
- Input $Nsen2$;
- Output includes two membership functions for controlling vehicle velocity.

There are two rules/conditions in this controller:

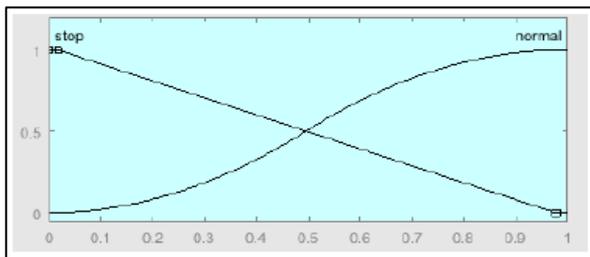


Figure 2.17: Controller conditions

1. if ($Nsen1$ is *safe*) AND ($Nsen2$ is *safe*) then the (*Speed* is not changed (*normal*));
2. if ($Nsen1$ is *caution*) OR ($Nsen2$ is *caution*) then the (*Speed* is reducing (*stop*));

This type of algorithm prevents crashing with obstacles which could appeared on vehicle trajectory. In particular, the system is able to guarantee a complete maneuver taking into account limitations and safety zone. [8]

2.2 FAST PARALLEL PARKING USING GOMPERTZ CURVES

The procedure of this method is based on the identification and preselection of a smooth sigmoid trajectory which is called Gompertz curve in parametric form.

The parameter of trajectory is real-time determined during the phase of path-planning by means a scheme which allows to generate an optimal candidate path taking into account the maximum steering angles physically realized.

The next step is to check if the candidate trajectory generates collisions and re-parametrized the trajectory to arc-length form through cubic interpolation method. The final step is following the parametrized path in reverse using odometry to park the vehicle with a single maneuver. This maneuver is one of the most arduous ones and it is needed for parallel parking in which a reversing movement into a parking space between two co-linearly parked vehicles happens.

From experience and several observations, it is possible to assert that once followed, exists a single trajectory that enables a precise parallel parking in a single motion. The additional maneuvers need for straightening the vehicle inside the parking berth.

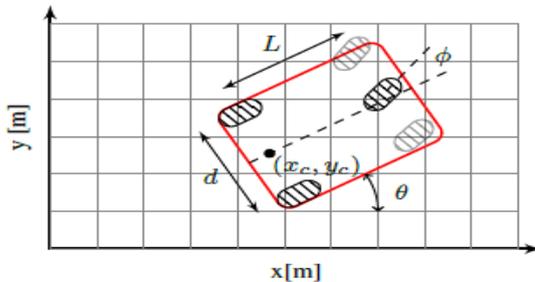
There are lots of effort and contributions for approaching the autonomous parking problem which face it in different way and from various point of view to arrive to a possible optimal final solution:

- hardware solutions;
- advanced fuzzy logic software solutions;
- fuzzy controllers;
- techniques involving stereo-vision parking space detection by 3D reconstruction;
- path planning based on overhead;
- Model Predictive Control (MPC);
- combination of probabilistic techniques with open and closed loop approaches.

Basically, considering the different cited aspects the method introduced in this chapter has several important features:

1. instantaneous laser scans and no priori information like overhead maps required;
2. variable, but safe speeds of the vehicle;
3. slip and odometry errors considered for kinematic model;
4. enter trajectory generated can be also used to exit from the parking berth;
5. vision is not required;
6. single-maneuver guaranteed;
7. user-input is not required;
8. parameters involved are less than other methods;
9. simplified path planning due to pre-selected path model in which only the parameters are determined;
10. path model based on *Gompertz* curve.

For this method a standard four-wheeled car-like robot, rear-wheel driven with Ackerman steering is considered and modelled as a bicycle with the following kinematic equation:



$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \frac{v}{L} \tan \gamma \end{bmatrix}$$

Figure 2.18: Four-wheeled car-like robot modelled as a bicycle

where q is the generalized coordinates and v the driving velocity. d_1 and d_2 are the distances travelled by the two rear wheels (determined by means the encoder ticks) and the associated midpoint (x_c, y_c) has a travel distance of:

$$d_s = \frac{d_1 + d_2}{2}$$

which is the reference point for the trajectory generation process.

Given a trajectory expressed in parametric form, its parametrized equations (by arc-length) and input information derived from d_s it is possible to determine the location of this reference point on the considered trajectory.

Slip-affected position of the vehicle on trajectory and slip pre-emption are involved in the results in terms of odometry errors and in lateral and longitudinal directions of travel:

$$E = \begin{bmatrix} k_1 & d_s \\ k_2 & d_s \end{bmatrix}$$

with E error vector, $k_1 > k_2$ such that $E(1)$ and $E(2)$ compose the major and minor axes of an ellipse. In particular:

$$1 = [x \quad y] \begin{bmatrix} \frac{1}{E(1)^2} & 0 \\ 0 & \frac{1}{E(2)^2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Describes the error ellipsoid related to the predicted dimensions centered on the in theory correct vehicle position (x_c, y_c) . Given the lateral slip in the y direction, it is possible to understand that it is greater than the longitudinal slip in the x direction and since the odometry error has a proportional evolution in which increase with the distance travelled, lateral slip $E(1)$ will be $k_1\%$ of distance travelled and longitudinal slip $E(2)$ as $k_2\%$.

The main part of the approach is using the *Gompertz* curve as model trajectory to realize a single manoeuvre parallel parking (sigmoid-like trajectory). One extremity of the curve is the reference point on the vehicle at starting position and the end extremity is where the reference point rest after the parking. While a perfect parking was executed, an external laser range scanner tracks the vehicle motion with a particular attention to robot centre by acquiring the edge of the vehicle during the manoeuvre.

The figure shows how the vehicle centre moved itself in sigmoid-like form accordingly with selected and controlled *Gompertz* curve based on specific parameters.

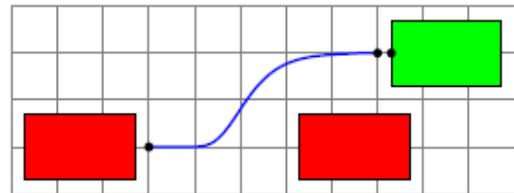


Figure 2.19: Gompertz curve movement of the vehicle centre

The *Gompertz* curve $\hat{g}(t) = \langle x(t), y(t) \rangle$ is part of sigmoid family and it is expressed in parametric form with t parameter on the domain $[0, t_{end}]$ as:

$$\hat{g}(t) = \langle t, ae^{be^{ct}} \rangle$$

or in non-parametric form:

$$g(x) = ae^{be^{cx}}, \text{ for } \forall t \in 0 \leq t \leq t_{end}$$

with $x(t)$ and $y(t)$ smooth and continuous in an interval.

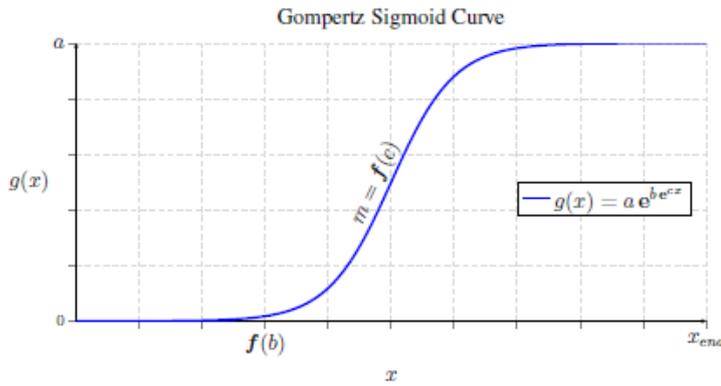


Figure 2.20: Gompertz Sigmoid curve function

The parameters (a, b, c) of the *Gompertz* curve allow to modify and influence the geometrical shape of the curve and to control a single property related to the curve. Basically, the path planning take place only on the parameters for determining the curve instead of the entire trajectory:

- *Parameter a* – for all $t > 1$, $\lim_{t \rightarrow \infty} e^{-ct} = 0$

The value of a affects and defines the width of trajectory, that is the upper asymptote of the curve. Basically, by considering the cited limit $e^{-\infty} = 0$ and thus $y = ae^0$ and since $e^0 = 1$, then $y = a$. So, the parameter a become a function of the parking space widths, initial position of the vehicle and surrounding ones.

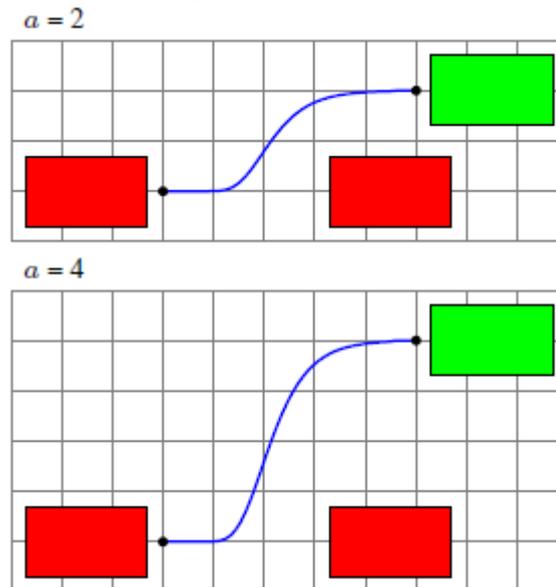


Figure 2.21: Parameter a evaluation

- *Parameter b* – non-positive number that defines the point of the curve in which it starts rising or in the real application of the parallel parking when the vehicle rear extremity starts aligning parallel to the stop line.

b may reach a range of values comparable to a which assume a single value instantly determined. In particular, b determines how much time the vehicle spend to complete

an aligned parking. A higher value than nominal one can determinate a need of an additional maneuver for parking, while a smaller value can cause a collision with the front vehicle.

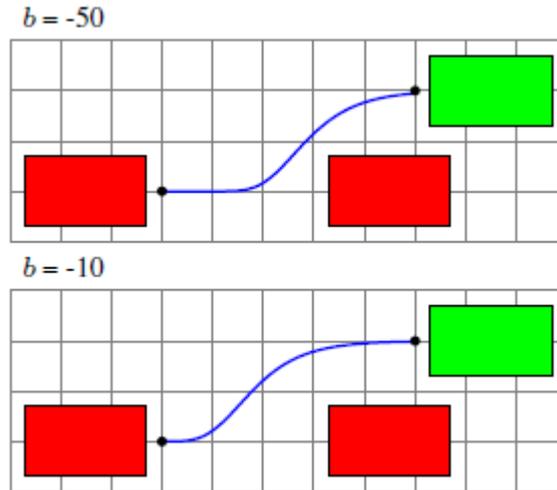


Figure 2.22: Parameter b evaluation

- *Parameter c* – non-positive number related to the slope of the curve. A higher value implies a gradual slope and soft turns, while lower value cause steeper inclination of the curve and abrupt turns. The value can depend on the constraint on the space. A constrained space implies lower value, contrarily a not constrained space involve higher value.

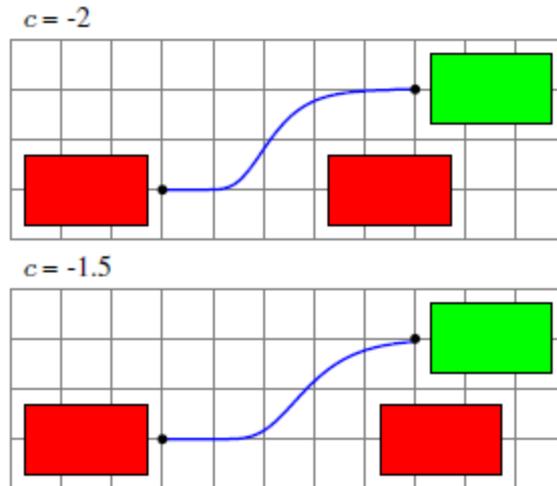


Figure 2.23: Parameter c evaluation

The information required to determinate the parameter value is extracted by means instantaneous laser scanner mounted on rear vehicle extremity midpoint, especially a and in iterative way the parameter b and c .

Once obtained the valid parallel-parking trajectory and the associated parameters, the *Gompertz* curve has to be re-parametrized as arc length s with this application formula:

$$s = \int_0^{t_{end}} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt$$

Re-parametrization is important because through the distance travelled d_s it is possible to determine the ideal error-free vehicle position on the arc-parametrized trajectory.

The previous formula produces an integral expression:

$$s = \int_0^{t_{end}} \sqrt{1^2 + (a^2 b^2 c^2 e^{2be^{ct} + 2ct})} dt$$

which cannot be solved analytically or reduced further.

Numerical integration allows to compute the value of the total trajectory length s_L , while for the above integral it is necessary to obtain a closed form result in order to have an expression for the arc length s as a function of t , that is $s(t)$.

It is also possible to write the expression in the contrary form, that is t as a function of s , $t(s)$ considering the original parametric equations with a parametrization based on t and re-parametrize to obtain the arc length in which s is the parameter.

In those cases where there is no simple closed form solution in which $t(s)$ can be derived from $s(t)$, the method of the cubic interpolation can be applied for approximating $t(s)$ that otherwise it is impossible to obtain analytically through cubic polynomial.

Therefore,

$$t(s) \approx c_1 s^3 + c_2 s^2 + c_3 s + c_4$$

where the four constants are so defined:

$$c_1 = \frac{1}{(s_L)^2} \left(c_3 + \frac{1}{\|c'(1)\|} \right) - \frac{2}{(s_L)^3}$$

$$c_2 = \frac{t_{end}}{(s_L)^2} - \frac{c_3}{s_L} - c_1 s_L$$

$$c_3 = \frac{1}{\|c'(0)\|}$$

$$c_4 = 0$$

By means a substitution of the previous relation using these coefficients in the first original parametric equations parametrized on t allows to obtain the re-parametrized *Gompertz* curve, as:

$$\hat{g}(t) = \langle x(s), y(s) \rangle = \langle c_1 s^3 + c_2 s^2 + c_3 s, a e^{be^{c(c_1 s^3 + c_2 s^2 + c_3 s)}} \rangle$$

Imposing $s = s_s$ a suitable sampling distance which is the travelled distance on the trajectory of the vehicle. When these distances increase $s_s, 2s, 3s, \dots, s_L$ and they are substituted in relation of $t(s)$ it is possible to determinate the error-free vehicle position at every sampling distance s .

The successive step is to find the tangent vector at each error-free position on the trajectory and so, for a curve $\hat{g}(t|s)$ parametrized by the parameter t or arc length s , there exist a unit tangent vector $\hat{T}(t|s)$ such that:

$$\hat{T}(t|s) = \frac{\hat{g}'(t|s)}{\|\hat{g}'(t|s)\|}$$

with $\|\hat{g}(t|s)\| = \sqrt{x'(t|s)^2 + y'(t|s)^2}$, where in this particular case are:

$$x'(s) = 3c_1s^2 + 2c_2s + c_3$$

and

$$y'(s) = abce^{be^{c(c_1s^3+c_2s^2+c_3s)}} e^{c(c_1s^3+c_2s^2+c_3s)} \times (3c_1s^2 + 2c_2s + c_3)$$

The unit tangent vector and the vehicle orientation has the same direction and the latter one is strictly related to the orientation of the vehicle θ which can be determined at each position based on $s_s, 2s, 3s, \dots, s_L$:

$$\theta(s)|_{global} = atan2\left(\frac{\hat{T}_y}{\hat{T}_x}\right)$$

for $\forall s: \theta \in [-180^\circ, 180^\circ]$ thanks to the sampling distance d_s . In this way the specific orientation of the vehicle to achieve in order to realize the *Gompertz* trajectory in a single maneuver is known.

The vehicle orientation is achieved by means the right magnitude of the steering angle (s) determined by considering the dimension of the system and the trajectory curvature k :

$$k(s) = \frac{\|\hat{g}'(s) \times \hat{g}''(s)\|}{\|\hat{g}'(s)\|^3}$$

the necessary steering angle is

$$\gamma(s) = atan\left(\frac{(L(y''(s)x'(s) - x''(s)y'(s)))}{\|\hat{g}'(s)\|^3}\right)$$

for all position in s .

The last step is the simulation of the vehicle motion through the vehicle model expression on the found trajectory considering slip and determine if collisions will occur or not.

