Generalized Feedback Control and Application to Vehicle Path Following Control

von Diplom Ingenieur
Frank-René Schäfer
aus Köln

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr. Ing. –
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Ing. Gühmann
Gutachter:	Prof. Dr. Ing. Buss
Gutachter:	Prof. Dr. Ing. King

Tag der wissenschaftlichen Aussprache: 15. April 2004

Berlin 2004
D 83

Abstract

In this dissertation, a discrete control approach is introduced and defined as *Generalized Feedback Control* (GFC). Traditionally, feedback control is associated with a control signal being computed proportionally to the error between the achieved response of the plant and the desired response. A summation point providing the difference between both becomes a central element in the traditional paradigm. Constraining the feedback to a pure subtraction is essential for the analysis of the control system by means of linear system theory.

The Generalized Feedback Control approach does not restrict the processing of feedback to a functional processing but to algorithms. Two essential types of feedback are identified: motivation matching and circumstance cognition. A *geometric-dynamic planning unit* processes the plant's state and interferes it with a *short term motivation.* As a result, it delivers a set of parameters specifying the nominal behavior for the following control interval, called *gd-plan.* This gd-plan is then passed to a so called *plan-to-action mapper.* This observes the plant's state in order to determine the control parameters to achieve the gd-plan for the given circumstances.

The depart from the summation point means at the same time a depart from classical methods of analysis to test the reliability/stability of the controller. This monography introduces an algorithmic approach to test for the reliability of the controller: the criteria *containability.* It is based on the idea to determine the ability to stay inside an admissible domain of system states.

A mathematical modeling of the short term motivation requires in depth consideration. Therefore, an integrated mathematical approach to derive one distinct preferred gd-plan from a set of preferences on the set of possible gd-plans is introduced. The procedure is then defined as the method of *target and limit maps.*

In order to clarify introduced concepts and in order to show its utility, the approach is applied to the problem of combined lateral and longitudinal control of a vehicle at limit handling conditions.

Contents

Prolog

This research project has been accomplished at the department EW-3 at BMW in Munich, the Technical University of Berlin, Germany, and the Mechanical Engineering Department at Clemson University, South Carolina, USA.

The author wishes to thank Dr. Juergen Schuller for the suggestion of the research project on driver modeling and his reviewing of the dissertation. Further, special thanks to the research center of BMW in Munich (FIZ) who sponsored the project. The exertion of Professor Dr. Buss at the Technical University of Munich, as well as Professor Dr. King and Dr. Thomas Juergensohn at the Technical University of Berlin for being correctors of this dissertation is greatly appreciated. The author, also, wishes to thank Professor Dr. Haque at Clemson University for his advices with respect to vehicle dynamics and scientific publishing. Finally, the author wishes to thank Professor Dr. Onken from the University of Armed Forces in Munich for useful hints and discussions on the subject of vehicle control and human driver modelling.

Nomenclature

Symbols in this section are divided into latin letters, greek letters and functions. A derivative of a variable with respect to time is expressed by a dot, e.g. $\dot{x} = \frac{d}{dt}x$. A derivative of a variable with respect to distance is indicated by a prime, e.g. $\kappa' = \frac{d}{ds}\kappa$.

Latin Letters

$\mathcal{A}$ Domain of admissible system states.

$\mathcal{A}^c$ Domain of inadmissible system states.

$\mathcal{A}_{gd}$ Domain of admissible gd-plans.

$\mathcal{A}_m$ Domain of manageable system states.

$\mathcal{A}_s$ Domain of safe target system states.

B Width of the vehicle.

b_{gd} Geometric-dynamic behavior of the plant during one control interval.

c_p Control parameters.

$\mathcal{D}_\infty^*$ Domain that envelops $\mathcal{D}_\infty$.

$\mathcal{D}_\infty$ Domain that envelops all dilemma domains $\mathcal{D}_k$.

$\mathcal{D}_k$ Dilemma domain of k-th order.

F_r Force related to air drag.

F_p Propulsive force.

$F_{s,f}$ Side force on the front wheel.

$F_{s,r}$ Side force on the rear wheel.

m Mass of the vehicle.

p Parameterization index.

$\mathcal{P}_{gd}$ Space of gd-plans.

p_{gd} GD-Plan. Describes nominal behavior over the subsequent control interval.

$\underline{p}^*$ State dependent input parameters for control parameter computation inside the plan-to-action mapper.

$\underline{\tilde{p}}$ Fixed input parameters for circumstance cognition inside the plan-to-action mapper.

$\Re$ Set of real numbers.

$\vec{r}_f$, $\vec{r}_r$ Vector from center of gravity to center of front and rear axles.

s State of the plant. Note, that local to some sections this index can be used to indicate a geometric length.

t Time.

T_c Length of control interval. Time between two control impulses.

T_s Length of the shaping function for control impulses.

v Velocity of the vehicle.

$\vec{v}$ Velocity vector in the center of gravity.

$\vec{v}_f$, $\vec{v}_r$ Velocity vectors in front and rear.

$\vec{x}$ Position of the vehicle's center of gravity.

x_{cg}, y_{cg} X- and y-coordinates of the vehicle's c.g.

$\ddot{y}$ Lateral acceleration in the center of gravity.

$\ddot{y}_{max,course}$ Maximum lateral acceleration limit for considerations based on curvature profile of the nominal course.

$\ddot{y}_{max,curv}$ Maximum lateral acceleration limit for considerations of short term curvature profile.

$\ddot{y}_{max,nestle}$ Maximum lateral acceleration limit for Nestle Curves.

$\ddot{y}_{max}$ Absolute value of maximum lateral acceleration limit.

Greek Letters

α_f, α_r Slip angle at the front and read tire group.

β Slip angle in the center of gravity of the vehicle.

$\Delta\psi$ Yaw angle deviation. Angle between the vehicle's longitudinal axes versus the angle x-axis of the frame of reference.

$\Delta\vartheta$ Angular deviation of velocity vector in vehicle's c.g. to angle of tangent in the nearest point on nominal course.

$\Delta\vartheta_p$ Predicted angular deviation of velocity vector in vehicle's c.g. to angle of tangent in the nearest point on nominal course.

Δd Lateral deviation of vehicle's c.g. from nominal course.

Δd_p Predicted lateral deviation of vehicle's c.g. to nearest point on nominal course.

δ Term that is proportional to front wheel angle. It is used to describe the front wheel input. Generally, it represents the front wheel angle itself.

κ Curvature of trajectory.

ψ Yaw angle of the vehicle.

Operators, Functions and Databases

$\chi_i(s)$ Attribute number i of a state s of the plant.

$dist(a, b)$ Scalar function indicating the distance from a to b (and vice versa).

$f_s(\tau)$ Generalized shaping function for control impulses.

$\mathbf{G}(s)$ Function that represents the gd-plan construction. It maps a given state s to a desired gd-plan p_{gd}.

$\kappa_{nc}(p)$ Curvature profile of nominal course.

$\mathcal{L}[b_{gd} : c_p](s_0)$ Table that associates a geometric-dynamic behavior b_{gd} with control parameters c_p for a given initial state s_0.

$L(p_{gd})$ Limit map. Assigns to each gd-plan p_{gd} a admissibility correspondent limit desires.

$\vec{n}(p)$ Parameterization of nominal course. p is the parameterization index.

$\mathbf{P}(s, p_{gd})$ Function the represents the plan-to-action mapper. It maps a given state s and a gd-plan p_{gd} that has to be performed to control parameters c_p.

$R(c_p,\, s)$ Response of the plant on control parameters c_{gd} when applied in state s.

$\mathcal{R}_m(p_{gd}, s)$ Range of motion. Set of possible system states that may be reached when the gd-plan p_{gd} is sent to the plan-to-action mapper. The range of motion also depends on the actual system state s.

$R^*(p)$ Simplified notation for $R(p,\, s)$ for a fixed state $s \,=\, s_0$.

R^{*-1} Inverse of R^* for a fixed state s_0. $R^{*-1}\, b_{gd}$ describes the control parameters c_p that caused the geometric-dynamic behavior of the plant b_{gd}.

$s_d(p_{gd},\, s)$ State of the plant that is reached when the gd-plan p_{gd} is perfectly implemented starting from state s.

$T(p_{gd})$ Target map. Assigns to each gd-plan p_{gd} a utility correspondent target desires.

$U(p_{gd})$ Function that assigns to each gd-plan p_{gd} a utility combining the target map $T(p_{gd})$ and the limit map $L(p_{gd})$.

$v_{nc}(p)$ Velocity profile of nominal course assuming a maximum lateral acceleration $\ddot{y}_{max,course}$.

Chapter 1

Introduction

Control, that is modifying the environment corresponding to a specific motivation[1] is an essential characteristic attributed to living beings. It is so essential that enhancing the ability to influence the environment has become a motivation on its own. For this reason tools such as hammers and flint stones were developed. Humankind, though, came up with a type of tools different from tools developed by any other species: tools that perform control autonomously. As an example for an early control system one may consider the greek Ktebios' float regulator for a water clock [Mayr, 1971]. A cornerstone in control systems marks the book 'Pneunematica', written by Heron of Alexandria in the first century C.E. [Mayr, 1970]. A cornerstone in the industrial age was set 1769 by James Watt with his flyball governor to control the speed of a steam engine. In the nineteenth century, first sophisticated mathematical formulations on control systems were accomplished by Maxwell [Maxwell, 1868] and Vyshnegradskii [Vyshnegradskii, 1877]. With advances made in mathematics, physics and computational sciences, artificial control systems gain more and more autonomy and flexibility. The robotic hand developed at MIT [Mason and Salisbury, 1985, Grupen and Coelho, 2000] and developments towards autonomous walking robots [Buss et al., 2003, Menzel and D'Aluiso, 2000] demonstrate the possibilities of todays highly advanced control systems.

Control systems are usually divided into *open-loop* and *closed-loop* systems [Bode, 1964]. An open-loop system does not receive any feedback from the environment while it attempts to set control parameters. A closed-loop, however, perceives the plant's state and can observe the difference to a desired output. Animals in nature usually perform closed-loop control since they see, hear, smell or grope the consequences of their actions[2].

Technically, closed-loop systems are more complex, since they require addi-

[1] A detailed definition of motivation is given later in section 2.2.

[2] However, animals not always process the received information. An interesting example is the dung beetle applying a dung ball to plug its nest. When removing its dung ball en route, the dung beetle keeps pantomiming plugging its nest with it, never noticing that something is missing [Hanski and Cambefort, 1991].

tional components such as sensors and filters. Moreover, a closed-loop control system is a recursive system and therefore potentially *chaotic*[3], i.e. it may show unpredictable and therefore potentially unstable behavior. However, the advantages of a closed-loop system generously outweigh these disadvantages [Black, 1977, Newton et al., 1957]. Since the plant is under permanent observation, closed-loop systems can compensate for disturbance signals and variation of parameters of the plant. For the same reason, they facilitate an adaption of the transient response to design specifications. After the transient response has decayed, a so called steady-state error may occur. It is easier to be compensated when feedback is provided.

A traditional closed-loop system using linear systems is shown in figure 1.1. The input to the system is a nominal value R that is pre-filtered by G_p. The plant's state is filtered through G_r and the difference of both results is passed to a linear system G_c that produces the control parameters for the plant. A *summation point* is the essential element to consider a nominal value in relation to the actual plant's state.

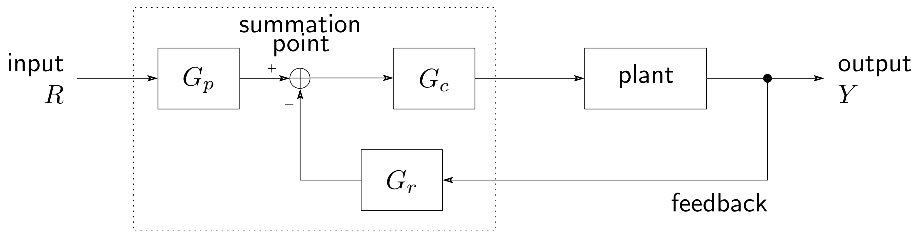


Figure 1.1: Structural diagram of a closed-loop system

The advantage of a system representation as in figure 1.1 is that its behavior can be investigated efficiently by means of the Laplace transform [Sarachik, 1997]. To compute the control signal based on the difference between an actual value and a nominal value is very intuitive and at the same time facilitates the mathematical handling of the problem. One disadvantage of a controller such as in figure 1.1 is that the large majority of physical systems are not linear.

Linearizations can be accomplished through limiting the range of states around an operating point [Kadiyala, 1993]. The major restriction focused on in this dissertation is the functional processing of feedback. The following text steps away from an understanding of a control system in terms of signals and functions towards an understanding in terms of *data* and *algorithms*.

[3]Mathematically, a chaotic system is defined as an iterated map that is sensitive to its initial conditions [Hilborn, 1993, Mandelbrot, 1988].

1.1 Generalized Feedback Control

Generalized Feedback Control tries to generalize certain concepts of control towards its original meaning as mentioned in the first section. The first, central hypothesis in this approach is that the observation of the plant serves basically two purposes as can be seen in figure 1.2:

Motivation Matching: The control system has to relate the system's state to its motivations (i.e. nominal values or design specifications). It has to have an idea of how the plant could transit into a state that is consistent with its motivations.

Circumstance Cognition: The mapping from control parameters to the plant's output depends on its current state. In order to set the appropriate control parameters, the current state of the plant (i.e. the 'circumstances') has to be known.

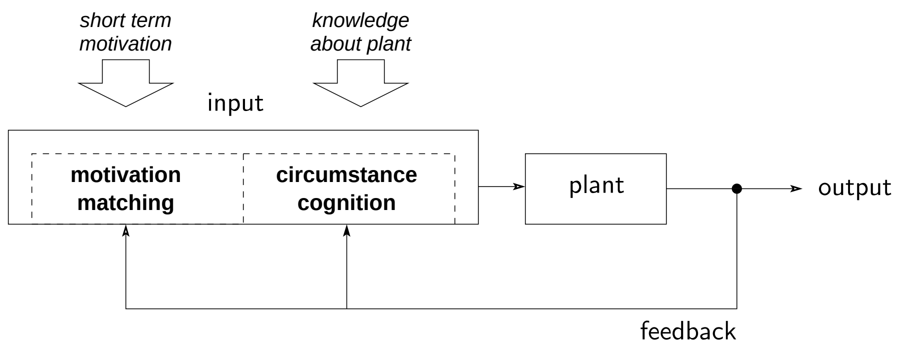


Figure 1.2: Two types of feedback in Generalized Feedback Control: motivation matching and circumstance cognition.

The second hypothesis is that these two types of feedback can be treated in two separate units inside the control system. The third hypothesis is that they are sequential. The forth hypothesis is that whenever a control signal is computed both types of feedback are processed.

The input to this time discrete controller is a *short term motivation*. In vehicle control the short term motivation can be, for example, to drive as close as possible to the nominal course, to reduce the angular distance to the nominal course and not to exceed a maximum lateral acceleration. This input directly includes what would be called the *design specifications* in classical control systems [Graham and Lathrop, 1953,Close and Frederick, 1993]. The output of the plant can be measured by any means: state parameters of the plant, trajectories, velocity profiles etc.

The present dissertation uses the idea of two types of feedback processing to develop an algorithmic control structure based on two basic units for motivation matching and circumstance cognition. The application of this structure in vehicle control emphasizes its abilities as an automated control system.

1.2 Overview and Outline

The chapters in this dissertation are lined up in a way, so that they reflect the process of building a controller based on Generalized Feedback Control. The flow of discussion is as follows:

1. A definition of the nominal behavior for one control interval has to be defined. Therefore, section 2 introduces the concept of a *gd-plan*. This concept is essential for the separation of the controller into two units: *plan-to-action mapper* and *gd-plan construction unit*.

2. The plan-to-action mapper is identified as the unit for circumstance cognition. It has the task to find appropriate control parameters to achieve a given nominal behavior in a given state of the plant. The development and application of a plan-to-action mapper is discussed in chapter 3.

3. A mathematical description for the short term motivation has to be found, before one can discuss the interaction of motivation with the observed plant's state. A methodology to quantitatively describe motivation based on utility functions is introduced in chapter 4, called the method of *target and limit maps*.

4. Being able to describe motivation in mathematical terms, chapter 5 describes the module handling the interaction between the short term motivation and the information perceived about the plant, i.e. motivation matching. The result of this interaction is the nominal behavior for the next control interval, the gd-plan. Therefore, this module is called the gd-plan construction unit.

5. When the gd-plan construction unit is build the controller is able to function. Its reliability, however, cannot be investigated by means of traditional *stability* measures, due to its rather algorithmic nature. Chapter 7 introduces a reliability criteria based on the idea of not leaving an *admissible domain of system states*: the criteria of *containability*.

As practical reference in these chapters, the control of an automatically guided vehicle offers a non-trivial problem. Step by step, it is described how to build a path following system using the derived concepts. During the discussion of step four, the performance of this controller with respect to *precision* and *speed* can be observed.

The review of the historical background opens the appendices. Further, the appendices describe the vehicle model being used for investigations and the details of several mathematical derivations.

Chapter 2

The Geometric-Dynamic Plan

The basic assumption of General Feedback Control is that the feedback from the plant can be divided into two categories (section 1.1). The first type of feedback is used for *circumstance cognition* in order to specify the exact mapping from control parameters to the plant's output for a given state. The second type of feedback is used for *motivation matching*, i.e. to relate the current state of the plant to a short term motivation.

Before control parameters can be determined, it must be specified what behavior of the plant is to be achieved. A nominal behavior, though, can be determined based on the relation between the short term motivation and the current state of the plant. A description of the nominal behavior is therefore a good candidate as an interface between the unit for *circumstance cognition* and *motivation matching*. This is the central role of the 'gd-plan', the geometric-dynamic plan, indicating the nominal trajectories and velocity profiles of the plant's motion for the subsequent control interval. It is by this interface that it is possible to treat *circumstance cognition* and *motivation matching* in two separate, sequential units as shown in section 2.1.

The following sections discuss what a gd-plan represents and how it has to be specified for a particular problem. The term 'episodic environment', as discussed in section 2.2 allows one to relate the gd-plan to a global design specification or control goal. In section 2.3 it is discussed what requirements apply on a set of physical parameters in order to represent a gd-plan.

2.1 Motivation Matching and Circumstance Cognition

The gd-plan defines the interface between the two entities for *circumstance cognition* and *motivation matching*. With this gd-plan the input and output of two units can be distinctly defined as shown in figure 2.1. One unit has the task to match the plant's state with the short term motivation. It has to elaborate an idea how the plant's state could transit into a state more conform with its motivation, i.e. it has to compute a gd-plan p_{gd}. This nominal behavior for the following control interval is the input to the unit that finds the appropriate control parameters c_p to accomplish the gd-plan correspondent to the current state s of the plant. Congruously, the two entities called *gd-plan construction unit* and *plan-to-action mapper* are defined as follows:

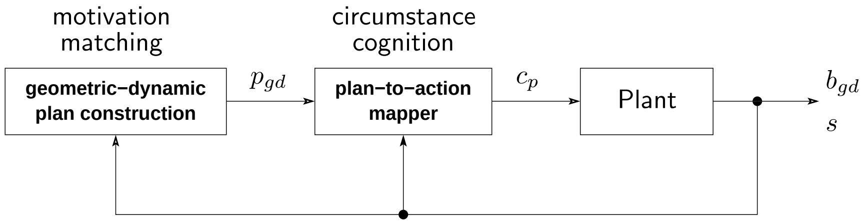


Figure 2.1: The concept of a gd-plan p_{gd} allows the division of the control process into gd-plan construction and plan-to-action mapping.

Geometric-dynamic plan construction: This unit uses information about the plant's state in order to *relate it to its short term motivation.* As a result it computes short term plans for the system's state transition over the subsequent control interval. A geometric-dynamic planning unit performs the mapping $\mathbf{G}(s)$

$$\mathbf{G}: \; s \quad \longrightarrow \quad p_{gd}. \tag{2.1}$$

where s is the plant's state and p_{gd} the plan for the system's state transition over the subsequent control interval, i.e. the 'geometric-dynamic plan', or 'gd-plan'.

Plan-to-action mapping: A plan-to-action mapper has the task to compute control parameters so that the plant performs the gd-plan p_{gd} during the following control interval. Information about the plant's state is necessary *to specify the mapping from the gd-plan p_{gd} to the control parameters c_p.* A plan-to-action mapper $\mathbf{P}(s,\, p_{gd})$ performs the mapping

$$\mathbf{P}: \; (s,\, p_{gd}) \quad \longrightarrow \quad c_p. \tag{2.2}$$

where c_p are the control parameters to be sent as input to the plant. The function $\mathbf{P}(s,\, p_{gd})$ can be considered to incorporate *knowledge* about the plant.

Correspondingly to the two concepts of feedback, *motivation matching* and *circumstance cognition*, the two control units incorporate concepts of *motivation* and *knowledge* about the system to be controlled. In the following subsection it is discussed how the concept of *compensation* in classical control fits into the scheme of Generalized Feedback Control.

2.1.1 Compensation

The main idea behind the summation point in classical control is *compensation*. By this means, the control inputs are reinforced or attenuated correspondent to the difference between nominal and achieved values. If a front wheel is not inclined enough to achieve a certain curvature, it is intuitive to incline it more correspondent to the deviation from the nominal curvature. Through this mechanism the controller compensates for its inability to precisely match the plant's characteristics. Let the compensation process be represented by the equation

$$i_2 = i_1 + c\,(o_n - o_1), \tag{2.3}$$

where i_1 is a first 'guess' of a control input to achieve the nominal output o_n. The output o_1 is the output actually achieved by i_1. The difference between nominal and achieved value $(o_n - o_1)$ is then multiplied by a compensation factor c in order to reinforce or attenuate the control input. This process is shown in figure 2.2.

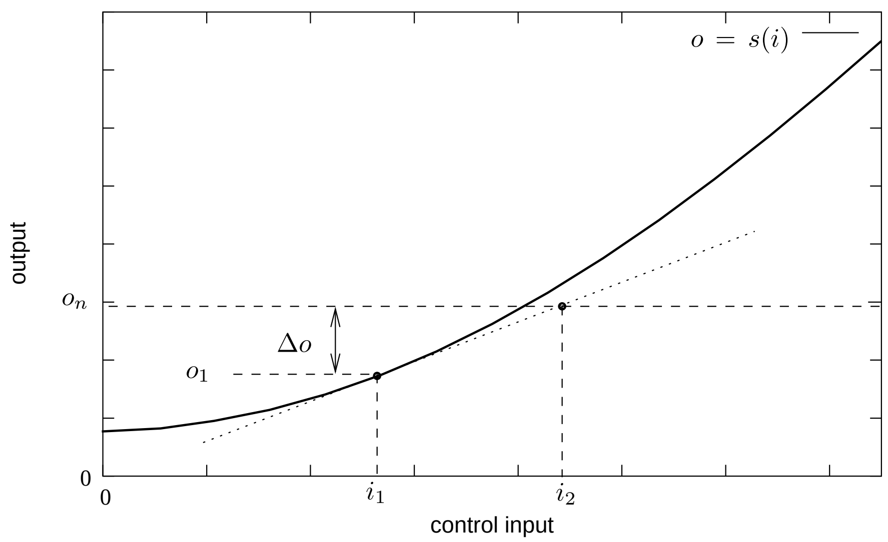


Figure 2.2: Compensation in the classical sense: trying to achieve a nominal value o_n by following the derivative.

Now let the plan-to-action mapper

$$\mathbf{P}(o_n) \;=\; P_0(o_n) + c_x\, o_n + d_x. \tag{2.4}$$

be the function that computes control input i_1 in order to achieve the nominal value o_n, where $P_0(o_n)$ is an arbitrary function that does the mapping. c_x and d_x are two coefficients that exist but are zero for now. Thus, the first control parameter i_1 was produced by the function $P_0(o_n)$ only. In order to reduce the error, the subsequent 'guess' is made adding the compensation term as in (2.3)

$$i_2 \;=\; i_1 + c(o_n - o_1) \;=\; P_0(o_n) + c\, o_n - c\, o_1 \tag{2.5}$$

If i_2 is to be computed by $\mathbf{P}(o_n)$, then the coefficients c_x and d_x have to be adapted to

$$c_x \;=\; c, \text{and} \quad d_x \;=\; -c\, o_1. \tag{2.6}$$

Accordingly, compensation in the classical sense performs a *parameter modification* in the sense of Generalized Feedback Control. One type of modifications has already been mentioned in Generalized Feedback Control: the circumstance cognition. For circumstance cognition the mapping from the gd-plan to control parameters is adapted due to the current state of the plant. This relates somehow to the adaption to an operating point in classical control.

The compensation discussed above, however, includes a modification of parameters due to an *insufficient* or *erroneous* description about the plant. It appears, for example, if the load transfer on the axles of a vehicle was neglected, or the frictional coefficient of a tire is not correct. For the sake of clarity, let this type of compensation be called *parameter adaption* or *learning* in the frame of Generalized Feedback Control.

Parameter adaption requires a memory or a database inside the controller so errors can be identified and corrected. As shown in figure 2.3 the gd-plan p_{gd}, the control parameters c_p and the outcome s and b_{gd} have to be stored in a database in order to be able to adapt the parameters of the plan-to-action mapper. As can be seen in the figure parameter adaption solely applies on plan-to-action mapping.

Based on the difference between the desired and the actual geometric-dynamic behaviors p_{gd} and b_{gd}, conclusions can be drawn on the need of adaption. Relating c_p to b_{gd} allows to conclude on the parameter settings that have to be made to reflect the plant's system function. Where compensation in the sense of *circumstance cognition* is extensively discussed in this dissertation, the subject of *parameter adaption* or *learning* is left for further research.

2.2 The Episodic Environment

The plant's state can be interpreted in terms of its geometric-dynamic behavior b_{gd}. Using b_{gd} the task of a plan-to-action mapper can be defined as to minimize the error between b_{gd} and and the desired geometric-dynamic plan p_{gd}.

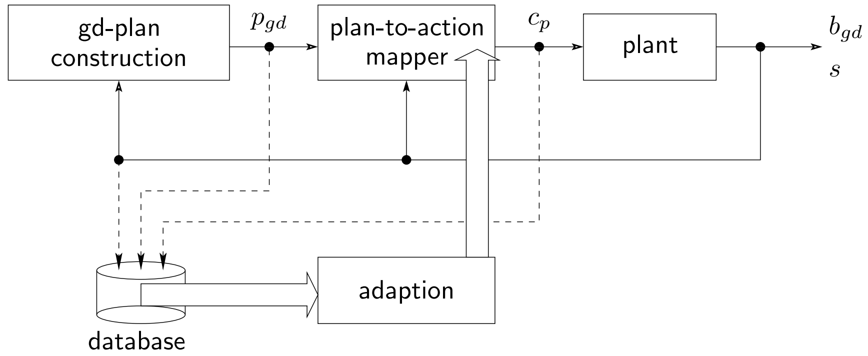


Figure 2.3: Generalized Feedback Control with adaption.

However, a task description for a gd-plan construction unit requires more in depth considerations. Before one can talk about motivation matching, the term motivation has to be defined.

Definition: 1 (Motivation) *In the frame of this dissertation,* MOTIVATION *is understood as a spur to action. A motivation assigns preferences to possible changes of the environment state (e.g. specified as a gd-plan). The preferences provided by Motivations are the bases for decision making.*

Control problems are usually specified through global design specifications on the time response of the system [Graham and Lathrop, 1953, Dorf, 1988]. Roughly speaking, the gd-plan construction unit has the task to develop a nominal behavior so that the plant's state develops as much as possible towards the global control goals. An essential facilitation towards the determination of a short term nominal behavior, i.e. a gd-plan, conform to long term control goals is the concept of the *episodic environment*[1]. In an episodic environment, long term goals can be projected onto isolated moments in time resulting in episodic goals.

Example: Vehicle Control.

Global Goal: Driving a vehicle as fast and as precisely as possible around a nominal course, i.e. continuous parameterized graph $\vec{n}(p)$.

Using this goal definition as a reference for one single moment in time where control parameters have to be computed is extremely difficult. When choosing a target point too close ahead on the nominal course, it is crossed with an angle too high, and the vehicle might start oscillating. With a faulty choice of acceleration the lateral acceleration may get too

[1]A definition of episodic environment can be found in [Russel and Norvig, 1995, section 2.4].

high and the control system looses control over the vehicle. Many other physical phenomena have to be handled at the same time. This is very difficult as long as one always has to refer to the global goal.

Episodic Goals: The above control goal, however, can be reformulated for a particular moment where control parameters can be set. It yields the following definition of a situational driving motivation, as introduced in this dissertation:

Definition: 2 (SDM) *Given a nominal course, for a particular isolated moment the* Situational Driving Motivation SDM *can be expressed by four concurrently active goals:*

- *Minimize lateral deviation* Δd.
- *Minimize deviation* $\Delta\vartheta$ *between the angle of the velocity in the center of gravity (c.g.) of the vehicle and the angle of the nominal course.*
- *Minimize lateral acceleration* $\ddot{y}(t)$.
- *Maximize velocity* v.

These goals may have different priorities depending on the specific situation.

This particular redefinition of the driving task is carried through the whole dissertation. Figure 2.4 illustrates the idea of the SDM[2]. Minimizing Δd targets nearness to the nominal course. Aiming to reduce the difference angle $\Delta\vartheta$ between the tangent of the trajectory and the angle of the nominal course ensures that the car does not diverge from the course. Minimizing the lateral acceleration results in smooth short term paths. Practically, it restricts the amount of 'sensitivity' of the vehicle's input/output behavior, which facilitates the control task. On the other hand, it is advantageous to optimize velocity.

The situational driving motivation facilitates the task to define parameters of a gd-plan, since it makes concrete assumptions about state variables in a single isolated moment (displacement Δd, lateral acceleration $\ddot{y}$, etc.). It leads towards a trajectory and velocity profile design defining the nominal behavior of the plant for the following control interval.

In summary, it can be said that the reformulation of global control goals in terms of episodic goals has two advantages. First of all, it allows to concretize a *short term motivation* as a basis to build a gd-plan construction unit. Second, it leads towards a parameterization of the nominal behavior, the gd-plan.

[2] The four goals are considered to be elementary pillars of a driving motivation. They may result in different secondary goals dependent on the priority of each one of them. In racing, for example, the goal to optimize velocity has a high priority. Therefore, one tries to minimize the lateral acceleration only as much as to stay under the maximum lateral acceleration $\ddot{y}_{max}$ that the driver can handle. This way, the curvature profile of the nominal course $\kappa_{nc}(s)$ together with the maximum lateral acceleration $\ddot{y}_{max}$ result in a velocity profile $v_{nc}(s) = \sqrt{\ddot{y}_{max}/\kappa_{nc}(s)}$. The secondary goal derived for each particular situation is therefore: 'Keep velocity under velocity profile'.

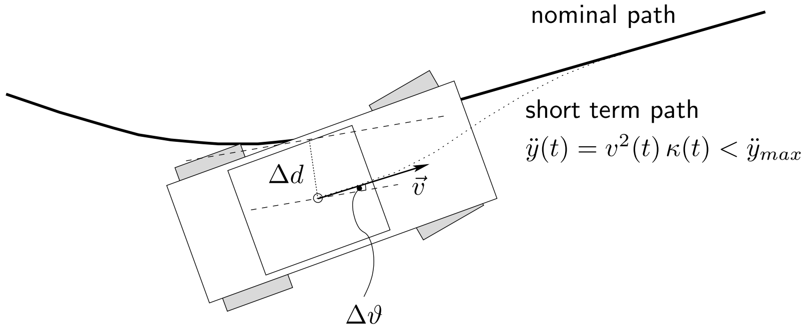


Figure 2.4: Practical example of an episodic environment: the situational driving motivation SDM.

2.3 Definition of a GD-Plan

The purpose of a gd-plan is to specify the nominal behavior of system components for one single control interval. Since, during the control interval no further information can be passed from the gd-plan construction unit to the plan-to-action mapper, physical parameters have to be chosen that are *close to constant*. In general terms, a parameter $\tilde{\gamma}$ can only be part of a gd-plan, if it can be optimized so that

$$dist\left(\Gamma_{[0,T_c]}, \tilde{\Gamma}_{[0,T_c]}\right) \quad < \quad \epsilon_\gamma, \tag{2.7}$$

where $dist(a,b)$ is a user defined distance measure for two graphs a and b. $\Gamma_{[0,T_c]}$ is the graph of the 'real' $\gamma(t)$ over the control interval of length T_c. $\tilde{\Gamma}_{[0,T_c]}$ is the graph over the control interval with a constant $\tilde{\gamma}$. ϵ_γ is a boundary value that is appropriate for the particular problem. Second, the set of gd-plan parameters should *distinctly define* all possible motions of the system for one control interval. Using these two conditions a gd-plan can be defined as follows[3].

Definition: 3 (GD-Plan) *A* GD-PLAN *is a set of parameters describing the nominal behavior of a plant for one single control interval. The set of parameters has to conform two conditions:*

- *Each one of the parameters in the* GD-PLAN *is sufficiently constant over the time span of one control interval.*

[3]It is advantageous for a gd-plan to be as concise as possible, i.e. to describe the nominal behavior with the fewest amount of parameters. In chapter 4 a methodology for gd-plan construction is introduced that works best, when parameters for the geometry of the trajectory are independent from parameters for the velocity profile. At this point these two requirements, though, cannot be part of the general definition of a gd-plan.

- *The whole set of parameters distinctly describes the motion of all trajectories and velocity profiles of system components to be controlled.*

The following paragraphs show that a rate of curvature change κ' and and acceleration $\dot{v}$ are suitable candidates to build a gd-plan in to control a vehicle.

Example Vehicle Control.

The motion of a ground vehicle under normal conditions can be considered as a motion in a two dimensional plane. The center of gravity moves along a trajectory, with a continuous curvature profile[4]. At this point it is assumed, though, that the controller can influence the derivative of the curvature over travel distance $\kappa' = d/ds\,\kappa$. Based on the instantaneous curvature κ_0 and the rate of curvature change κ' a trajectory can be determined (see section 5.3). A possible distance measure for equation (2.7) is based on the position of the vehicle at the end of the following control interval:

$$dist\left(\{\kappa'(t)\}_{t\in[0,T_c]}, \{\kappa'\}_{t\in[0,T_c]}\right) \equiv \tag{2.8}$$
$$|\,\vec{\tau}(\kappa(t),\, T_c) - \vec{\tau}(\kappa'\, t,\, T_c)\,|, \tag{2.9}$$

where $\vec{\tau}(f(t),\, t_{end})$ delivers a vector indicating the position in space that is reached at time t_{end} with a curvature profile $\kappa(t)$. The particular value of this function depends on the specific vehicle being used. Experience in observing vehicle trajectories, though, makes it plausible that the displacement between the two graphs are negligible for short control intervals T_c (i.e. in the range of 0 to 0.5 seconds). Therefore, κ' is a good candidate to determine the geometric trajectory for short time periods. Considering the relatively slow changes in velocity during driving, a similar argumentation leads to the longitudinal acceleration $\dot{v}$ as a suitable parameter to describe the velocity profile. The nominal motion of the vehicle for one single control interval can therefore be defined as:

$$p_{gd} \equiv (\kappa',\, \dot{v}). \tag{2.10}$$

Note, that the set of parameters in the gd-plan spans a multi-dimensional space $\mathcal{P}_{gd}$ of possible gd-plans[5]. Physical aspects can directly be expressed using the coordinates of this space. Consider a restriction of lateral acceleration on the short term trajectory. As shown in a later section (equations (5.43) to (5.45), page 80) the lateral acceleration with respect to time, can be computed as

$$\ddot{y}(t) = (v + \dot{v}\,t)^2 \left(\kappa_0 + \kappa'_0 \left(v + \frac{1}{2}\,\dot{v}\,T_c\right)\, t\right). \tag{2.11}$$

[4]As long as there is no impact with an obstacle in the environment.

[5]This may be reminiscent the signal space $\mathcal{W}$ in the Willem's behavioral approach [Willems and Polderman, 1997]. The space of possible gd-plans, here, corresponds the the space of admissible trajectories, i.e. behavior $\mathcal{B}$, in the Willem's terminology. In the frame of this dissertation, though, no considerations are made concerning the physical feasibility of the whole space $\mathcal{P}_{gd}$.

The restriction of a maximum lateral acceleration $\ddot{y}_{max}$ at the end of the control interval T_c leads to a relation between κ' and $\dot{v}$

$$\kappa'_{min,\,max}(\dot{v}) \quad = \quad \frac{\pm\,\ddot{y}_{max}}{(v+\dot{v}\,T_c)^2\,T_c} - \frac{\kappa}{T_c}. \tag{2.12}$$

By this equation a set of admissible gd-plans [6] can be defined where for a given $\dot{v}$ the boundaries of κ' are given by $\kappa'_{min}(\dot{v})$ and $\kappa'_{max}(\dot{v})$. The representation of a gd-plan in a multi-dimensional space facilitates therefore the consideration of physical phenomena. However, a particular representation of a gd-plan defined under the constraints in definition (3) may not be suitable for some specific views on the problem.

2.4 Interpretation of a GD-Plan

A gd-plan was defined as a set of parameters that describe the trajectory and velocity profiles with as few parameters as possible. In this section, it is discussed how a gd-plan can be re-interpreted so that its description fits more investigations required for a particular problem.

In the previous section, a gd-plan for vehicle control was determined in terms of rate of curvature change κ' and acceleration $\dot{v}$. An important concept in vehicle control, though, is that of a target point, i.e. a point to be driven trough after a certain amount of time. The target point has then to be related to a rate of curvature change κ' as can be observed in figure 2.5. GD-Plan Construction depends on the search for an appropriate target point $(t_x,\, t_y)$. Thus, a relation between the gd-plan $(\kappa',\, \dot{v})$ and the coordinates t_x and t_y is required[7]. Let a general interpretation of a gd-plan be defined the following way:

Definition: 4 (Interpretation of a GD-Plan) *An* INTERPRETATION OF A GD-PLAN i_{gd} *is distinctly related to a given gd-plan* p_{gd}*, i.e. it is based on a function*

$$f: \quad \mathcal{P}_{gd} \quad \longrightarrow \quad \mathcal{I}_{gd} \tag{2.13}$$

where $\mathcal{I}_{gd}$ *is the space of possible values for* i_{gd}*.* i_{gd} *incorporates some aspects of the gd-plan in a manner more suited for a particular problem.*

Interpretations of gd-plans are very useful in many places. During gd-plan construction different physical phenomena require different representations of the gd-plan (e.g. a target point, target velocities, etc.). Since the output of the gd-plan construction unit must be a gd-plan parameterized in its original

[6] See figure 4.3, page 41 for an example plot of the set of admissible gd-plans.

[7] Section E discusses how to determine a point on the trajectory with a constant rate of curvature change, and vice verse, how to determine a rate of curvature change for a given target point.

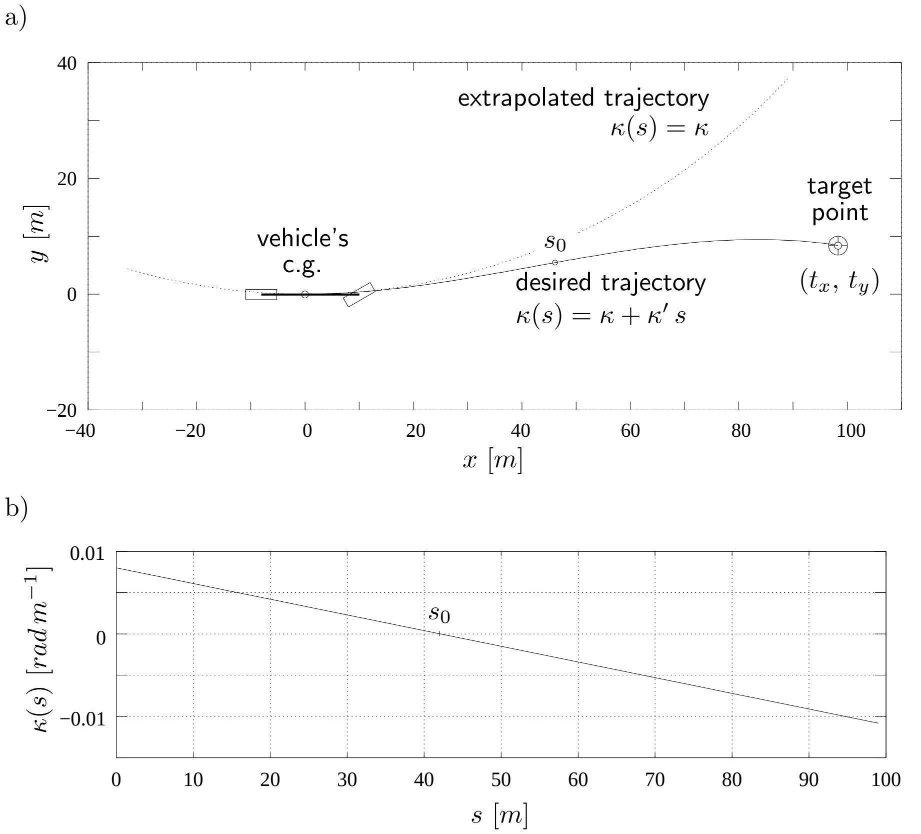


Figure 2.5: Interpretation of a gd-plan: A target point relating to a constant rate of curvature change κ'. a) x- and y-coordinates. b) curvature profile.

way one has to require that $f(p_{gd})$ is 'invertible'[8]. Another application for interpretations of gd-plans is the comparison between a desired nominal behavior p_{gd} and a actually shown behavior of the plant b_{gd}. For vehicle control it is more intuitive to consider lateral displacements, for example, rather than errors in rate of curvature change. In the latter type of application no injectivity is required for $f(p_{gd})$.

2.5 Conclusion

This chapter introduced the concept of a gd-plan as a nominal behavior of the plant for one control interval. It was demonstrated how this concept allows to separate *circumstance cognition* from *motivation matching* into two separate sequential units, supporting hypothesis two and three in section 1.1. It was further shown that *compensation* in the classical sense can be understood as a combination of *circumstance cognition* and *adaption*.

The notion of an episodic environment facilitates the concretization of a gd-plan through a set of parameters, and at the same time leads to a mathematical formulation of a short term motivation. In section 2.3 constraints on a set of parameters where defined, in order to constitute a gd-plan. Finally, it was discussed how a gd-plan can be re-interpreted to fit the purpose of a specific view on the control problem.

The main purpose of this chapter was to demonstrate the plausibility to construct a Generalized Feedback Control system based on the two units gd-plan construction and plan-to-action mapping. Having also clarified the concept of a gd-plan, this chapter prepared the discussion of how to map from a gd-plan to control parameters as an input to the plant. In this sense, the following chapter discusses the construction of a plan-to-action mapper.

[8] See section 3.1 for an explanation of the concept of 'invertibility' without requiring injectivity.

Chapter 3

The Plan-to-Action Mapper

A plan-to-action mapper has to find appropriate control parameters c_p as input to the plant so that it performs a given gd-plan p_{gd} for a given state of the plant $s(t)$. The consideration of the plant's state to achieve this is called circumstance cognition. In optimal control [Kalman, 1960] and model predictive control [Rawlings et al., 1994] it is strived for an optimal input function to achieve a desired output over some fixed time interval $[t,\, t + T_c]$. The plan-to-action mapper, however, only computes control inputs which are *constant* for the length of one control interval. The inflexibility of the rigid shapes of the resulting time profiles can be compensated by using derivatives of control inputs or using control impulse template profiles that are amplified by the plan-to-action mapper's output. The essential idea for Generalized Feedback Control, though, is that a minimum of control happens without the involvement of geometric-dynamic planning. The *constant* control inputs, therefore, correspond to the *constant* terms in the gd-plan (section 2.3). Figure 3.1 illustrates the problem of plan-to-action mapping.

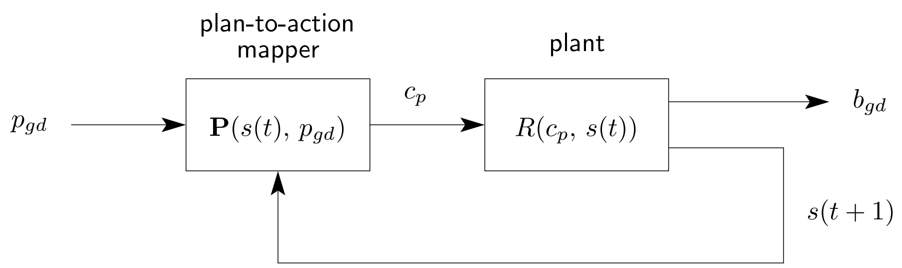


Figure 3.1: The problem of plan-to-action mapping.

A plant's behavior can be interpreted in geometric-dynamic terms b_{gd}, i.e. trajectories and velocity profiles of system components. Additionally, it's state is identified through a set of state indices $s(t + 1)$, where $t + 1$ indicates the end of the subsequent control interval. Let $R(c_p,\, s(t))$ be the response of the plant on control parameters c_p when applied in a given initial state $s(t)$ over one single control interval.

Constructing a plan-to-action mapper means finding a mapping $\mathbf{P}(s,\, p_{gd})$ so that the difference between b_{gd} and p_{gd} becomes minimal, i.e.

$$p_{gd} - R(c_p,\, s(t)) \quad = \quad min. \tag{3.1}$$

$$p_{gd} - R(\mathbf{P}(s,\, p_{gd}), s(t)) \quad = \quad min. \tag{3.2}$$

Using operator notation for a common $s(t)$, i.e. defining $R\, c_p \;\equiv\; R(c_p,\, s(t))$ and $\mathbf{P}\, p_{gd} \;\equiv\; \mathbf{P}(p_{gd},\, s(t))$, results in

$$p_{gd} - R\,\mathbf{P}\, p_{gd} \quad = \quad min. \tag{3.3}$$

it becomes clear that (3.1) is best fulfilled when $R\,\mathbf{P}$ becomes equal to identity. Therefore, the plan-to-action mapper $\mathbf{P}$ has to be close to the inverse of the system response function R^{-1}.

In most cases, however, the equations that describe the plant's behavior do not lead to a closed analytical formula for R^{-1}. In order to acquire, nevertheless, a mathematical description for R^{-1} and its dependency on the initial state, section 3.1 discusses an empirical method based on a state-space equation of the plant.

In parallel to the theoretical discussion a plan-to-action mapper for a vehicle control system is developed. A vehicle model with three degrees of freedom performing a planar motion serves as a plant for the controller. It is controlled by two parameters:

- $\dot{\delta}$: a front wheel angle velocity that is distributed to the two front tires.
- F_p: a propulsive force that is applied on the rear tires.

The geometric-dynamic plan for the subsequent control interval specifies a short term trajectory and a velocity profile specified through

- κ': a rate of curvature change of the vehicle's trajectory. This specifies the geometric curve to be driven.
- $\dot{v}$: an acceleration specifying the velocity profile.

The vehicle control problem again serves as a experimentation platform. Section 3.1 explains how to build a plan-to-action mapper with the derived formula from section 3.1. It discusses in detail how to actually build a plan-to-action mapper module for a vehicle controller.

3.1 Theoretical Considerations

The concern of the following sections is to find an expression for the inverse system response $R^{-1}(b_{gd},\, s)$ that determines appropriate control parameters

c_p to fulfill a desired gd-plan p_{gd} for a given state of the plant s. For simple systems it may be possible to find a closed mathematical expression. In the general case, however, closed analytical solutions cannot be achieved without simplifications of the system's equations. The more degrees of freedom, or number of equations, are involved in the description of the plant, the more likely it is that suitable simplifications neglects basic characteristics of the system. The following sections introduce an approach that circumvents the complications of solving for a closed mathematical formula. A systematical method is presented based on hierarchical curve fitting of empirical data.

In a first step, for fixed initial states s_0 of the plant the relationship between control parameters c_p and geometric-dynamic behavior b_{gd} is investigated. Using curve fitting, template functions are adapted so that they approach as close as possible the shape of the sampled data points. For a particular initial state the relationship between control parameters and geometric-dynamic behavior is then distinctly defined through the set of parameters $\underline{p}^*$ of the adapted template functions as shown in figure 3.2a.

In a second step, it is investigated how the parameters $\underline{p}^*$ of the template function change with respect to a given initial state s_0. Again by means of curve fitting techniques, template functions are adapted and the relationship is described through a set of parameters $\underline{\tilde{p}}$ as shown in figure 3.2b.

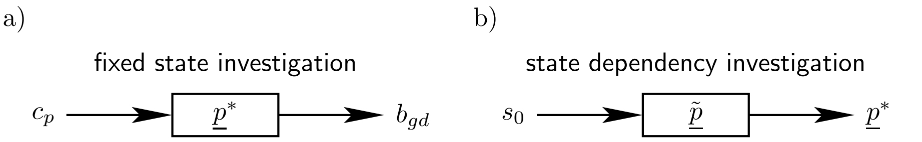


Figure 3.2: Two steps in the process of building a plan-to-action mapper: a) investigating the behavior for a fixed initial state of the plant. b) investigating the change of the plant's behavior with respect to the initial state.

With the parameter set $\underline{\tilde{p}}$ and the appropriate template functions control parameters for a specific gd-plan can be found for any given state. This process of plan-to-action mapping is displayed in figure 3.3. Using the set of parameters $\underline{\tilde{p}}$, the state s is used to compute the parameters $\underline{p}^*$. These parameters define the relation between control parameters and geometric-dynamic behavior of the plant for the specific state s. Now, the appropriate control parameters c_p can be computed for the specific gd-plan p_{gd}.

The empirical data is obtained through massive simulation experiments sampling the plant's system response function

$$R : \quad (c_p,\, s) \quad \longrightarrow \quad b_{gd}, \tag{3.4}$$

where c_p are the control parameters sent as input to the plant, s the initial system state, and b_{gd} the geometric-dynamic description of the behavior of the plant during the control interval. Operator notation for a fixed initial state s_0 facilitates the discussion, e.g.

$$R^* : \quad c_p \quad \longrightarrow \quad b_{gd}. \tag{3.5}$$

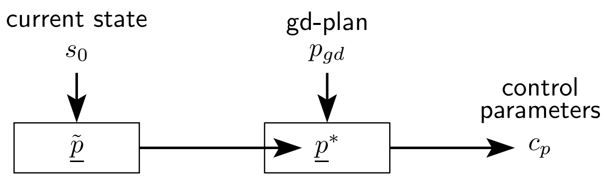


Figure 3.3: Plan-to-action mapping in two steps.

is equivalent to $R(c_p,\, s_0)$. Let all operators and variables with a star as in R^* be initial state specific.

In the example of vehicle control, $R_v^*(c_p)$ is the mapping from an applied front wheel angle velocity $\dot{\delta}$ and a propulsive force F_p to the achieved average rate of curvature change κ' and the average acceleration $\dot{v}$, i.e.

$$R_v^* : \quad \left(\dot{\delta},\, F_p\right) \quad \longrightarrow \quad (\kappa',\, \dot{v})\,. \tag{3.6}$$

As mentioned earlier, the construction of a plan-to-action mapper requires to search for a representation of the inverse system response, i.e. a mapping from a geometric-dynamic plan p_{gd} to control parameters c_p. The inverse system response, as it is used in this dissertation does not require the system response R^* to be injective. However, R^{*-1} determines for a given geometric-dynamic behavior b_{gd} in a given range the appropriate control parameters c_p. This idea is demonstrated in figure 3.4 where the system function $s(i)$ is not injective (i_1, i_2, and i_3 produce the same output o). But, an inverse $s_i(o)$ finds a distinct i for a desired output o. This dissertation proposes a systematic approach to

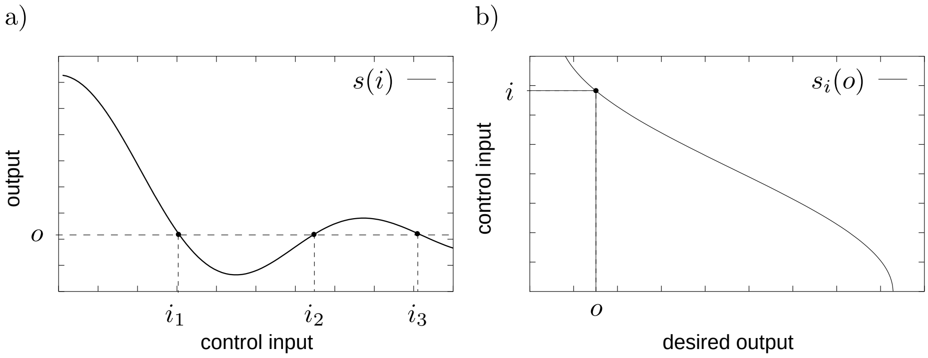


Figure 3.4: Concept of a system function. Injectivity of $s(o)$ is not required to determine a 'inverse' $s_i(o)$ that finds for a desired output o a required input i.

acquire $R^{-1}(b_{gd},\, s)$ separated into three steps:

Fixed initial state investigations: as described in section 3.1.1. In this step the response behavior $R^{*-1}\,b_{gd} = c_p$ for one particular initial state s_0 and changing c_p is observed.

Initial state dependency investigations: as described in section 3.1.2. This section tries to find a description of how R^{*-1} changes with respect to the initial state s. At this point the function $R^{-1}(p_{gd},\, s)$ is fully determined.

Identification of physical relationships: In this step the relationships between physical parameters of the plant to $R^{-1}(p_{gd},\, s)$ have to be investigated. This topic, though, is not covered by this dissertation.

It is important to note that the gd-plan p_{gd} as well as the control inputs c_p have to be described through parameters that can be considered to be *constant* for the duration of one control interval. Intuitively, replacing them with their average values should only cause negligible errors. In the vehicle control example, it is assumed that the front wheel angle velocity $\dot{\delta}$, the propulsive force F_p, the rate of curvature change κ', and the acceleration $\dot{v}$ to be constant for the time period of one control interval. Due to the dynamics of the system, the acceleration $\dot{v}$, for example, is not exactly a constant term during the control interval. However, since velocity changes relatively slow a constant acceleration describes the system behavior sufficiently precise.

3.1.1 Fixed Initial State Investigation

The first step towards a formula for a plan-to-action mapper is to investigate the plant's input/output behavior in a particular initial state s_0. Starting from this state a set of control inputs c_p is applied for the time of one control interval and the correspondent geometric-dynamic behaviors b_{gd} are measured. The results are stored in a table

$$\mathcal{L}^*[b_{gd} : c_p] \quad \equiv \quad Sampling(R^*,\, c_p). \tag{3.7}$$

containing samples of the relation between geometric-dynamic behaviors $b_{gd} = R^*\,c_p$ for different settings of control parameters c_p when applied in a specific initial state s_0.

Figure 3.5 shows the geometric-dynamic response $R^*\,c_p$ of a vehicle when exposed to a front wheel angle velocity $\dot{\delta}$ and a propulsive force F_p for a certain amount of time (here 0.5 *sec*). When building a database a whole grid of possible inputs $\{\dot{\delta}_i,\, F_{p,k}\}_{i,k}$ has to be applied to the system and its output measured in terms of average rate of curvature change κ' and acceleration $\dot{v}$. An example of a table resulting from such simulation experiments can be viewed in figure 3.6.

Before one can apply curve fitting techniques, scalar template functions must be provided. In case that $R^{*-1}\,b_{gd}$ is multi-dimensional, i.e. if c_p consists of more than one control parameter, then $R^{*-1}\,b_{gd}$ has to be broken up into a set

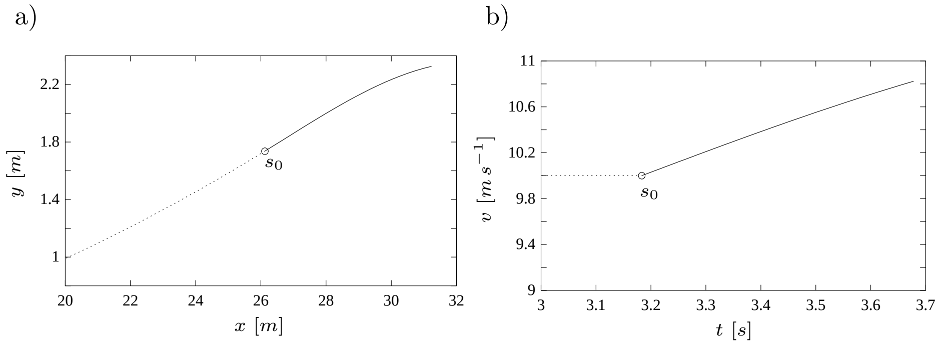


Figure 3.5: Geometric-dynamic response of a vehicle being exposed to control inputs starting from the initial state s_0. a) resulting trajectory of the vehicle's c.g. b) velocity profile of the vehicle.

control inputs c_p		gd-behavior b_{gd}	
$\dot{\delta}$ [rad/s]	F_p [N]	$\dot{v}$ [m/s]	κ' [rad/m^2]
2.6179	−5000	−3.928	0.000353
2.6179	−4000	−3.6275	0.000278
2.6179	−3000	−3.322	0.000121
2.6179	−2000	−3.0291	0.000049
2.6179	−1000	−2.716	0.000012
⋮	⋮	⋮	⋮

Figure 3.6: Table recording relationship between control inputs $c_p = (\dot{\delta},\, F_p)$ and the plant's geometric-dynamic response $b_{gd} = (\kappa',\, \dot{v})$.

of scalar functions:

$$R^{*-1}\, b_{gd} \quad \equiv \quad \begin{cases} R_0^{*-1}\, \beta_0 \\ R_1^{*-1}\, \beta_1 \\ \vdots \\ R_N^{*-1}\, \beta_N \end{cases} . \tag{3.8}$$

where β_i is the 'key' number i for a specific input/output configuration of the plant. It is called 'key' because it opens the possibility to compute a factor. This factor can then be used to build subsequent keys, and so on, until all control parameters are computed.

A key β_i can consist of any function of the geometric-dynamic behavior b_{gd} and the control parameters c_p excluding the control parameter number i that is to be computed. However, the first function R_0^{*-1} in the sequence needs to work with the original b_{gd}, i.e. $\beta_0 = b_{gd}$ since this is the original input to the plan-to-action mapper. Later functions can be based on inputs β_i, $i = 1 \ldots N$ provided that they appear above in the sequence.

In order to capture vehicle behavior, it is advantageous[1] to use the front wheel angle velocity $\dot{\delta}$ to compute the propulsive force F_p for a desired acceleration $\dot{v}$. The physical reason behind this is the decelerating friction force of the inclined front wheels. However, before it can be applied, the front wheel angle velocity for the next control interval has to be computed based on the desired rate of curvature change κ' and acceleration $\dot{v}$. Hence, the propulsive force F_p can only be computed after the front wheel angle velocity $\dot{\delta}$. Therefore,

$$\beta_{v,0} \quad \equiv \quad (\kappa',\, \dot{v}), \tag{3.9}$$

is the first key. It computes the factor $\dot{\delta}$ which can now be used in a key $\beta_{v,1}$ to compute the propulsive force.

$$\beta_{v,1} \quad \equiv \quad (\dot{\delta},\, \dot{v}). \tag{3.10}$$

The inverse of geometric-dynamic response of the vehicle $R_v^{*-1}\, b_{gd}$ is then specified as

$$R_v^{*-1}\, b_{gd} \quad = \quad \begin{cases} \dot{\delta}^*(\kappa',\, \dot{v}) \\ F_p^*(\dot{\delta},\, \dot{v}) \end{cases} . \tag{3.11}$$

where $\dot{\delta}^*$ and F_p^* are the operators representing $\dot{\delta}(\kappa',\, \dot{v},\, s)$ and $F_p(\dot{\delta},\, \dot{v},\, s)$ for a fixed situation $s = s_0$ (in analogy to R^{*-1}). Once R^{*-1} is identified, the table $\mathcal{L}^*[b_{gd} : c_p]$ can be used to find a functional representation of $R^{*-1}\, b_{gd}$ by means of curve fitting techniques.

Now, template functions have to be found that can be adapted to the shape of the scalar functions in (3.8). A template function t_η^* has a certain characteristic

[1] 'Advantageous' here means that the resulting curve fitting can be accomplished with fewer coefficients.

shape, i.e. a functional skeleton, that can be adapted through a set of coefficients $\underline{\eta}$. For the vehicle example, front wheel angle velocity and propulsive force may be described by the template functions

$$t_a^* = \dot{\delta}^*(\kappa', \dot{v}) \tag{3.12}$$
$$t_b^* = F_p^*(\dot{\delta}, \dot{v}). \tag{3.13}$$

The diamonds in figure 3.7 show the propulsive forces for a sample set of front wheel angle velocities $\dot{\delta}$ and acceleration $\dot{v}$ given a specific initial state s_0. Knowing this shape, it is possible to arrange an appropriate template function

$$t_b^*(\dot{\delta}, \dot{v}) \equiv b_0^* + b_1^* \dot{v} + b_2^* \dot{\delta} + b_3^* \dot{\delta}^2 + b_4^* \dot{\delta}^3 \tag{3.14}$$

The parameter set $\underline{b}^* = (b_0^*, b_1^*, b_2^*, b_3^*, b_4^*)$ can now be used to adapt the template t_b^* to the specific shape of the samples. The solid line surface in figure 3.7 shows the adapted template function $t_b^*(\dot{\delta}, \dot{v})$ once the parameters $\underline{b}^*$ are optimized through curve fitting.

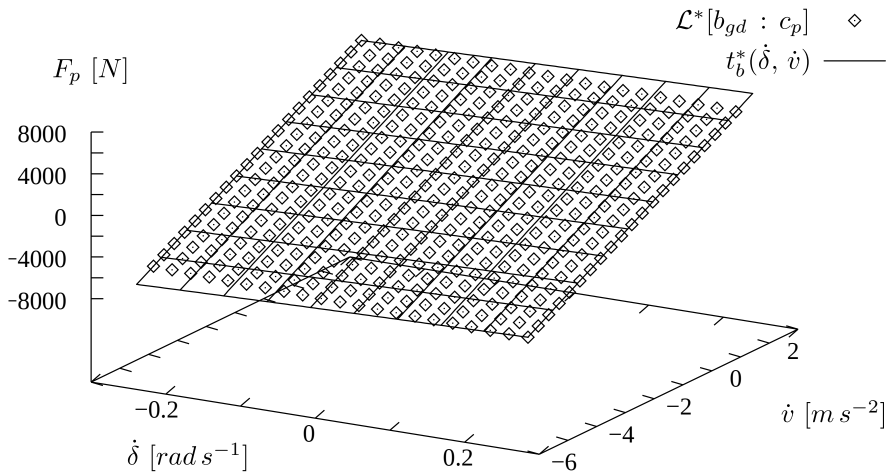


Figure 3.7: Sampled data of $F_p^*(\dot{\delta}, \dot{v})$ together with the adapted template $t_b^*(\dot{\delta}, \dot{v})$.