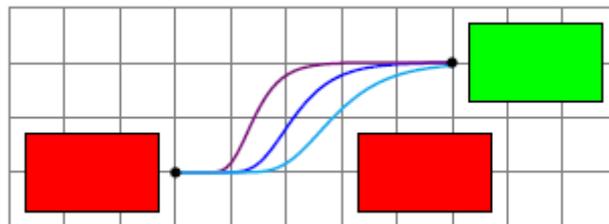


Figure 2.24: Visualization of the optimal scheme result

The entire process is shown and summarized in this figure where there is a simple visualization of the optimal scheme result.

The total procedure can be used together with other constraints in iterative way to determine the best values of b and c varying them.

The first step is defining constraints and the necessary conditions:

$C = 0$, for example, is the condition when collisions are not present and γ belongs to the range $[-30^\circ, 30^\circ]$ is another hard constraint that need to be satisfied in order to have the required steering angles.

The parameter c is the more critical of the two parameters, in fact in the scheme the parameter b is fixed for first to a low value while the first one (c) will be optimized. The desirable result is to obtain the value of c which minimizes the *Gompertz* curve slope and at the same time to not have a steering angle greater than the maximum possible and thus to have collisions.

$$\arg \min_c \hat{g}'(a, b, c) \{C = 0, \gamma \in [-30^\circ, 30^\circ]\}$$

for parameter b the objective is to have the maximum possible value using the minim value of c obtained with the above relation which avoid collisions:

$$\arg \max_b \left(\ln \left(\frac{\hat{g}}{a} \right) e^{-c \min x} \right) \{C = 0\}$$

If for either of the above relation there are not solutions, a single trajectory parallel parking maneuver doesn't exist without collision.

Generally, in this method a lot of different possible trajectories are generated based on test through valid values of b and c . [9]

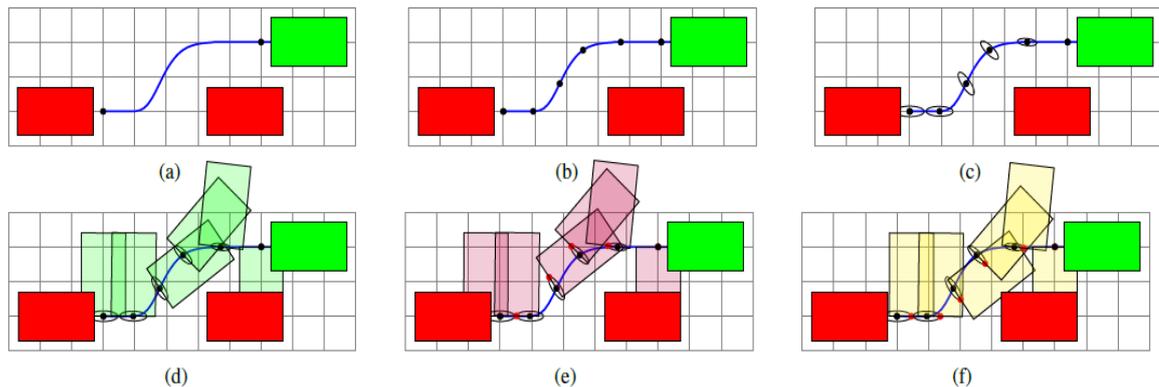


Figure 2.25: Evolution of trajectory based on b and c

CHAPTER 3

LATERAL DYNAMICS PROBLEM

LATERAL DYNAMICS PROBLEM

3. LATERAL DYNAMICS ANALYSYS

In the automotive field one of the major problems which the vehicle faces day by day are those related to the lateral dynamics mostly involved in the parking scenario. These problems are amplified especially for autonomous vehicles.

In general, lane departures are one the main cause of fatal accidents, especially in the United States. Carelessness can be attributed as one of the principal reasons of lane departures accidents.

For such a purpose the automotive industries in the last years develop three type of lateral systems in order to solve the problem of lane departure accidents:

- lane departure warning system (LDWS);
- lane keeping system (LKS);
- yaw stability control system.

This development involves a significant amount of university researchers which conduct study on these types of systems.

3.1 YAW STABILITY CONTROL SYSTEMS

Spinning and drifting out are two phenomena which can be prevent by means vehicle stability control systems developed by lot of automotive manufacturers. Generally, when it comes to stability control systems one refers to yaw control systems or electronic stability control systems.

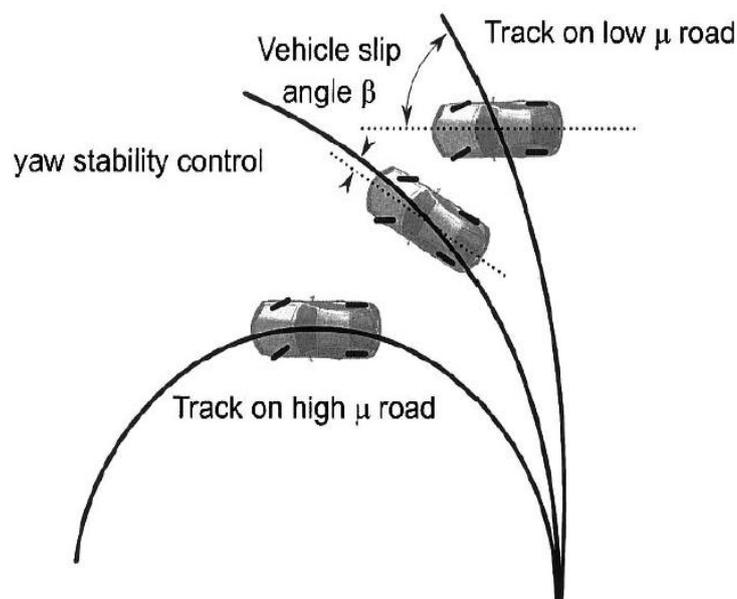


Figure 3.1: The functioning of a yaw control system

The lower curve of the figure shows the desirable trajectory to follow as response of a steering input come from the driver in dry-road condition and high tire-road friction coefficient. In this specific case it is possible to obtain the required lateral force due to the e high friction coefficient which allows to approach the curved road. The upper curve shows as in cases in which the friction coefficient is small, or the vehicle speed is too high, the is uncapable to follow the nominal motion required and it will travel on a larger radius trajectory and so there will be a smaller curvature.

The expected nominal motion requested by the driver can be respected if the yaw control system become able to restore the right yaw velocity of the vehicle. The objective is to obtain the maximum result closer to the expected nominal yaw rate even if once can be in not favorable situation like in low friction coefficient scenario.

In the last ten years many automotive industries carried on several study about yaw control system to develop through vehicle experiment and simulations.

The main types of stability control system which the companies have been proposed and developed to solve the yaw dynamics control problem are three and they are the following ones:

- differential braking systems, which is a system that use the ABS braking system on the vehicle for applying differential braking between the right and left wheels in order to control the yaw moment;
- steer-by-wire systems, which acts on the driver's steering angle input by adding if needed a correction on the angle to the wheels;
- active torque distribution systems, which use an active differentials and all-wheel drive to control in independent way the drive torque distributed to each wheel and then supply an active control of both yam moment and traction.

3.2 KINEMATIC MODEL OF LATERAL VEHICLE MOTION

A mathematical description of the vehicle motion without putting into play the forces which affect the motion can be developed. The equations will be based only on geometric relationships which act on the system.

A bicycle model is the most important used model to describe and approximate the vehicle motion. In this model the four wheels of the vehicle are placed by one single wheel for each wheel couple (rear and front) of the first one at two points, A and B.

The rear steering angle δ_r and the front one δ_f are associated to a system in which we assume that both wheels can be steered with the front only steering and the rear one can be set to

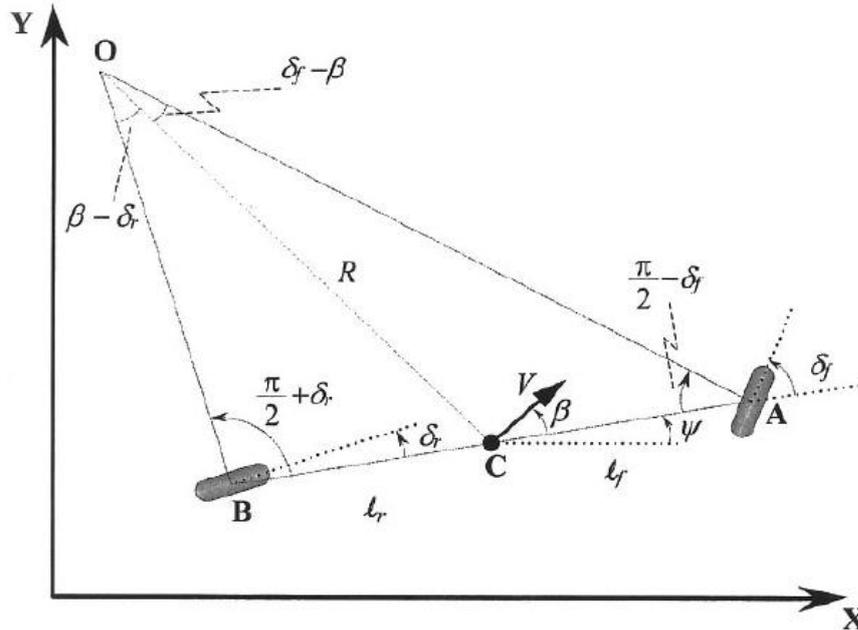


Figure 3.2: Kinematics of lateral vehicle motion

zero. The point C is the centre of gravity of the vehicle and the distance points between A or B and C are l_f and l_r respectively. The vehicle wheelbase is $L = l_f + l_r$.

Under the assumption of planar motion of the vehicle, it is defined a three-coordinate system (X, Y, ψ) to describe its motion. The first two coordinates (X, Y) are inertial coordinates of the centre of gravity location of the vehicle, while ψ is the heading angle of the vehicle and it is associated to the vehicle orientation and V denote the vehicle velocity at the c.g. The angle that makes the vehicle with respect to the longitudinal axes is the slip angle which is defined with the variable β .

The velocity vectors at the points A and B are in the direction of the orientation of both wheels (front and rear) and it is the main assumption on the kinematic model development. The velocity vector at the front wheel produce an angle δ_f with the longitudinal axis, while the one at the rear wheel make an angle δ_r with the longitudinal axes, that corresponds to assume the slip angle equals to zero at both wheels (reasonable assumption for low speed motion of the vehicle). The lateral force generated by the tires is small at low speeds.

Generally, the total lateral force from both tires to drive on any circular R radius road is:

$$\frac{mV^2}{R}$$

which varies in quadratic form with the speed and it becomes obviously small at low speed. When the lateral forces are small it is necessary to assume that the velocity vector is in direction of the wheel at each wheel.

The instantaneous rolling centre for the vehicle is the point O defined as the intersection of AO and BO lines drawn perpendicularly to the two rolling wheels orientation. The length of the line OC connecting the rolling centre O with the centre of gravity C defines the radius R of the vehicle's path and the velocity is perpendicular to the connecting line and the direction. The course angle γ is the sum of the heading angle and the slip angle, $\gamma = \psi + \beta$.

By applying the sine rule to the OCA triangle:

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin\left(\frac{\pi}{2} - \delta_f\right)}{R}$$

And to the OCB triangle:

$$\frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin\left(\frac{\pi}{2} + \delta_r\right)}{R}$$

from the first equation one obtains:

$$\frac{\sin(\delta_f) \cos(\beta) - \sin(\beta) \cos(\delta_f)}{l_f} = \frac{\cos(\delta_f)}{R}$$

and from the second one:

$$\frac{\cos(\delta_r) \sin(\beta) - \cos(\beta) \sin(\delta_r)}{l_r} = \frac{\cos(\delta_r)}{R}$$

By multiplying both sides of the first trigonometric relation by $\frac{l_f}{\cos(\delta_f)}$:

$$\tan(\delta_f) \cos(\beta) - \sin(\beta) = \frac{l_f}{R}$$

and by multiplying both sides of the second trigonometric relation by $\frac{l_r}{\cos(\delta_r)}$:

$$\sin(\beta) - \tan(\delta_r) \cos(\beta) = \frac{l_r}{R}$$

Summing the two last equations one obtains:

$$\tan(\delta_f) - \tan(\delta_r) \cos(\beta) = \frac{l_f + l_r}{R}$$

Under the assumption of slow path changes due to the low speed of the vehicle, the orientation change rate of the vehicle must be equal to the angular velocity of the vehicle $\frac{V}{R}$:

$$\dot{\psi} = \frac{V}{R}$$

Combining the last two passages:

$$\dot{\psi} = \frac{V \cos(\beta)}{l_f + l_r} \tan(\delta_f) - \tan(\delta_r)$$

The final equations of motion are:

$$\dot{X} = V \cos(\psi + \beta)$$

$$\dot{Y} = V \sin(\psi + \beta)$$

$$\dot{\psi} = \frac{V \cos(\beta)}{l_f + l_r} \tan(\delta_f) - \tan(\delta_r)$$

where δ_f , δ_r and V are the three inputs. The velocity V is an external variable which can be obtained from a longitudinal vehicle model or assumed to be a variable that changes in time.

From the previous equations it is possible to estimate the slip angle β :

$$\beta = \tan^{-1}\left(\frac{l_f \tan \delta_r + l_r \tan \delta_f}{l_f + l_r}\right)$$

The radius of each path of the two wheels travels is different despite the left and right steering angles are in general approximately equal.

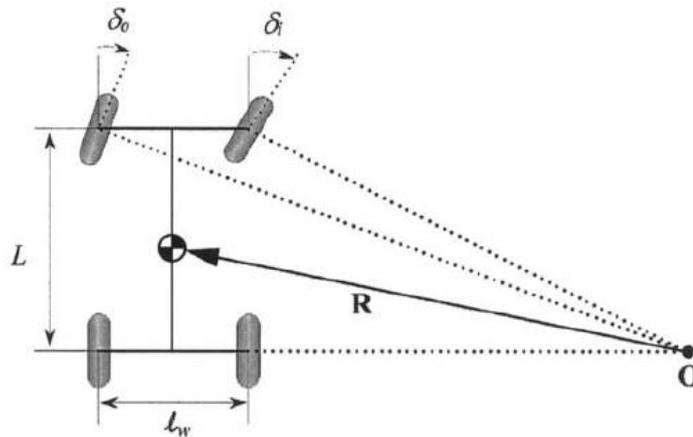


Figure 3.3:Ackerman turning geometry

Assume that l_w is the vehicle track width, δ_o and δ_i the outer and inner steering angles and $L = l_f + l_r$ the wheelbase which for assumption small with respect to the radius R , if the slip angle β is small the $\dot{\psi}$ equation can be approximated by:

$$\frac{\dot{\psi}}{V} = \frac{1}{R} = \frac{\delta}{L}$$

or

$$\delta = \frac{L}{R}$$

It is needed to be considered that the inner and outer wheels radius are different, so one has:

$$\delta_o = \frac{L}{R + \frac{l_w}{2}}$$

$$\delta_i = \frac{L}{R - \frac{l_w}{2}}$$

Approximatively, the average front wheel steering angle is:

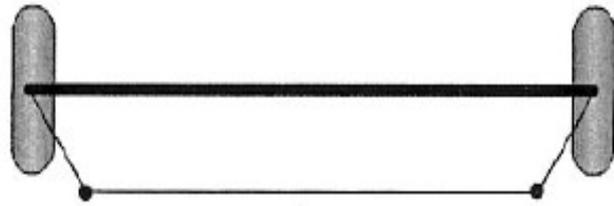
$$\delta = \frac{\delta_o + \delta_i}{2} \cong \frac{L}{R}$$

while the difference between the inner and the outer steering angle of the front wheels is:

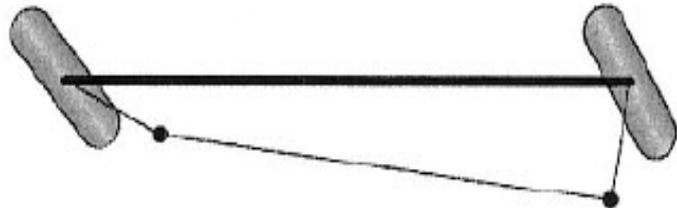
$$\delta_i - \delta_o = \frac{L}{R^2} l_w = \delta^2 \frac{l_w}{L}$$

where the difference is strictly related to the square of the average steering angle.