The choice of appropriate template functions at this point is essential in order to achieve sufficient *precision* and to *reduce the number of coefficients* when adapting the sampled data. Once, appropriate template functions are found, then the union of the adapted parameter sets

$$\underline{p}^* \equiv (a_0^*, a_1^*, \ldots, b_0^*, b_1^*, \ldots). \tag{3.15}$$

distinctly describes the relationship R^{*-1} between control parameters c_p and the geometric-dynamic behavior b_{gd} *for one specific situation* s_0. All parameters such as a_0^*, a_1^*, ..., b_0^*,, describing an initial state specific mapping, are called *star parameters*. This term is chosen to distinguish them from parameters on the initial state dependency level as shown in the subsequent section.

3.1.2 Initial State Dependence

Regarding the relationship between front wheel angle velocity $\dot{\delta}$ and rate of curvature change κ' when driving a passenger car, it becomes clear that it depends on situational circumstances, e.g. lateral acceleration and speed. This section shows how to deal with changes in the input/output behavior dependent on the initial state.

In equation (3.15) a set of star parameters $\underline{p}^*$ was identified that distinctly determines the relationship between control parameters c_p and the performed geometric dynamic behavior b_{gd} for one particular initial state s_0. Therefore, the influence of specific circumstances on the mapping can be investigated through changes in the star parameters. The idea is to find a function that describes the relationship between any initial state s and each single star parameter in $\underline{p}^*$. The parameters that shape these functions are then called the *tilde parameters* $\underline{\tilde{p}}$.

The very first step to approach such a mapping is to identify key values of an initial state that physically relate to $R(c_p, s)$ as directly as possible. The roll angle ϕ of a vehicle may have some influence on the relationship between $\dot{\delta}$ and κ'. However, from vehicle dynamics it is known that the tire forces saturate and that the curvature depends highly on the initial lateral acceleration $\ddot{y}_0$. The initial curvature κ_0, the velocity v_0 or the initial front wheel angle δ_0 might as well be decisive factors in this example. These values are called *state indices* in the frame of this dissertation.

In section 3.1.1 a database $\mathcal{L}^*[b_{gd} : c_p]$ was build to sample the input/output relationship for one specific initial state s_0. As a result of curve fitting the parameter vector $\underline{p}^*$ was determined that defines the mapping for a given initial state. Now, this process has to be accomplished for a set of initial states so that one gets a table $\mathcal{L}[\underline{\chi}(s) : \underline{p}^*]$ recording the relationship between state indices and the star parameters $\underline{p}^*$ (see figure 3.8). Similar to the procedure in the previous section, curve fitting has to be accomplished by means of template functions.

Usually, the graph of the star parameters behaves very roughly over the domain of state indices. A trick to reduce the amount of coefficients required during curve fitting is to transform the state indices by any kind of senseful function. At this point, for each star parameter in $\underline{p}^*$ a set of keys $\chi_i(s)$ has to be determined. Similar to the keys β_i that were used in the previous sections, these keys allow to sequentially compute the star parameters based on state indices. Now, the template functions $t_i(\chi_i(s))$ have to be determined that are able to capture the dependency of a p_i^* on the keys $\chi_i(s)$:

$$\begin{pmatrix} p_0^* \\ p_1^* \\ \vdots \\ p_N^* \end{pmatrix} = \begin{pmatrix} t_0(\chi_0(s)) \\ t_1(\chi_1(s)) \\ \vdots \\ t_N(\chi_N(s)) \end{pmatrix}. \tag{3.16}$$

Using curve fitting techniques, the template functions $t_i(\chi_i(s))$ are adapted to find the value of the star parameters $\underline{p}^*$ based a given initial state.

state indices s				star parameters $\underline{p}^*$			
v_0 $[m/s]$	δ_0 $[rad]$	$\ddot{y}$ $[m/s^2]$	κ_0 $[rad/m]$	a_0^*	a_1^*	a_2^*	a_3^*
10	0.0977	0.831	0.00831	-8.338e-07	1.1830e-06	5.1673e-03	-1.3841e-04
11	0.0821	0.732	0.00605	-1.631e-06	2.3647e-06	5.0881e-03	-1.3714e-04
12	0.0712	0.671	0.00494	-2.373e-06	3.5496e-06	4.9556e-03	-1.3475e-04
13	0.0534	0.542	0.00316	-3.069e-06	4.7273e-06	4.7808e-03	-1.3187e-04
14	0.0401	0.432	0.00205	-3.771e-06	5.9019e-06	4.5642e-03	-1.2821e-04

Figure 3.8: Table containing for each initial state its state indices as well as the parameters used to adapt the template function.

The previous section used the template t_b^* to fit the function F_p^* for a given state. As a result, a parameter set $\underline{b}^*$ was determined, that adapted the template to the particular shape of F_p^*. To describe the change of the mapping F_p^* with respect to the initial state s_0, one has to consider the changes in the parameter set $\underline{b}^*$. Let the key χ_{b0} set for b_0^* be

$$\chi_{b0}(s_0) \equiv \left(\log(v_0),\ \sqrt{|\delta_0|}\right). \tag{3.17}$$

where v_0 is the initial velocity, and δ_0 the initial front wheel angle. The diamonds in figure 3.9 display the samples of the set $\mathcal{L}[(v_0,\ \delta_0) : b_0^*]$. Using the template

$$\begin{aligned} t_{b0}(x,\ y) \equiv\ & \tilde{b}_{0,0}\, y^4 + \tilde{b}_{0,1}\, y^8 + \tilde{b}_{0,2}\, x^2\, y^7 + \tilde{b}_{0,3}\, x^3\, y^3 \\ & + \tilde{b}_{0,4}\, x^3\, y^4 + \tilde{b}_{0,5}\, x^3\, y^6 + \tilde{b}_{0,6}\, x^4 + \tilde{b}_{0,7}\, x^5\, y^2 \\ & + \tilde{b}_{0,8}\, x^5\, y^3 + \tilde{b}_{0,9}\, x^5\, y^4 + \tilde{b}_{0,10}\, x^6\, y^3 + \tilde{b}_{0,11}\, x^6\, y^5 \\ & + \tilde{b}_{0,12}\, x^7 + \tilde{b}_{0,13}\, x^7\, y + \tilde{b}_{0,14}\, x^7\, y^4 + \tilde{b}_{0,15}\, x^7\, y^6 \\ & + \tilde{b}_{0,16}\, x^8\, y^2 + \tilde{b}_{0,17}\, x^8\, y^5 + \tilde{b}_{0,18}\, x^8\, y^9 + \tilde{b}_{0,19}\, x^8\, y^{10} + \tilde{b}_{0,20}\, x^9 \\ & + \tilde{b}_{0,21}\, x^9\, y^2 + \tilde{b}_{0,22}\, x^9\, y^6 + \tilde{b}_{0,23}\, x^{10}\, y^3 + \tilde{b}_{0,24}\, x^{10}\, y^7, \end{aligned} \tag{3.18}$$

where $x = log(v_0)$ and $y = \sqrt{|\delta_0|}$ allows one to fit through the data sufficiently close as shown by the solid line in figure 3.9 indicating the adapted template. For the case of fixed initial state the keys, β_i can contain control parameters as

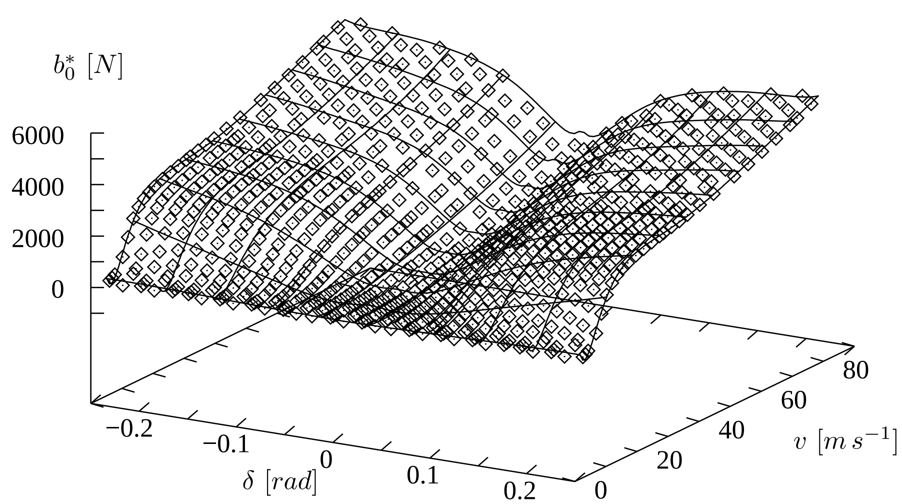


Figure 3.9: Sampled data of $a_0^*(v\,\delta)$ together with the adapted template $t_{b0}(\chi_{b0}(s))$ where $\chi_{b0}(s) = (log(v_0),\ \sqrt{|\delta_0|})$.

soon as they were computed by an earlier template. The same is obviously true for the key sets χ_i that can contain star parameters $\underline{p}^*$ as far as a template t_h with $h < i$ computes it earlier.

Once, all the templates in equation (3.16) are parameterized the function $R^{-1}(p_{gd},\, s)$ is determined. Everything is set up for the plan-to-action mapper to work. Figure 3.10 shows the whole process of control parameter calculation for a given state s in order to achieve a specific gd-plan p_{gd}. The tilde parameters of each template function in (3.16) directly relate to the characteristics of the plant. They are fixed during the whole control process.

For a given state s, with the given tilde parameters, the template functions $t_{a0}(\chi_{a0})$, $t_{a1}(\chi_{a1})$, … compute the star parameters $\underline{p}^*$ that specifies the behavior of the templates $t_a^*(\beta_a)$, $t_b^*(\beta_b)$, … for R^{*-1} (see section 3.1.1). Now, the correspondent control parameters c_p can be computed, in order to achieve the desired gd-plan p_{gd}.

The first step, i.e. the mapping from the given state s to the parameter set $\underline{p}^*$ can be interpreted as *circumstance cognition.* The result of this operation is a vector $\underline{p}^*$ specifying the mapping from the gd-plan p_{gd} to control parameters c_p based on the current state s.

Now, the plan-to-action mapper is fully functional. Its precision is only limited by the choice of appropriate template functions. The problem at this point is that it requires an enormous amount of input parameters

$$\underline{\tilde{p}} \;\equiv\; (a_{0,0},\, a_{0,1},\, \ldots\, b_{0,1},\, \ldots). \tag{3.19}$$

These parameters have to be specified by the user and change whenever the

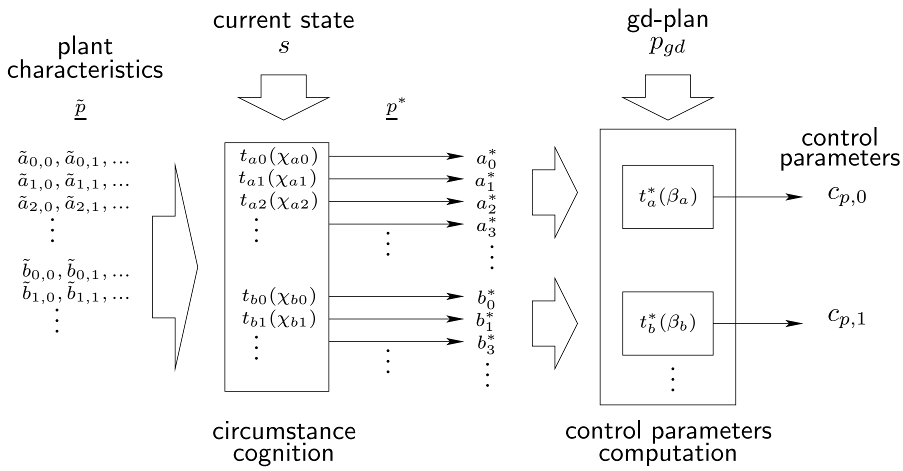


Figure 3.10: Diagram to show how to compute control parameter c_p to achieve the geometric-dynamic behavior p_{gd} for a given state s.

plant configuration changes. In order to avoid this inconvenience, a similar curve fitting can be done with respect to these parameters dependent on changes in the configuration parameters. As a result only the configuration parameters have to be specified when a different plant is used. This step is essential when it is tried to add *adaption* to the controller as discussed in section 2.1.1. However, is not investigated in this dissertation.

3.2 Practical Application

The following sections describe the construction of a plan-to-action mapper for vehicle control. The plant used for the investigation is a three degree of freedom vehicle model (see section B, page 135). It is controlled by the front wheel angle velocity $\dot{\delta}$ and the propulsive force F_p. The geometric-dynamic plan for the plan-to-action mapper is specified through a rate of curvature change κ' and an acceleration $\dot{v}$ to be achieved during the subsequent control interval.

As mentioned earlier the task of the plan-to-action mapper is to compute control parameters so that the geometric-dynamic behavior of the plant b_{gd} matches as much as possible the given gd-plan p_{gd}. Thus, the first thing required is an intuitive measure that tells the distance between two gd-plans using an *interpretation of a gd-plan* (see definition 4, page 13).

In order to judge performance in lateral and longitudinal control, the rate of curvature change κ' and the acceleration $\dot{v}$ are associated with a lateral displacement y_e and an end velocity v_e to be expected at the end of the subsequent control interval, the interpretation follows

$$(\kappa',\, \dot{v}) \quad \longrightarrow \quad (y_e,\, v_e). \tag{3.20}$$

v_e can be computed through $v_0 + \dot{v}\, T_c$. The computation of Δy_e is explained in section 5.3. As can be seen in figure 3.11, the difference Δy_e between lateral displacement and the difference Δv_e between the achieved velocities provide an intuitive tool to judge the performance of the mapping. A lateral displacement error allows to estimate the capability of path following. The error in velocity allows to estimate the probability that a certain lateral acceleration is exceeded (since $\ddot{y} = \kappa\, v^2$). However, considering only errors in the original gd-plan parameters, i.e. rate of curvature change or acceleration, does not provide any physical idea about the performance.

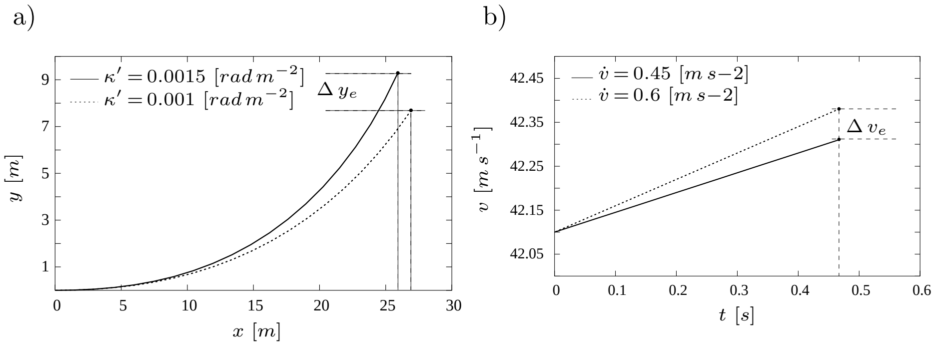


Figure 3.11: Quality of plan-to-action mapping: a) rating by means of lateral displacement $\Delta\, y_e$. b) rating by means of error in velocity $\Delta\, v_e$.

3.2.1 Fixed Initial State Investigation

The first step is to find template functions for a fixed initial state (see section 3.1.1). The criteria of quality for a template function is expressed by the number of star parameters (cost) and the precision by which it is able to approximate a given database samples (gain). Another problem of template function has to deal with convergence during the process of curve fitting. As long as linear multi-dimensional polynomials are applied, though, the data can be fit by solving a linear system of equations [Lawson and Hanson, 1974].

Before scalar template functions can be searched for, the mapping has to be separated into a set of scalar mappings correspondent the number of control parameters. As explained in (3.11) for vehicle control $\dot{\delta}^*(\kappa',\, \dot{v})$ and $F_p^*(\dot{\delta},\, \dot{v})$ are good candidates to represent the inverse system function. As a starting point let the following polynomials specify the templates

$$t_a^*(x,\, y) \quad = \quad \sum_{i=0}^{N_a} \sum_{k=0}^{M_a} a^*_{(M_a+1)\, i+k}\, x^i\, y^k \tag{3.21}$$

$$t_b^*(x,\, y) \;=\; \sum_{i=0}^{N_b} \sum_{k=0}^{M_b} b^*_{(M_b+1)\,i+k}\, x^i\, y^k \tag{3.22}$$

N_a and N_b are the orders of the polynomials in x-direction. M_a and M_b are the orders of the polynomials in y-direction. $t_b^*(\dot\delta,\, \dot v)$ is supposed to model $F_p^*(\dot\delta,\, \dot v)$. $t_a^*(\dot\delta,\, \dot v)$ is supposed to model $\kappa'^*(\dot\delta,\, \dot v)$. This means that it has to be inverted with respect to κ' in order to get $\dot\delta^*(\kappa',\, \dot v)$. Next, it is required to build a databases

$$\mathcal{L}^*[b_{gd} : c_p] \;=\; \mathcal{L}^*[(\kappa',\, \dot v) : (\dot\delta,\, F_p)] \tag{3.23}$$

sampling the space of the vehicle's input/output behavior for specific initial states. For each initial state the correspondent database is used for curve fitting in order to parameterize the template functions in (3.21).

Reducing the number of star parameters in the template functions is crucial in order to reduce complexity[2]. The error that occurs when a coefficient is set to zero, is a good indicator for its importance. Using the interpretations of the gd-plans from the last section, figure 3.12 shows candlestick plots of the absence error of each coefficient. A candlestick displays maximum and minimum values that appeared for a certain setting as lines, while a rectangle covers a certain range of the majority of cases. In our experiments the boxes covered by default a range of 80%.

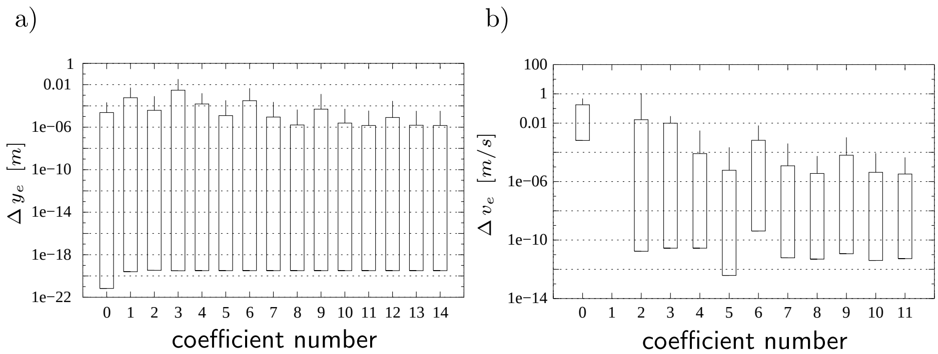


Figure 3.12: Error distributions expressed through candlesticks. a) Displacement error for coefficients of $t_a^*(x,\, y)$. b) Velocity error for coefficients of $t_b^*(x,\, y)$.

Obviously some star parameters only produce a tiny error when left out. These are the coefficients that can be deleted from the template polynomial. Through iterative deletion of star parameters, the following polynomials evolve:

$$t_a^*(x,\, y) \;=\; a_0^*\, y + a_1^*\, y^2 + a_2^*\, x + a_3^*\, x\, y + a_4^*\, x\, y^2 + a_5\, x^2, \tag{3.24}$$
$$t_b^*(x,\, y) \;=\; b_0^* + b_1^*\, y + b_2^*\, x + b_3^*\, x^2. \tag{3.25}$$

[2]Recall, that each coefficient has to be described by a function of state indices during initial state dependency investigations

These functions perform the mapping from a desired gd-plan to correspondent control parameters. To verify that the total precision with which they work is sufficient, the error distributions can be considered as shown in figure 3.13. A maximum lateral deviation of $0.0005\,m$ for lateral control and a precision in longitudinal control of $5 \cdot 10^{-5}\,m/s$ can, indeed, be considered as sufficient.

At this point, one still uses star parameters that are stored in a database for each initial state. The next section discusses how to generalize the information about the discrete set of initial states in order to find for any given state a set of star parameters for (3.24) and (3.25). The precision, of course, will decrease since one has to approximate the database through template functions. It is, however, indispensable, when the controller has to be able to deal with any given state.

3.2.2 Initial State Dependence

The next step in the construction of a plan-to-action mapper is to investigate how the mapping from control parameters to the geometric-dynamic motion of the plant changes with respect to the initial state. The star parameters in equations (3.24) and (3.25) determine the mapping for one particular initial situation. Relating the star parameters to state indices of an initial state allows exactly to describe the desired relationship.

For this step, the template functions have to be computed and the star parameter have to be stored in a table together with the state indices of the initial state where they were computed. In the vehicle control example we use the state indices velocity v_0, front wheel angle δ_0, lateral acceleration $\ddot{y}_0$ and curvature κ_0'. Computing the star parameters for a given set of initial states results then in a table[3]

$$\mathcal{L}[\underline{p}^* : s] \quad = \quad \mathcal{L}[(\underline{a}^*, \underline{b}^*) : (v_0, \delta_0, \ddot{y}_0, \kappa_0')]. \tag{3.26}$$

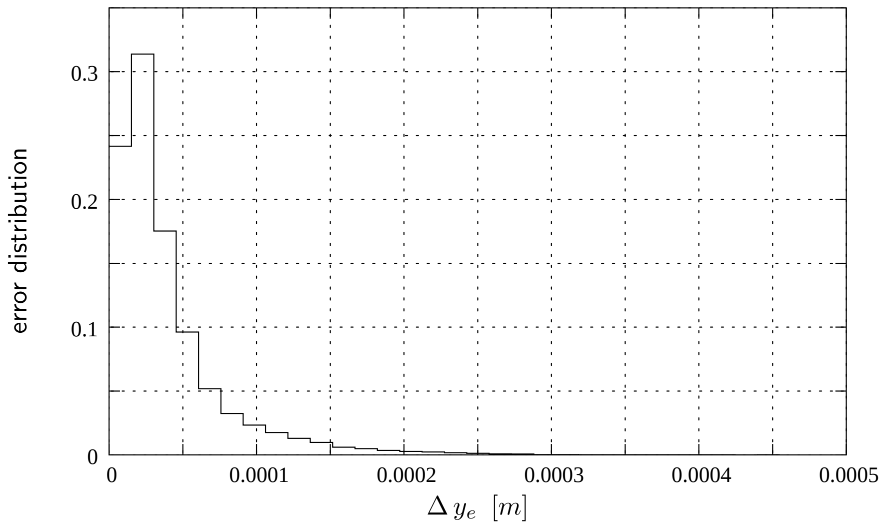
For each of the parameters in $\underline{a}^*$ and $\underline{b}^*$ a function has to be found based on the state indices of the initial states. Again, determining appropriate template functions one has to consider the tradeoff between number of coefficients and the precision achieved. So, the first step is to investigate the importance of each coefficient for the global formula.

Figure 3.14 shows the importance of each single parameter. For example, parameter two in (3.24) is very important, since a very high error occurs when it is left out. Parameter one and four, however, are almost negligible. Dependent on the importance of the parameter for the global formula the precision can be determined with which it is described dependent on the state indices.

The next step is to watch the sample data of each coefficient with respect to state indices. Figure 3.15a displays parameter five from (3.24) as the initial lateral acceleration and the initial velocity changes. Since surfaces with edges and

[3] For practical reasons it is advantageous to split this table into two files: one for $\underline{a}^*$ and one for $\underline{b}^*$.

a)



b)

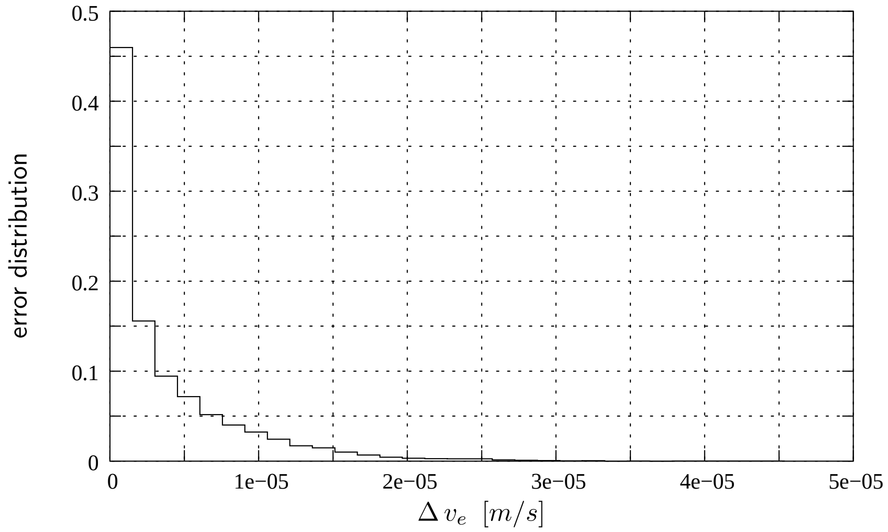


Figure 3.13: Error distributions when using formulae (3.24) and (3.25) for lateral and longitudinal plan-to-action mapping.

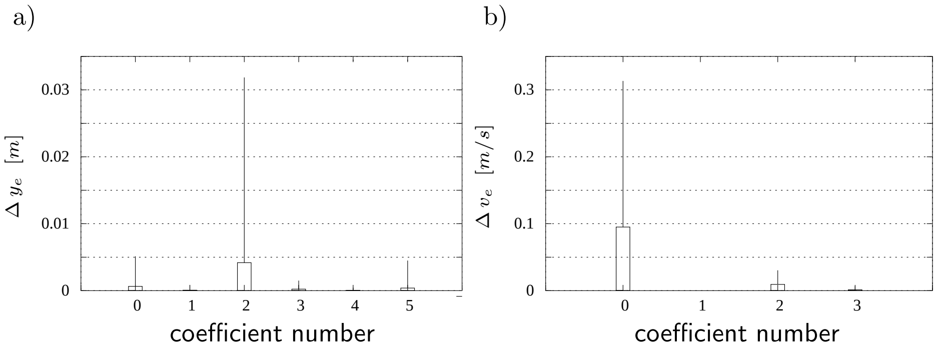


Figure 3.14: Error distributions for final parameters of fixed initial state templates in a) formula (3.24) and b) formula (3.25) where parameter one is indispensable.

long flat planes are expensive to curve fit with polynomials it is advantageous to stretch the domain. In the above example the choice was made to determine the x and y coordinates of the template function by

$$x_t = e^{\frac{-v_0}{10}}, \quad y_t = \ddot{y}_0. \tag{3.27}$$

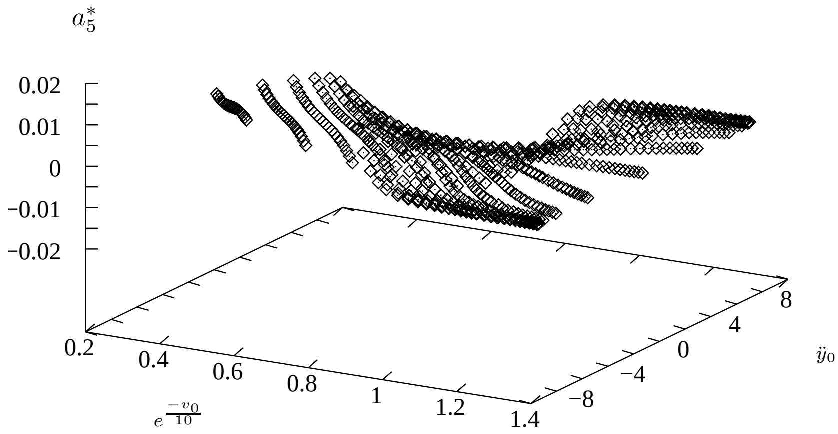
A software tool developed in the frame of this dissertation allows to determine an optimized polynomial structure for a given number of desired coefficients. This means that for the given number of coefficients the approximation error is minimized. As a result, one can display the number of coefficients in a template function versus curve fitting error achieved with the best possible polynomial. Figure 3.15b shows the error in parameter five versus the number of coefficients in the optimized configuration.

Another feature of the software tool mentioned above is the ability to produce C and Python code for the structure optimized polynomials where the user only specifies the number of coefficients he desires. With the automatically produced modules it is then possible to compute an error distribution for the total plan-to-action mapper as done in figure 3.13 with original star parameters, i.e. star parameters as they result directly from the curve fitting process. The error distributions for the case that the star parameters are computed as functions based on state indices is of course higher, since an approximation is performed. Indeed, the plots in figure 3.16 show that the error distribution using interpolated star parameters moves towards higher errors comparing to the results with original star parameters. However, for both parts it can be stated the provided precision is largely enough for the task of vehicle control.

3.3 Conclusion

In this chapter the unit performing circumstance cognition was discussed. A so called plan-to-action mapper was build based on a state space description

a)



b)

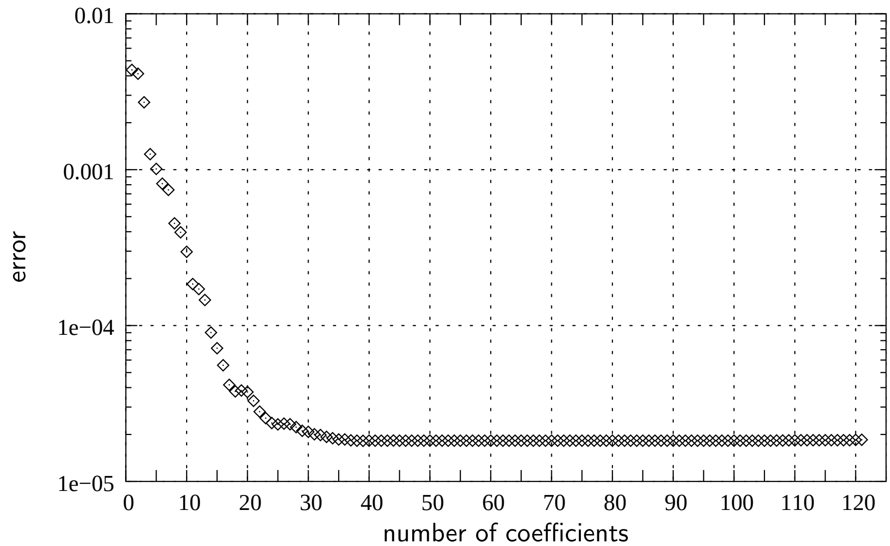


Figure 3.15: Behavior of parameter five in (3.24). a) Parameter five as a function of state indices v and $\ddot{y}$. b) Error of curve fitting parameter five with respect to the number of coefficients in an struture optimized polynomial.

a)

b)

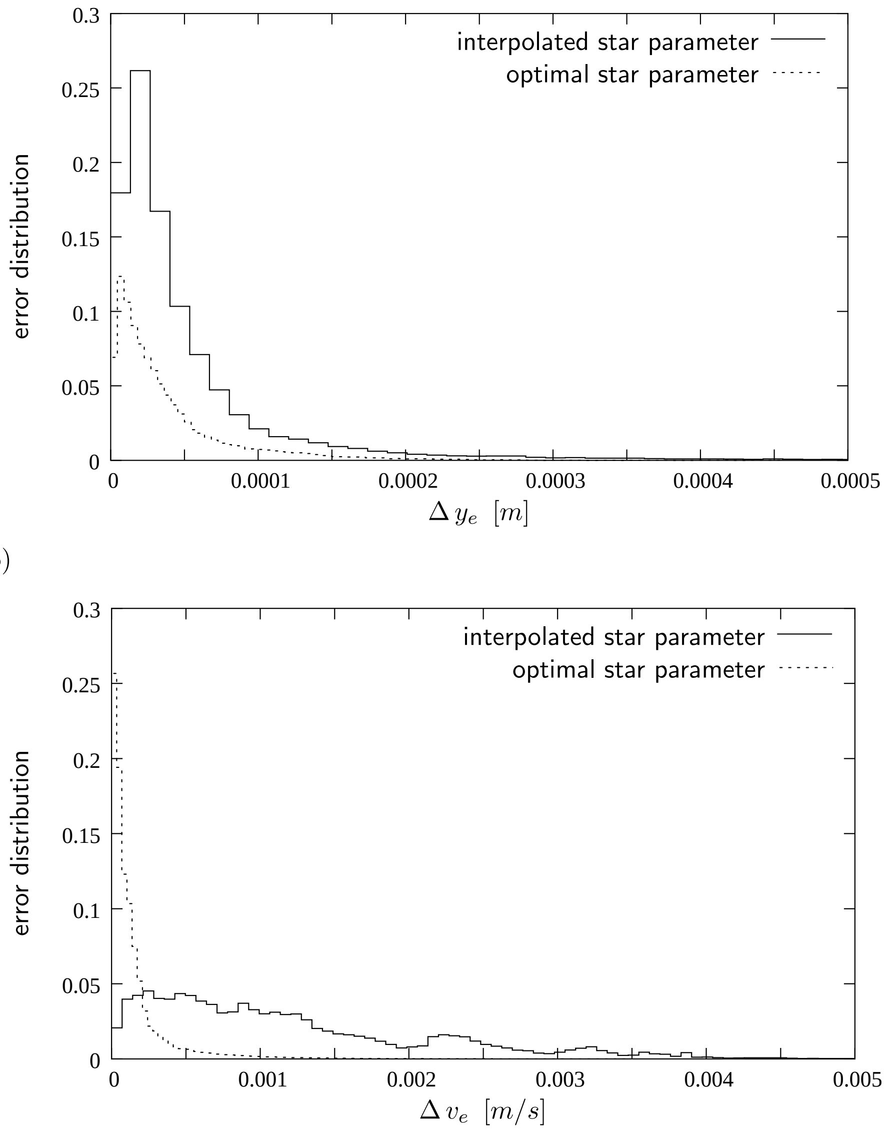


Figure 3.16: Comparing error distributions when using formulae (3.24) and (3.25) with original and interpolated star parameters. a) Errors of later displacement. b) Errors of velocity.

of the system to be controlled. It was shown, how to build a formula for plan-to-action mapping in two basic steps once the system function is broken up into scalar functions. Then, template functions have to be found that describe the relationship between control parameters and geometric-dynamic output for a fixed initial state. The parameters of those template functions are found through curve fitting techniques. In a second step, the dependency of those parameters on state indices are investigated for a set of representative initial states. The result is a formula that allows to find control parameters for a given gd-plan for any situation that is circumscribed in the database that was used. The previous sections outlined a way to automize the construction of a plan-to-action mapper[4].