Trapezoidal geometry



Left turn



Right turn

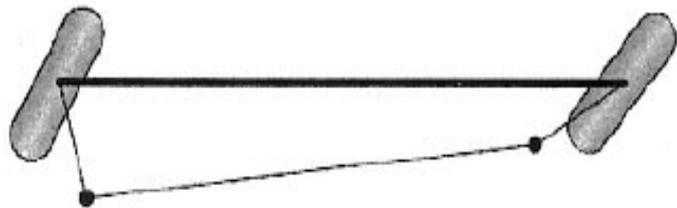


Figure 3.4: Differential steer from a trapezoidal tie-rod arrangement

The inner wheel always produces a large steering angle than the outer one.

LATERAL VEHICLE DYNAMICS BY MEANS BICYCLE MODEL

The assumption related to the fact that the velocity at each wheel is in the same direction of the wheel is no longer valid at higher speeds. For this reason, a dynamic model must be defined instead of a kinematic model considering a bicycle model of the vehicle which has two degrees of freedom represented with the vehicle lateral position and yaw angle (y, ψ):

- y is measured along the lateral axis of the vehicle to the vehicle center of rotation O
- ψ is measured with respect to the global X axis.

The longitudinal velocity of the vehicle center of gravity is defined as V_x .

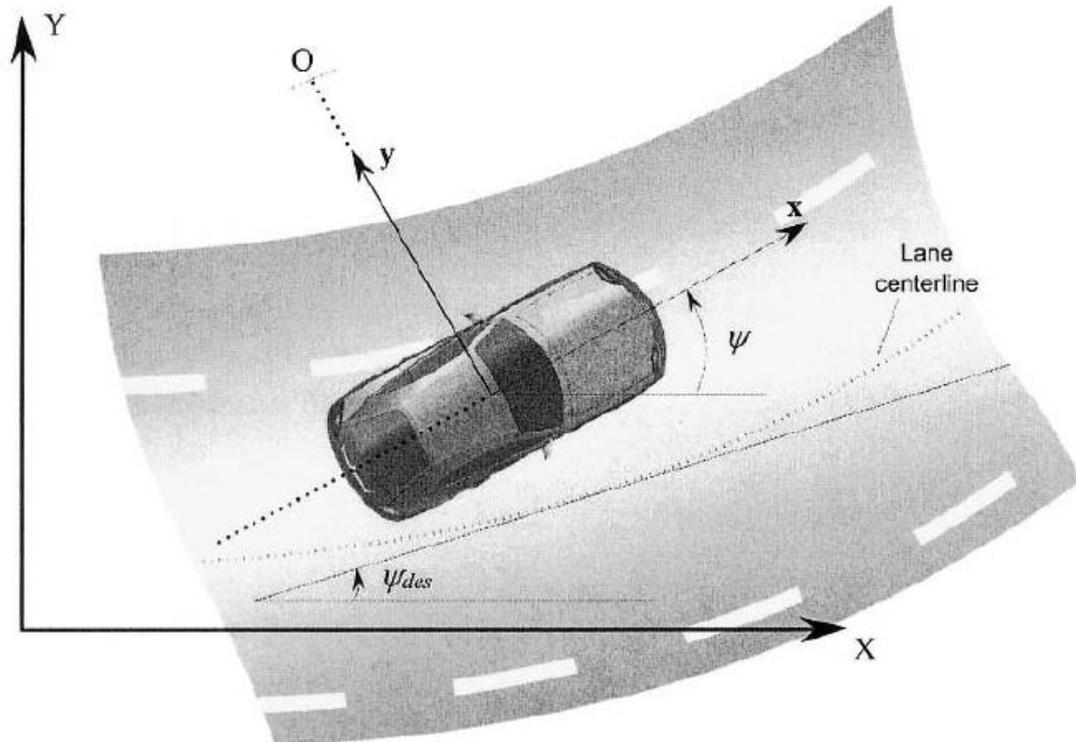


Figure 3.5: Lateral vehicle dynamics

By applying the Newton's second law for motion along the y axis and ignoring the road bank angle one has:

$$ma_y = F_{yf} + F_{yr}$$

in which:

- F_{yf} and F_{yr} are the front and rear lateral tire forces;
- $a_y = \left(\frac{d^2y}{dt^2}\right)_{inertial}$ is the inertial acceleration at the vehicle center of gravity in the y direction obtained thanks two contributes:
 - \ddot{y} acceleration related to the motion along the y axis;
 - $V_x\dot{\psi}$ centripetal acceleration.

therefore:

$$a_y = \ddot{y} + V_x\dot{\psi}$$

By using the lateral translation motion of the vehicle equation (Newton's second law) inside this previous relation, one obtains:

$$m(\ddot{y} + V_x\dot{\psi}) = F_{yf} + F_{yr} \quad (1)$$

the moment balance about the z axis produces the equation for the yaw dynamics:

$$I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \quad (2)$$

The next step is dedicated for modelling the lateral tire forces F_{yf} and F_{yr} that act on the vehicle, which are proportional to the slip angle when it is small. The slip angle of a tire is the angle between the tire orientation and the velocity vector of the wheel orientation:

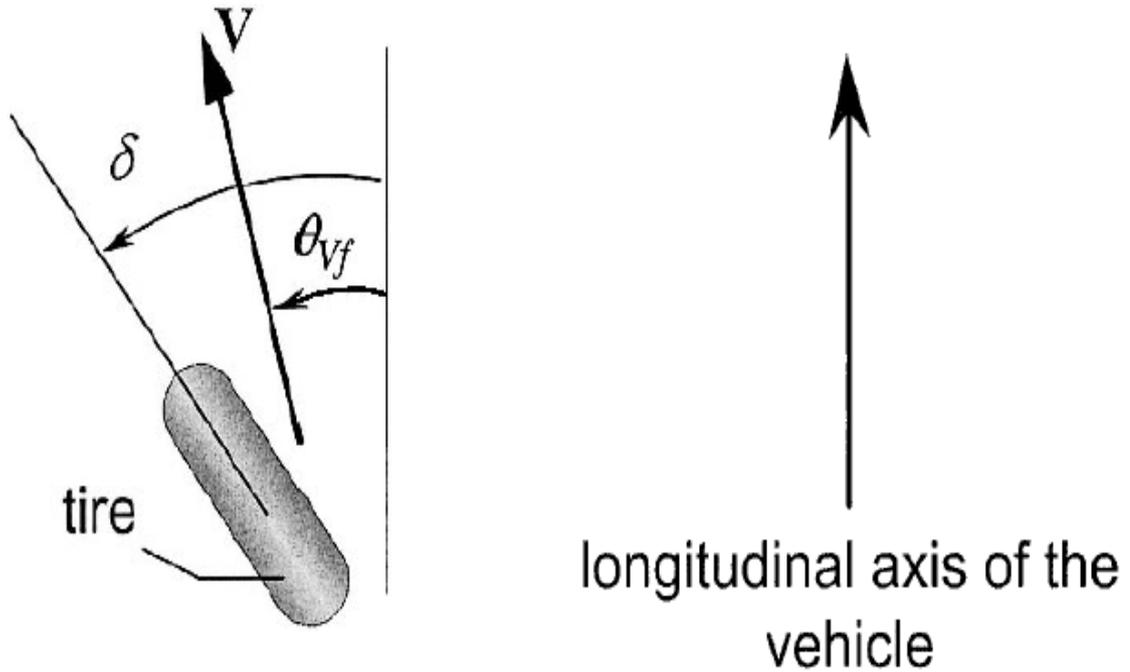


Figure 3.6: Tire slip angle

The front slip angle is:

$$\alpha_f = \delta - \theta_{vf}$$

where δ is the front wheel steering angle, while θ_{vf} is the angle between the vector velocity direction and the longitudinal axis of the vehicle, the front tire velocity angle.

Similarly, the rear slip angle is:

$$\alpha_r = -\theta_{vr}$$

where θ_{vr} is the rear tire velocity angle.

The slip angle results zero when the vehicle travels straight ahead and is not steering and so, the two components of the slip angle are both zero.

In the static region of the contact zone, there is a contact of the tip of each tread with the ground and this remains stationary. So, the top of the tread moving with respect to the tread tip causes a tread deformation. If one denotes the velocity at the wheel as V_w , the lateral component of the velocity is $V_w \sin(\alpha)$. The magnitude of the tread lateral deflection is proportional both to the lateral velocity and to the amount of time spent by the tread in the contact zone. Finally, the lateral tread deflection is proportional only to the slip angle.

The lateral tire force on the tire is strictly related and proportional to the magnitude of lateral deflection of the treads in the contact zone, while for small slip angles it is proportional to the slip angle.

Given these considerations, it is possible to define the lateral tire force for the front wheels of the vehicle:

$$F_{yf} = C_{\alpha}(\delta - \theta_{vf})$$

in which C_{α} is denoted as cornering stiffness and it is a proportionality constant.

Similarly, for the rear wheels of the vehicle it is definable the lateral tire force:

$$F_{yr} = C_{\alpha}(-\theta_{vr})$$

where also in this case C_{α} is the cornering stiffness.

The lateral and longitudinal velocity ratio at each wheel is useful to calculate the velocity angle at that wheel by exploiting these two relations to determinate θ_{vf} and θ_{vr} :

$$\tan(\theta_{vf}) = \frac{V_y + l_f \dot{\psi}}{V_x}$$

$$\tan(\theta_{vr}) = \frac{V_y - l_r \dot{\psi}}{V_x}$$

with V_y and V_x lateral and longitudinal velocity at the center of gravity of the vehicle, $\dot{\psi}$ yaw rate of the vehicle and l_r and l_f longitudinal distances from center of gravity to the rear and front wheels respectively.

Under the assumption of small angle:

$$\theta_{vf} = \frac{V_y + l_f \dot{\psi}}{V_x}$$

$$\theta_{vr} = \frac{V_y - l_r \dot{\psi}}{V_x}$$

Thus,

$$F_{yf} = C_{\alpha} \left(\delta - \frac{V_y + l_f \dot{\psi}}{V_x} \right)$$

$$F_{yr} = C_{\alpha} \left(-\frac{V_y - l_r \dot{\psi}}{V_x} \right)$$

Definitely, starting from the (1) and (2) relations

$$\begin{cases} m(\dot{y} + \dot{\psi}V_x) = F_{yf} + F_{yr} \\ I_z \ddot{\psi} = l_f F_{yf} - l_r F_{yr} \end{cases}$$

Denoting \dot{y} as \dot{V}_y , using the tire forces relations and the inverse relation of the front and rear tire velocity angle:

$$F_{yf} = C_\alpha(\delta - \theta_{vf})$$

$$F_{yr} = C_\alpha(-\theta_{vr})$$

$$\theta_{vf} = \tan^{-1}\left(\frac{V_y - l_r\dot{\psi}}{V_x}\right)$$

$$\theta_{vr} = \tan^{-1}\left(\frac{V_y - l_r\dot{\psi}}{V_x}\right)$$

one obtains:

$$\begin{cases} m(\dot{V}_y + \dot{\psi}V_x) = F_{yf} + F_{yr} \\ I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \end{cases}$$

that is:

$$\begin{cases} \dot{V}_y = \frac{1}{m}(-V_x\dot{\psi} + F_{yf} + F_{yr}) \\ \ddot{\psi} = \frac{1}{I_z}(l_f F_{yf} - l_r F_{yr}) \end{cases}$$

which is at last:

$$\begin{cases} \dot{V}_y = \frac{1}{m}\left(-V_x\dot{\psi} + C_\alpha\left[\delta - \tan^{-1}\left(\frac{V_y + l_f\dot{\psi}}{V_x}\right)\right] - C_\alpha\left[\tan^{-1}\left(\frac{V_y - l_r\dot{\psi}}{V_x}\right)\right]\right) \\ \ddot{\psi} = \frac{1}{I_z}\left(l_f C_\alpha\left[\delta - \tan^{-1}\left(\frac{V_y + l_f\dot{\psi}}{V_x}\right)\right] - l_r C_\alpha\left[\tan^{-1}\left(\frac{V_y - l_r\dot{\psi}}{V_x}\right)\right]\right) \end{cases}$$

the final nonlinear vehicle model.

The main idea when there is an approach to a nonlinear model is that to “convert” it in a correspondent linearize one.

In this regard, in the next chapter there is an explanation on how a nonlinear system can be traduced in a linear one. [10]

CHAPTER 4

LINEARIZATION OF NONLINEAR SYSTEMS

LINEARIZATION OF NONLINEAR SYSTEMS

4. FEEDBACK LINEARIZATION

In recent years many researchers have been attracted by feedback linearization which is an approach to nonlinear control design.

The basic idea of the approach is to transform algebraically a nonlinear system dynamic into a linear (complete or partial) so that linear control methods can be applied. This type of linearization differs from traditional linearization, as feedback linearization is achieved through exact state transformations and feedbacks rather than linear approximations of dynamics. The idea of simplifying the form of the system dynamics by choosing a different representation of state is not entirely unusual. In mechanics, for example, it is known that the complexity and the form of a system model depend substantially on the choice of coordinate systems or reference frames. Feedback linearization techniques can be assimilated as method to obtain equivalent model of simpler form of an original system model. Some practical problem such as biomedical devices, industrial robots, high performance aircraft and control of helicopters can be successfully solved by Feedback linearization, even if it is still in developing phase in industry for more applications. Nevertheless, this approach has a number of important deficiencies and limitations which are aim of study of current research.

4.1 CANONICAL FORM OF FEEDBACK LINEARIZATION

Feedback linearization set itself the goal to cancel the nonlinearities in a nonlinear system imposing a desired linear dynamic in order to have the closed-loop dynamics is in a linear form. This idea can be simply applied to a class of nonlinear systems described by the so-called *companion form*, or *controllability canonical form*. A system is said to be in companion form if its dynamics is represented by

$$\dot{x}^{(n)} = f(x) + b(x)u$$

where $f(x)$ and $b(x)$ are nonlinear functions of the states, x is the scalar output of interest, $x = [x, \dot{x}, \dots, \dot{x}^{(n-1)}]^T$ and u is the scalar control input, This form is unique because the derivative of the input u is not present, although derivatives of x appear in this equation, no derivative of the input u is present.

The state-space representation of the equation can be written in this way:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \dots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \dots \\ x_n \\ f(x) + b(x)u \end{bmatrix}$$

Using the control input (assuming b to be non-zero):

$$u \approx \frac{1}{b} [v - f]$$

for all those systems which can be expressed in the controllability canonical form, it is possible to cancel the nonlinearities obtaining the simple input-output relation:

$$x^{(n)} = v$$

Then, the control law:

$$v = -k_0 - k_1\dot{x} - \dots - k_{n-1}x^{(n-1)}$$

with the k_i chosen so that the polynomial $p^n + k_{n-1}p^{n-1} + \dots + k_0$ has all its roots strictly in the left-half complex plane, leads to the exponentially stable dynamics which implies that $x(t) \rightarrow 0$. For tasks involving the tracking of a desired output $x_d(t)$, the control law:

$$v = x_d^{(n)} - k_0e - k_1\dot{e} - \dots - k_{n-1}e^{(n-1)}$$

(where the tracking error is $e(t) = x(t) - x_d(t)$ is the tracking error) leads to exponentially convergent tracking.

When the system has a nonlinear dynamic, which is not in a controllability canonical form, it is important to use algebraic transformations to have the dynamics into the controllability form before using the feedback linearization design or to exploit partial linearization of the original system dynamics, instead of full linearization.

4.2 INPUT-STATE LINEARIZATION

Considering the problem of control input design for a single-input u nonlinear system of the form:

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

this problem can be solved in two steps thanks to the input-state linearization technique:

1. finds a state transformation $z = z(x)$ and an input transformation $u = u(x, v)$ in order to transform the nonlinear system dynamics into an equivalent linear time-invariant dynamic in the form $\dot{z} = Az + bv$;
2. design v using standard linear techniques.