As an application, a plan-to-action mapper for vehicle control was developed. In this case, two scalar functions were defined determining front wheel angle velocity and propulsive force. Following the above mentioned procedure statistical plots showed the suitability of the developed plan-to-action mapper for the vehicle control task. When relying on *containability* as a measure for the controller's reliability (to be shown section 7, page 103), constraints on precision are sufficient. However, in the case that containability it not an acceptable reliability measure, no systematic method has been presented to proof the formal stability of the plan-to-action mapper itself.

With the ability to compute control parameters based on a given gd-plan, the task of motivation matching, i.e. gd-plan construction, can be targeted. However, before this can be done it has to be discussed how motivations can be mathematically described and through what procedure it is possible to find a distinct gd-plan that matches all existing motivations. This is the focus of the next chapter.

[4]Another example of a tool for identification of system models can be found in [King et al., 2002].

Chapter 4

Target and Limit Maps

Before motivation matching can be discussed, a methodology has to be established to express motivation in mathematical terms. Much work has been accomplished in the field of decision theory. A very basic assumption in decision theory is that a rational agent chooses from a set of alternatives the one which provides the maximal prefered outcome [Wellman and Doyle, 1991]. A handy tool to describe preference relationships is the *utility function* [Ramsey, 1931, Keeney and Raiffa, 1976].

The aim of this chapter is to define a simple mathematical terminology to handle the modeling of motivations in terms of utility functions for the task of gd-plan construction. An important requirement is that it can be easily implemented, fastly executed and possesses a transparent intuitive structure. Where the argumentation consists purely of a chain of logical conclusions, the following assumptions build the bases for the method of *target and limit maps*:

- Two basic kinds of motivations exists. One type of motivations restrict the space of gd-plans, in order to avoid domains of disastrous system states. These motivations are related to *fear* as shown in figure 4.1a. Another type of motivation targets a specific system state and correspond to a specific gd-plan. Motivations of this type are related to *desire* as shown in figure 4.1b.

- It is assumed that all motivations can be described through scalar functions, i.e. the correspondent preferences obey *transitivity* and *orderability.*

Section 4.1 discusses motivations of fear and the appendant *limit maps.* Section 4.2 discusses desire motivations and the appendant *target maps.* In section 4.3 it is discussed how the two types of maps can be combined into one utility function. Section 4.4 finally discusses how to determine one distinct gd-plan consistent with all related motivations.

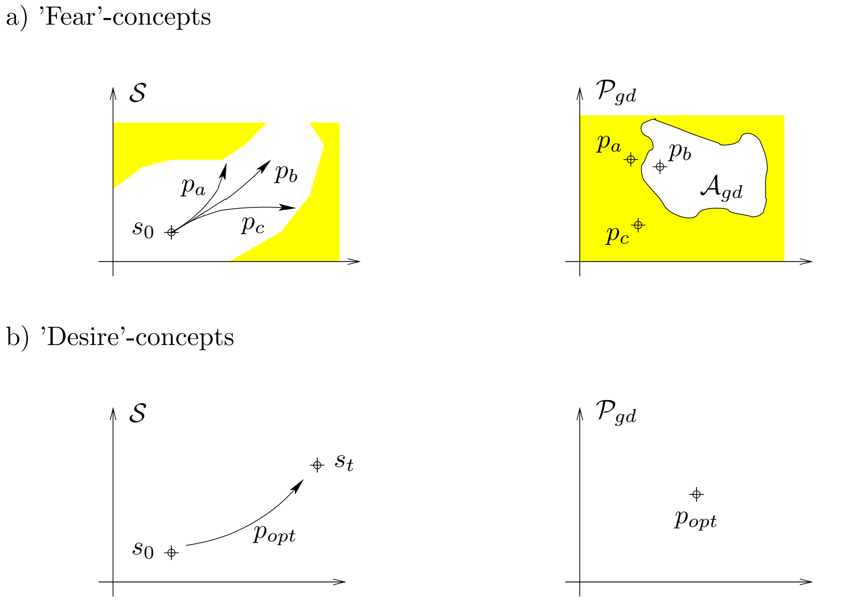


Figure 4.1: Concepts of 'desire-' and 'fear-motivations', with respect to the space of plant states $\mathcal{S}$ and the space of possible gd-plans $\mathcal{P}_{gd}$.

4.1 Limit Map

The following term defines a 'fear' motivation as a bases for limit maps:

Definition: 5 (Fear Motivation) *A* FEAR MOTIVATION *is a motivation that manifests itself in a separation of the space of gd-plans $\mathcal{P}_{gd}$ into regions of different admissibilities.*

Consider vehicle control, for example: The fear to exceed a certain lateral acceleration $\ddot{y}_{max}$ forbids whole sets of desired motions where lateral acceleration $\ddot{y}$ becomes higher than $\ddot{y}_{max}$. The definition of a fear motivation directly leads to the definition of a scalar function that expresses the preferences that are caused by this motivation:

Definition: 6 (Admissibility Function) *An* ADMISSIBILITY FUNCTION *is a function that assigns to each element p_{gd} of the the space of gd-plans $\mathcal{P}_{gd}$ a value in $[0, 1]$, i.e.*

$$A(p_{gd}) : \mathcal{P}_{gd} \rightarrow [0,\, 1]. \tag{4.1}$$

to indicate the amount of admissibility for each gd-plan p_{gd}.

Due to the fact that, in some cases, it might be possible to precisely measure the admissibility of a gd-plan $A(p_{gd})$ is left to the continuous set of numbers between 0 and 1. The use of the binary range $\{0, 1\}$, however, could simplify subsequent calculations tremendously[1].

The question that arises now is: If there are multiple fear motivations, then how can the corresponding admissibility functions be combined ? As insinuated earlier, fear motivations express the fear of running into disaster. Therefore, a gd-plan that is, to a certain degree, inadmissible by one fear motivation cannot gain admissibility because another fear motivation fears it less, i.e. assigns a higher admissibility to it. So, the overall admissibility shall never exceed the lowest admissibility for a given gd-plan. Hence, the upper boundary for a function that combines all admissibility functions is the *minimum*-operator. The following definition specifies a limit map:

Definition: 7 (Limit Map) *A* LIMIT MAP *is a scalar function*

$$L(p_{gd}) \,:\, \mathcal{P}_{gd} \,\rightarrow\, [0,\, 1]. \tag{4.2}$$

that represents all fear motivations. The correspondent admissibility functions $\{A_i(p_{gd}) : i = 0, 1, 2, ...\}$ *are combined to one single utility function* $L(p_{gd})$ *that obeys the constraint*

$$L(p_{gd}) \leq min\{A_i(p_{gd}) : i = 0, 1, 2, ...\}. \tag{4.3}$$

Restricting the range of each A_i to [0, 1] allows to use the $\prod$-operator for the combination of multiple maps[2], e.g.

$$L(p_{gd}) \quad = \quad \prod_{i=0}^{N} p_i\left(A_i(p_{gd})\right), \qquad \text{with} \qquad p_i(x) < x. \tag{4.4}$$

Figure 4.2 explains how fear motivations are modeled. For a given situation, each fear motivation has to be expressed by an admissibility function over the space of possible motions $\mathcal{P}_{gd}$. The deliberately chosen combination operator finally produces one limit map, that includes considerations about all existing fear motivations.

An example from vehicle control helps the understanding. In figure 4.3, a gd-plan is defined as the composition of desired rate of curvature change κ' and desired acceleration $\dot{v}$. One constraint that exists has to do with longitudinal dynamics. In the simplest case, acceleration and deceleration are limited to constant values. The dotted lines in figure 4.3 represent these limits. No admissible gd-plan lies to the right of the line at $\dot{v} = 5.7m\, s^{-2}$ (max. acceleration) or left to $\dot{v} = -8.3m\, s^{-2}$ (max. deceleration). Another constraint comes from lateral dynamics. To avoid situations that are 'too difficult' the lateral acceleration has to be less then a certain maximum $\ddot{y}_{max}$. This introduces limits on curvature

[1] A binary set is also sufficient to treat aspects of containability as explained later in chapter 7.

[2] This is because multiplying a value 'x' with a value that is between 0 and 1 results always in something that is less than or equal 'x'. So, the result is always less than or equal the result of the minimum-operator.

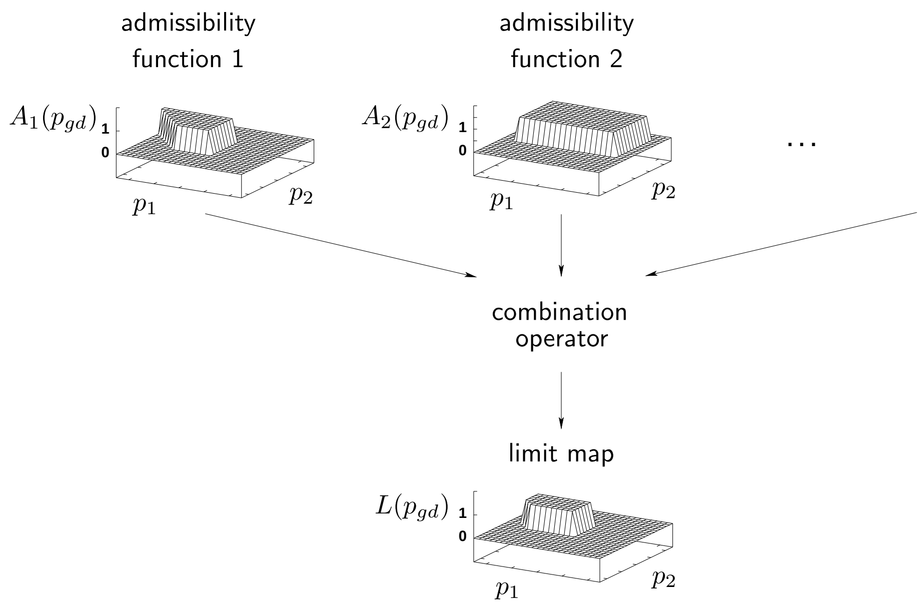


Figure 4.2: Two-dimensional example for combination of different admissibility functions $A_1(p_{gd})$, $A_2(p_{gd})$, etc. to one single limit map $L(p_{gd})$.

dependent on the desired acceleration. In figure 4.3 these limits are indicated by dashed lines. No admissible gd-plan lies above the upper dashed line or underneath the lower dashed line. The intersection of both domains results in an admissible domain, indicated by the shaded area.

4.2 Target Map

In order to classify motivations that have a preference for one specific nominal motion, i.e. a 'prefered gd-plan,' *desire motivations* are defined as follows:

Definition: 8 (Desire Motivation) *A motivation that determines for each situation one distinct preferred gd-plan is called a* DESIRE MOTIVATION. *Its primary manifestation is the* PREFERRED GD-PLAN $p_t \in \mathcal{P}_{gd}$.

It cannot be assumed that the preferred gd-plan p_t is always practicable, since it may lie outside the admissible domain $\mathcal{A}_{gd}$. In the general case, desire motivations produce a preference structure in the whole space $\mathcal{P}_{gd}$. So it is possible to define an influence of the desire motivation, even if the preferred gd-plan is outside the allowed domain. This is done by assigning each gd-plan p_{gd} a constructiveness value $C(p_{gd})$.

In vehicle control, for example, one strives to drive along the nominal course as accurately as possible. This motivation results in one specific gd-plan p_t,

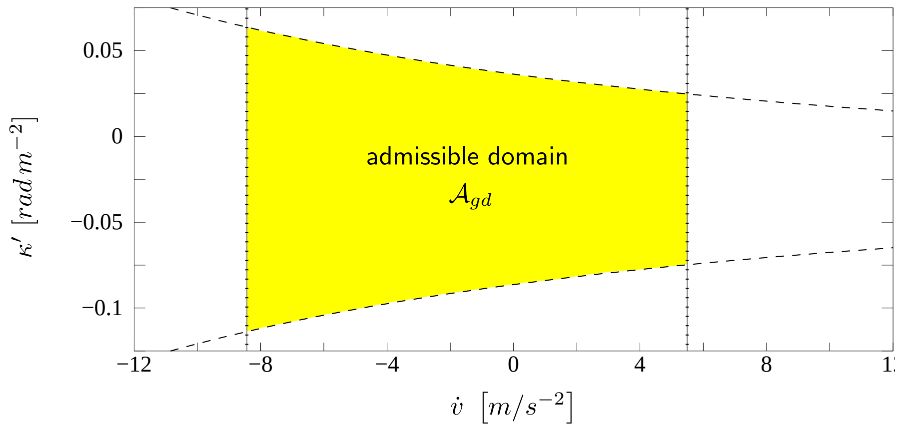


Figure 4.3: Limit map in vehicle control: The gd-plan consists of rate of a curvature change κ' and an acceleration $\dot{v}$.

represented by a rate of curvature change κ'_t and an acceleration $\dot{v}_t$ that leaves the vehicle traveling as accurately and as quickly as possible along the nominal course. The constructiveness of a gd-plan decreases with the deviation between its rate of curvature change and κ'_t and the deviation between its acceleration and $\dot{v}_t$.

It is essential to understand that target maps only exist to describe the preference relationship between two arbitrary gd-plans in $\mathcal{P}_{gd}$. So adding an offset to the constructiveness value of all gd-plans does not change the greater, less or equal relationships. The constructiveness of a gd-plan has to always be finite; therefore, a finite minimum over $\mathcal{P}_{gd}$ must exist. Since constant offsets do not change the relationship between the gd-plans, one can require that the constructiveness must always be greater than zero. Such a unipolar treatment of desires[3] facilitates the dicussion below. It expresses also that desire motivations only describe the amount of 'constructiveness' and never the 'destructiveness' of a gd-plan with respect to a specific desire motivation. The restriction $C(p_{gd}) > 0$ allows later to combine multiple constructiveness functions in an efficient way.

Definition: 9 (Constructiveness Function) *Given a specific desire motivation, the* CONSTRUCTIVENESS FUNCTION

$$C(p_{gd}) : \mathcal{P}_{gd} \rightarrow \Re, \tag{4.5}$$

expresses the constructiveness of each gd-plan in $\mathcal{P}_{gd}$ with respect to this specific motivation by a real number. The constructiveness of a gd-plan decreases with the distance to the preferred gd-plan p_t, i.e.

$$dist(p_0, p_t) > dist(p_1, p_t) \Leftrightarrow C(p_0) < C(p_1), \tag{4.6}$$

[3]For a bipolar example of modelling goals see [van der Torre and Weydert, 1998].

where the distance function dist : $(\mathcal{P}_{gd}, \mathcal{P}_{gd}) \rightarrow \Re$ *indicates the amount of difference between two gd-plans by a real number. Furthermore, one must be certain that,*

$$C(p_{gd}) > 0 \quad \forall \quad p_{gd} \in \mathcal{P}_{gd}. \tag{4.7}$$

Applications may impose further constraints on the shape of the constructiveness function. Figure 4.4a shows an example from vehicle control, where it is desired to follow a nominal course. The constructiveness $C(\kappa')$ is drawn with respect to a rate of curvature change κ' of a possible trajectory. Assuming that $C(\kappa')$ is symmetric to the preferred gd-plan κ'_{opt} would not reflect a realistic judgment. In this case, κ'_1 and κ'_{-1} have the same constructiveness. However, figure 4.4b shows that the trajectory resulting from κ'_1 is very likely able to follow the nominal course, where else κ'_{-1} simply guides the vehicle away from the track. A realistic shape of $C(\kappa')$ should therefore result in a higher constructiveness for κ'_1 than for κ'_{-1}. This shows that the shape of the constructiveness function is very important and at the same time relatively easy to determine. In the example of figure 4.4, a first approach would be to decrease the constructiveness value of a gd-plan proportionally to the predicted angular deviation between the velocity vector and the nominal course.

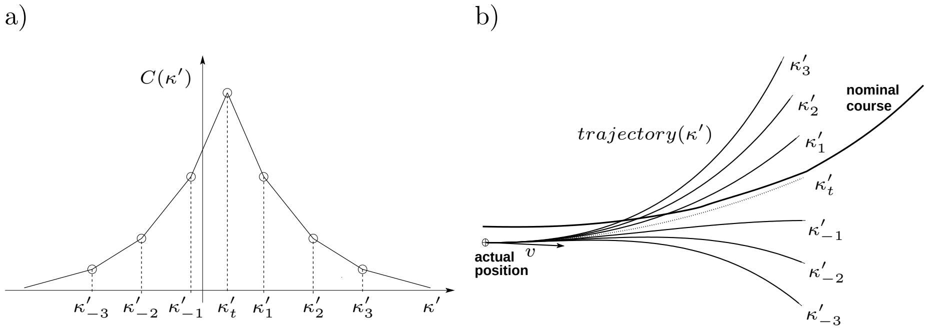


Figure 4.4: Example from vehicle control: Different gd-plans expressed by the rate of curvature change κ'. a) A symatric constructiveness function $C(\dot{\kappa})$ b) Corresponding trajectories resulting from the different κ'.

It is possible that there is more than one desire motivation. Therefore, multiple preffered gd-plans can also exist. The question then is: how can we combine all desire motivations to one constructiveness function $T(p_{gd})$?

As mentioned earlier, desire motivations only express constructiveness and never destructiveness, i.e. negative constructiveness. Thus, a specific gd-plan p that is of a certain constructiveness $C_1(p_{gd})$ with respect to a target motiviation cannot be less constructive, because another constructiveness function $C_2(p_{gd})$ assigns a lower value to it. Equivalently, the combined constructiveness has to be greater than or equal to the *maximum*-operator. We are now ready to define a target map.

Definition: 10 (Target Map) *A* TARGET MAP *is a scalar function*

$$T(p_{gd}) : \mathcal{P}_{gd} \rightarrow \Re. \tag{4.8}$$

that combines the preference relationships of all desire motivations. This is done by combining the independent constructiveness functions $C_i, i = 0, 1, 2, ...$ into one function $T(p_{gd})$. The total utility has to satisfy

$$T(p_{gd}) \geq max\{C_i(p_{gd}) : i = 0, 1, 2, \ldots\}. \tag{4.9}$$

Since, by definition the constructiveness is always a positive value, the $\sum$-operator may be used to combine multiple target maps, e.g.

$$T(p_{gd}) \quad = \quad \sum_{i=0}^{N} q_i\, C_i(p_{gd}), \qquad \text{with} \qquad q_i \,\geq\, 1. \tag{4.10}$$

The chosen operator for the combination of different constructiveness functions is very important. Figure 4.5 shows how two constructiveness functions are combined by the maximum-operator and the '$\sum$'-operator. The use of a maximum-operator ensures that no gd-plan has a higher constructiveness than the gd-plans that directly relate to desire motivations. This may be an advantage, since in this case the optimum of the total utility will always be related to a gd-plan that is precisely calculated as best for a certain target motivation. The disadvantage is that there cannot be a compromise in the sense of choosing a gd-plan in the 'middle'. Such a compromise can be achieved by the summation operator as depicted in figure 4.5b. With this operator, the resulting gd-plan satisfies, to a certain degree, all desire motivations. It requires, however, that the shape of the constructiveness functions is precise, i.e. reflects the preference relationships in the space of gd-plans correctly. Otherwise, the compromise may not be in the sense of any target motivation.

a) $T(p_{gd}) = max\{C_1(p_{gd}),\, C_2(p_{gd})\}$ b) $T(p_{gd}) = C_1(p_{gd}) + C_2(p_{gd})$

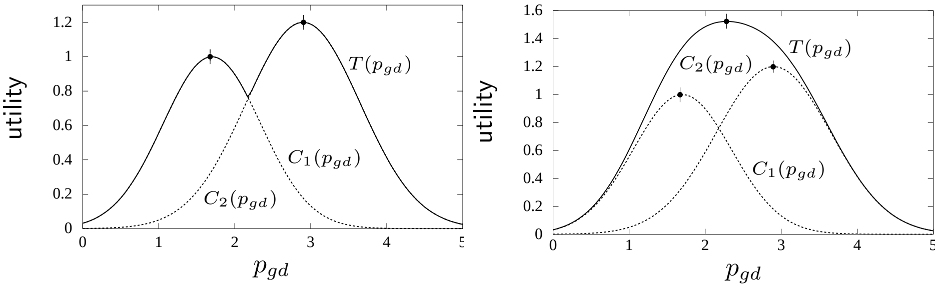


Figure 4.5: Combining multiple target based utilities over a one-dimensional space of gd-plans. a) Using the maximum-operator. b) Using the '$\sum$'-operator.

The process of calculating a target map is depicted in figure 4.6. Each target motivation results in a constructiveness function that decreases with distance to the preferred gd-plan. To get a representation of all target motivations, they are combined into one single target map.

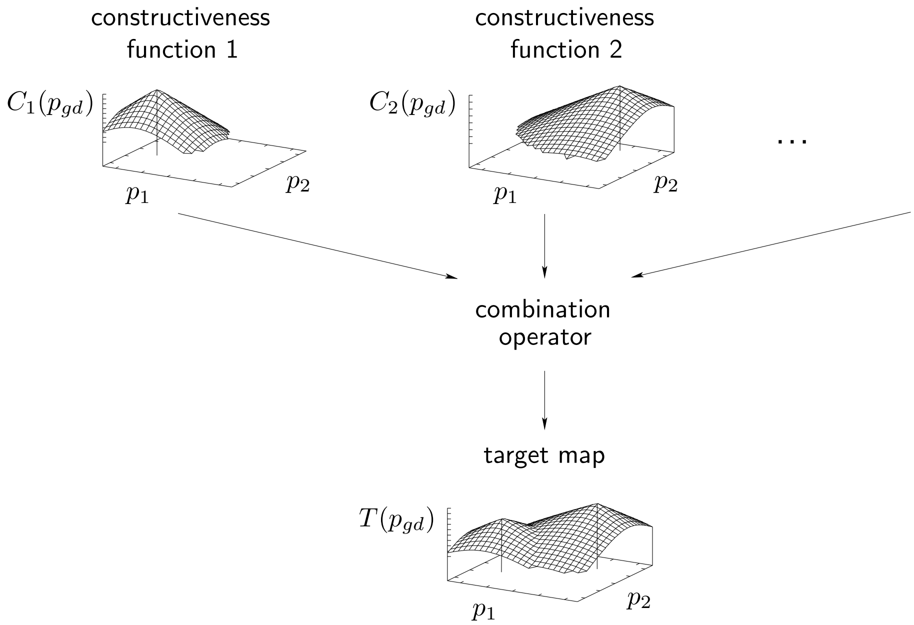


Figure 4.6: Two-dimensional example of combination of constructiveness functions $C_1(p_{gd})$, $C_2(p_{gd})$, etc. to one single target map $T(p_{gd})$. As indicated, the preferred gd-plans act like pillars that span the tent of the constructiveness function.

4.3 Combining Target and Limit Maps

The previous sections introduced how fear and desire motivations, can be modeled by utility functions. As a first step towards the gd-plan consistent with all motivations, a combined utility function must be found. This section now shows how the target map $T(p_{gd})$ and the limit map $L(p_{gd})$ can be combined into a single utility function $U(p_{gd})$.

The domains where the limit map is equal to zero form the absolutely inadmissible domain. No utility should be less than the utility of absolutely inadmissible gd-plans.

It is logical to say, that if two gd-plans are equal with respect to constructiveness, then the preference relationship is exclusively determined by their admissibility. In the same way, if two gd-plans are of equal admissibility then their constructiveness is decisive for their preference relationship. This means

$$T(p_1) = T(p_2) \quad \Rightarrow \quad \big(L(p_1) > L(p_2) \;\Leftrightarrow\; U(p_1) > U(p_2)\big), \tag{4.11}$$

$$L(p_1) = L(p_2) \quad \Rightarrow \quad \big(T(p_1) > T(p_2) \;\Leftrightarrow\; U(p_1) > U(p_2)\big) \tag{4.12}$$

and

$$L(p_1) = L(p_2) \cap T(p_1) = T(p_2) \quad \Rightarrow \quad U(p_1) = U(p_2). \tag{4.13}$$

It follows that for all gd-plans p_{gd} that have the same constructiveness T_0 the total utility function increases consistently alongside the admissibility; for gd-plans of the same admissibility the total utility is steadily increasing with constructiveness, i.e.

$$T(p_{gd}) = T_0 \quad \Rightarrow \quad U(p_{gd}) = f_{T_0}(L(p_{gd})), \tag{4.14}$$

$$L(p_{gd}) = L_0 \quad \Rightarrow \quad U(p_{gd}) = f_{L_0}(T(p_{gd})). \tag{4.15}$$

where f_{T_0} and f_{L_0} are steadily increasing functions. Now, the combined utility function can be defined:

Definition: 11 (Combined Utility Function) *A* Combined Utility Function *$U(p_{gd})$ is a mapping from $\mathcal{P}_{gd}$ to $\Re$, that combines the constructiveness and admissibility considerations of the target map and the limit map. The following conditions have to be met:*

- *The absolutely inadmissible regions ($L(p_{gd}) = 0$) must have the lowest utility in the whole set, i.e.*

$$L(p_1) > 0 \;\wedge\; L(p_2) = 0 \quad \Rightarrow \quad U(p_1) > U(p_2). \tag{4.16}$$

- *Indifference with respect to one map causes the 'rating' to be based on the other map, i.e.*

$$T(p_1) = T(p_2) \quad \Rightarrow \quad \big(L(p_1) \geq L(p_2) \;\Leftrightarrow\; U(p_1) \geq U(p_2)\big), \tag{4.17}$$

$$L(p_1) = L(p_2) \quad \Rightarrow \quad \big(T(p_1) \geq T(p_2) \;\Leftrightarrow\; U(p_1) \geq U(p_2)\big). \tag{4.18}$$

The formulation of (4.14) enables a simple definition of the combined utility function for the case of a binary range of admissibility, i.e. the case where gd-plans are either admissible or inadmissible, with nothing in between. In this case, the total utility can be set to zero for all gd-plans p_{gd} where $L(p_{gd}) = 0$. The total utility of gd-plans that are admissible can be set equal to the constructiveness value $T(p_{gd})$. Since in this case the constructiveness is the only factor that influences the preference relationships nothing is gained by inventing a more complex function to conclude on the total utility. Note, that the constraint on constructiveness $C(p_{gd}) > 0$ (equation (4.7)) ensures that no gd-plan that is allowed has a utility of zero. The combination process can be considered as cutting inadmissible domains out of the target map. An example of such a combination for a two-dimensional domain is shown in figure 4.7. At this point, the decision making process has to find the plan $p_{gd} \in P_{gd}$, where $U(p_{gd})$ is maximal.

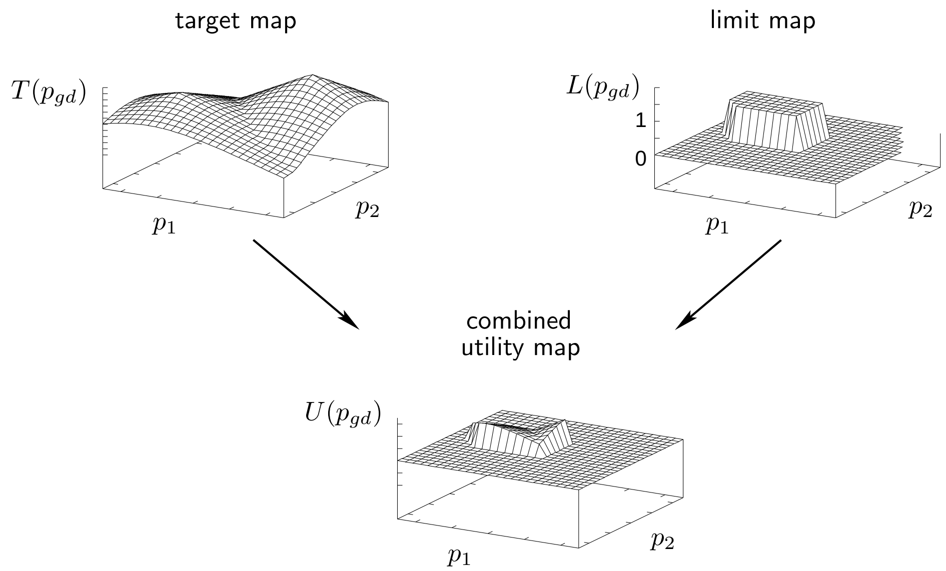


Figure 4.7: Combining target map $T(p_{gd})$ and limit map $L(p_{gd})$ into one single utility function $U(p_{gd})$.

4.4 The Single Consistent GD-Plan

The task of gd-plan construction is to determine one single gd-plan for the subsequent control interval. This gd-plan has to be consistent with all related motivations. The single consistent gd-plan is now to be determined through an optimum search in the combined utility function. The definition of the admissibility functions (definition 6) determines the possible choice of optimum search algorithm. A definition of the range of $L(p_{gd})$ as the binary set $\{0, 1\}$ simplifies the calculations significantly as opposed to the definition with a range as the continuous interval [0, 1]. First of all, the binary range of the limit map supports a very simple definition of the combined utility

$$U(p_{gd}) \equiv \begin{cases} 0 & \forall \quad L(p_{gd}) = 0 \\ T(p_{gd}) & \forall \quad L(p_{gd}) = 1 \end{cases} . \tag{4.19}$$

Assuming that the target map is described as a differentiable function the set of local optima $\mathcal{P}_{opt}$ in $T(p_{gd})$ can be determined analytically. Two cases are possible:

- An optimum of $T(p_{gd})$ determined by setting the derivative to zero. This is true as long as the extrema ly inside the addmissible domain.
- The optimum lies on the border $\partial\mathcal{A}_{gd}$ of the addmissible domain.

This motivates a definition of admissibility functions directly in terms of borderlines $\partial\mathcal{A}_{gd}$ of the admissible domain of gd-plans. Practically, in the two-dimensional space a simple polygon chain may represent the limit map.

If the range of the limit map is left to be continuous, then things require much more concentration on the design of the combination operator. As long as target map, limit map and the combination procedure relies on differentiable functions, the optimum search in $U(p_{gd})$ may be accomplished straightforwardly. However, the limit map has an influence on the shape of the combined utility function. To ensure that the total utility function soundly combines all target and fear motivations, the shapes of the functions have to be precisely defined, i.e. they have to express the existing preference relationships in the space of gd-plans correctly. Non-binary addmissiblity functions require much more sophisticated algorithms to determine the target and the limit map.

Even with a binary range of $L(p_{gd})$, though, the shape of the target map is important if the space of gd-plans is multidimensional. It is by the shape of the target map that preferences in precision between the parameters of the gd-plan are expressed. This is best explained by an example: figure 4.8 demonstrates how two different shapes of a target map, that are based on the same target gd-plan, produce a different preferred gd-plan p_{opt}. Since the target gd-plan lies outside the admissible domain the optimum of the function has to be searched on the border $\partial\mathcal{A}_{gd}$ of the allowed domain. In figure 4.8a $T(p_{gd})$ decreases very slowly with the distance in rate of curvature change $\dot{v}$ as expressed by the equipotential lines. In consequence, the rate of curvature change of the preferred gd-plan p_{opt} is very similar to the one of the gd-plan of the target pillar. Figure 4.8b however shows a function $T(p_{gd})$ that decreases very slowly with the distance in $\dot{\kappa}$ which causes that the acceleration of the chosen gd-plan's acceleration is very similar to the one of the target gd-plan.

4.5 Conclusion

The method of target and limit maps establishes a way to describe the interference of motivations with the current state of the system to be controlled. Motivations are classified into two basic types:

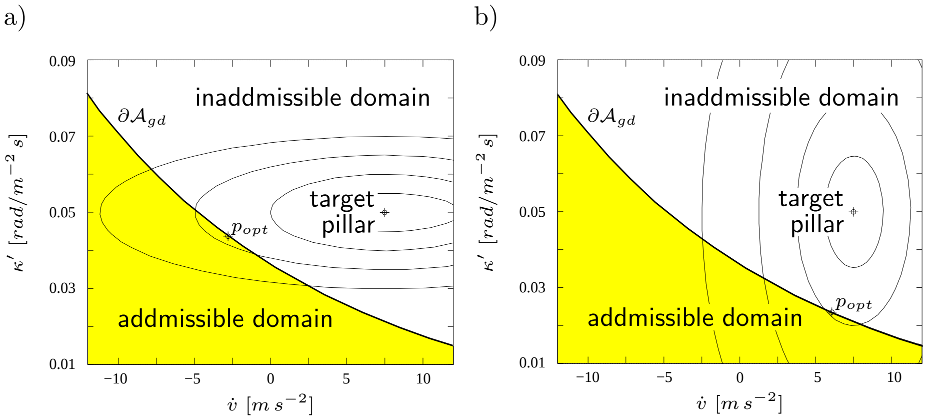


Figure 4.8: Influences of shapes of utility function on the choice of single consistent gd-plan from the example of vehicle control.

- Desire related motivations introduce preference relationships in the space of gd-plans based on a specific preferred gd-plan. They are related to performance of the controller. Each desire motivation is represented by a constructiveness function.

- Fear motivations introduce preference relationships for subsets of gd-plans striving to avoid inadmissible system states. A fear motivation is represented by an admissibility function.

In the general case, multiple fear and desire motivations exist. Therefore, the constructiveness and admissibility functions have to be combined into one target map $T(p_{gd})$ and one limit map $L(p_{gd})$.

The characteristics of both maps are summarized in table 4.1. For both types of functions limits in the range can be defined. The limiting operators for the combination of admissibility and constructiveness functions show a certain symmetry. The limit map as a result of multiple fear motivations has to be less or equal to the minimum operator applied on all related admissibility functions. The target map has to be greater or equal to the maximum operator applied on all related constructiveness functions. Based on the restrictions on the ranges of the limit map and the target map, the multiplication operator is a valid candidate for a combination of admissibility functions. On the other hand, the summation operator is a valid candidate for the combination of constructiveness functions.

Finally, it was discussed how the limit map and the target maps have to be combined. It was derived that for binary admissibility functions, it is advantageous to describe the limit map by border lines of the admissible domain, rather than a utility function. Based on the combined utility function, an optimum search has to find the gd-plan that is consistent with all related motivations.

It must be mentioned that the method of target and limit maps is not the ultimate way to do gd-plan construction. Where it is possible to define one

Table 4.1: Overview on modeling of fear and desire motivations.

	Fear Motivations	**Desire Motivations**
Basic Idea	Fears - Avoiding *sets* of disastrous states.	Desires - Striving for a *specific* state.
Functions	Admissibility functions $A_i(p_{gd}) : \mathcal{P}_{gd} \rightarrow [0, 1]$.	Constructiveness functions $C_i(p_{gd}) : \mathcal{P}_{gd} \rightarrow [0, \infty)$.
Restriction on map	$L(p_{gd}) \leq \min_{i=0,1,2,\ldots} A_i(p_{gd})$	$T(p_{gd}) \geq \max_{i=0,1,2,\ldots} C_i(p_{gd})$
Example	$L(p_{gd}) = \prod_{i=0}^{N} p_i\left(A_i(p_{gd})\right),$ with $p_i(x) \leq x$.	$T(p_{gd}) = \sum_{i=0}^{N} q_i\, C_i(p_{gd}),$ with $q_i \geq 1$.

single algorithm or one single formula that combines all aspects of motion planning, this approach is preferable to the method of target and limit maps. However, problems where multiple physical aspects are involved in gd-planning or problems of a higher-dimensional space of gd-plans, are most likely difficult to describe in one single formula. Here, the method target and limit maps carries a set of advantages:

- Aspects of admissibility and aspects of performance are discussed independently. This allows to work on performance (i.e. target map construction) without having to worry about admissibility (i.e. limit map construction) and vice versa.
- The understanding of gd-plan construction as a process that combines multiple motivations is very intuitive.
- The ability to model an unlimited amount of different concepts of gd-planning in an isolated manner supports a conquest of the problem by division into its different aspects.

In summary, the method of target and limit maps provides a deterministic and precise method to accomplish gd-plan construction. The way how the influences of motivation on the preference structure over the space of gd-plans is modeled, is reminiscent of processes related to human motivations.

Chapter 5

The Geometric Dynamic Planning Unit

In the chapter dealing with *circumstance cognition*, i.e. plan-to-action mapping, it was discussed how to determine the appropriate control parameters c_p for a given gd-plan p_{gd}. This chapter, now, discusses how to determine a gd-plan which is consistent with all related design specifications. This process was previously identified as *motivation matching*. Using the method of target and limit maps it is possible to implement algorithms that model the interference of motivations with informations about the environment resulting in one distinct gd-plan.

Where the previous chapter was theoretical in nature, this chapter applies the introduced concepts to vehicle control. The problem of vehicle control is related to a variety of different concepts, such as target point search, compliance of a maximal lateral acceleration, boundaries on curvature and so on. It is shown, how target and limit maps can be applied to deal with each of these problems separately. In particular, the investigation on target point search demonstrates how subtle shortcomings related to purely geometric-dynamic issues can be identified and expunged.

5.1 Overview

Previously, the situational driving motivation SDM was defined as a guideline for gd-plan construction (definition 2, page 10). When applied to vehicle control it results in multiple secondary concepts as they are: target point search, short term curvature profile compliance, compliance of the curvature profile of the nominal course, minimum and maximum rates of curvature change, and optional velocity constraints. To combine all these concepts in a suitable way target and limit maps are used. The structure of the gd-plan construction unit is shown in figure 5.1.

Environment information about course geometries (nominal course $\vec{n}(p)$,

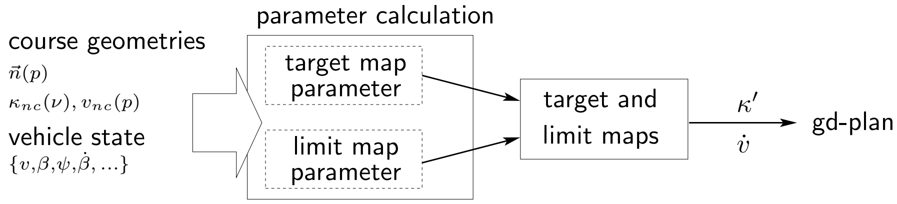


Figure 5.1: Overview over gd-plan construction unit for vehicle control.

$\vec{\kappa}_{nc}(p)$ and $\vec{v}_{nc}(p)$) and the vehicle state $(v, \beta, \psi, \dot{\beta}, \ldots)$ are now used for the determination of parameters related to each concept. With these parameters the shapes of the target and the limit map are given and the method of target and limit maps can be applied. This results, finally, in one distinct gd-plan (κ', $\dot{v}$) that is consistent with all concurrent motivations involved.

5.1.1 Parameter Calculation

The concept that most obviously plays a significant role in gd-plan construction is the target point, i.e. a point that lies a certain distance ahead and that the driver strives to surpass. Striving to drive through a target point is supposed to keep the vehicle close to the nominal course. Since striving to drive through a target point results in a specific preferred gd-plan it corresponds to a desire motivation (definition 8, page 40). In figure 5.2a it is shown how to calculate the required rate of curvature change κ'_t and the required acceleration $\dot{v}_t$ to pass through a target point.

First of all, a target point must be found that is appropriate for the current situation. This is explained in section 5.2. Second, a rate of curvature change $\kappa'_t = \frac{d}{ds}\kappa$ is determined in order to specify a trajectory through the target point (section 5.3). Third, to prevent the vehicle from becoming uncontrollable, a certain maximum lateral acceleration $\ddot{y}_{max,curv}$ has to be respected when driving the desired trajectory, as expressed by κ'_t. The process for determination of a maximum acceleration $\dot{v}_t$ to respect the limit $\ddot{y}_{max,curv}$ is illustrated in section 5.4. At this point, the gd-plan (κ'_t, $\dot{v}_t$) specifies the preferred gd-plan of the target point search algorithm. It is the position of the 'pillar' in the target map.

On the other hand, there are concepts related to fear motivations (definition 5, page 38). In order to construct a limit map, methods have to be found that allow its parameterization. Figure 5.2b shows how the parameters of the limit map are calculated. It is assumed that the SDM-goal of maximizing velocity has a high priority. By consequence the SDM-goal to minimize lateral acceleration is reduced to the goal of respecting a maximum lateral acceleration limit. This goal expresses the 'fear' of losing control of the vehicle and is thus a limit motivation. The lateral acceleration limit $\ddot{y}_{max,course}$ together with the curvature profile of the nominal course results in a velocity profile by $v_{nc}(p) = \sqrt{\ddot{y}_{max,course}/\kappa_{nc}(p)}$. Now, a maximum acceleration can be determined in order to respect this velocity

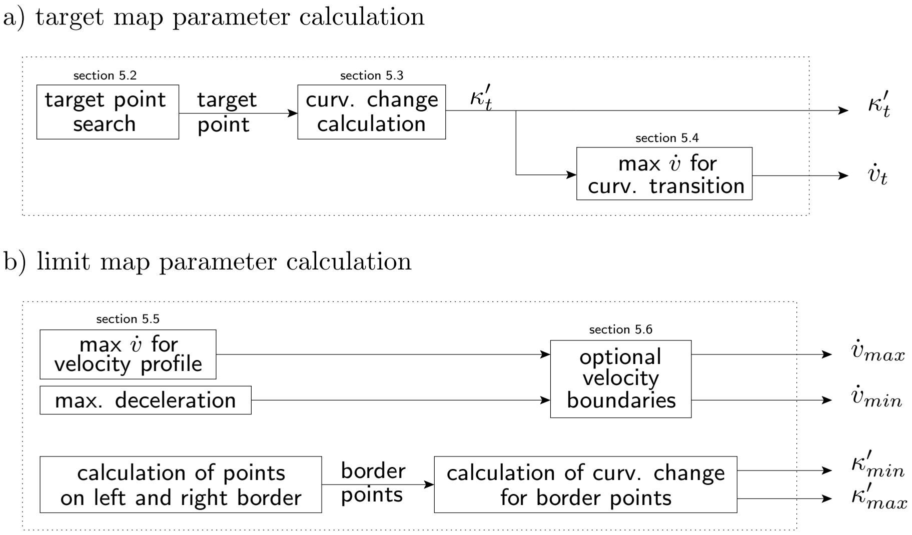


Figure 5.2: Calculation of parameters required to construct target and limit maps.

profile. The method for doing this is explained in section 5.5.

To introduce lower limits on acceleration resulting from wheel slip, for example, the maximum deceleration can be specified by the user. For some maneuvers it is necessary to define additional lower and upper borders of velocity. The way how optional velocity restrictions cut the admissible domain of accelerations is explained in section 5.6. Finally, the output of this module defines the acceleration boundaries $\dot{v}_{min}$ and $\dot{v}_{max}$.

Limits on rate of curvature change are calculated in a very simple way. First, a position that lies a certain preview distance $t_p\, v$ ahead on the nominal course is determined. From there one moves a distance ϵ_b to the left and the right to get two points representing the left and the right border. The rates of curvature change that are needed to pass through these two points establish the limits κ'_{min} and κ'_{max} for the rate of curvature change.

The following sections focus on the algorithms for the parameter calculation. Then, in Section 5.7 the application of target and limit maps is discussed. There, it is explained how the calculated parameters are used to shape the target map and the limit map.

5.2 Target Point Search

As shown in figure 5.2a, the parameters of the target map are computed based on the concept of a target point. The aim of this procedure is to define constant rate of curvature change κ'_t and an acceleration $\dot{v}_t$ for the subsequent control interval which is consistent with all related physical and geometrical concepts. This section treats the details of target point search (see also [Schaefer et al., 2000]). A target point must be defined in such a way that

- the resulting trajectory is physically feasible for the short and long term and
- the vehicle is guided as close and as fast as possible along the track.

The *situational driving motivation* SDM (page 10) provides a direct guideline for the design of a trajectory. According to the SDM, trajectory design has to take into account the minimization of lateral deviation, angular deviation, lateral acceleration and maximization of speed.

This section first introduces three target point search methods that are intuitively practicable but do not explicitly imply the SDM. The first one, *Preview Point*, is based on a specific element of the vehicle state: velocity. The second one, *End Of Sight*, is an approach based solely on course geometries. A third method uses lateral and angular deviations from the nominal course to adapt the preview time of the Preview Point method. It is therefore called *Deviation Dependent Preview Point*. All of them show an unsatisfactory performance. Their malfunction is then discussed by means of the SDM. Before new target point methods are introduced a fundamental error is identified that causes all three target point methods to fail.

As a consequence, yet another target point method is developed that directly uses the SDM as a guideline. This method is termed *Nestle Curves*. It immediately causes a higher performance that is much less dependent on the specific maneuver.

The quality of each of the methods was determined through simulation with the absolute value of lateral deviation from the nominal course as the measure for precision. Simulation experiments presented below were accomplished with a vehicle model as described in appendix B. A test course, named 'Moby Dick' (figure 5.3), contains a variety of different maneuvers for investigations of driving behavior. The parameter settings are always chosen so that the driver model has to manage situations with lateral acceleration of about $6 - 8\,m\,s^{-2}$.

The experiments that are mentioned here are, of course, only reproducible with a structure of a driver model as introduced in this dissertation. However, the results of the investigations are *independent* of the control structure that is used. This is because the trajectory of any vehicle can be approximated by a constant rate of curvature change over small control intervals[1].

The extensive discussion on target point search demonstrates the advantage of dealing with geometric-dynamic problems without dealing with the state space equation of the plant explicitly. The treated concepts are rather physical than mathematical in nature. This results in simple formulae and short intuitive algorithms for the processes involved.

[1] The control intervals used in the developed driver model were chosen between 0.02 seconds and 1 second.

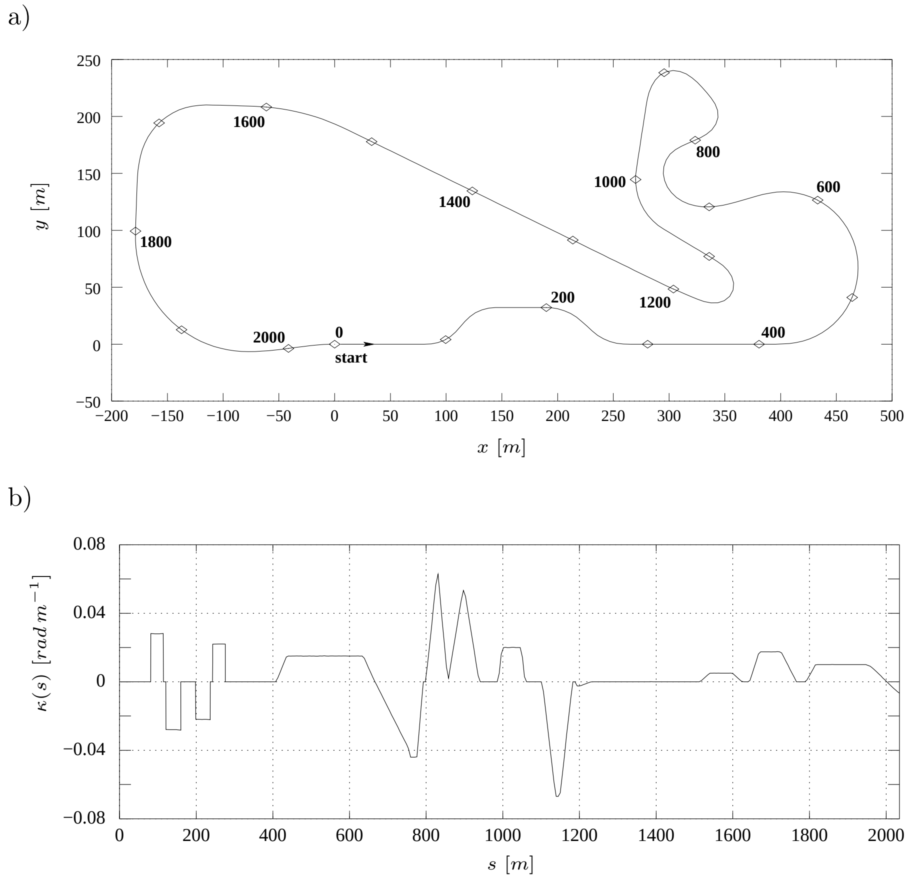


Figure 5.3: The fantasy course 'Moby Dick' designed as simulation environment. a) x- and y-coordinates. b) Curvature profile $\kappa(p)$.

5.2.1 Intuitive Target Point Search Methods

Preview Point

The first target search method to be examined is Preview Point, which is by far the simplest and most intuitive target point search method. This or similar methods are used in many driver models[2] (e.g. [Kondo and Ajimine, 1968] and [Voegel, 1997]). In this target search, first the point $\vec{n}(p_0)$ on the nominal course has to be determined that is nearest to the vehicle's c.g. From this position one looks a distance ahead on the course, equal to the current velocity v multiplied by some specified preview time t_p, and places the target point $\vec{n}(p_p)$ there (figure 5.4). Preview Point makes sense, considering the parallels in the way human drivers consider their view points. A human driver would look far ahead on a road when he is moving at a high velocity, such as on a straight-away or in a curve of low curvature, but would concentrate closer in front when greater road curvatures require a lower speed.

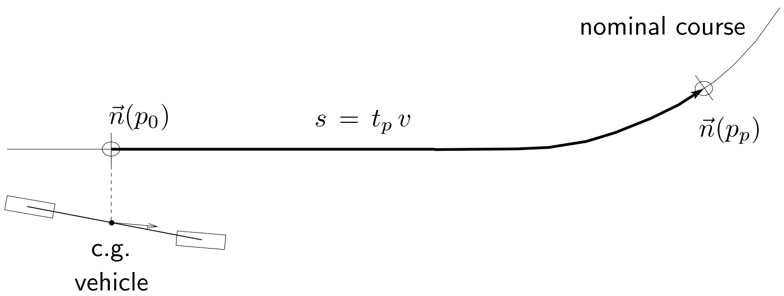


Figure 5.4: Preview Point search method.

Definition: 12 (Preview Point) *Let $\vec{n}(p_0)$ be the nearest point to the vehicles position on the nominal course $\vec{n}(p)$. Then a* Preview Point *is the point $\vec{n}(p_p)$ ahead on the nominal course, where*

$$\int_{p_0}^{p_p} |\, \vec{n}'(p)\, |\, dp \;=\; t_p\, v\,. \tag{5.1}$$

With the given target point, the appropriate rate of curvature change can now be determined in order to pass through it. This way, the desired trajectory is determined.

[2] In fact, in the mentioned driver models the target point is not used to calculate a desired trajectory in terms of a desired rate of curvature change. In [Voegel, 1997], for instance, the target point is used to define a desired lateral displacement. Still the term target point is referred to as a point to be passed through.

Consider the results from a simulation run shown in figure 5.5, where the absolute lateral deviation from the nominal course is depicted with respect to travel distance on the course. Hand-optimized parameters are used in order to achieve precision in path following. The control interval sizes T_c were chosen to 0.4 and 0.2 seconds. It is shown that smaller control intervals do not cause an observable increase in precision. On the contrary, using a control interval of $T_c = 0.2\,s$ causes frequent deviations that are higher than $1\,m$. In any case, the deviation vacillates around $0.4\,m$, which is not acceptable. The following reasons can be suspected to cause this unsatisfactory behavior:

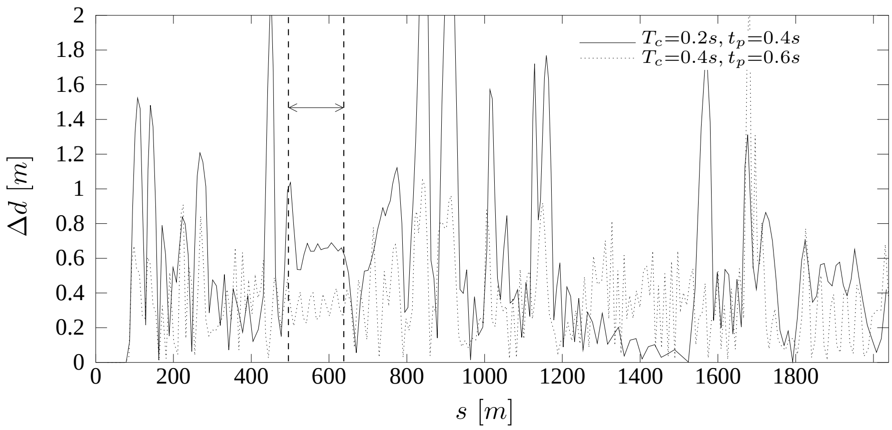


Figure 5.5: Absolute value of lateral deviation from nominal course using the Preview Point target point search method.

- The Preview Point method does not take into account the lateral deviation of the trajectory that is laid out to pass through the target point. If the velocity is high, then the target search picks a target point $\vec{n}(p_p)$ that is far ahead on the course as shown in figure 5.6a. In the case of high road curvatures, the vehicle's position $\vec{p}_2$ after the next control interval will be even further removed from the nominal course than the actual position $\vec{p}_1$. In the figure, it is shown how this problem causes the vehicle to cut the curve. However, it is the same way, possible that the choice of a distant target point results in planned trajectories that cause the vehicle to drive outside the curve.

  In segments on the course, where curvature changes quickly this leads to a gradual return to the nominal course. In segments of constant curvature this leads to a more or less constant deviation as for example in between 500 to 650 meters in figure 5.5.

  These errors demonstrate the fact that the Preview Point method does not explicitly conform to SDM. Namely, it does not ensure a minimization of the lateral deviation.

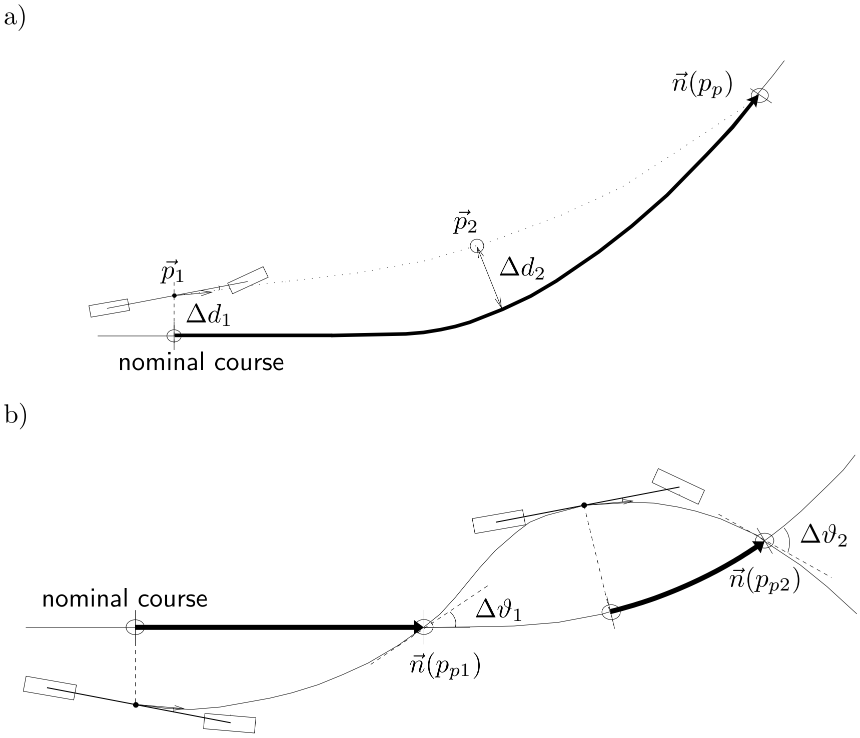


Figure 5.6: Two problems with Preview Point. a) The target point lies too far in front. b) The target point comes too close.

- A second failure of Preview Point is that it does not take into account the angular deviation $\Delta\vartheta$ between the planned trajectory and the nominal course. This becomes a problem if the distance of the target point ahead of the vehicle becomes short, as shown in figure 5.6b. Target points that lie extremely close ahead, together with some deviation from the nominal course, lead to high rates of curvature change to pass through them. This can cause extremely high curvatures of the trajectory and therefore ends in exceeding the maximum lateral acceleration.

  This problem can be attributed to a conflict with the SDM. The Preview Point method does not ensure a minimization of the angular error $\Delta\vartheta$ and does not consider a maximum curvature to restrict lateral acceleration.

The preview time t_p is the only means to configure the target search behavior. To prevent the first problem, the preview time t_p would have to be decreased. Avoiding the second problem requires the increase of preview time. Both rules are contradictory and necessitate the development of new concepts for target point search.

End of Sight

The Preview Point method is mainly oriented towards the vehicle state, given by the current velocity. Alternatively, a concept is developed in this dissertation, that is purely *geometry-oriented*. The idea is that the target point must be placed at a position ahead on the nominal course where the course itself becomes too unpredictable. This is done using an arc of constant radius to approximate and 'predict' the nominal course until a certain distance ahead. The minimal distance where the approximation deviates too much from the nominal course defines the target point. The following paragraphs define this method precisely as the End of Sight concept for target point search.

For this method, an arc of constant curvature is calculated that passes through the point on the nominal course $\vec{n}(p_0)$ closest to the vehicle, is tangent to the course at this point, and subsequently passes through a point $\vec{n}(p_c)$ ahead on the course. The point $\vec{n}(p_c)$ is moved as far ahead as possible while the error between the midpoint of the approximation and the nominal course is less than some specified error ϵ_d. The forward-most point $\vec{n}(p_c)$ where the error limit ϵ_d is still not exceeded becomes the target point as shown in figure 5.7.

Definition: 13 (End of Sight Point) *Let $\vec{n}(p)$ be a parameterization of the nominal course and $\vec{n}(p_0)$ be its nearest point to the vehicle's c.g. Then let $\vec{c}(q)$ be a circle that approximates the current nominal course to point $\vec{n}(p_c)$ with following constraints:*

$$\vec{c}(q_0) = \vec{n}(p_0)\,, \quad \vec{c}(q_1) = \vec{n}(p_c) \tag{5.2}$$

and

$$\alpha(\vec{c}(q_0)) = \alpha(\vec{n}(p_0)), \tag{5.3}$$

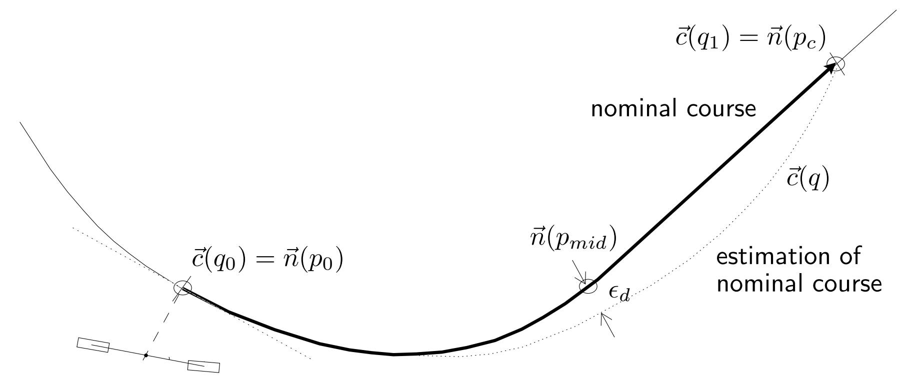


Figure 5.7: End of Sight target point search method.

where $\alpha(\vec{c}(q_0))$ indicates the angle of the circle at q_0 and $\alpha(\vec{n}(p_0))$ the angle of the tangent to the nominal course at p_0. Let $\vec{n}(p_{mid})$ be at the middle of the stretch between $\vec{n}(p_0)$ and $\vec{n}(p_c)$. Let $dist(\vec{c}, \vec{n}(p_{mid}))$ be the distance between $\vec{n}(p_{mid})$ and the nearest point on the circle. The END OF SIGHT POINT *$\vec{n}(p_{eos})$ is defined by*

$$dist(\vec{c}, \vec{n}(p_{mid})) > \epsilon_d, \tag{5.4}$$

$$\int_{p_0}^{p_{eos}} |\,\vec{n}'(p)\,|\,dp = min. \tag{5.5}$$

With the End of Sight method the target is placed near the end of a segment of the course that has an approximately constant curvature. For instance, if the vehicle is on a long curve of constant curvature, the target is chosen at the end of this curve where the curvature begins to decrease. This behavior can be interpreted as striving to bring the vehicle on a trajectory of constant curvature. Since the front wheel angle and trajectory's curvature are stationary proportional this corresponds to a driver striving to hold the front wheel constant. It theoretically has the effect of eliminating unwanted oscillations. Furthermore, the target is placed very close to the vehicle if the course ahead of it has a high rate of curvature change, and thus a high degree of difficulty. Despite these theoretical advantages, however, the End Of Sight method does create significant errors.