The closed-loop system under the control law obtained is represented in this block diagram:

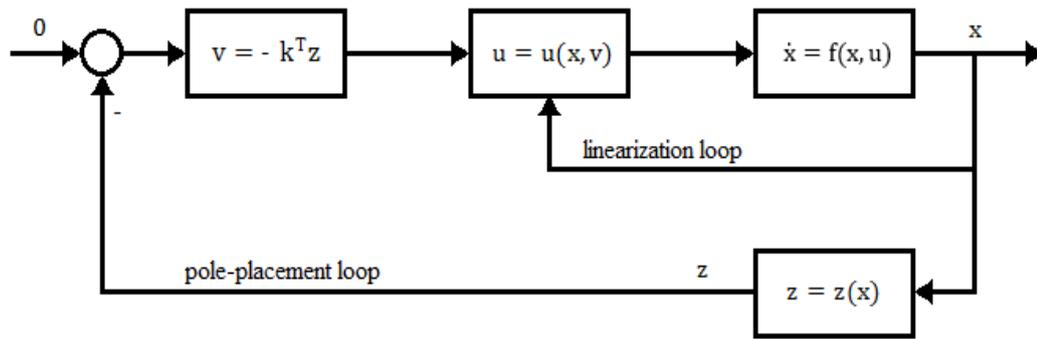


Figure 4.1: Closed-loop system

In this control loop two loops with two different functions are detected:

- the inner loop for the linearization of the input-state relation;
- the outer loop for the stabilization of the closed-loop dynamics.

4.3 INPUT-OUTPUT LINEARIZATION

For this type of linearization, it is possible consider the tracking control problem and in particular the system:

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

$$y = h(x)$$

and with the assumption that the objective is to make sure that the output $y(t)$ track a desired trajectory $y_d(t)$ while keeping the whole state bounded, where $y_d(t)$ its time derivatives until a sufficiently high order are assumed to be known and bounded. One difficulty associated with this model is that the output y is only indirectly related to the input u , by the state variable x and the nonlinear state equations.

Thus, the design of the input u for the control of the output y start to be complicated even if the difficulty can be reduced to the research of a simple and explicit relationship between the control input u and the system output y . To generate a direct relation between the output y and the input u , it is necessary to differentiate the output until the relation is found. If we require to differentiate r times the output of the system, it is said that the systems have *relative degree* r (in linear system coincides with the excess of poles over zeros). It is possible take at most n differentiations of the output y to make appear the input u , $r \leq n$. If the n differentiations are exceeded, and the control input never appeared, the system would be of order higher than n and not controllable.

In the input-output linearization a part of the system dynamics called *internal dynamics* seems “unobservable”, because it cannot be seen from the external input-output relation. If the internal dynamics is stable, the tracking control design problem can be solved.

Moreover, the input-output linearization can be applied also on the stabilization problem as well as the output tracking.

In stabilization problems:

1. there is no reason to limit the choice of the output $y = h(x)$ to be a physically important quantity (in tracking problems the physical task determines the choice of output). To generate a linear relation input-output it is possible exploit an artificial output using any function of x .
2. different choices of output function allow to obtain different internal dynamics. One choice of output can lead to a stable internal dynamic (or no internal dynamics), while another one leads to having an unstable one. The best choice is the output function which guarantees an associated stable internal dynamic.

A particular case is when the order of the system is the same as the system relative degree r , that is when n differentiations are applied on the output y to obtain a linear input-output relation. In this case, the variables $y, \dot{y}, \dots, \dot{y}^{(n-1)}$ becomes the new state variables for the system and thus, the consequences are that there is no internal dynamics associated with this input-output linearization that in particular leads to input-state linearization, and both state regulation and output tracking (for the particular output) can be achieved easily.

4.4 THE INTERNAL DYNAMICS OF LINEAR SYSTEM

In general, to direct determine the stability of the internal dynamics is very difficult because it is nonlinear, non-autonomous and coupled to the “external” closed-loop dynamics. The objective for the internal dynamics is finding the simpler ways for determining its stability. By evaluating two apparently similar linear systems, but with an associated transfer function that differs exclusively for zeros (equal poles), it is possible to notice how a tracking design method is applicable only in one of the two cases. In particular, the design is successful in the case of the transfer function with left half-plane zero, but fails in the case of the function with right half-plane zero. From here it is inferred that the internal dynamics is stable if the plant is "minimum-phase", that is in the left half-plane and this is applicable to all linear systems. As for the "non-minimum phase" systems, a perfect tracking of arbitrary trajectories requires infinite control effort.

For summarize, if the systems are phase minimum, then with the zeros in the left-half plane, this will imply that the internal dynamics are stable regardless of the magnitudes of the desired $y, \dots, y^{(r)}$ (with r relative degree) and of the initial conditions.

4.5 THE ZERO-DYNAMICS

One important aspect to be considered is how to determine the stability of the internal dynamics in the case of nonlinear system. Since for linear system this stability depends on the locations of the zeros, it is necessary as first requirement to extend the notion of zeros in order to understand this concept in the nonlinear case. For nonlinear system, transfer function cannot be defined. In particular, zeros are intrinsic properties of linear plant systems, while the stability of the internal dynamics for nonlinear systems depends on the specific control input. For nonlinear systems a way to treat these difficulties is to define a so called *zero-dynamics* that are the internal dynamics of the system when the input keeps the systems output at zero. The zero-dynamics become an intrinsic feature for nonlinear systems, which is independent from the desired trajectories or the chosen control law. The result is general for linear systems, in particular the global asymptotic stability of the zero-dynamics is guaranteed when the systems has all zeros in the left-half plane and implies the global stability of the internal dynamics.

In case of nonlinear system, in stabilization problem, the local stability of the zero-dynamics guarantees the local stability of the internal dynamics, while in tracking problem this relation for the global stability doesn't exist.

Definitely, to examine the local stability of the zero-dynamics may be much easier and simpler than evaluating the internal dynamics stability, because only the internal states are relating to the zero-dynamics, while the internal dynamics is linked in some way to the desired trajectories and the external dynamics.

The control design method based on input-output linearization can be summarized in three steps:

1. differentiate the output y until the input appears;
2. choose the input u in order to guarantee the tracking convergence, but most of all to cancel the nonlinearities;
3. stability study of the internal dynamics.

For what concerns the topology the input-output linearization, it is possible to outline in the following way:

- if $r = n$, then the nonlinear system is fully linearized, and the procedure allows to have a satisfactory controller
- if $r < n$, then the nonlinear systems are partly linearized, and the controller can be applied based on the stability of the internal dynamics.

4.6 INPUT-STATE LINEARIZATION OF SISO SYSTEMS

The input-state linearization for single-input nonlinear systems is represented by the state equations:

$$\dot{x} = f(x) + g(x)u$$

with f and g smooth vector fields on \mathbb{R}^n . Systems in this form are said to be *affine* or *linear in control*. In particular, if a nonlinear system is in the form:

$$\dot{x} = f(x) + g(x)w[u + \varphi(x)]$$

with w invertible scalar function and φ arbitrary functional.

The substitution $v = w[u + \varphi(x)]$ leads the dynamics into the first form. Thus, it is possible design a control law for v and compute u by inverting w obtaining $u = w^{-1}(v) - \varphi(x)$.

4.6.1 INPUT-STATE LINEARIZATION DEFINITION

A single-input nonlinear system in the form $\dot{x} = f(x) + g(x)u$ is *input-output linearizable* if exist a region Ω in \mathbb{R}^n , a diffeomorphism $\varphi: \Omega \rightarrow \mathbb{R}^n$ and a nonlinear feedback control law of this type:

$$u = \alpha(x) + \beta(x)v \quad \text{input transformation}$$

in order to have the new state variables $z = \varphi(x)$ (or $z = z(x)$ state transformation) called *linearizing state* and the new input v satisfy a linear invariant relation called *linearizing control law*:

$$\dot{z} = Az + bv \quad (3.2)$$

where:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

In particular, a diffeomorphism φ is a generalization of the coordinate transformation concept and its formal definition said that φ it's a smooth function having a smooth inverse φ^{-1} with $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and defined in a region Ω . $\varphi(x)$ denoted also as $z(x)$ is a global diffeomorphism if the region Ω is the whole space \mathbb{R}^n (general situations deal with local diffeomorphism). A matrix and b vector correspond to a linear companion form.

From (3.2) canonical form it is highlighted that feedback linearization is a special subcase of the input-output linearization, in which the output leads to a relative degree n . This imply that a system must be input-state linearizable if it is input-output linearizable with relative

degree n . Nevertheless, there is a relationship between the input-state linearization and input-output linearization according to which a nonlinear system is input-state linearizable if, and only if, the first new state $z_1(x)$ which represents the output in the case of input-state linearization.

Furthermore, a nonlinear system with $f(x)$ and $g(x)$ smooth vector fields, is input-state linearizable if, and only if, a region Ω exists such that the following conditions are respected:

- $\{g, ad_f g, \dots, ad_{f^{n-1}} g\}$ vector fields are linearly independent in Ω ;
- $\{g, ad_f g, \dots, ad_{f^{n-2}} g\}$ set is involutive in Ω .

The first condition is almost a reinterpretation of a simply controllability condition for nonlinear system (for linear systems the vector fields are $\{b, Ab, \dots, A^{n-1}b\}$ and the independence corresponds to the invertibility of the controllability matrix). The second condition is not generally satisfied and in particular, a linearly independent set of vector fields is said to be *involutive* if, and only if, there are a scalar function $\alpha_{ijk}: \mathbb{R}^n \rightarrow \mathbb{R}$ such that:

$$[f_i, f_j] = \sum_{k=1}^n \alpha_{ijk}(x) f_k(x) \quad \forall i, j$$

A condition of integrability derives from that of involutivity (necessity and sufficiency).

Obviously, it is needed to define the *adjoint* that is a vector fields also known as *Lie bracket*. Particularly, starting from two vector fields f and g on \mathbb{R}^n , the Lie bracket of f and g will be the third vector field determined by:

$$[f, g] = \nabla g f - \nabla f g$$

commonly written as $ad_f g$ (where *ad* stands for “adjoint”). By repeating Lie brackets it is defined recursively cases of this type:

$$ad_{f^0} g = g$$

$$ad_{f^i} g = [f, ad_{f^{i-1}} g] \quad \text{for } i = 1, 2, \dots$$

Lie brackets benefit of the following proprieties:

1. bilinearity:

$$[\alpha_1 f_1 + \alpha_2 f_2, g] = \alpha_1 [f_1, g] + \alpha_2 [f_2, g]$$

$$[f_1, \alpha_1 g_1 + \alpha_2 g_2] = \alpha_1 [f_1, g_1] + \alpha_2 [f_1, g_2]$$

where f, f_1, f_2, g, g_1, g_2 are vector fields, while α_1 and α_2 are constant scalars.

2. skew-commutativity:

$$[f, g] = -[g, f]$$

3. Jacobi Identity:

$$L_{ad_f g} h = L_f L_g h - L_g L_f h$$

where $h(x)$ is a smooth scalar function of x and L_f and the others are the Lie derivatives explained in the next paragraph.

4.7.1.1 HOW TO PERFORM INPUT-STATE LINEARIZATION

Input-state linearization of a nonlinear system is divided in four following steps:

- construction of the vector fields $g, ad_f g, \dots, ad_{f^{n-1}} g$ for the given system;
- controllability and involutivity conditions verification;
- if both conditions are satisfied, it is needed to find the new output function z_1 related to the input-output linearization of relative degree n :

$$\nabla_{z_1} ad_{f^i} g = 0 \quad i = 0, \dots, n - 2$$

$$\nabla_{z_1} ad_{f^{n-1}} g \neq 0$$

- state transformation $z(x) = [z_1 \ L_f z_1 \ \dots \ L_{f^{n-1}} z_1]^T$ and input transformation computation:

$$\alpha(x) = -\frac{L_{f^n} z_1}{L_g L_{f^{n-1}} z_1}$$

$$\beta(x) = \frac{1}{L_g L_{f^{n-1}} z_1}$$

With the transformed state equation into a linear form, it is possible design controllers for either tracking problems or stabilization purposes.

The scalar function $L_f h$ called *Lie derivative* (or simply, the derivative) of a generic scalar function $h(x): \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to a vector field $f(x): \mathbb{R}^n \rightarrow \mathbb{R}^n$. The Lie derivative of h with respect to f is a scalar function defined by $L_f h = \nabla h f$.

Therefore, the Lie derivative $L_f h$ is the directional derivative of h along the vector f direction, which has this property:

$$L_{f^0} h = h \qquad L_{f^i} h = L_f(L_{f^{i-1}} h) = \nabla(L_{f^{i-1}} h) f \quad \text{for } i = 1, 2, \dots$$

$$L_g L_f h = \nabla(L_f h) g$$

[11]

CHAPTER 5
SYSTEM CONTROL DESIGN

SYSTEM CONTROL DESIGN

5. TIME-STATE CONTROL FORM

In the last decades the automatic control of vehicles has been debated and studied intensively, in order to reach high efficiency as well as advanced safety. An example of treated argument is when front steering vehicles run at low speed, with a negligible side slip angles of tires and thus regarded as a kind of nonholonomic systems. In general, these systems which cannot be stabilized by continuous control and therefore requires either discontinuous input or discontinuous coordinates transformation in order to use linear feedback control technique. There are several control methods for nonholonomic systems, but time-state control form (TSCF) suggested by Sampei et.al. is known as a helpful method for that of nonholonomic systems including front steering vehicles.

This particular technique allows the transformation of the nonlinear dynamics described as a differential equations with respect to the time of a system into two linear subsystems, in which one is a differential equation of a special state, the time-state, which increases or decreases monotonously as the time grows and the other one is a differential equation (a vector first order system) with respect that time-state, instead of time. The method plans to start from an original system not linearizable and then transforming it in a subsystem that becomes linearizable and suitable for linear design methods without including explicitly constraint on steering angle.

Another demonstrated method directly applied to nonlinear systems dynamic is the Model Predictive Control (MPC) which is able to provide an optimal control input satisfying various types of constraints. The principal difficulty linked to the MPC is that requires high computational load, even if in the last decade significant progress in the algorithms alleviates the computation load. MPC has been applied to various problems for the vehicle control, including obstacle avoidance control, path-tracking control, optimal velocity control. In addition, the computational load may still large for real time control in some kind of mechanical systems which requires fast response. To face these issues, for front steering vehicles, the TSCF allow to divide the original system into reduced order two subsystems successively linearized using coordinates and input transformation. One of the most difficult driving tasks in narrow road in urban area is determine where and when to conduct the switchback motion and parking control which requires capable driving technique since the dynamics is divide into the forward and backward motion. There is a certain number of researches on parking control of vehicles with nonholonomic constraints.

In particular, for front steering vehicles, there are researches which deal with parallel parking control and garage parking control.