Results of a simulation run with control interval sizes $T_c = 0.4\,s$ and $T_c = 0.2\,s$ can be observed in figure 5.8. Parameter ϵ_d is optimized by hand in order to improve precision of control. As predicted theoretically, this method enables the vehicle to stabilize quickly in a segment of constant curvature, such as the stretch between 400 to 500 m. Much more obvious, however, is the poor performance on the straight-ways at $300 - 450\,m$ and $1200 - 1600\,m$. Here, a fundamental error appears that has not been previously discussed. This primarily results from the fact that the target point is chosen too far ahead, so that the trajectory

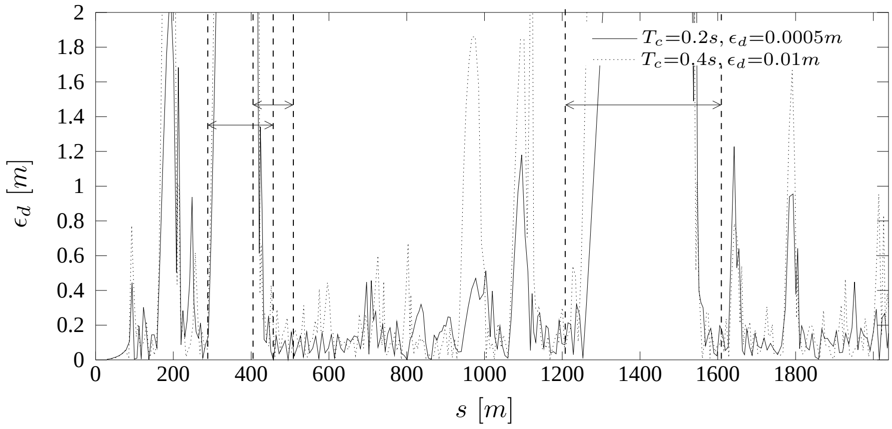


Figure 5.8: Absolute value of lateral deviation of the vehicle's trajectory from nominal course using the End Of Sight target point search method.

that it tries to pass through is not reasonable. A detailed discussion of this phenomenon, however, is addressed in section 5.2.2.

Other errors can be explained by similar causes as mentioned in the discussion concerning Preview Point. When curvature changes too abruptly, than it is difficult to find an approximation of the nominal course, and target points may approach to much. This explains the high errors in between $150 - 280\,m$ and $1050 - 1150\,m$. Again this method does not include a way to prevent high angular deviation. Further, there is nothing that ensures that the chosen trajectory does not cause the lateral acceleration to get so high that the vehicle is no longer manageable. The disregard of the goals of the SDM again leads to unreliable system behavior and unsatisfactory results.

Deviation Dependent Preview Point

A third search method, called *Deviation Dependent Preview Point*, was derived, as the name says, from *deviations*. It adapts the preview time t_p to a given situation by a simple formula. Two coefficients increase the preview time proportionally to lateral and angular deviations. This is to prevent the target point from coming too close, if one of those deviations becomes high. For the case that both deviations are close to zero, the target point lies close ahead on the nominal course. Since, in this case, both the vehicle and the target point lie close together on the nominal course, the trajectory through the target point can be planned very precisely. This is the theoretical idea behind this method. The definition of the method follows:

Definition: 14 (Deviation Dependent Preview Point) *Let Δd be the absolute value of the vehicle's current lateral deviation from the nominal course, let $\Delta\vartheta$ be the absolute value of the current angular deviation from the nominal* *course. A* Deviation Dependent Preview Point $\vec{n}(p_p)$ *is a point that lies a certain distance* $t_p \cdot v$ *ahead correspondent to*

$$\int_{p_0}^{p_p} |\, \vec{n}'(p) \,|\, dp \;=\; t_p \, v \,. \tag{5.6}$$

The preview time t_p *is given by*

$$t_p \;=\; c_1 \;+\; c_2 \, \Delta d \;+\; c_3 \, \Delta\vartheta. \tag{5.7}$$

with

$$c_1 \;\geq\; 0. \tag{5.8}$$

The parameters c_2 *and* c_3 *may be chosen arbitrarily.*

The coefficient c_1 has to be positive, because otherwise the target point would lie backwards, if lateral displacement Δd and angular error $\Delta\vartheta$ are both zero.

Results from a sample simulation with two optimized parameter sets for $\{c_1, c_2, c_3\}$ that produce minimal deviation for control intervals $T_c = 0.4$ and $T_c = 0.2$ are shown in figure 5.9. The graphs are produced based on optimized coefficients c_1, c_2 and c_3. The comparison with figure 5.5 makes it clear that no improvements were achieved. By a simple formula, like (5.7), it is apparently not possible to incorporate all required relationships that have to be considered to choose an appropriate target point. The results of the last three sections indicate that a much more systematic approach is required to handle target point search.

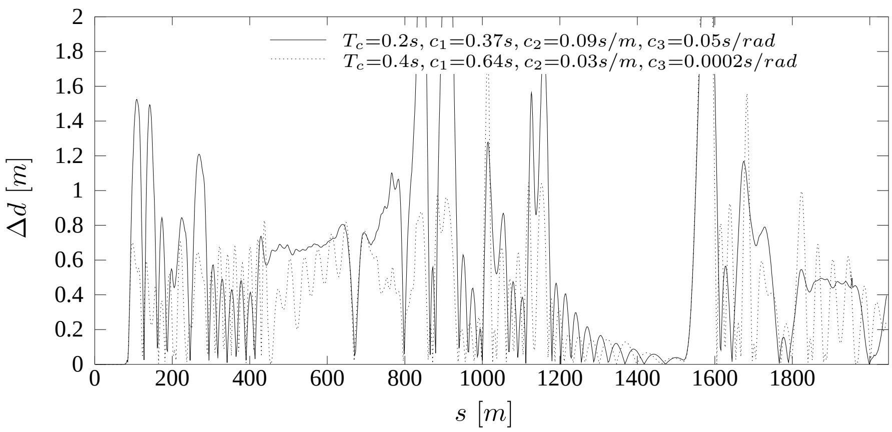


Figure 5.9: Absolute value of lateral deviation from nominal course using the Deviation Dependent Preview Point search method.

5.2.2 Fundamental Problem with Target Point Search

At this point, it is beneficial to search for a principal error in these target point search methods. There is, in fact, one fundamental problem that causes many inaccuracies:

> The target point is being used in a way that violates the assumptions on which it was founded. A target point is a means to provide a constant rate of curvature change κ' for exclusively the *subsequent* control interval. However, each of the previously mentioned target search methods indirectly uses the target point concept to make plans with constant rates of curvature changes that span much longer than one control interval, and this leads to disaster.

Consider the situation shown in figure 5.10. The target point is placed at the end of the straight-away as desired, theoretically to eliminate oscillations. What happens instead is that the driver makes a plan which spans several control intervals into the future based on a constant rate of curvature change κ' and the initial curvature κ_0. Of course, only the portion of this planned trajectory that spans the subsequent control interval is actually driven (see point at '$v\,T_c$'), after which a new target is chosen. Nevertheless, it is still laid out as a segment of a longer plan.

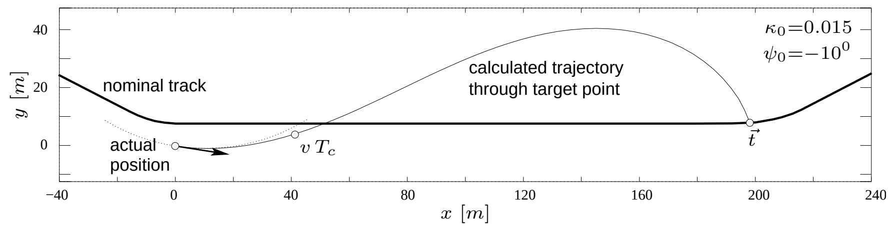


Figure 5.10: Mathematical problem with a target point $\vec{t}$ that lies much further ahead than one control interval. The planned trajectory is a swooping arc.

Furthermore, it is the disregard of the rules of the SDM that caused the previous three target point search methods to fail. Figure 5.11a illustrates the disparity between a geometric plan resulting from a target point $\vec{t_p}$ using the Preview Point method (dotted line) and an idealized geometric plan (solid line), which guides the vehicle smoothly back to the nominal course. The geometric plan that results from the Preview Point target $\vec{t_p}$ does not respect angular deviations and leads to very high curvatures (figure 5.11b). High curvatures produce high lateral accelerations that threaten the stability of the system. Smoothly guiding back means that all rules of the SDM are striven to be attained. Driving along the solid line the vehicle approaches the nominal course and turns so that it is parallel to the course angle. At the same time, the geometric plan does not contain high curvatures so that the lateral acceleration to drive it is relatively small.

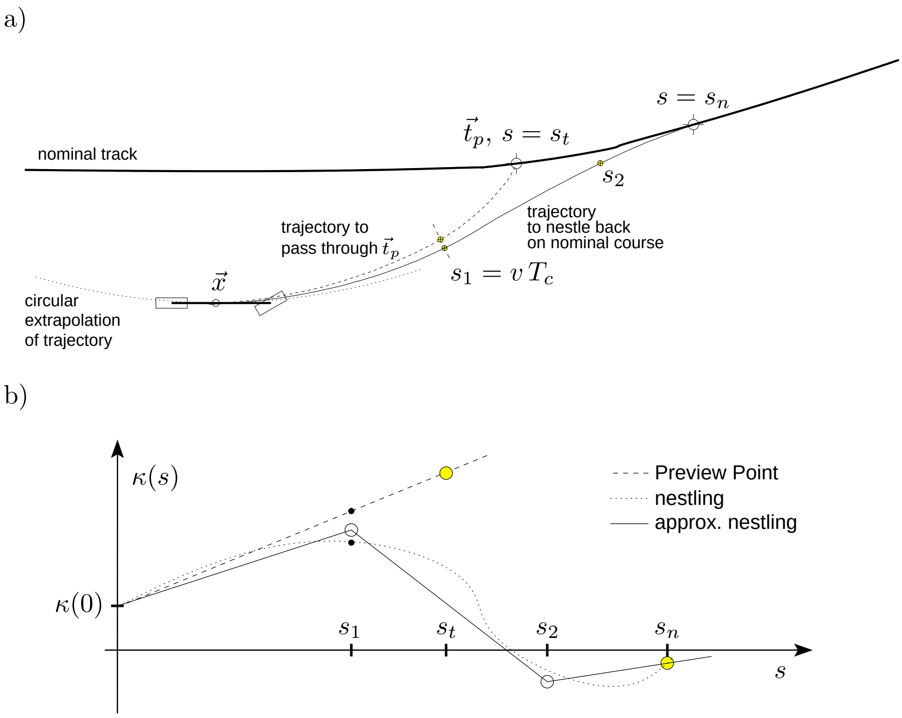


Figure 5.11: a) Comparison of an ideal trajectory and a trajectory derived from a Preview Point target on the nominal course. b) Comparison of curvature profile of geometric plans resulting from Preview Point, an ideal maneuver to recover the nominal course and its approximation by segments of constant rate of curvature change.

It is clear that the curvature profile of this ideal trajectory can not be reached by one single segment of constant rate of curvature change. However, figure 5.11b shows how, properly used within multiple control intervals, a sequence of target points at s_1 and s_2 can very closely approximate the previously mentioned idealized behavior. It is therefore necessary to formulate an improved target point search method, which is capable of properly placing the target point to achieve accurate control. This target point does not necessarily lie on the nominal course.

5.2.3 Systematic Target Point Search Methods

Nestle Curves of 1^{st} Order

'Nestle Curves' are a systematic target point search method developed in this dissertation. It takes into account the lateral and angular deviation of the vehicle to form a geometric plan that allows a smooth return to the nominal course. It is as direct as possible and explicitly satisfies three requirements of the SDM. To accomplish this, a smooth curve, parameterized by q, is interpolated between the vehicle and the nominal course subject to the following constraints:

- The curve starts at $q = 0$ in the c.g. of the vehicle and ends at $q = 1$ somewhere ahead on the nominal course.

- It is tangent to the vehicle's actual trajectory at $q = 0$ and to the nominal course at $q = 1$.

- It does not contain a curvature that causes, together with the current velocity, a higher lateral acceleration than a predefined limit $\ddot{y}_{max,nestle}$.

- It is as short as possible, i.e the point on the nominal course lies as closely as possible ahead of the vehicle.

In this way, the lateral and angular errors are reduced as quickly as possible without exceeding the lateral acceleration limit. Once this curve is found, then the target point is placed some specified preview time t_p, theoretically in the range of one control interval T_c, ahead on this curve. It is now assumed that the nestle curve is a possible trajectory of the vehicle. Choosing a point on the nestle curve makes it therefore plausible, that the vehicle follows the curve precisely when driving through the target point. This was not necessarily the case for the intuitive target point methods. A schematic picture of this idea is shown in figure 5.12.

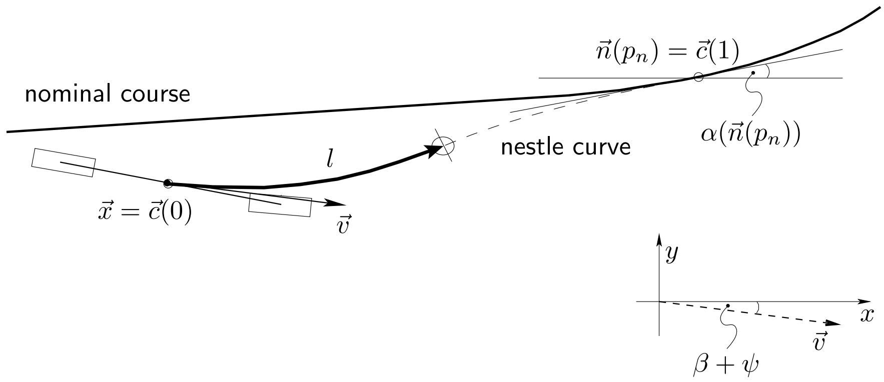


Figure 5.12: Nestling to the nominal course.

Definition: 15 (1st Order Nestle Curve) *Let $\vec{x}$ be the actual position of the vehicle's c.g., ψ and β its yaw and slip angle, v its velocity and $\ddot{y}_{max,nestle}$ the desired maximum lateral acceleration. Let $\vec{n}(p)$ denote the nominal course. A* Nestle Curve *is defined as a continuous curve $\vec{c}(q)$, $q \in [0,1]$ that has the following properties:*

$$\begin{array}{rclrcl} \vec{c}\,(0) & = & \vec{x}, & \vec{c}\,(1) & = & \vec{n}(p_n), \\ \alpha(\vec{c}(0)) & = & \beta + \psi, & \alpha(\vec{c}(1)) & = & \alpha(\vec{n}(p_n)), \end{array} \tag{5.9}$$

so that

$$\max_{q \in [0,1]} \{|\kappa(\vec{c}(q))|\} \;\leq\; \kappa_{lim} \;\equiv\; \frac{\ddot{y}_{max,nestle}}{v^2} \tag{5.10}$$

and

$$\int_0^1 |\vec{c}\,'(q)|\, dq = min. \tag{5.11}$$

Nestle curves are restricted to no more than two sign changes in curvature from $\vec{c}(0)$ to $\vec{c}(1)$.

As a particular implementation of nestle curves, the following mathematical parameterization of the form $\vec{c}(q) = (x_c(q)\, y_c(q))^T$ is used. The problem is first transformed into a local coordinate system with the origin in the vehicle's c.g. and the x-axis along the velocity vector. In this coordinate system, the conditions (5.9) transmute to

$$\begin{array}{rclrcl} x_c(0) & = & 0, & y_c(0) & = & 0, \\ y_c(1) & = & t_x, & y_c(1) & = & t_y, \end{array} \tag{5.12}$$

$$\begin{array}{rclrcl} tan(\alpha(\vec{c}(0))) & = & 0 & \Rightarrow & y_c'(0) & = & 0, \\ tan(\alpha(\vec{c}(1))) & = & \alpha(\vec{n}(p_n)) & \Rightarrow & \frac{y_c'(1)}{x_c'(1)} & = & \alpha(\vec{n}(p_n)) - (\beta + \psi) \equiv \xi_1. \end{array} \tag{5.13}$$

with $(t_x\, t_y)^T$ as the point $\vec{n}(p_n)$ on the nominal course, transformed into the local coordinate system. There are six constraints, and therefore, six unknown variables which are required. In order to avoid loops in the nestle curve, the x-parameterization is left as linear (since then 'x' cannot go forward and backward). The four other variables required are provided by a third order polynomial for the y-coordinate, such that the parameterization for the nestle curve becomes

$$\vec{c}(q) \quad \equiv \quad \begin{pmatrix} x_c(q) \\ y_c(q) \end{pmatrix} = \begin{pmatrix} a_x\, q + b_x \\ a_y\, q^3 + b_y\, q^2 + c_y\, q + d_y \end{pmatrix}. \tag{5.14}$$

The constraints (5.12) and (5.13) result in the following setting of the coefficients

$$\begin{array}{rcl} a_x & \equiv & t_x, \\ a_y & \equiv & t_x\, tan(\xi_1) - 2\, t_y, \\ b_y & \equiv & -t_x\, tan(\xi_1) + 3\, t_y, \\ c_y & \equiv & d_y \;\equiv\; b_x \;\equiv\; 0. \end{array} \tag{5.15}$$

The curvature of a nestle curve is given by

$$\kappa(q) \quad = \quad \frac{d}{d\,l}\alpha(\vec{c}(q)) = \left(\frac{dl}{dq}\right)^{-1} \frac{d}{d\,q} \arctan\left(\frac{y_c'(q)}{x_c'(q)}\right). \tag{5.16}$$

It follows with $dl^2 = (x_c'(q)\, dq)^2 + (y_c'(q)\, dq)^2$ and $x_c''(q) = 0$

$$\kappa(q) \quad = \quad \frac{y_c''(q)\, x_c'(q) - y_c'(q)\, x_c''(q)}{((x_c'(q))^2 - (y_c'(q))^2)^{\frac{3}{2}}} = \frac{y_c''(q)\, x_c'(q)}{((x_c'(q))^2 - (y_c'(q))^2)^{\frac{3}{2}}}, \tag{5.17}$$

$$= \quad \frac{(6\, a_y\, q + 2\, b_y)\, a_x}{(a_x^2 + (3\, a_y\, q^2 + 2\, b_y\, q)^2)^{\frac{3}{2}}}. \tag{5.18}$$

Each point ahead on the nominal course defines a specific nestle curve by which it can be nestled to the nominal course. A nestle curve is only allowed, if its maximum curvature is less then the maximum curvature $\kappa_{\lim} = \ddot{y}_{max,nestle}/v^2$. Thus, the first point on the nominal course where this condition holds defines the nestle curve. The process of searching for a suitable nestle curve in displayed in figure 5.13. Having determined the nestle curve, nestle points can easily be defined.

Definition: 16 (Nestle Point) *Let $\vec{c}(q)$ describe a nestle curve that nestles from the vehicle to the nominal course as defined in definition 15. Then a* Nestle Point *is a point $\vec{c}(q_n)$, $q_n \in [0, 1]$ that lies a certain distance $v\, t_p$ in front on the nestle curve, thus*

$$\int_0^{q_n} |\, \vec{c}'(q)\,| dq \; = \; v\, t_p. \tag{5.19}$$

t_p should be set in the range of the control interval size T_c, since the resulting curvature profile is also only laid out for one control interval.

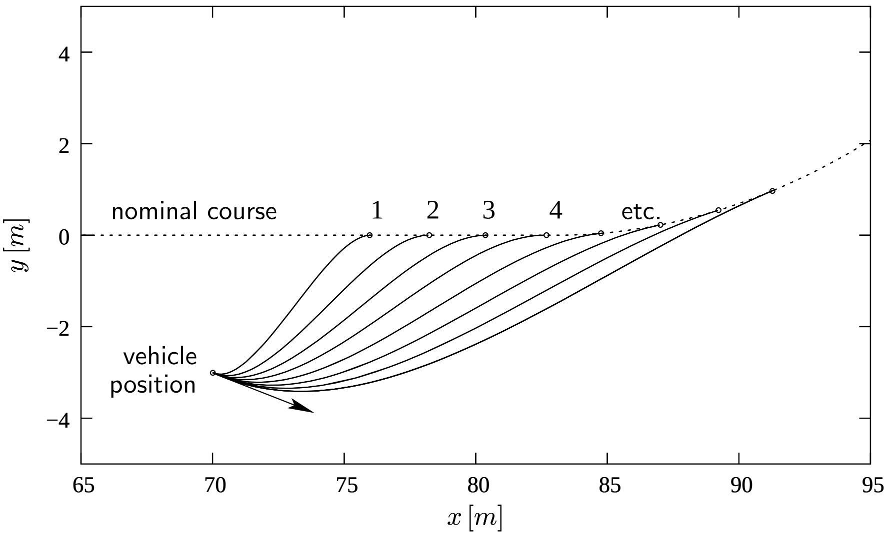


Figure 5.13: Iterative construction of nestle curves.

This search method has proven to be very powerful, independent of the requirements of different situations. Figure 5.14 shows that this method produces excellent performance for almost every maneuver on the course 'Moby Dick.' Parameters t_p and t_{pm} are optimized to minimize lateral deviation. t_{pm} indicates the minimum preview distance to look ahead on the course. Where Preview Points produce deviations that vacillate around $0.4\,m$ it is possible to achieve deviations of less than $0.05\,m$ with Nestle Curves and control interval sizes $T_c = 0.1\,s$. In addition, the deviation error seems to be *scalable* by means of the control interval size T_c. With this method the driver model is also capable of returning to the nominal course rapidly after large deviations, as for instance at $s = 1000\,m$. One must not forget that during these cornering maneuvers the lateral acceleration is large, and therefore the vehicle is very sensitive to front inputs. Finally, constant low deviations indicate that the resulting gd-plan seldom runs the plan-to-action mapper into states that are out of its manageable domain.

However, there is a potential for malfunction that comes from the fact that always the shortest nestle curve possible is chosen. It is always the curve that has the maximum curvature equal to the desired limit, even if the deviation to the nominal course is small. The resulting geometric plans, with their high curvature, prevent an increase in speed and can therefore be self-sustaining. This phenomenon can be controlled by defining a minimum preview time t_{pm} that defines a distance $v\,t_{pm}$ ahead on the nominal course from which to start searching for a nestle curve.

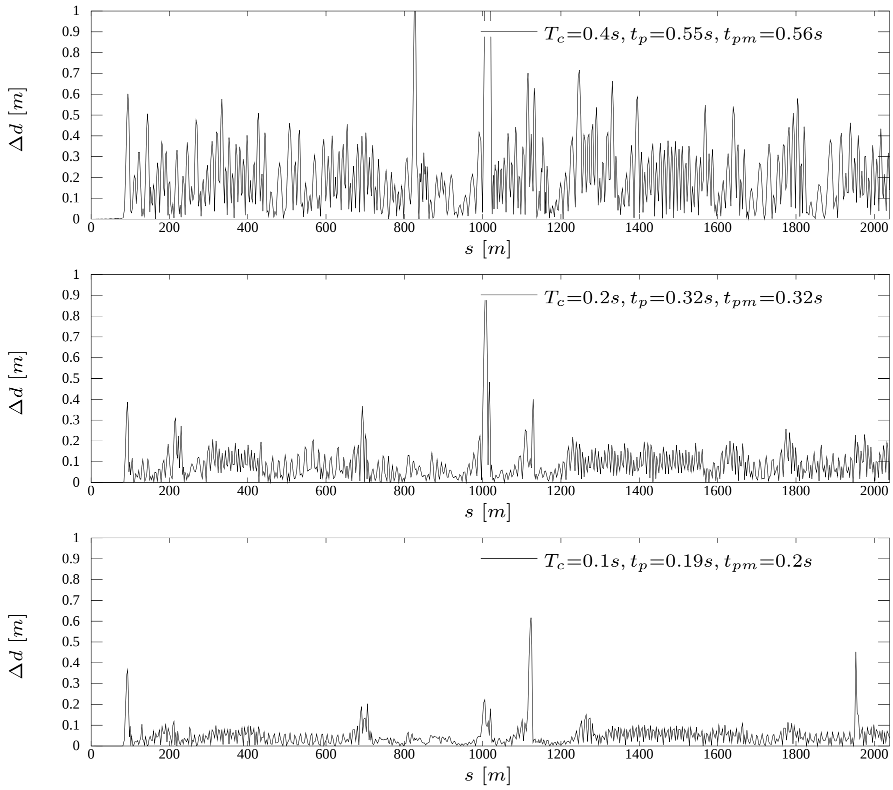


Figure 5.14: Absolute value of lateral deviation from nominal course using the Nestle Curve target point search method.

The primary problem with nestle curves, however, is the fact that in many situations, the geometric plan defined by a constant rate of curvature change through a target point on the nestle curve is not always similar to the interpolated polynomial. This problem is a result of differing initial curvatures of the trajectory and the nestle curve, and nearly always causes the lateral acceleration on the geometric plan to be higher than if the vehicle had followed the nestle curve itself. Figure 5.15 shows an example of how it is impossible for the geometric plan to be similar to the nestle curve. A second problem that arises from this issue is that the curvature of the calculated path to the target point becomes higher than expected from the shape of the nestle curve. These problems cause the majority of errors produced by nestle curves; therefore, a new mathematical definition for nestle curves with an initial curvature constraint is required.

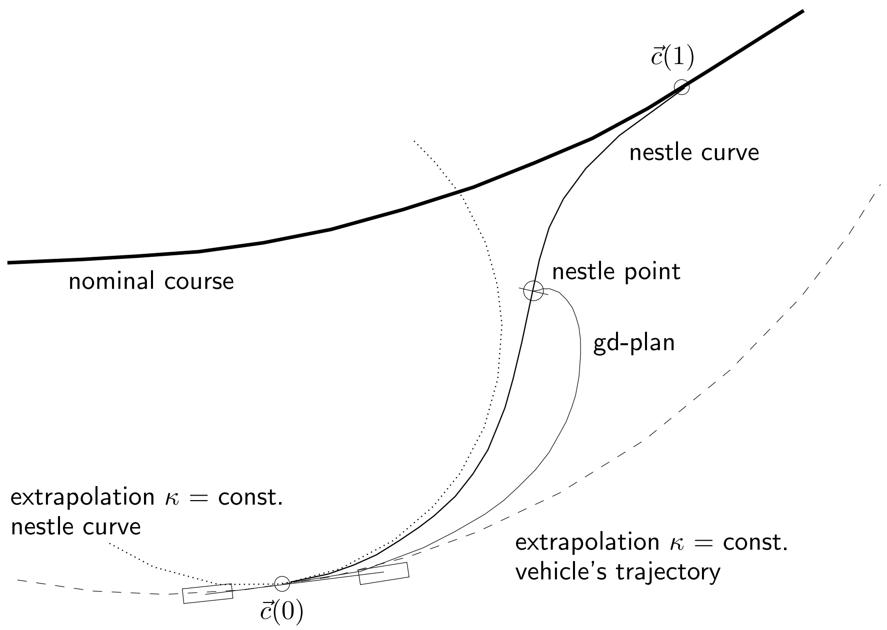


Figure 5.15: Problem with first order nestle curves.

Nestle Curves of 2^{nd} Order

It is clear that an initial curvature constraint must be placed on the nestle curve. The addition of this constraint creates what is called a 2^{nd} order nestle curve.

Definition: 17 (2^{nd} Order Nestle Curve) *A 2^{nd} ORDER NESTLE CURVE is a curve that obeys the constraints of a 1^{st} order nestle curve. Furthermore, it ensures that the curvature at the starting point of the nestle curve is the same* *as the actual curvature of the vehicle's trajectory, i.e.*

$$\frac{d}{ds}\kappa(\vec{c}(s))\bigg|_{s=0} = \kappa_0 = \frac{\dot{\psi}+\dot{\beta}}{v}, \tag{5.20}$$

where s is a variable indicating the length of the curve.

The search for an appropriate parameterization develops into a complicated affair. At first glance, only one degree of freedom has to be added to the parameterization of (5.14). The constraint that there may not be loops in the nestle curve forbids a quadratic term for the x-parameterization. However, a polynomial of forth order for the y-coordinate can behave *very* strangely[3]. On the other hand, a stretching of the x-parameterization would not fit for cases, where the initial curvature is negative. To achieve a parameterization a more complex idea is developed in this dissertation, as described in detail in appendix D. The result is a parameterization of the following form for the vehicle local coordinate system:

$$\vec{c}(s) \equiv \frac{1}{2}\begin{pmatrix} R\,s\,\cos(\xi_0(s)) - R\,(1-s)\cos(\xi_1(s)) + d_x \\ R\,s\,\sin(\xi_0(s)) - R\,(1-s)\sin(\xi_1(s)) + d_y \end{pmatrix}, \tag{5.21}$$

with

$$\xi_0(s) \equiv a_x\,(s-1)^3 + c_x, \tag{5.22}$$

$$\xi_1(s) \equiv a_y\,s^3 + b_y\,s^2 + c_y. \tag{5.23}$$

The solution for a given target $(t_x\,t_y)^T$, target angle α_t and initial curvature κ_0 becomes

$$\begin{array}{l} a_x = 2\,A, \quad c_x = A, \quad d_x = t_x, \\ a_y = 2\,\alpha_t - 8\,A + \kappa_0\,|\cos(A)|\,R, \quad b_y = -\kappa_0\,|\cos(A)|\,R + 6\,A, \\ c_y = A, \quad d_y = t_y, \\ A = arctan(t_y, t_x), \quad R = \sqrt{t_x^2 + t_y^2}. \end{array} \tag{5.24}$$

The expression for curvature $\kappa(s)$ becomes

$$\kappa_c(s) = \frac{2}{R}\frac{\kappa_{nom}(s)}{(\kappa_{denom}(s))^{\frac{3}{2}}}. \tag{5.25}$$

with

$$\kappa_{nom}(s) \equiv \Big[(1-s)\sigma_1(s)\xi_1^{'2}(s) - s\,\sigma_0(s)\xi_0^{'2}(s)$$

[3]For cases, where there is no root of curvature the forth order polynomial may have a very rough shape. As a consequence, it is possible that the angle constraint at the endpoint is fulfilled, but the angle until an extremely small distance before does not change its direction as wanted. Such a nestle curve does not really nestle. Furthermore, it contains extremely high curvatures that are absolutely not necessary to nestle back to the course.

$$
\begin{aligned}
&+\big(2\,\xi_0'(s) + s\,\xi_0''(s)\big)\gamma_0(s) + \big(2\,\xi_1'(s) - (1-s)\xi_1''(s)\big)\gamma_1(s)\Big] \\
&\cdot\Big[\gamma_0(s) + \gamma_1(s) - s\,\sigma_0(s)\xi_0'(s) + (1-s)\,\sigma_1(s)\xi_1'(s)\Big] \\
&+\Big[\,(1-s)\,\gamma_1(s)\xi_1'^{2}(s) - s\,\gamma_0(s)\xi_0'^{2}(s) \\
&+\big(2\,\xi_0'(s) + s\,\xi_0''(s)\big)\sigma_0(s) + \big(2\,\xi_1'(s) - (1-s)\xi_1''(s)\big)\sigma_1(s)\Big] \\
&\cdot\Big[\sigma_0(s) + \sigma_1(s) + s\,\gamma_0(s)\xi_0'(s) - (1-s)\,\gamma_1(s)\xi_1'(s)\Big], \qquad (5.26)
\end{aligned}
$$

$$
\begin{aligned}
\kappa_{denom}(s) \;\equiv\; &\Big[\gamma_0(s) + \gamma_1(s) - s\;\sigma_0(s)\xi_0'(s) + (1-s)\,\sigma_1(s)\xi_1'(s)\Big]^2 \\
&+\Big[\sigma_0(s) + \sigma_1(s) + s\;\gamma_0(s)\xi_0'(s) - (1-s)\,\gamma_1(s)\xi_1'(s)\Big]^2, \qquad (5.27)
\end{aligned}
$$

and

$$
\sigma_0(s) \;=\; sin(\xi_0(s)), \quad \gamma_1(s) \;=\; cos(\xi_1(s)), \qquad (5.28)
$$
$$
\gamma_0(s) \;=\; cos(\xi_0(s)), \quad \sigma_1(s) \;=\; sin(\xi_1(s)). \qquad (5.29)
$$

A derivation of this formula is shown in appendix D. As for first order nestle curves, a near point ahead on the nominal course has to be chosen. Then a nestle curve is calculated from the vehicle's c.g. to that point and checked for a curvature greater than the desired maximum. If the curvature exceeds the maximum at any point, the nestle curve is rejected and the subsequent point on the nominal course has to be taken. This process is repeated until a nestle curve is found that nestles back to the nominal course satisfying all constraints.