This figure shows a front steering vehicle model represented as a single-track model in which

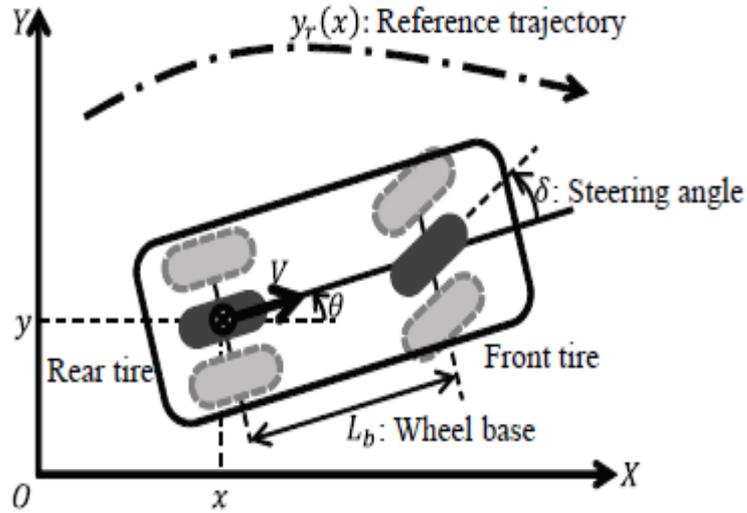


Figure 5.1: Single track vehicle model

both left and right wheels are combined together by neglecting the tire force difference of two wheels. Since the considered scenario is that at low driving speed for parking control, an assumption on the side-slip angles of the front and rear wheels has been made and in particular they are assumed equal to zero. By means the triad (x, y, θ) is possible to represent the rear wheel position and orientation (pose) of the vehicle on $X - Y$ coordinates as (x, y, θ) . V is the velocity at the point of rear wheel, δ is the steering angle of the front wheel and L_b represents the wheelbase. The point of rear wheel of the vehicle tracks a reference path y_r . The kinematic model of the vehicle is described by the following equations (1)(2)(3):

$$\begin{cases} \frac{dx}{dt} = V \cos(\theta) & (1) \\ \frac{dy}{dt} = V \sin(\theta) & (2) \\ \frac{d\theta}{dt} = \frac{V}{L_b} \tan(\delta) & (3) \end{cases}$$

Time-state control form is used to realize the path-tracking control of the vehicle, in which the dynamics is represented as a differential equation with respect to the state instead of time. Assuming that the reference path of the vehicle lies along the x-axis, x is taken as *time-state* and dividing (2) and (3) by (1), we get the following time-state control form:

$$\begin{cases} \frac{dy}{dx} = \tan(\theta) & (4) \\ \frac{d\theta}{dx} = \frac{1}{L_b \cos(\theta)} \tan(\delta) & (5) \end{cases}$$

The next steps are the nonlinear state transformation (6) and input transformation (7) and (8):

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x \\ dy/dx \\ y \end{bmatrix} = \begin{bmatrix} x \\ \tan(\theta) \\ y \end{bmatrix} \quad (6)$$

$$\mu_1 = V \cos(\theta) \quad (7)$$

$$\mu_2 = \frac{1}{L_b \cos^3(\theta)} \quad (8)$$

The transformation is defined in the range $-\pi/2 < \theta < \pi/2$. Starting from (6)(7)(8), nonlinear state space equation (1)(2)(3) is transformed into the two following linear subsystems:

$$\frac{dz_1}{dt} = \mu_1 \quad (9)$$

$$\frac{d}{dz_1} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu_2 \quad (10)$$

where (9) equation is the differential equation of the time-state z_1 with respect to the time t , and (10) equation is the differential equation of other states z_2 and z_3 with respect to the time-state z_1 . Using θ and μ_2 is possible to compute the actual steering δ by means the following equation transformed from (8):

$$\delta = \tan^{-1}(L_b \cos^3(\theta) \mu_2)$$

Since the linearized time-state control form (10) is a double integrator system with respect to the time-state x , it is straightforward to realize the path-tracking control if the desired path y_r is described as a function of the time-state. In fact, if we apply

$$\mu_2 = \frac{d^2 y_r}{dz_1^2} - k_2 \left(z_2 - \frac{dy_r}{dz_1} \right) - k_1 (z_3 - y_r)$$

with positive k_1 and k_2 , then $z_3 \rightarrow y_r$ as $z_1 \rightarrow \infty$, which is achieved for positive μ_1 in (9). For $z_1 \rightarrow -\infty$, $k_1 > 0$ and $k_2 < 0$ is used to ensure stability. [12]

5.1 MATHEMATICAL ANALYSIS AND SIMULINK TSC DESIGN

Starting from these two relations:

$$\frac{d}{dz_1} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu_2$$

$$\mu_2 = \frac{d^2 y_r}{dz_1^2} - k_2 \left(z_2 - \frac{dy_r}{dz_1} \right) - k_1 (z_3 - y_r)$$

it is possible to determinate the values of k_1 and k_2 to guarantee the stability of the system.

For this reason, the Routh criterion will be used after a substitution of μ_2 in the first relation:

$$\begin{aligned} \frac{d}{dz_1} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left(\frac{d^2 y_r}{dz_1^2} - k_2 \left(z_2 - \frac{dy_r}{dz_1} \right) - k_1 (z_3 - y_r) \right) = \\ &= \begin{bmatrix} z_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{d^2 y_r}{dz_1^2} - k_2 z_2 + k_2 \frac{dy_r}{dz_1} - k_1 z_3 + k_1 y_r \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{d^2 y_r}{dz_1^2} - k_2 z_2 + k_2 \frac{dy_r}{dz_1} - k_1 z_3 + k_1 y_r \end{bmatrix} = \\ &= \begin{bmatrix} z_2 \\ -k_2 z_2 - k_1 z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{d^2 y_r}{dz_1^2} + k_2 \frac{dy_r}{dz_1} + k_1 y_r \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} \begin{bmatrix} z_3 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left(\frac{d^2 y_r}{dz_1^2} + k_2 \frac{dy_r}{dz_1} + k_1 y_r \right) \end{aligned}$$

The first step is to calculate the characteristic polynomial:

$$P_K^{(A)} = |A - \lambda I| = \left| \begin{pmatrix} 0 & 1 \\ -k_1 & -k_2 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right| = \begin{vmatrix} \lambda & 1 \\ -k_1 & -k_2 - \lambda \end{vmatrix} = \lambda^2 + k_2 \lambda + k_1$$

and so, one obtains:

$$\Delta = k_2^2 - 4k_1$$

Thus, the roots of the polynomial are:

$$\lambda_{1,2} = \frac{-k_2 \pm \sqrt{k_2^2 - 4k_1}}{2}$$

which are the eigenvalues of the matrix.

By applying the Routh criterion to the characteristic polynomial, one arrives to find the stability values:

$$\begin{aligned} \lambda^2 + k_2 \lambda + k_1 &= 0 \\ \begin{vmatrix} 1 & k_1 \\ k_2 & 0 \end{vmatrix} &\rightarrow \frac{\begin{vmatrix} 1 & k_1 \\ k_2 & 0 \end{vmatrix}}{-k_2} = \frac{-k_2 k_1}{-k_2} = k_1 \\ &\begin{vmatrix} 1 & k_1 \\ k_2 & 0 \\ k_1 & \end{vmatrix} \end{aligned}$$

from this table, it is needed to evaluate:

- sign permanence \rightarrow roots with negative real part
- sign change \rightarrow roots with positive real part

thus:

$$+ \begin{vmatrix} 1 & k_1 \\ k_2 & 0 \\ k_1 & \end{vmatrix}$$

that is, $k_1 > 0$ and $k_2 > 0$.

These conditions allow to have a control system which guarantees the overall stability.

For this purpose, $k_1 = 1$ and $k_2 = 0.1$ are chosen.

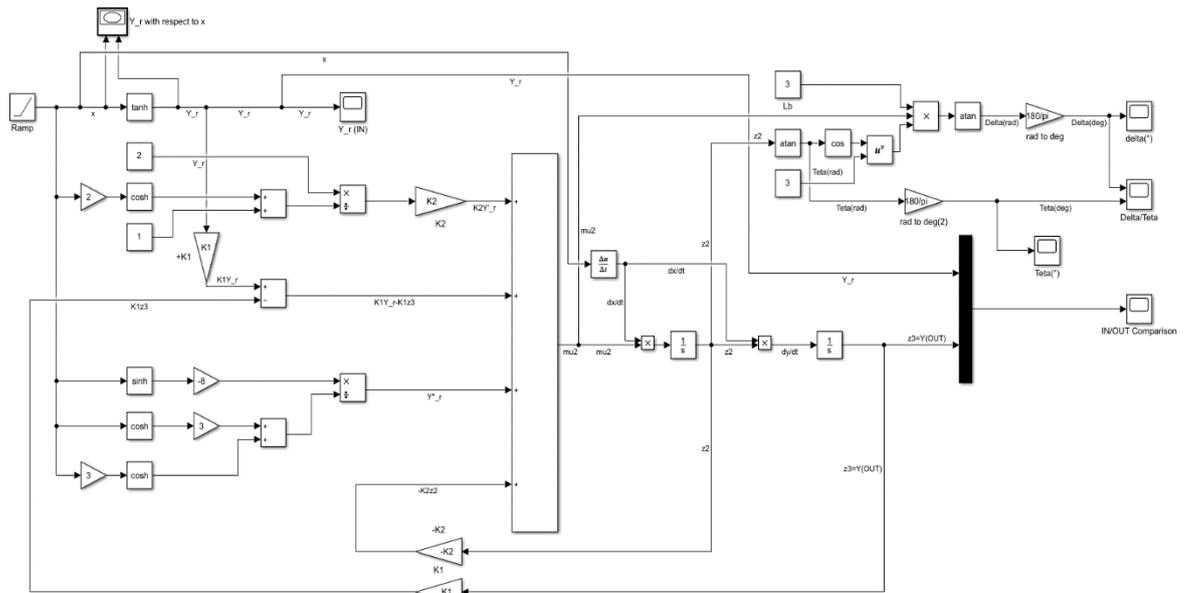


Figure 5.2: Time State Control scheme

This control scheme in a Time-State Control Form is the external generator of the reference signal and it provides those which are the ideal signal to obtain and to be followed in order to have a final parking.

Obviously, the first objective in terms of control is to follow in the right way the reference trajectory as lateral position y . The below graph shows the right parallel parking between two parked vehicles.

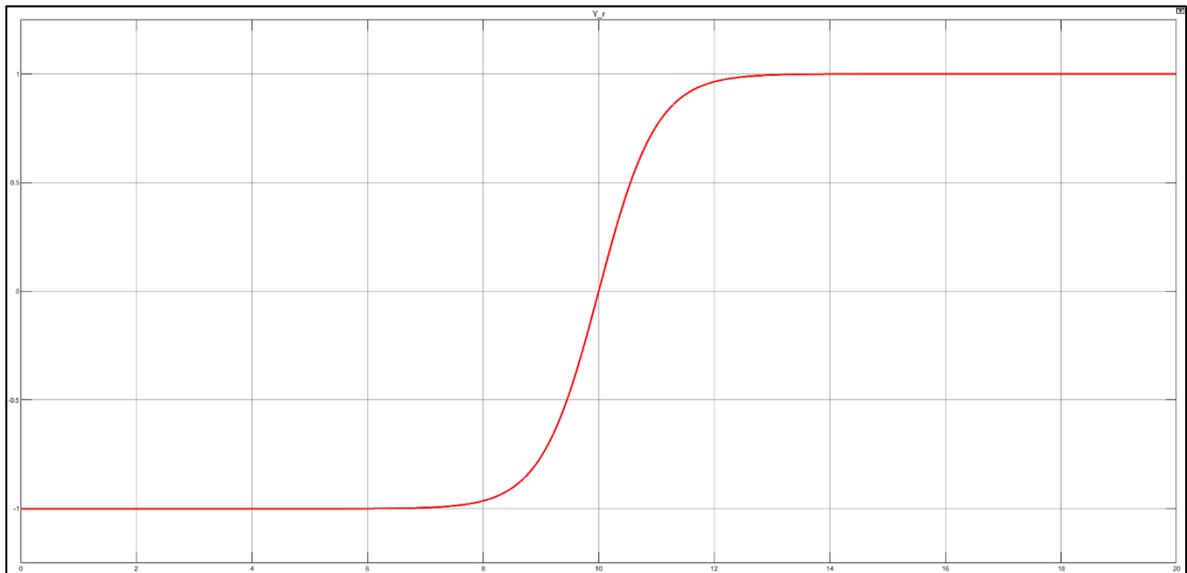


Figure 5.3: Ideal parallel trajectory

This S-trajectory which symmetrically spans from a positive value to the correspondent negative one is specifically created by means a ramp signal multiplied by a hyperbolic tangent.

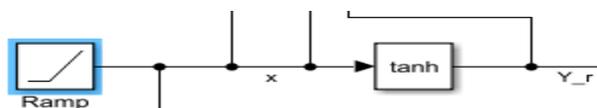


Figure 5.4: Trajectory generation in Simulink

The second signal to be followed is the yaw angle ψ . The imposition of this second following is needed because only the trajectory is not enough for the perfect parking.

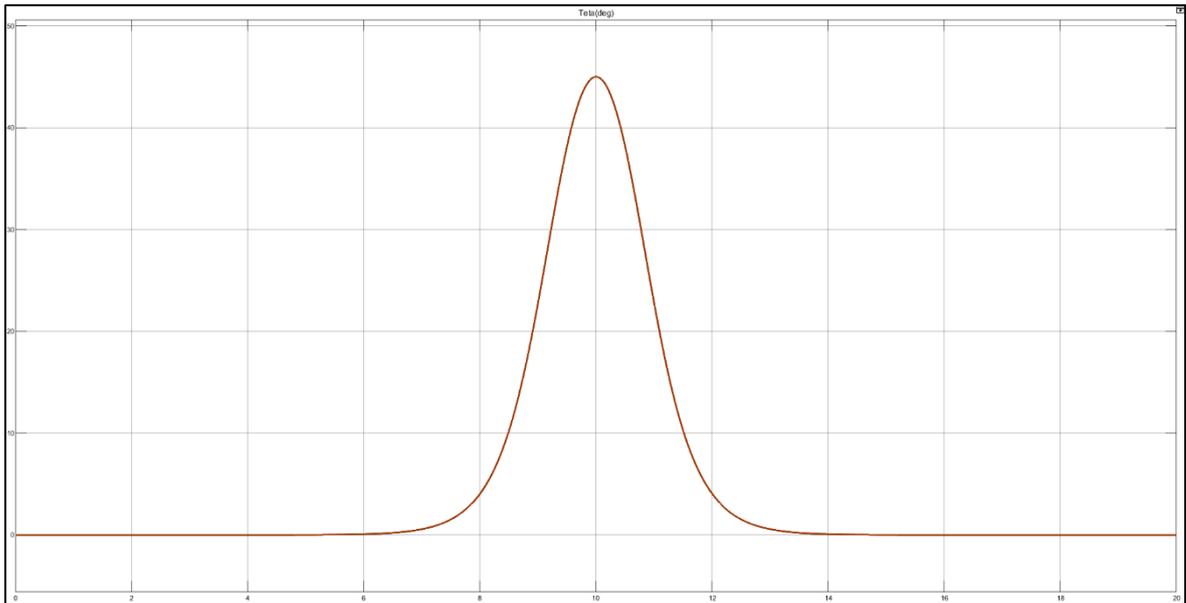


Figure 3: Ideal yaw angle

This is the trend that it must have the yaw rate during the parallel parking manoeuvre, a sort of bell shape in which the signal rises to the maximum value when the vehicle approaches the half part of the S-trajectory and decrease during the second half part.

The final steering angle δ generated has a graph of this type:

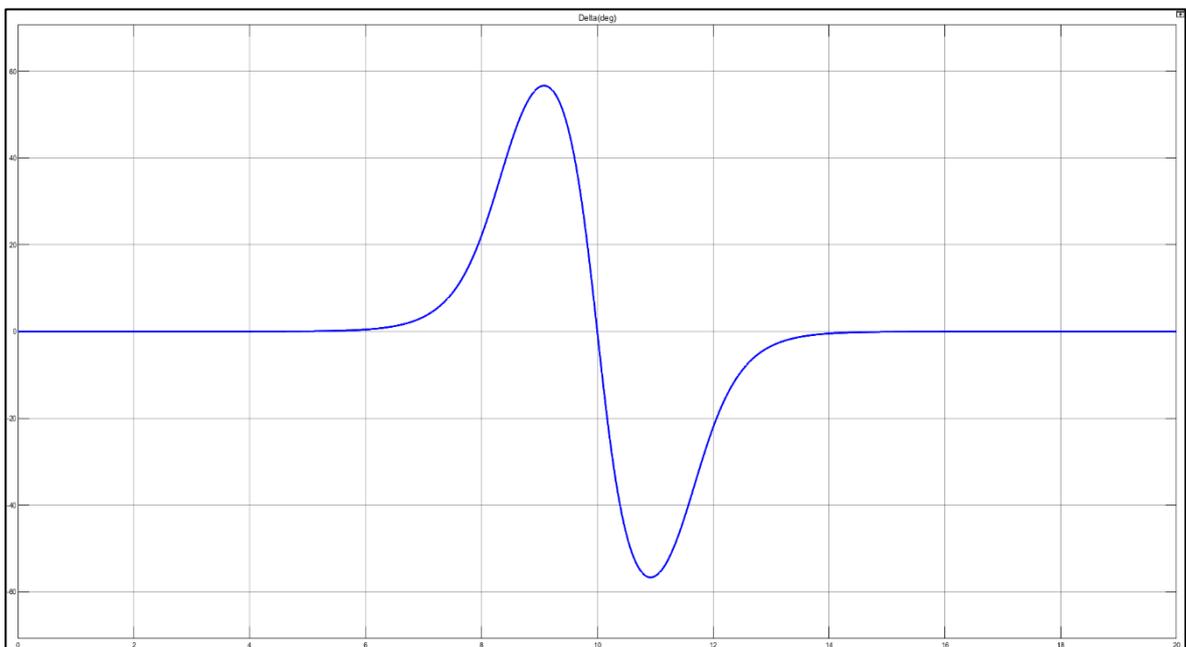


Figure 5.6: Ideal resultant steering angle

A symmetrical signal where there is a rise to the maximum value which is about 60° for the first half of the trajectory and a decrease to the minimum value (about -60°) passing from the value of 0° during the second half part.

This reference ideal result is very plausible with respect to the real parking situation.

In the following scope graph, it is possible to note the simultaneous behaviour of the steering angle δ and the yaw angle ψ during the ideal parallel parking situation.

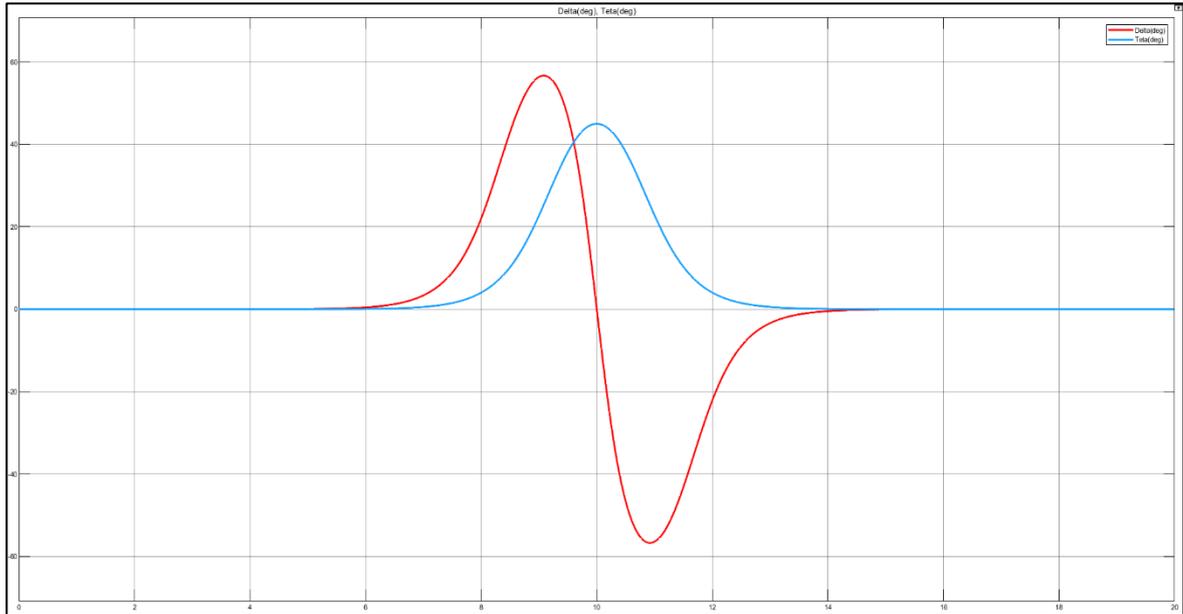


Figure 5.7: Steering angle and Yaw angle

5.2 LINEAR PLANT MODEL AND CONTROL DESIGN

Starting from the nonlinear vehicle model:

$$\begin{cases} \dot{V}_y = \frac{1}{m} \left(-V_x \dot{\psi} + C_\alpha \left[\delta - \tan^{-1} \left(\frac{V_y + l_f \dot{\psi}}{V_x} \right) \right] - C_\alpha \left[\tan^{-1} \left(\frac{V_y - l_r \dot{\psi}}{V_x} \right) \right] \right) \\ \dot{\psi} = \frac{1}{I_z} \left(l_f C_\alpha \left[\delta - \tan^{-1} \left(\frac{V_y + l_f \dot{\psi}}{V_x} \right) \right] - l_r C_\alpha \left[\tan^{-1} \left(\frac{V_y - l_r \dot{\psi}}{V_x} \right) \right] \right) \end{cases}$$

which has the following Simulink scheme:

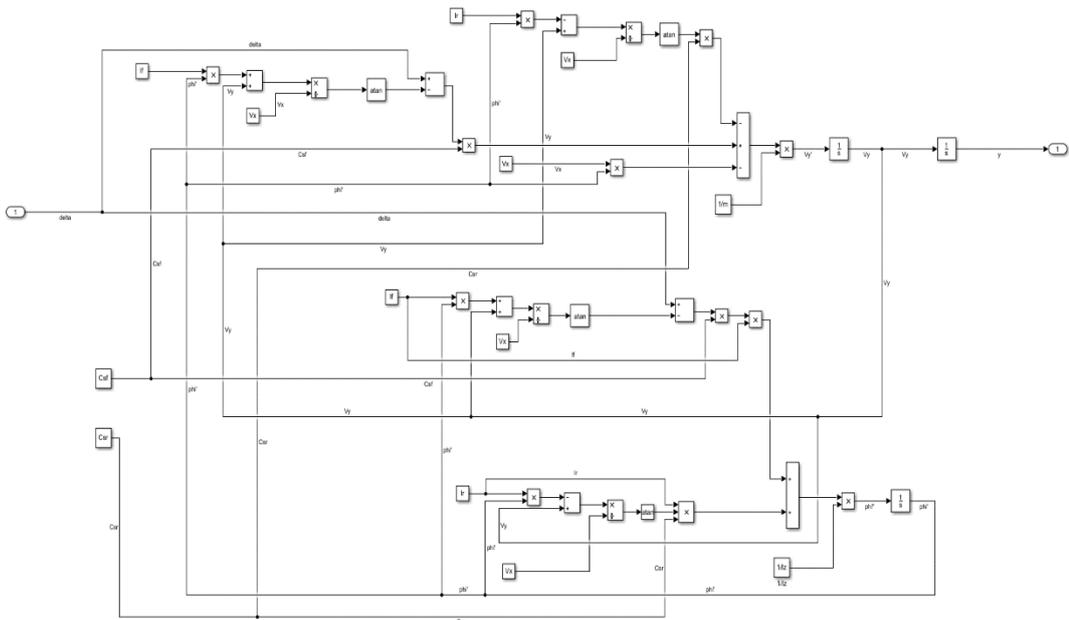


Figure 5.8: Nonlinear bicycle Simulink scheme

The schemes have been used for the preliminary simulation in which the Time State control is direct connected to the above system.

Obviously, with this scheme the total system is more difficult to manage. Generally, the vehicle dynamics can be described by a nonlinear model with 6-degrees of freedom obtained by means the Newton – Euler equations in which the focus point is to find an equilibrium

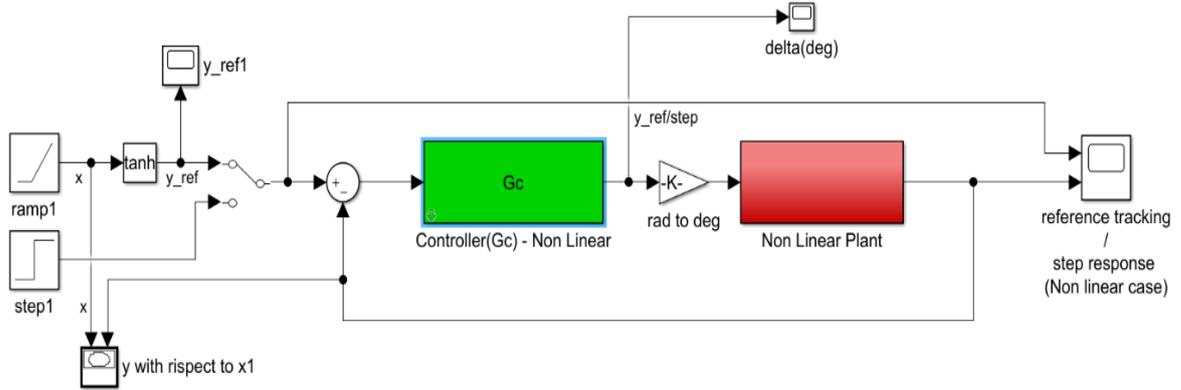


Figure 5.9: Controller connected to nonlinear plant

of forces and torques, but in this case, it is better to use a simplified and linearized model base on only lateral dynamics description. The final practical solution is a 2-degrees of freedom linear single-track model, which is a more treatable system:

$$\begin{bmatrix} \dot{v}_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_1 & \frac{a_2 - v_x^2}{a_5 v_x} \\ v_x & \frac{a_4}{v_x} \end{bmatrix} \begin{bmatrix} v_y \\ \psi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \delta_v$$

where v_y is the lateral velocity, ψ is the yaw angle, δ_v is the steering-wheel angle and the coefficients values are:

$$\{a_1, a_2, a_3, a_4, a_5, b_1, b_2\} = \{-127.24, 82536, 43.44, -148.36, 1226, 0.0475, 0.0317\}. \quad [9]$$

These equations are the outcome of three important hypotheses:

1. the vehicle is reduced to a rigid body which moves in a plane and so, no pitch, roll and heave motions are considered, and small steering angles and constant longitudinal velocity are allowable. In addition, the linearization requirement is small side slip angles such that to describe the tires in linear way.
2. there is a decoupling lateral and longitudinal dynamics in the approximation from the nonlinear model;
3. the equations are determined by linearization of the single-track model based on longitudinal velocity considered as parameter and this reduction is possible only under the assumption of the vehicle symmetrical with respect to the longitudinal plane containing x and z axes and the rolling friction neglectation. [13]

The linear bicycle model represents the plant of the system:

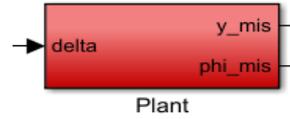


Figure 5.10: Plant block

which is modeled in Simulink by means the associated above equations:

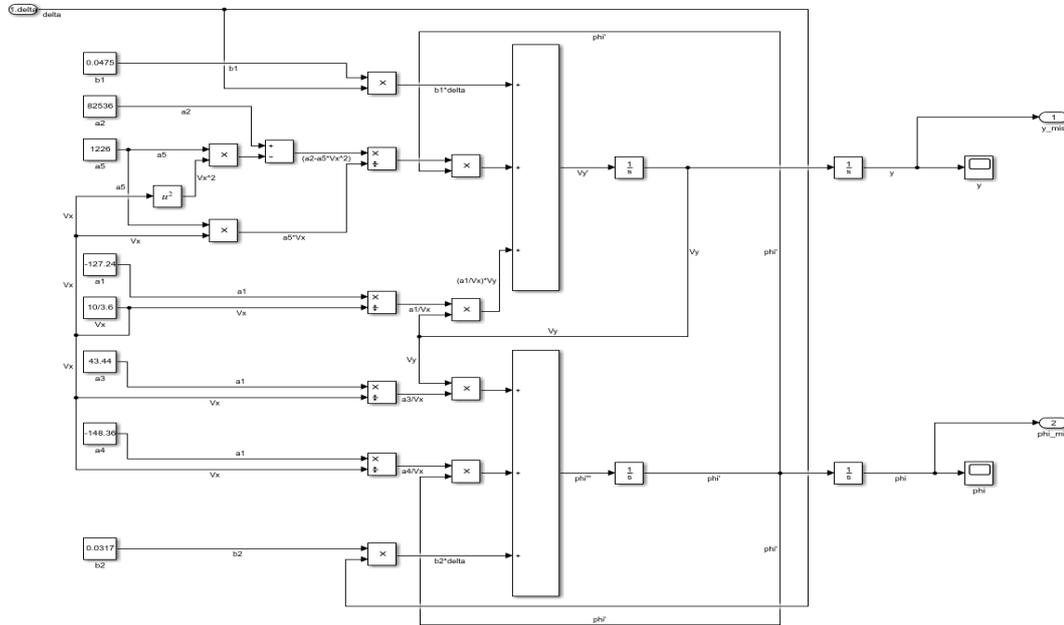


Figure 5.11: Linear bicycle Simulink scheme

The associated plant transfer functions need to be calculated:

$$Gp = [Gp_1 \quad Gp_2]^T$$

In particular, the linear bicycle plant model can be written in state-space representation in order to have a better vision of the single contribution of each element within the equations and to simpler determinate the A, B, C, D matrices for the next computations:

$$\begin{cases} \dot{x} = Ax + Bu \\ w' = Cx + Du \end{cases}$$

where:

- \dot{x} is the state vector:

$$\dot{x} = \begin{bmatrix} \dot{v}_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix}$$

- w' and thus w are the output vector and u is the input (or control) vector:

$$w' = \begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix} \text{ and } w = \begin{bmatrix} y \\ \psi \end{bmatrix} = \begin{bmatrix} Gp_1 \\ Gp_2 \end{bmatrix} \delta_v$$

- A is the state (or system) matrix:

$$A = \begin{bmatrix} \frac{a_1}{v_x} & \frac{a_2 - v_x^2}{a_5 v_x} \\ \frac{a_3}{v_x} & \frac{a_4}{v_x} \end{bmatrix} \cong \begin{bmatrix} -45.81 & 21.46 \\ 15.64 & -53.41 \end{bmatrix}$$

- B is the input matrix:

$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 0.0475 \\ 0.0317 \end{bmatrix}$$

- C is the output matrix and it is imposed to be:

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- D is the feedforward (or feedthrough) matrix and it is imposed to be:

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

From the output vector, one calculates the Gp vector:

$$Gp = \begin{bmatrix} Gp_1 \\ Gp_2 \end{bmatrix} = \begin{bmatrix} \frac{Gp_1'}{s} \\ \frac{Gp_2'}{s} \end{bmatrix}$$

that is equal to calculate the terms of the H matrix:

$$H = \begin{bmatrix} Gp_1' \\ Gp_2' \end{bmatrix} = C(sI - A)^{-1} \cdot B + D \rightarrow \begin{bmatrix} Gp_1 \\ Gp_2 \end{bmatrix} = \begin{bmatrix} \frac{Gp_1'}{s} \\ \frac{Gp_2'}{s} \end{bmatrix} \rightarrow \begin{matrix} Gp_1 = \frac{H(1,1)}{s} \\ Gp_2 = \frac{H(2,1)}{s} \end{matrix}$$

imposing that the output vector:

$$w' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u$$

Thus, the computations are:

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \left(s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{a_1}{v_x} & \frac{a_2 - v_x^2}{a_5 v_x} \\ \frac{a_3}{v_x} & \frac{a_4}{v_x} \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \left(s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -45.81 & 21.46 \\ 15.64 & -53.41 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 0.0475 \\ 0.0317 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

well-defined in the MATLAB script:

```
A=[a1/Vx, (a2-a5*Vx^2)/(a5*Vx); a3/Vx, a4/Vx];
B=[b1;b2];
D=zeros(2,1);
C=eye(2,2);
H=C*(inv(s*eye(2,2)-A))*B+D;
Gp1=minreal(zpk((H(1,1))/s),1e-2);
Gp2=minreal(zpk((H(2,1))/s),1);
```

and the final plant computed transfer functions are:

$$Gp_1 = \frac{0.0475}{s(s + 30.9)}, \quad Gp_2 = \frac{0.0317}{s(s + 30.9)}$$

the found transfer functions are those of a SITO system, where the single input is the steering angle δ , while the two outputs to be controlled are the lateral trajectory and the yaw angle.

5.2.1 SINGLE INPUT TWO OUTPUT (SITO) SYSTEMS

The problem of feedback control for Multiple Input, Multiple Output (MIMO) feedback systems is object of study from many years. Within the category of the MIMO systems one can find and approach to the SITO systems, which represents a class of problem both highly analyzed and practice relevant.