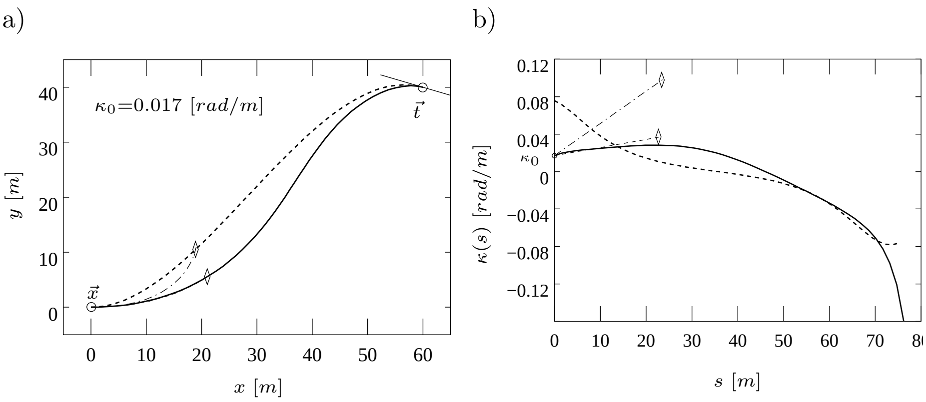


Figure 5.16: Nestle curves of first (bold dashed line) vs. nestle curves of second order (bold line). a) x- and y-coordinates of nestle curves and constant curvature segments. b) Curvature profiles.

A comparison between first and second order nestle curves is illustrated in figure 5.16. The dash-dotted line indicates the segment of constant curvature change to pass through the target point on the first order nestle curves. The dashed line represents the segment to pass through the target point on the second order nestle curve. Diamonds indicate target points. Having defined a nestle curve, a target point has to be chosen on the nestle curves. Based on this target point a segment of constant curvature, i.e. the geometric plan, is determined in order to pass through it. In figure 5.16a, it becomes obvious that the geometric plan resulting from the first order nestle curve matches the nestle curve much less than the geometric plan based on second order nestle curves. The geometric deviation between nestle curve and geometric plan becomes even more obvious if one considers the curvature profiles in figure 5.16b. The second order nestle curve and the resulting geometric plan have almost the same curvature profile. The curvature profile of the first order nestle curve and its resulting geometric plan, however, do not match at all.

Even though the considerations of this kind of nestle curves are based on a very detailed analysis, there is a fundamental problem. It is not as significant as any problems pinpointed before, however, it decreases performance. Consider figure 5.17 where a simple solution is required to nestle to a point 150 meters ahead with the same angle as the actual. The second order nestle curve calculates a trajectory as an arc starting with an initial curvature of $\kappa_0 = 0.0005\, rad\, m^{-1}$ and ending with and angle of zero. Correspondingly, the vehicle is guided away from the track. In figure 5.17a it is predictable that it may be displaced soon more than $0.2\, m$ from the nominal course.

a) b)

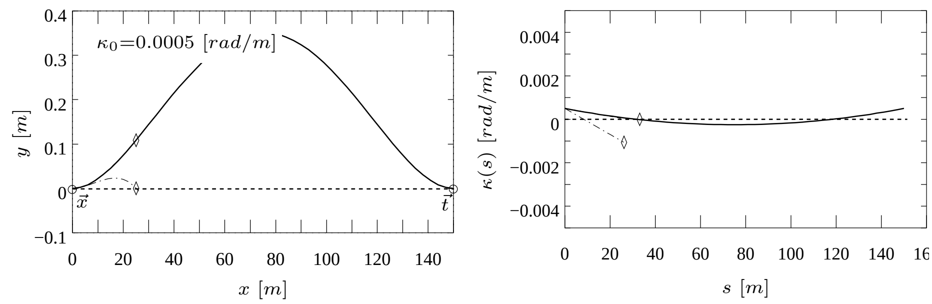


Figure 5.17: Nestling to a point that lies 150m straight ahead. The dash-dotted line indicates the segment of constant curvature change to pass through the target point on the first order nestle curves. The dashed line represents the segment to pass through the target point on the second order nestle curve. Diamonds indicate target points. a) Geometric behavior of the curves. b) Curvature profiles.

First order nestle curves simply result in a straight line from the start to the end. The resulting curvature profile of the segment of constant rate of curvature change contains higher curvatures and does not really fit the nestle curve.

However, this phenomenon becomes unimportant when curvatures are as low as that. The fact that the vehicle is guided $0.2\,m$ away from the nominal course by second order nestle curves is much more significant than fitting curvature profiles of curves that can be considered to be straight lines.

Thus, the process of nestle curve determination has to switch between first and second order nestle curves. For the following experiments, the first order nestle curve is only taken, if the actual curvature has the same sign as the curvature of the nestle curve and is greater than that. Otherwise, the second order nestle curve is used.

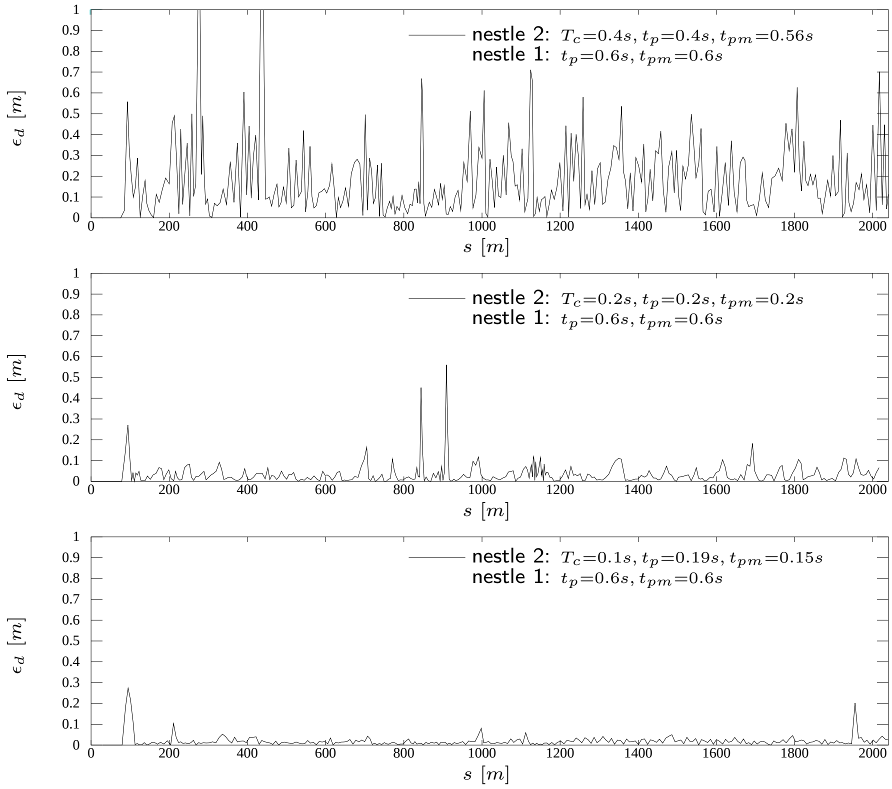


Figure 5.18: Absolute value of lateral deviation from nominal course using the second order nestle curves.

A sample run with second order nestle curves is shown in figure 5.18. Like the first order nestle curves, the second order nestle curves provide a mean to scale the deviation error. Smaller control intervals really result in higher precision. It is interesting to see that the preview times t_p on the nestle curves can now be chosen close or equal to the control interval length T_c. This is because the approximation done by a segment of constant rate of curvature change is much closer to the real curve, than it was with the first order nestle curves. With control intervals of 100 ms it is possible to reduce the error practically to zero.

5.2.4 Conclusion on Target Points

The preceding sections discussed the problems of target point search. First, a set of intuitive methods was developed. Three different intuitive methods were derived to calculate the position of the target point. Preview Point used the system state of the vehicle, i.e. the velocity, as a basis for target point placement. End Of Sight placed the target point based on the geometry of the nominal course. Deviation Dependent Preview Point uses lateral displacement and angular deviation to adapt the distance to the target point. The three methods were not convincing in terms of their performance.

Fundamental errors were attributed to a mathematical problem that arises from initial curvature and target points that lie too far ahead as well as the disregard of the situational driving motivation SDM. A new method, called Nestle Curves, was introduced to immediately boost the performance of the driver model. Nestle Curves, however, introduce a new piece of data in the target point search procedure: a nominal course that is local to the current situation. Where the previous three methods *directly* referred to the long term nominal course for target point placement, the nestle curve method places the target point on a temporary trajectory.

A nestle curve acts very similarly to a nominal course. However, a nestle curve and a nominal course are fundamentally different concepts. Nestle curves are constructed in order to nestle back to the nominal course. For every control interval they are calculated in order to reduce the lateral and angular deviations from the nominal course. The nominal course itself is designed by a strategic unit correspondent to road geometries. Due to these considerations, target point placement has to be understood as a three level process as depicted in figure 5.19:

1. Nominal course construction: Every time that errors exceed a certain boundary, a new nominal course $\vec{n}(p)$ has to be created related to the road geometries close ahead. The error check happens every control interval, while the nominal course is only constructed when the threshold check triggers.

2. Short term path planning: As long as errors are small enough, the vehicle only has to be nestled back to the nominal course. The construction of a short term plan $\vec{c}(p)$ happens every control time interval.

3. Target point determination: A suitable target point has to be chosen, that can be reached in the time range of one control interval. This also has to be accomplished every control interval. The target point $\vec{t}$ is then directly related to a desired rate of curvature change κ' for the next control interval.

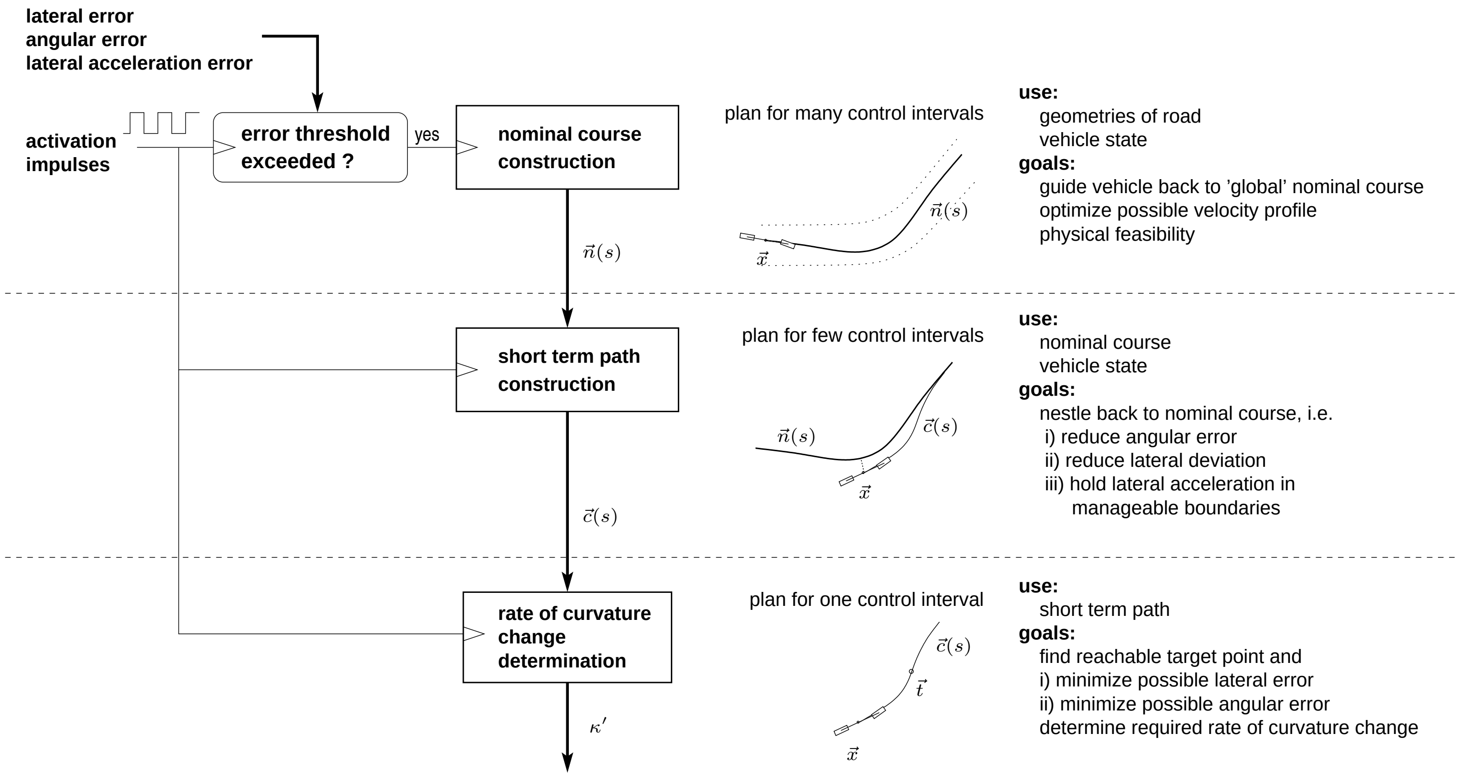


Figure 5.19: Three levels of geometric plan construction.

The development of a strategic unit that constructs a nominal course according to road geometries is the next step in order to improve geometric dynamic planning. It has to be located, however, on a layer above the Generalized Feedback Controller.

5.3 Rate of Curvature Change for Target Point

As can be seen in figure 5.2a, the calculation of parameters of the target map is based on a target point. The search for an appropriate target point was discussed in the previous section. This section discusses the computation of a constant rate of curvature change κ' that is required in order to pass through the target point. Starting from an initial vehicle position $\vec{X}_0$, with an angle α and a curvature κ, a curve with a constant rate of curvature change κ' has to be determined in order to pass through the target point $\vec{X}_t$. A constant rate of curvature change κ' results in a profile of the tangent angle of the trajectory as given by

$$\alpha(s) = \alpha + \kappa\, s + \tfrac{1}{2}\kappa'\, s^2, \tag{5.30}$$

where s is the trajectory length. Such a curve is called a *spiral*. Examples for opening and closing spirals can be observed in figure 5.20.

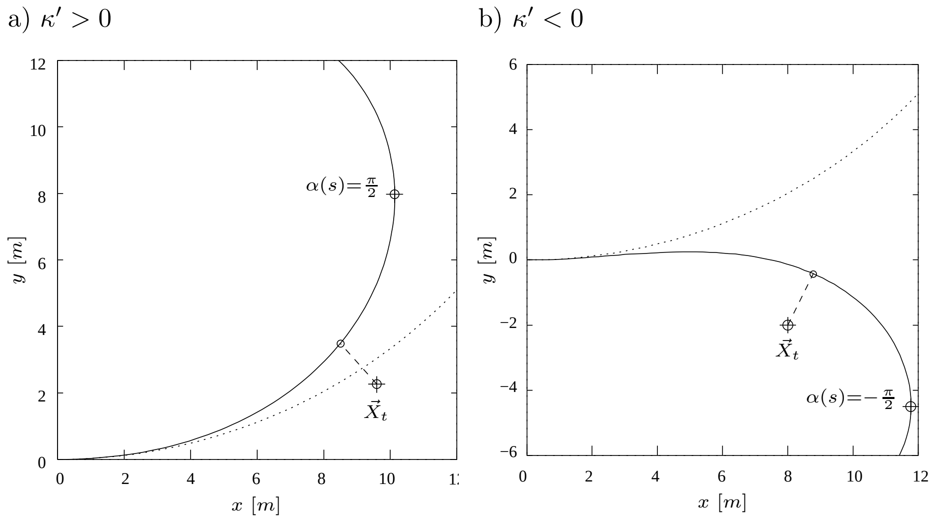


Figure 5.20: Pictures of closing ($\kappa' > 0$) and opening ($\kappa' < 0$) spirals as solid lines. The extrapolated curve with constant curvature ($\kappa > 0$) is depicted as dotted line. a) Closing spiral. Rate of curvature change has the same sign as curvature at $s = 0$. b) Opening spiral. Rate of curvature change has a different sign than curvature at $s = 0$.

To find the appropriate κ' to pass through the target point $\vec{X}_t$, a formula is derived that describes the point on a spiral defined by κ and κ' dependent on the trajectory length s. The resulting expression makes it clear that an analytical solution to find the κ' to pass through $\vec{X}_t$ is impossible. Therefore, an approximation method has to be developed to find a numerical solution for κ'. This is achieved by means of a two level minimization algorithm.

For the sake of simplicity, the problem is transformed into a local coordinate system. It is assumed that the initial position lies in the origin and that the initial angle is parallel to the x-axis, i.e.

$$\vec{X}_0 \;=\; (0\,0)^T \qquad \text{and} \qquad \alpha(s=0) \;=\; \alpha \;=\; 0. \tag{5.31}$$

In the general case, this initial condition is achieved by a simple linear transformation consisting of a rotation and a translation. Curvature is defined as the rate of angle change with respect to distance, i.e. $\kappa(s) = \frac{d}{ds}\alpha(s)$. The angle of a trajectory $\vec{X}_{\kappa'}(s)$ is defined as $\arctan(\frac{d}{dy}\vec{X}_{\kappa'}(s)/\frac{d}{dx}\vec{X}_{\kappa'}(s))$. Using equation (5.30) the tangent vector can be specified as

$$\vec{T}_{\kappa'}(s) = \begin{pmatrix} T_x(s) \\ T_y(s) \end{pmatrix} = \begin{pmatrix} cos(\kappa s + \frac{1}{2}\kappa'\, s^2) \\ sin(\kappa s + \frac{1}{2}\kappa'\, s^2) \end{pmatrix}. \tag{5.32}$$

Integrating $\vec{T}_{\kappa'}(s)$ yields the position

$$\vec{X}_{\kappa'}(s) = \int \vec{T}_{\kappa'}(\sigma) d\sigma = \begin{pmatrix} \int_0^s T_x(\sigma) d\sigma \\ \int_0^s T_y(\sigma) d\sigma \end{pmatrix}. \tag{5.33}$$

The integrals over $T_x(s)$ and $T_y(s)$ contain expression of the form s^2 inside the trigonometric functions *cos* and *sin*. Expanding the sine and cosine expressions by Euler's rule reveals the integral over e^{-x^2} which is analytically not solvable. Instead, the arguments are transformed so that Fresnel Cosine and Sine can be used to express the solution. Fresnel Cosine $C(s)$ and Fresnel Sine $S(s)$ are defined as

$$C(s) \;\equiv\; \int_0^s cos\left(\tfrac{\pi}{2}\, x^2\right) dx \qquad \text{and} \qquad S(s) \;\equiv\; \int_0^s sin\left(\tfrac{\pi}{2}\, x^2\right) dx, \tag{5.34}$$

To be able to use these expressions, the argument of the sine and cosine function must now be reformulated:

$$\kappa s + \tfrac{1}{2}\kappa'\, s^2 \quad \longrightarrow \quad \left(s + \frac{\kappa}{\kappa'}\right)^2 - \frac{\kappa^2}{2\kappa'}. \tag{5.35}$$

The term $-\frac{\kappa^2}{2\kappa'}$ is independent of s and can thus be understood as the influence of a *rotation* of curve $\vec{X}_{\kappa'}(s)$ by a constant angle $\lambda = -\frac{\kappa^2}{2\kappa'}$. Let $A(\lambda)$ be the necessary rotation matrix to get a simplified vector $\vec{T}^*_{\kappa'}(s) = A(\lambda)\vec{T}_{\kappa'}(s)$ by applying equation (5.35) in equation (5.32). $\vec{T}^*_{\kappa'}(s)$, therefore, becomes

$$\vec{T}^*_{\kappa'}(s) \;\equiv\; A(\lambda)\vec{T}_{\kappa'}(s) \;=\; \begin{pmatrix} cos\left(\left(s + \frac{\kappa}{\kappa'}\right)^2\right) \\ sin\left(\left(s + \frac{\kappa}{\kappa'}\right)^2\right) \end{pmatrix}. \tag{5.36}$$

Since $A(\lambda)$ is independent of s, it behaves in integration like a constant factor. Thus, first $\vec{X}^*_{\kappa'}(s)$ can be calculated as the integral of $\vec{T}^*_{\kappa'}(s)$. Second, $\vec{X}^*_{\kappa'}(s)$ has to be rotated by $A^{-1}(\lambda)$ to get $\vec{X}_{\kappa'}(s)$. Applying the substitutions $u_1 \equiv s + \frac{\kappa}{\kappa'}$ and $u_2 \equiv \sqrt{\kappa'/\pi}\, u_1$ allows the integral to be expressed as:

$$\vec{X}^*_{\kappa'}(s) = \int \vec{T}^*_{\kappa'}(\sigma) d\sigma = \sqrt{\frac{\pi}{\kappa'}} \left(\begin{array}{c} C\left(\frac{\kappa'\sigma+\kappa}{\sqrt{\pi\kappa'}}\right) \\ S\left(\frac{\kappa'\sigma+\kappa}{\sqrt{\pi\kappa'}}\right) \end{array} \right) \Bigg|_{\sigma=0}^{\sigma=s} . \tag{5.37}$$

Since $A(\lambda)$ carries out a pure rotation one can directly state that $A^{-1}(\lambda) = A(-\lambda)$ which is a rotation in the opposite direction. $\vec{X}_{\kappa'}(s)$ is then given by

$$\begin{aligned} \vec{X}_{\kappa'}(s) &= A^{-1}(\lambda)\vec{X}^*_{\kappa'}(s), & (5.38) \\ &= A(-\lambda)\sqrt{\frac{\pi}{\kappa'}} \left(\left(\begin{array}{c} C\left(\frac{\kappa' s+\kappa}{\sqrt{\pi\kappa'}}\right) \\ S\left(\frac{\kappa' s+\kappa}{\sqrt{\pi\kappa'}}\right) \end{array} \right) - \left(\begin{array}{c} C\left(\frac{\kappa}{\sqrt{\pi\kappa'}}\right) \\ S\left(\frac{\kappa}{\sqrt{\pi\kappa'}}\right) \end{array} \right) \right), & (5.39) \end{aligned}$$

where

$$A(-\lambda) \equiv \left[\begin{array}{cc} cos(\lambda) & -sin(\lambda) \\ sin(\lambda) & cos(\lambda) \end{array} \right] . \tag{5.40}$$

Equation (5.39) defines the point $\vec{X}_{\kappa'}(s)$ that is reached by applying a certain rate of curvature change κ' over a distance s. The task is now to solve the inverse problem, i.e. one has to find the appropriate κ' to pass through a given target point $\vec{X}_t$. The fact that these formulae include sines, cosines as well as their Fresnelian equivalents makes it impossible to find analytic solutions. To solve this problem, a two level minimization procedure is derived:

1. First, a *distance measure* is defined that describes the distance between a target point and a spiral of fixed κ' as the minimum distance between both, i.e.

$$dist(\vec{X}_t, \{\vec{X}_{\kappa'}(s)\}) \equiv \min_{s\in D_s} \left\{ \left| \vec{X}_t - \vec{X}_{\kappa'}(s) \right| \right\} . \tag{5.41}$$

   Minimization algorithms are not discussed in the following text[4]. In general however, rapid minimum search algorithms (like the Brent Algorithm that was used in this case [Brent, 1973]) require that there is only one minimum inside the search interval. Otherwise, the algorithm might find a local minimum. Thus, section E.1 is dedicated to the elaboration of an appropriate search interval D_s.

   With the appropriate D_s a minimization algorithm can be run to find the minimum distance between the spiral and the target point. This minimum distance represents at the same time the distance measure between the spiral and the target point.

[4] Literature provides a large overview over this topic [Press et al., 1992, chapter 10].

2. Having a procedure to compute the distance between the target point and a spiral it is now possible to find the closest spiral to the target point by adapting κ'. The best spiral is the one where the distance between target point and spiral is zero. Again, the variation of κ' and searching for the closest distance is best done with a minimization algorithm to get the κ'_{min} so that

$$dist(\vec{X}_t, \{\vec{X}_{\kappa'_{min}}(s)\}) \;=\; \min_{\kappa' \in D_{\kappa'}} \Big\{ dist(\vec{X}_t, \{\vec{X}_{\kappa'}(s)\}) \Big\}. \tag{5.42}$$

   To use an appropriate algorithm, a domain $D_{\kappa'}$ is required that does not contain local minima. The derivation of this domain is illustrated in section E.2.

Briefly, two procedures are required. The first needs to determine for a fixed κ' how far the resulting spiral is to the target point. This requires a suitable domain D_s with a unique minimum over s. Then it is possible to apply a second procedure that searches for the κ' so that the resulting spiral hits the target point.

5.4 Acceleration for Curvature Transition

Referring to figure 5.2a, the next step towards the preferred gd-plan of the target map is to calculate an appropriate acceleration $\dot{v}$ for the curvature transition that is specified by κ'. For this, it is assumed that there is a maximum lateral acceleration $\ddot{y}_{max,curv}$ that must not be exceeded during the next control interval.

For the following discussion the velocity profile $v(t)$ over the next control interval is approximated by a linear term, i.e $v(t) \approx v + \dot{v}\,t$. A constant rate of curvature change κ' with respect to s results in an expression $\dot{\kappa}(t)$ as

$$\dot{\kappa}(t) \;=\; \frac{\partial \kappa}{\partial s} \cdot \frac{\partial s}{\partial t} \;=\; \kappa'\,(v + \dot{v}\,t). \tag{5.43}$$

Since κ' is constant and assuming that the velocity change during one control interval is relatively small, it is plausible that $\dot{\kappa}(t)$ can be effectively approximated by a constant. Concretely, this means that it is assumed that the trajectory that results from applying a constant $\dot{\kappa}(t) = \dot{\kappa}$ is sufficiently close to the shape of a spiral, given by a constant $\kappa'(t) = \kappa'$. Using equation (5.43) allows one to calculate the average $\langle \dot{\kappa}(t) \rangle$ that appears, if $\kappa'(t)$ is constant. Assuming the constant $\dot{\kappa}$ to be exactly this average $\langle \dot{\kappa}(t) \rangle$ minimizes the error between $\dot{\kappa}$ and $\dot{\kappa}(t)$ over the control interval $(0, T_c]$, i.e.

$$\dot{\kappa} \;=\; const. \;=\; \langle \dot{\kappa}(t) \rangle \;=\; \frac{1}{T_c} \int_0^{T_c} \kappa'\,(v + \dot{v}\,t)\,dt \;=\; \kappa' \left(v + \frac{\dot{v}\,T_c}{2} \right). \tag{5.44}$$

With the linear approximations of curvature and velocity the lateral acceleration profile $\ddot{y}(t) \;=\; v^2(t)\,\kappa(t)$ can be expressed as

$$\ddot{y}(t) \;=\; (v + \dot{v}\,t)^2 (\kappa + \dot{\kappa}\,t), \tag{5.45}$$

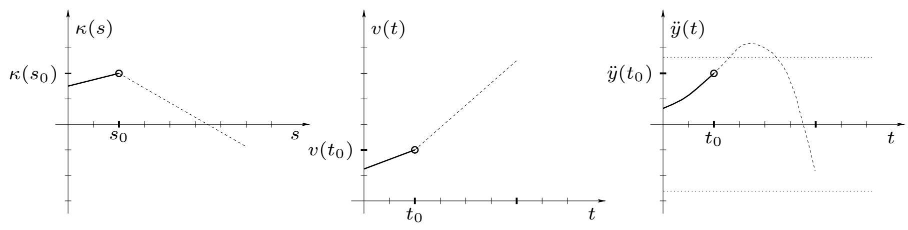


Figure 5.21: Principle diagrams to illustrate the derivation of a lateral acceleration profile $\ddot{y}(t)$ from a constant rate of curvature change κ' and a constant acceleration $\dot{v}$. The exact formula for $\ddot{y}(t)$ is given by equation (5.45).

$$= \quad (v + \dot{v}\,t)^2 \left(\kappa + \kappa' \left(v + \frac{\dot{v}\,T_c}{2}\right) t\right). \tag{5.46}$$

The lateral acceleration must not exceed the maximum lateral acceleration $\ddot{y}_{max,curv}$. Therefore, the extremum of the function $\ddot{y}(t)$ has to be determined. For a fixed $\dot{v}$ the lateral acceleration $\ddot{y}(t)$ has two extrema with respect to time.

$$t_a \quad = \quad -\frac{v}{\dot{v}}, \tag{5.47}$$

$$t_b \quad = \quad \begin{cases} -\frac{2\,\kappa'\,v^2 + v\,\kappa'\,\dot{v}\,T_c + 4\,\dot{v}\,\kappa}{3\,\dot{v}\,\kappa'\,(2\,v + \dot{v}\,T_c)} & \forall \quad \kappa \neq 0 \wedge \kappa' \neq 0 \\ -\frac{v}{3\,\dot{v}} & \forall \quad \kappa = 0 \wedge \kappa' \neq 0 \\ t_a & \forall \quad \kappa \neq 0 \wedge \kappa' = 0 \end{cases} . \tag{5.48}$$

Note that in the case that κ and κ' are both zero is a special case, where $\dot{v}$ can have an arbitrary value. In this case the vehicle is moving on a straight line and lateral acceleration profiles do not have to be considered. Equation (5.45) shows that lateral acceleration at the first extremum t_a is zero, so the absolute value can never be a maximum. The second extremum t_b, however, has to be considered as a possible candidate.

Let $\kappa > 0$ for all following considerations. In case of $\kappa < 0$ this assumption implies an inversion of κ to $\kappa^* = -\kappa$. Consequently inverting κ' thus gives a problem that is geometrically identical to the original.

Furthermore, it is supposed that the velocity is always positive[5]. Preventing the velocity from becoming zero in the time interval $(0, T_c)$ restricts the allowed domain of accelerations I_0 to

$$I_0 \equiv (-\tfrac{v}{T_c}, \infty). \tag{5.49}$$

[5]Restricting the velocity to be greater than zero and not to cross zero during the control interval does not at all restrict the validity of the discussion. Negative velocities can be handled the same way as mentioned, treating the desired trajectory to be driven backwards. As mentioned previously, velocity changes during the control interval are considered to be relatively small. Thus the zero-crossings of the velocity happen at very low velocities, i.e. $v(t) \approx 0$ during the control interval. In consequence of this, lateral acceleration $\ddot{y}(t) = v^2(t)\kappa(t)$ will be close to zero. In this case, the discussion about an exceeding of an lateral acceleration limit would be superfluous, since limiting a lateral acceleration that is approximately zero does not make sense.