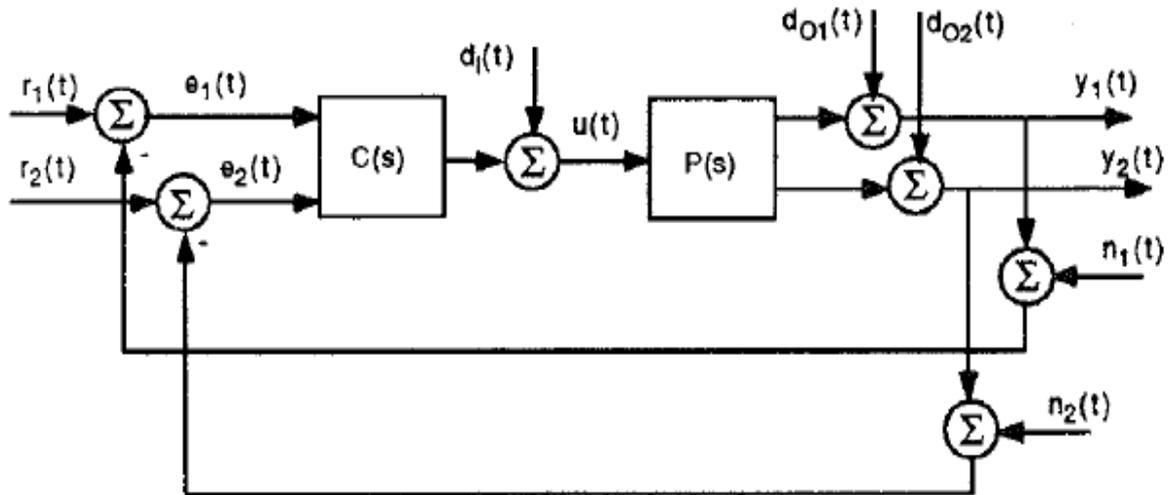


Figure 5.12: Feedback system with SITO plant

Given a matrix $M \in \mathbb{C}^{m \times n}$, one can define the range of M by $R(M)$, the right nullspace by $N_{right}(M)$, the left nullspace by $N_{left}(M)$ and the rowspace by $R_{row}(M)$.

The feedback system in figure represents a common system without assumptions and considering every type of variable which plays an important role in reality. In these systems one can denote:

- $P(s) = [p_1(s) \ p_2(s)]^T$ are the plant transfer functions;
- $C(s) = [c_1(s) \ c_2(s)]$ are the controller transfer functions;
- $r(t)$ is the reference input;
- $u(t)$ is the control input;
- $y(t)$ is the system output;
- $n(t)$ is the measurement noise;
- $e(t)$ is the measured error signal;
- $d_I(t)$ and $d_O(t)$ are the disturbance applied at the input and the output of the plant.

There different important associated transfer function of the system:

- Input and output loop transfer function

$$L_I(s) = C(s)P(s), \quad L_O(s) = P(s)C(s)$$

- Input and output sensitivity function:

$$S_I(s) = \frac{1}{1 + L_I(s)}, \quad S_O(s) = \frac{1}{1 + L_O(s)}$$

- Input and output complementary function:

$$T_I(s) = \frac{L_I(s)}{1 + L_I(s)}, \quad T_O(s) = \frac{L_O(s)}{1 + L_O(s)}$$

To describe the response of a SITO plant and a TISO controller (the same used in this working thesis) at each frequency it is important to use linear algebra concepts.

Basically:

1. let $P(s) = N_p(s)D_p^{-1}(s)$ a right polynomial plant factorization. By defining the direction of the plant at frequency w by $R(N_p(jw))$ and suppose that $P(jw) \neq 0$, then $R(N_p(jw)) = R(P(jw))$, if jw is not a pole of $P(s)$;
2. let $C(s) = D_c^{-1}(s)N_c(s)$ a left polynomial controller factorization. By defining the direction of the controller at frequency $s = jw$ by $R_{row}(N_c(jw))$ and suppose that $C(jw) \neq 0$, then $R_{row}(N_c(jw)) = R_{row}(C(jw))$.

It is needed to measure how closely the controller direction corresponds to that of the plant. For this reason, under the assumption of $P(jw) \neq 0$ and $C(jw) \neq 0$, one can define the alignment angle between the plant and the controller:

$$\varphi(jw) = \arccos\left(\frac{|C(jw)P(jw)|}{\|C(jw)\|\|P(jw)\|}\right)$$

which for definition needs to satisfy $\varphi(jw) \in [0^\circ, 90^\circ]$.

So, five different conditions are determined:

- *perfect alignment* between plant and controller if $\varphi(jw) = 0^\circ$;
- *misalignment* between plant and controller if $\varphi(jw) > 0^\circ$;
- *complete misalignment* between plant and controller if $\varphi(jw) = 90^\circ$;
- *well alignment* between plant and controller if $\varphi(jw) \approx 0^\circ$;
- *poor alignment* between plant and controller if $\varphi(jw) \approx 90^\circ$.

There are two necessary and sufficient conditions to satisfy for obtaining a perfect alignment between plant and controller and they are related to the gain and phase ratio:

$$|C_{rat}(jw)| = |P_{rat}(jw)|$$

And, if $p_1(s)$ and $p_2(s)$ are both different from zero:

$$\text{phase}(C_{rat}(jw)) = -\text{phase}(P_{rat}(jw))$$

If one of the two conditions are not respected, then a poor alignment is obtained. [14]

After the plant design, it is necessary to mathematically create thanks to a loop-shaping method the controller transfer functions of the TISO controllers:

$$Gc = [Gc_1 \quad Gc_2]$$

in MATLAB:

```
Gc1 = minreal(inv(Gp1)/s);
Gc2 = minreal(inv(Gp2)/s);
Gc1 = minreal(Gc1/(1+s/100));
Gc2 = minreal(Gc2/(1+s/100));
```

thus, the final controller transfer functions are:

$$Gc_1 = \frac{2105.3(s + 30.9)}{(s + 100)}, Gc_2 = \frac{3154.6(s + 30.9)}{(s + 100)}$$

The alignment angle computed in MATLAB is about 90° , so it is important to modify the design approach and later introduce something that compensate the problem by means particular filters useful for several purposes.

Is it significant to determine the output open loop transfer function:

$$L_o = Gp \cdot Gc = Gc_1 Gp_1 + Gc_2 Gp_2$$

which is needed to design Gc_1 and Gc_2 in order to properly shape the frequency response of:

$$S_o = (1 + L_o)^{-1} = \frac{1}{1 + L_o} = \frac{1}{1 + Gc_1 Gp_1 + Gc_2 Gp_2}$$

The next computation is that of T_o :

$$T_o = \frac{L_o}{1 + L_o} = \frac{Gp \cdot Gc}{1 + Gp \cdot Gc} = \frac{[Gp_1 \ Gp_2]^T \cdot [Gc_1 \ Gc_2]}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + Gc_1 Gp_1 + Gc_2 Gp_2}$$

finally equals to:

$$T_o = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} \frac{Gc_1 Gp_1}{1 + Gc_1 Gp_1 + Gc_2 Gp_2} & \frac{Gc_2 Gp_1}{1 + Gc_1 Gp_1 + Gc_2 Gp_2} \\ \frac{Gc_1 Gp_2}{1 + Gc_1 Gp_1 + Gc_2 Gp_2} & \frac{Gc_2 Gp_2}{1 + Gc_1 Gp_1 + Gc_2 Gp_2} \end{bmatrix}$$

It is necessary to uncouple the output y and ψ from the reference input y_{ref} and ψ_{ref} respectively so that y is never influenced by ψ_{ref} and ψ is never influenced by y_{ref} .

To solve this, one could introduce the so-called feedforward filter on the reference input:

$$F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$$

designed mathematically in that way:

$$F \cdot T_o = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \cdot \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} F_{11}T_{11} + F_{12}T_{21} & F_{11}T_{12} + F_{12}T_{22} \\ F_{21}T_{11} + F_{22}T_{21} & F_{21}T_{12} + F_{22}T_{22} \end{bmatrix} = \begin{bmatrix} Gr_{11} & Gr_{12} \\ Gr_{21} & Gr_{22} \end{bmatrix}$$

and so, in Simulink:

```
Gr11 = minreal(zpk(F11*T11), 1e-3) + minreal(zpk(F12*T21), 1e-3);
Gr22 = minreal(zpk(F21*T12), 1e-3) + minreal(zpk(F22*T22), 1e-3);
Gr12 = minreal(zpk(F11*T12), 1e-3) + minreal(zpk(F12*T22), 1e-3);
Gr21 = minreal(zpk(F21*T11), 1e-3) + minreal(zpk(F22*T21), 1e-3);
```

where *minreal* produce for a given LTI system model an equivalent minimal realization system where all cancelling pole/zero pairs or non-minimal state dynamics are eliminated, while *zpk* constructs a zero-pole-gain format model.

For the aim of the system, the four functions will have an imposed frequency response:

- sensitivity response for Gr_{11} and Gr_{22} , with a frequency band tending to 1

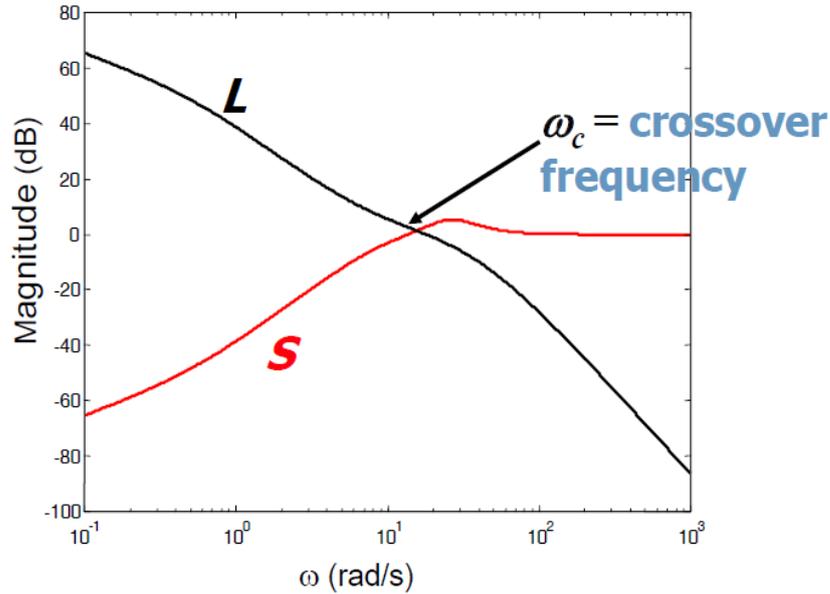


Figure 5.13: Sensitivity frequency response

- complementary sensitivity response for Gr_{12} and Gr_{21} , with a small frequency tending to 0

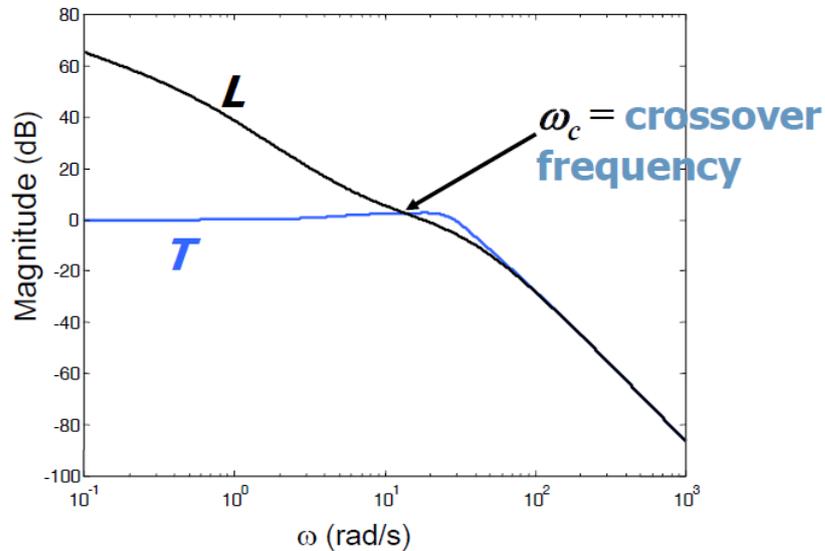


Figure 5.14: Complementary sensitivity frequency response

To obtain the desired frequency response, one need to assume the following mathematical relations:

$$1) Gr_{12} = F_{11}T_{12} + F_{12}T_{22} = 0 \rightarrow F_{11} = \left(-\frac{T_{22}}{T_{12}}\right)F_{12}$$

$$2) Gr_{21} = F_{21}T_{11} + F_{22}T_{21} = 0 \rightarrow F_{21} = \left(-\frac{T_{21}}{T_{11}}\right)F_{22}$$

So, the other two relations will be:

$$3) Gr_{11} = (F_{11}T_{11} + F_{12}T_{21}) \sim 1 \text{ (very small frequency band)}$$

$$4) Gr_{11} = (F_{21}T_{12} + F_{22}T_{22}) \sim 1 \text{ (very small frequency band)}$$

From the last two relations:

$$F_{11}T_{11} + F_{12}T_{21} = \left(-\frac{T_{22}T_{11}}{T_{12}}\right)F_{12} + T_{21}F_{12} = F_{12}\left(T_{21} - \frac{T_{22}T_{11}}{T_{12}}\right)$$

$$F_{21}T_{12} + F_{22}T_{22} = \left(-\frac{T_{21}T_{12}}{T_{11}}\right)F_{22} + F_{22}T_{22} = F_{22}\left(T_{22} - \frac{T_{21}T_{12}}{T_{11}}\right)$$

Since from the MATLAB computation:

```
%% FeedForward Filters
```

```
pp=1e4;
F12 = minreal(zpk(T12/(T21*T12-T22*T11)))/(1+s/pp)^6
F11 = minreal(zpk((-T22/T12)*F12))
F22 = minreal(zpk(1/(T22-T21/T11*T12)))/(1+s/pp)^4
F21 = minreal(zpk((-T21/T11)*F22))
```

the resultant four filter are improper and therefore closing high-frequency poles are posteriori added at about a decade beyond the feedback system band to fill the poles-zeros gap.

The final overall feedback scheme is a multivariable feedback control scheme, which is represented in Simulink environment in this way:

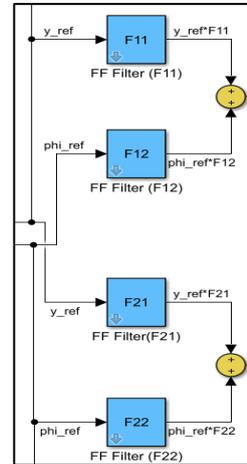


Figure 5.15: Feedforward filters

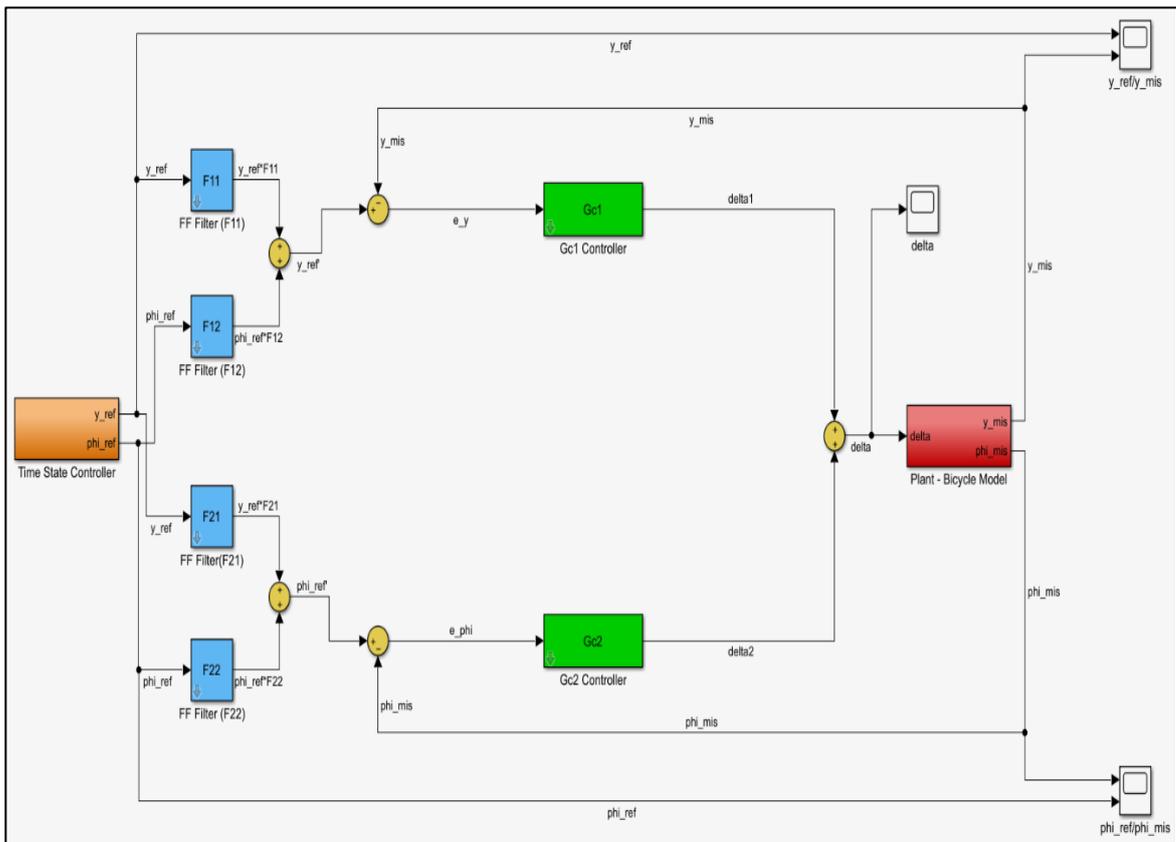


Figure 4: Final complete system Simulink scheme

In this scheme the feedforward filters have an important because allow to impose the key condition to obtain the desired behaviours in terms of controlled variable.

The resultant trajectory obtained with the scheme almost match the reference trajectory as is shown in the figure below:

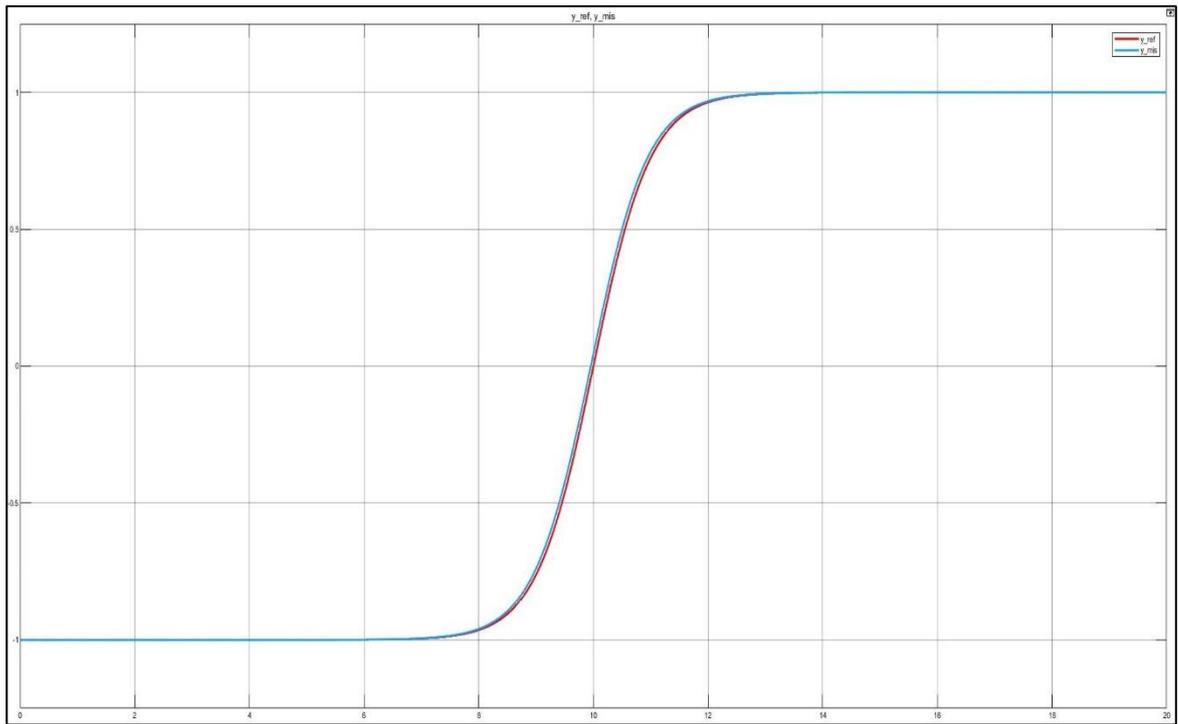


Figure 5: Scope of the reference trajectory and the resultant trajectory

The same matching is obtained for the yaw angle that is almost perfectly followed by the generated yaw angle. This is the last result after some correction related to the fact that the peak at the beginning was slower than the reference one.

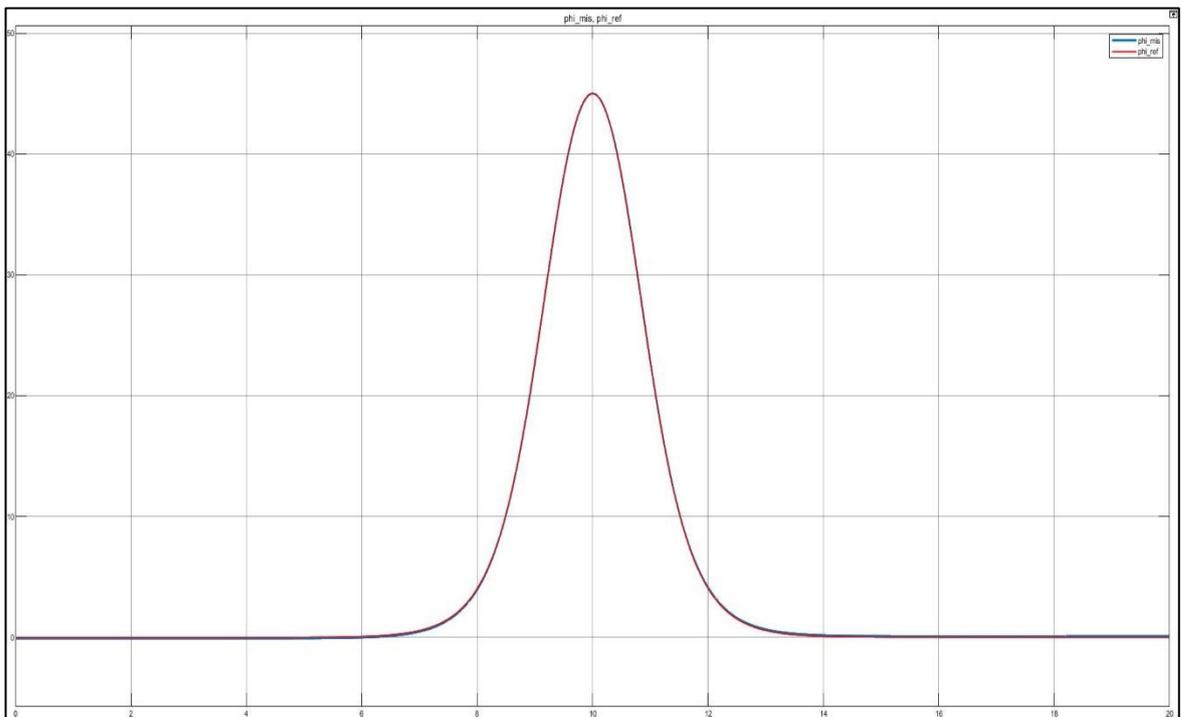


Figure 68: Scope of the reference yaw angle and the resultant yaw angle

The double control allows to obtain a right parallel parking in terms of both lateral trajectory and yaw angle.

The final consideration is related to the steering angle that need to space in the double manoeuvre from a positive value to the correspondent negative one, passing from the first steering to the second one to place the vehicle in the straight direction inside the lot.

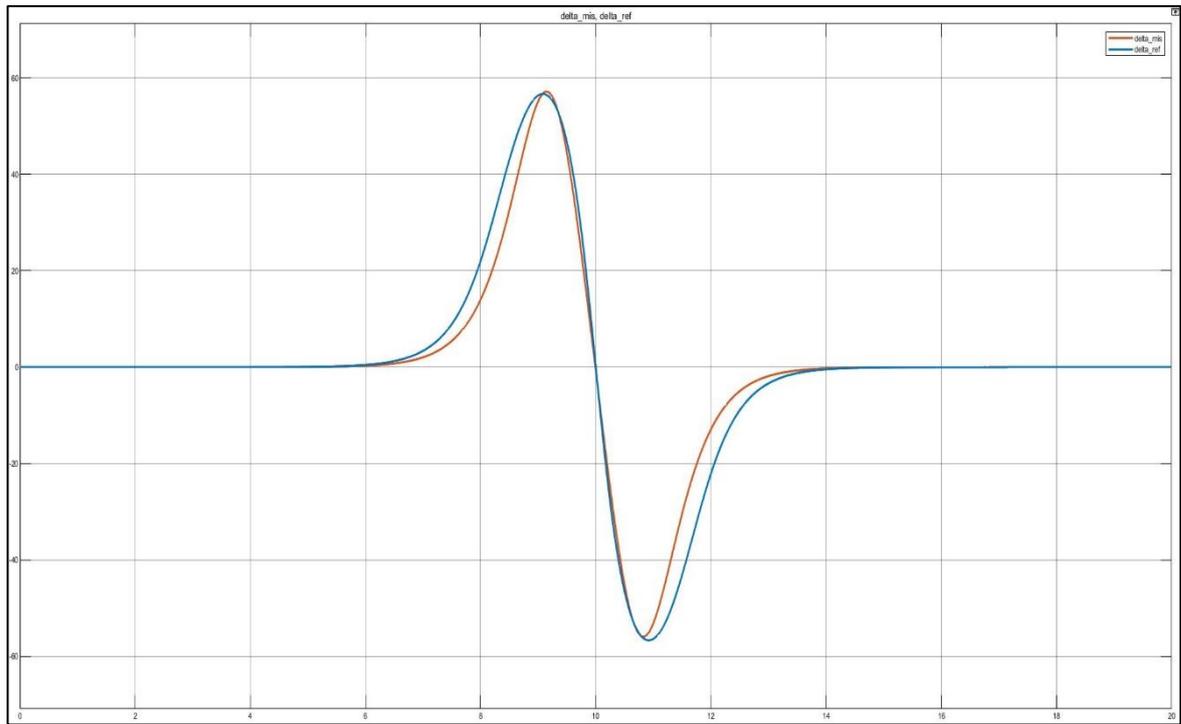


Figure 79: Scope of the ideal steering angle and resultant steering angle

CONCLUSIONS

The obtained results show how the experienced approach as results of mathematical and theoretical considerations can be used to face with a problem like a parking situation.

The analysis of the problem has highlighted the need to project a MIMO control system with 2-degrees of freedom composed by a feedback TISO controller and a feedforward filter used to perform the decoupling of the reference signals effects on the outputs.

The approach tested on a SITO bicycle model system has given positive results even if however, the performed tests are not very realistic because the model uncertainties have not been taken into account.

Thus, the future developments are those of analysing the effect of the model uncertainty presence and maybe considering no longer a linear model, but a nonlinear one.

LIST OF FIGURES

1.1 Remote-controlled vehicle.....	9
1.2 General Motors Prototype.....	9
1.3 Tsubuka autonomous vehicle.....	10
1.4 Dickmanns team vehicle.....	10
1.5 Level of integration of companies that works in autonomy field.....	11
1.6 Free driver from driving tasks.....	12
1.7 Autonomous vehicles levels.....	13
1.8 Perception of the environment.....	14
1.9 Lane line marking detection.....	16
1.10 LIDAR detection.....	17
1.11 LIDAR.....	18
1.12 LIDAR Image.....	19
1.13 DARPA CHALLENGE.....	21
1.14 Probabilistic Road Maps (PRM).....	24
1.15 Rapidly-exploring Random Trees (RRT).....	24
1.16 Decision making for obstacle avoidance.....	25
2.1 Toyota Prius sensors (2003).....	31
2.2 Parking vision camera.....	32
2.3 Volvo autonomous parking.....	32
2.4 Direct trajectory planning.....	33
2.5 Planning scheme.....	34
2.6 Bicycle model - Forward motion.....	35
2.7 Bicycle model - Backward motion.....	35
2.8 Possible trajectories.....	37
2.9 Specific trajectory.....	37
2.10 Steering angle sequence.....	37
2.11 From the initial to the finale state.....	38
2.12 Bicycle model representation.....	39
2.13 Steering wheels and maximum rotation angle.....	40
2.14 Sensors information.....	40
2.15 Fuzzy controller.....	41

2.16	Inputs and outputs of controller.....	42
2.17	Controller conditions.....	42
2.18	Four-wheeled car-like robot modelled as a bicycle.....	43
2.19	Gompertz curve movement of the vehicle centre.....	44
2.20	Gompertz curve function.....	45
2.21	Parameter a evaluation.....	45
2.22	Parameter b evaluation.....	46
2.23	Parameter c evaluation.....	46
2.24	Visualization of the optimal scheme result.....	48
2.25	Evolution of trajectory based on b and c	49
3.1	The functioning of a yaw control system.....	51
3.2	The functioning of a yaw control system.....	53
3.3	Ackerman turning geometry.....	55
3.4	Differential steer from a trapezoidal tie-rod arrangement.....	56
3.5	Lateral vehicle dynamics.....	57
3.6	Tire slip angle.....	58
4.1	Closed-loop system.....	64
5.1	Single track vehicle model.....	72
5.2	Time State Control scheme.....	75
5.3	Ideal parallel trajectory.....	75
5.4	Trajectory generation in Simulink.....	75
5.5	Ideal yaw angle.....	76
5.6	Ideal resultant steering angle.....	76
5.7	Steering angle and Yaw angle.....	77
5.8	Nonlinear bicycle Simulink scheme.....	77
5.9	Controller connected to nonlinear plant.....	78
5.10	Plant block.....	79
5.11	Linear bicycle Simulink scheme.....	79
5.12	Feedback system with SITO plant.....	81
5.13	Sensitivity frequency response.....	84
5.14	Complementary sensitivity frequency response.....	84

5.15 Feedforward filters.....	85
5.16 Final complete system Simulink scheme.....	85
5.17 Scope of the reference trajectory and the resultant trajectory.....	86
5.18 Scope of the reference yaw angle and the resultant yaw angle.....	86
5.19 Scope of the ideal steering angle and resultant steering angle.....	87

BIBLIOGRAPHY

- [1] Markus Maurer, J. Christian Gerdes, Barbara Lenz, Hermann Winner, “Autonomous Driving”, *Springer*, 2015
- [2] Isabel Harne, “The 5 Autonomous Driving Levels Explained”, <https://www.iotforall.com/5-autonomous-driving-levels-explained/amp/>, 2017
- [3] Aakash Goel, “What Tech Will it Take to Put Self-Driving Cars on the Road?”, <https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/13270/What-Tech-Will-it-Take-to-Put-Self-Driving-Cars-on-the-Road.aspx>, 2016
- [4] Landmark Dividend, <https://www.landmarkdividend.com/self-driving-car/>
- [5] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, Marcelo H. Ang Jr., “Perception, Planning, Control, and Coordination for Autonomous Vehicles”, in *machines*, 2017
- [6] Shubhankar Gautam, “A Brief History of Car Parking Technology”, <http://blog.getmyparking.com/2019/04/08/a-brief-history-of-car-parking-technology>, 2019
- [7] Wei Liu, Zhiheng Li, Li Li, Fei-Yue Wang, “Parking Like a Human: A Direct Trajectory Planning Solution”, in *IEEE Transactions on Intelligent Transportation Systems*, 2017
- [8] Denis M. Filatov, Elena V. Serykh, Michael M. Kopichev, Andrey V. Weinmeister, “Autonomous Parking Control System of Four-Wheeled Vehicle”, in *IEEE*, 2016

- [9] Aneesh N. Chand, Michihiro Kawanishi and Tatsuo Narikiyo, “Fast Parallel Parking for Autonomous Vehicles using Gompertz Curve”, in *IEEE*, 2014
- [10] Rajesh Rajamani, “Vehicle Dynamics and Control”, *Springer*, 2006
- [11] Jean-Jacques E. Slotine, Weiping Li, “Applied Nonlinear Control”, *Pearson plc.*, 1991
- [12] Kentaro Oyama, Kenichiro Nonaka, “Model Predictive Parking Control for Nonholonomic Vehicles using Time-State Control Form”, in *European Control Conference*, 2013
- [13] V. Cerone, A.Chinu, D. Regruto, “Experimental Results in Vision-Based Lane Keeping for Highway Vehicles”, in *Proceedings of the American Control Conference*, 2002
- [14] Jim Freudenberg, Rick Middleton, “Properties of Single Input, Two Output Feedback Systems”, in *Proceedings of the American Control Conference*, 1998