Starting from equation (5.45) several constraints can be defined on the lateral accelerations profile. The first two constraints follow the restriction that the maximum lateral acceleration limit $\ddot{y}_{max,curv}$ cannot to be exceeded. The third constraint follows from restrictions on the profile of the lateral acceleration over time. Each constraint results in a set of admissible accelerations. Finally, the intersection of all sets defines a domain of accelerations in which $\dot{v}$ can be chosen. The constraints are as follows:

1. Lateral acceleration at the end of the control interval has to be less than or equal to the maximum value. That means

$$|\,\ddot{y}(T_c)\,| \;=\; \left|(v+\,\dot{v}\,T_c)^2\left(\kappa+\kappa'\,\left(v+\frac{1}{2}\,\dot{v}\,T_c\right)\,T_c\right)\right| \;\leq\; \ddot{y}_{max}. \tag{5.50}$$

Correspondingly, one has to search for a subset $I_a \subset I_0$ with the property

$$I_a \;\equiv\; \{\dot{v}\;\in\; I_0 : \,|\,\ddot{y}(T_c)\,|\;\leq\; \ddot{y}_{max,curv}\}\,. \tag{5.51}$$

2. If the extremum of lateral acceleration t_a lies inside the control interval it has to be less than or equal to the maximum lateral acceleration, i.e.

$$\begin{aligned} |\,\ddot{y}(t_b)\,| \quad &= \quad \left|-\frac{2}{27}\,\frac{\left(2\,\kappa'\,v^2+\dot{v}\,v\kappa'\,T_c-2\,\dot{v}\,\kappa\right)^3}{\kappa'^2\,\left(2\,v+\dot{v}\,T_c\right)^2\,\dot{v}}\right| \;\leq\; \ddot{y}_{max} \\ \vee &\quad t_b\;\notin\;(0,\,T_c]. \end{aligned} \tag{5.52}$$

To specify the subset I_b where this equation holds one defines

$$\begin{aligned} I_{b,1} \quad &\equiv \quad \{\dot{v}\;\in\; I_0 : \,|\,\ddot{y}(t_b)\,|\;\leq\; \ddot{y}_{max,curv}\}\,, && (5.53)\\ I_{b,2} \quad &\equiv \quad \{\dot{v}\;\in\; I_0 : \,t_b\notin(0,\,T_c]\,\}\,. && (5.54) \end{aligned}$$

There are two special cases that have to be treated separately. These are the cases where either $\kappa = 0$ or $\kappa' = 0$. Naming the resulting set as $I_{b,spec}$ one can define I_b as

$$I_b\;\equiv\;\begin{cases} I_{b,spec} & \forall\quad \kappa\neq 0\,\wedge\,\kappa'\,=\,0 \\ & \quad\;\;\oplus\,\kappa=0\,\wedge\,\kappa'\,\neq\,0 \\ I_{b,1}\cup I_{b,2} & else \end{cases}\;. \tag{5.55}$$

3. Since the lateral acceleration at the minimum t_a is zero by (5.45) and (5.47) it is preferable to avoid allowing it to lie in the interval $(0,\,T_c)$. This would mean that there are oscillations in lateral acceleration of shorter periods than the control interval T_c. Such oscillations in lateral accelerations would make the front task enormously difficult. So must be required that

$$t_a \;=\; -\tfrac{v}{\dot{v}}\notin(0,\,T_c]. \tag{5.56}$$

By $v > 0$ it follows that

$$t_a\;\leq\;0\;\;\Leftrightarrow\;\;\dot{v}\;>\;0,\qquad\text{and}\qquad t_a\;>\;T_c\;\;\Leftrightarrow\;\;\dot{v}\;>-\frac{v}{T_c}. \tag{5.57}$$

> So the restriction of not to perform a full brake, i.e. $\dot{v} > -v/T_c$ includes totally the requirement of not having a minimum of lateral acceleration between $t = 0$ and $t = T_c$. Recall, that for $\dot{v} = 0$ the minimum does not exist.

The total allowed domain I, where $\dot{v}$ satisfies all conditionscomputes to

$$I \quad \equiv \quad I_a \cap I_b. \tag{5.58}$$

The acceleration of the taget map's preferred gd-plan is then chosen as the maximum $\dot{v}$ inside the domain I. The exact mathematical formulation of the domain I_a and I_b is illustrated in appendix F.

5.5 Curvature Profile of Nominal Course

Now, that the algorithms for calculating parameters for the target map have been discussed, the following sections focus on the calculation of parameters for the limit map. As can be seen in figure 5.2b, it is necessary to respect a velocity profile corresponding the nominal course. The velocity profile $v_{nc}(s)$ of the nominal course can be derived from the curvature of the nominal course $\kappa(s)$ and a limiting lateral acceleration $\ddot{y}_{max,course}$. Let the maximum velocity $v_{nc}(s)$ with respect to the distance s be defined as

$$v_{nc}(s) \quad \equiv \quad \sqrt{\frac{\ddot{y}_{max,course}}{\kappa_{nc}(s)}}, \tag{5.59}$$

where $\kappa_{nc}(s)$ describes the curvature of the nominal course. The algorithm that is presented here assumes a maximum amount of negative acceleration a_0, i.e. a deceleration, which the driving agent never desires to exceed. The velocity profile of a vehicle that is decelerated with this negative acceleration a_0 is given by

$$v(s) \quad = \quad \sqrt{v_0^2 + 2a_0\, s}. \tag{5.60}$$

A velocity can only be considered to be admissible, if it is always possible to decelerate enough, so that $v_{nc}(s)$ is never exceeded. Using (5.60) this means that

$$\sqrt{v_0^2 + 2\, a_0\, s} \leq v_{nc}(s) \quad \forall \quad s \in (0,\, s_{max}]. \tag{5.61}$$

s_{max} indicates the maximum distance that the algorithm looks ahead on the nominal course. A large value of s_{max} can represent a very experienced driver since he would know many of the curves ahead. A small value for s_{max} represents a less experienced driver since he might have difficulty anticipating the curves lying ahead of him.

For the geometric-dynamic plan, the driver has to find an admissible acceleration. An admissible acceleration has to ensure that the velocity during the next control interval lies under the velocity profile derived from the nominal course. Additionally, it must be confirmed that the velocity after the next control interval can be adapted in a way so that the maximum deceleration is enough to avoid a violation of the velocity profile in future.

First, a braking graph is introduced that describes the velocity profile that reaches a point s_x ahead with the velocity $v_{nc}(s_x)$ by applying the acceleration a_0. A discussion follows which explains how to determine if a braking graph lies above or beneath another one. Finally, having found the most critical braking curve, the maximum acceleration that does not exceed this braking graph is determined. For a given point s_x ahead on the nominal course with a velocity $v_{nc}(s_x)$, the braking graph resulting from a negative acceleration a_0 trough this point is given by

$$v_{s_x}(s) \equiv \sqrt{v_{nc}^2(s_x) + 2a_0\,(s - s_x)}, \qquad \text{with} \qquad s \,\in\, (0, s_x]. \tag{5.62}$$

Figure 5.22 shows such braking graphs that end up at given limiting velocities on the nominal course. The dotted lines indicate braking graphs with acceleration $a_0 = -1.9\,m\,s^{-2}$. Assuming a constant maximum deceleration a 'braking graph' represents the needed velocity profile to break down to the limit velocity at a certain distance s_x ahead. If the vehicle's velocity is ever higher than one of the braking graphs of a point s_x ahead on the course, it is no longer possible to respect the maximum velocity $v_{nc}(s_x)$ at this point. To avoid this, it must be determined that *none* of the braking graphs is ever exceeded. These considerations yield a specification of the dilemma domain $\mathcal{D}^*_\infty$ as defined in the containability discussion in section 7.2. Fortunately, it can be proven that if a braking graph lies underneath another one at one specific point s^*, it lies underneath for all s:

> If at some distance s^* ahead two braking graphs $v_{s_1}(s)$ and $v_{s_2}(s)$ are related by
>
> $$v_{s_1}(s^*) \;<\; v_{s_2}(s^*), \tag{5.63}$$
>
> then it follows by equation (5.62) that
>
> $$\begin{aligned} v_{nc}^2(s_1) + 2a_0\,(s^* - s_1) \quad &< \quad v_{nc}^2(s_2) + 2a_0\,(s^* - s_2), & (5.64)\\ v_{nc}^2(s_1) - 2a_0\,s_1 \quad &< \quad v_{nc}^2(s_2) - 2a_0\,s_2, & (5.65)\end{aligned}$$
>
> which is always true *independent* of the primarily considered distance s^*. So, if one braking graph lies under another braking graph at a certain point s^*, then it lies under it for all possible s, i.e.
>
> $$v_{s_1}(s^*) \;<\; v_{s_2}(s^*) \quad \Rightarrow \quad v_{s_1}(s) \;<\; v_{s_2}(s) \quad \forall \quad s \in \Re. \tag{5.66}$$

Without any loss of generality, one can therefore restrict the comparison of the braking curves to $s = 0$ and define the velocity of a braking graph at $s = 0$ as a reference, i.e.

$$v_0(s_x) \quad \equiv \quad \sqrt{v_{nc}^2(s_x) - 2\,a_0\,s_x}. \tag{5.67}$$

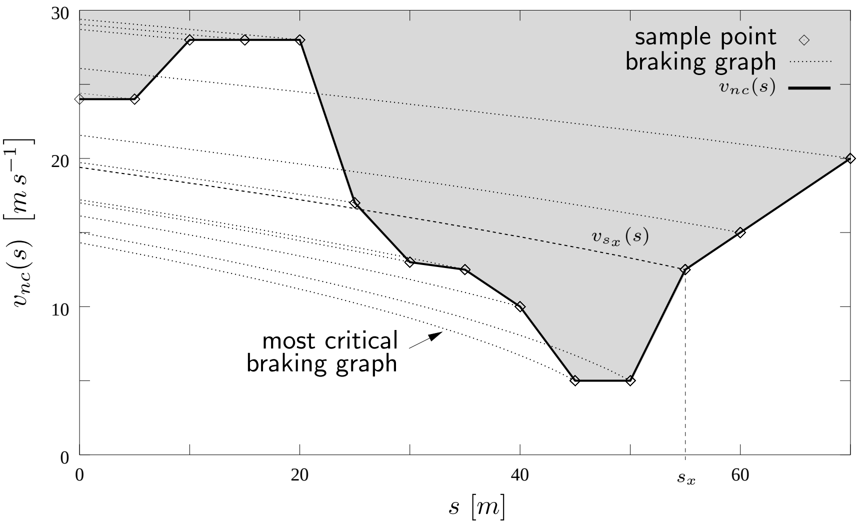


Figure 5.22: Limiting velocity profile for nominal course ahead of the car's position indicated by a bold line.

Thus, it is sufficient to respect the braking graph that has the lowest velocity at $s = 0$. This most critical braking graph directly relates to the critical distance s_{crit} where the velocity has to be lower than $v_{nc}(s_{crit})$

$$s_{crit} \quad \text{whereby} \quad v_{s_{crit}}(0) = \min_{s_x \in (0, s_{max}]} \{v_0(s_x)\}. \tag{5.68}$$

Any velocity profile that is admissible has to lie, therefore, under the most critical braking graph defined by equation (5.62) for $s_x = s_{crit}$. Since an acceleration must finally be determined, the results have to be considered with respect to time and not travel length. The braking curve for s_{crit} with respect to time can be defined as

$$v_{crit}(t) \quad = \quad v_0(s_{crit}) + a_0\, t. \tag{5.69}$$

This is the velocity profile with respect to time that has to be followed. During the next control interval, the following condition has to be fulfilled:

$$v(t) \quad \leq \quad v_{crit}(t) \qquad \forall \qquad t \in (0, T_c). \tag{5.70}$$

Thus, for a linear transition of $v(t)$, i.e. a constant acceleration $\dot{v}$, it has to hold

$$v(t) \;=\; v + \dot{v}\, t \quad \leq \quad v_0(s_{crit}) + a_0\, t \qquad \forall \qquad t \in (0, T_c). \tag{5.71}$$

It is now possible to define a maximum acceleration $\dot{v}_{max}$ that ensures that the vehicle can be slowed down to any limiting velocity ahead.

$$\dot{v}_{max} \quad \equiv \quad \frac{v_0(s_{crit}) - v}{T_c} + a_0. \tag{5.72}$$

If the acceleration $\dot{v}$ during the next control interval is not greater than this $\dot{v}_{max}$, it is safe to assume that the vehicle never enters domains of unwanted lateral acceleration caused by the curvature of the nominal course.

The previous paragraphs specified a dilemma domain $\mathcal{D}^*_\infty$ as introduced in the containability discussion (section 7.1). A admissible acceleration has to avoid to enter the dilemma domain, determined by the most critical braking graph which is based on a maximal deceleration. For 'real' vehicle guidance, however, this dilemma domain is not sufficient, because it does not include other important considerations about the frictional ellipse and influences of load transfer.

5.6 Optional Velocity Restrictions

As illustrated in figure 5.2b, limit map construction includes a feature that imposes limits on velocity. This feature is introduced in order to satisfy requirements raising from maneuvers like the lane-change maneuver ISO-3888 that demands a constant velocity. If the acceleration would cause a violation of the upper or lower velocity boundary, then its desired acceleration $\dot{v}$ is adapted to respect the boundaries. Figure 5.23 compares a velocity profile of the agent driving along a test course with and without velocity restrictions. In the first case, figure 5.23a the velocity planning is only restricted by the velocity profile of the nominal course and (not visible) restrictions coming from the curvature profile through the local target point. Figure 5.23b shows a velocity profile for the case that the driver model includes the velocity boundaries in its considerations.

5.7 Application of Target and Limit Maps

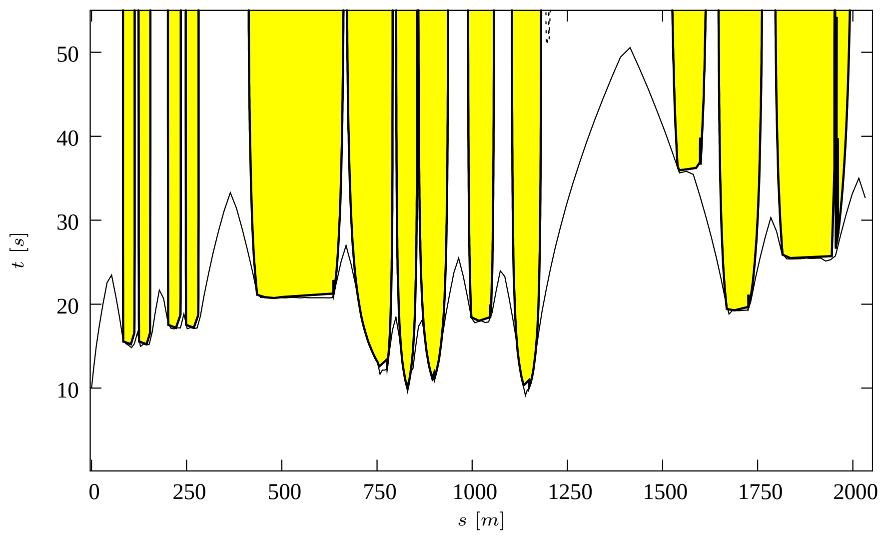
With the parameters, specified in the previous sections, the target map $T(p)$ and the limit map $L(p)$ can be constructed. Both maps have to be combined into one single utility function $U(p)$. The gd-plan consistent with all related motivations is then determined as the gd-plan that has the highest utility in $U(p)$. An overview of this procedure is shown in figure 5.24.

The parameter pair $(\dot{\kappa}_t, \dot{v}_t)$ that results from the target point concepts is now used as the preferred gd-plan of a target motivation utility function. As mentioned earlier, this gd-plan acts like a pillar that spans the tent of the constructiveness function $C(p_{gd})$. The constructiveness decreases with distance to the preferred gd-plan. By definition, the constructiveness function has to have only positive values (see section 4.2, definition 9). Since there is only one constructiveness function $C(p_{gd})$ it directly represents the target map $T(p_{gd})$. Due to the aforementioned restrictions, it is defined as

$$T(p_{gd}) \quad = \quad exp\left(-\sqrt{\kappa_r^{-2}\,(\kappa' - \kappa'_t)^2 + c_{\kappa',\dot{v}}\,(\dot{v} - \dot{v}_t)^2}\right), \qquad (5.73)$$

where $\kappa_r = 1/m$ is a reference value that is introduced to make the product $\kappa_r^{-2}\,(\kappa' - \kappa'_t)^2$ dimensionless. $c_{\kappa',\dot{v}}$ is a factor that indicates the importance

a)



b)

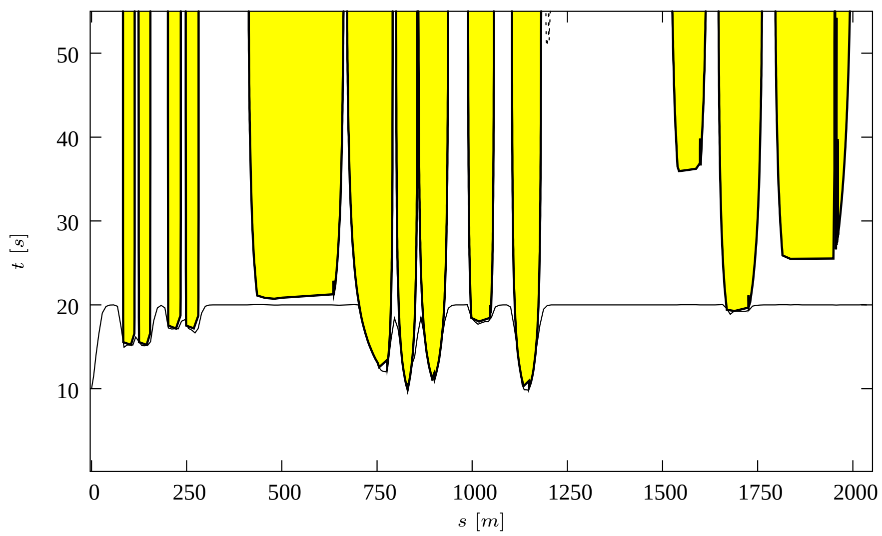


Figure 5.23: Influence of restrictions on velocity. The shaded areas indicates the velocity profile resulting from maximum lateral acceleration considerations. The solid line represents the vehicle's velocity profile. a) Vehicle's velocity if the driver agent has no restriction on velocity. b) Velocity profile with minimum and maximum borders chosen as $v_{min} = 10\frac{m}{s}$ and $v_{max} = 20\frac{m}{s}$.

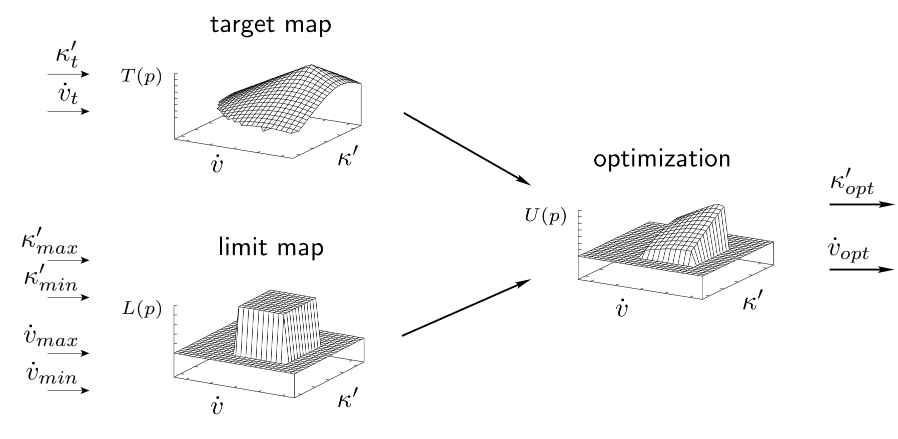


Figure 5.24: Construction of target and limit maps, their combination and optimum search.

of precision in curvature over precision in velocity. Such a factor is required, anyway, because the values of κ' in $[rad\, s^{-1}\, m^{-1}]$ are not in the same range as the values of the acceleration $\dot v$ in $[m\, s^{-2}]$. Since the shape of the trajectory is much more important than velocity constraints, $c_{\kappa',\dot v}$ is chosen as a value close to zero.

The boundaries for rate of curvature change κ'_{min} and κ'_{max} together with the boundaries for acceleration $\dot v_{min}$ and $\dot v_{max}$ enable a limit map definition. With these parameters the space of gd-plans can be separated into allowed and disallowed regions. Thus, the limit map is defined as

$$L(p) \equiv \begin{cases} 1 & \forall \quad \kappa' \geq \kappa'_{min} \wedge \kappa' \leq \kappa'_{max} \\ & \quad \wedge\, \dot v \geq \dot v_{min} \wedge \dot v \leq \dot v_{max}. \\ 0 & else \end{cases} \tag{5.74}$$

The combination of the target and limit maps $T(p)$ and $L(p)$ is accomplished by simply cutting the domains out of the target map where $L(p)$ is zero. Thus, the combined utility is chosen as

$$U(p) = \begin{cases} T(p) & \forall \quad L(p) = 1 \\ 0 & \forall \quad L(p) = 0 \end{cases} . \tag{5.75}$$

The procedure to find the single consistent gd-plan (see section 4.4) deals with two cases. If the preferred gd-plan of the target map $(\dot\kappa_t,\, \dot v_t)$, i.e. the pillar, lies inside the boundary of the admissible domain, than $(\dot\kappa_t,\, \dot v_t)$ is the optimal gd-plan with respect to the combined utility function. If it doesn't, one has to search for the maximum of $T(p)$ along the borderlines of the domain where $L(p) = 1$.

5.8 Conclusion

The construction of a suitable gd-plan for a control interval requires a consideration of various different 'motivations', such as maximum lateral acceleration, target points, curvature constraints and so on. The process to compute a distinct gd-plan is associated with the interference of a controller's motivation and information about the plant's state.

Using the method of target and limit maps motivations are represented through scalar functions. The function value for a particular gd-plan represents its utility or admissibility with respect to a specific motivation. A target map is a utility functions based on a preferred gd-plan. A limit map makes assumptions about the admissibility of sets of gd-plans. I this chapter it was discussed what motivations are involved when building a gd-plan construction unit for vehicle control and how they are represented.

The target map is based on a preferred gd-plan of a target motivation $(\dot{\kappa}_t, \dot{v}_t)$. It acts like a pillar in the utility function. The gd-plan is developed based on the target point concept. Excessive investigations were done in order to derive a systematic target point search method that improved the performance of the controller tremendously.

The limit map includes limits on longitudinal acceleration based on the velocity profile of the nominal course. Furthermore, the left and right border of the road may be included that result in a maximum and minimum rate of curvature change. Finally, optional velocity restrictions allow one to constrain possible velocity profiles.

The gd-plan construction unit together with the plan-to-action mapper build a fully functional Generalized Feedback Control unit. As an example, the performance of the vehicle controller can be perceived from various plots in previous sections. In order to judge the reliability of the controller the next chapter derives the criteria of *containability*.

Chapter 6

Observation of Control Behavior

The algorithmic nature of Generalized Feedback Control directly leads to a modular structure. The division of functionality into independent modules with well defined interfaces, however, supports systematic analysis and error tracing. In the following sections it is shown how problematic situations can be identified. The investigation of several plots allow to trace issues as being problems of geometric-dynamic planning or problems of plan-to-action mapping. Further, the performance of target and limit map parameter computation can be observed.

6.1 Situational Observation

6.1.1 State Variables

The following paragraphs elaborate on data produced by the driver model and its environment in order to identify misbehavior. Being able to relate quality of control to categories of situations is a key for further analysis. A transparent display of processes and transitions allow the parameterization of problematic situations in terms of lateral displacement, curvature, amount of understeer, etc.

For the identification of problematic situations the consideration of the vehicle state is indispensable. The driver model environment produces data about the position (see figure 6.1a), the lateral acceleration, velocity, longitudinal acceleration, and front wheel angle directly together with the position of the center of gravity (see figure 6.1b). The lateral acceleration is illustrated as an arrow with the direction of the lateral acceleration starting from the vehicle's center of gravity. Lateral acceleration is an intuitive means to measure the difficulty of a situation. One is now able to relate the amount of understeer to the driver's performance in terms of precision and speed.

Also recorded are slip and yaw angle as well as their derivatives. As can be seen in figure 6.2, slip angles at the rear and the front allow one to judge the under-/oversteer behavior of the vehicle. In the figure, the absolute value of the rear slip angle is always less than absolute value of the front wheel's slip angle. This is typical for *understeer*. By displaying the lateral and angular

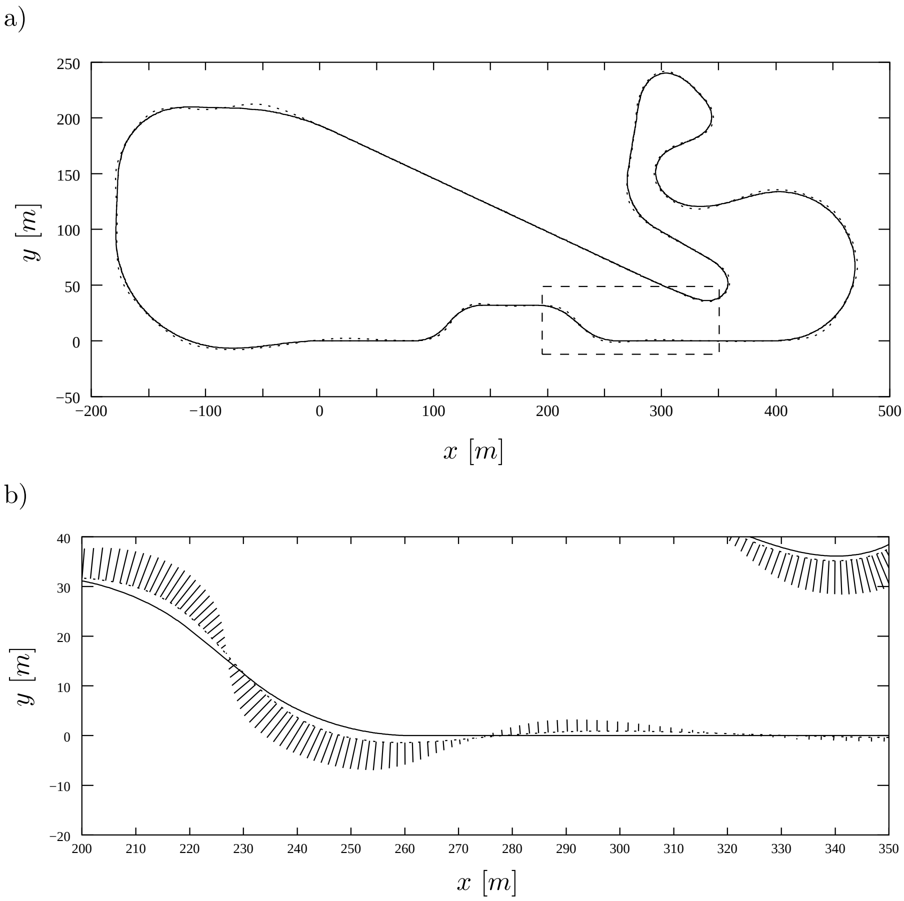
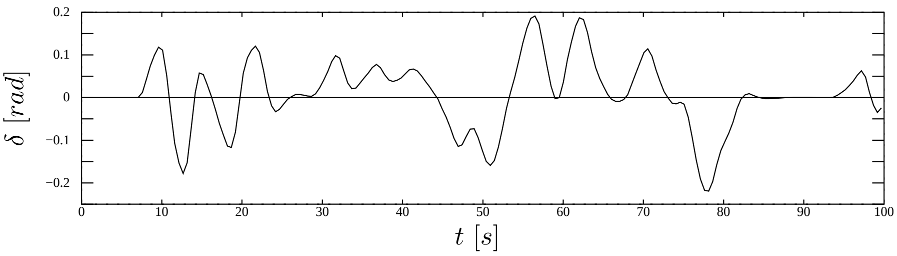


Figure 6.1: Display of vehicle states. a) Driven trajectory (dotted line) of the vehicle together with the nominal course (solid line). b) Zoom into the dashed rectangle in figure 6.1a. Lateral acceleration is displayed by lines from center of gravity of the vehicle to the side where it acts.

displacement to the nominal course, one is able to judge the overall performance of the vehicle controller. However, these plots do not permit a detailed analysis of the driver. They allow one to quickly determine situations where the driver has problems, such as in figure 6.1 where the driver deviates enormously from the nominal course.

a)



b)

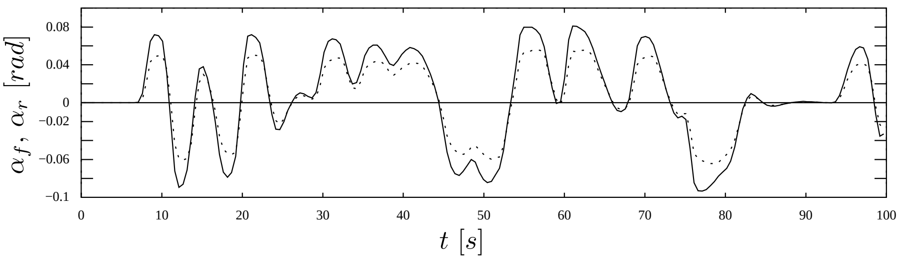


Figure 6.2: Display of several angles. a) Front wheel angle. b) Slip angle in front (solid line) and rear (dotted line).

6.1.2 Vehicle Picture

The illustration of the vehicle picture provides an intuitive view on specific situations as shown in Figure 6.3. These plots make it possible to simultaneously observe a whole set of system parameters, such as front wheel angle, vehicle orientation, vehicle position, lateral displacement and angular error to the nominal course. In many situations plot of the vehicle picture are far more effective than observations on time profiles of system parameters.

The abovementioned tools allow to identify situations of misbehavior. It can be identified in terms of deviations, oscillations, etc. At the same time, one develops first ideas about the circumstances under which certain errors occur. In the vehicle example one might identify domains of yaw ratios, curvatures or speeds that cause the control system to fail.

6.2 Plan-to-Action Mapping

Having identified problematic situations for the controller, it is now possible to trace errors and unsatisfactory behavior down to the units of gd-plan construction and plan-to-action mapping. The quality of the plan-to-action mapper can be determined by comparing the desired gd-plan (or one of its interpretations, see section 2.4) to the actually performed geometric dynamic output.

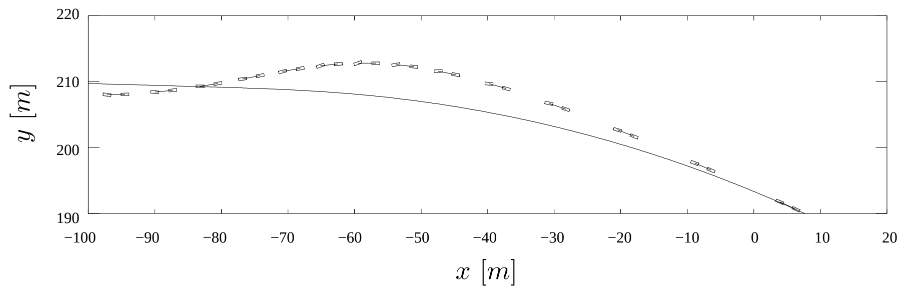


Figure 6.3: Graphical display of the vehicle as a bicycle model. The inclination of the front wheel is five times amplified.

In order to localize errors and trace them further down into submodules of the plan-to-action mapper, the driver model provides the following informations:

- The difference between desired and achieved geometric-dynamic behavior.
- The star parameters $\underline{p}^*$ that were computed for each specific instant.
- Information about miscellaneous parameters inside the plan-to-action mapper.

The difference between expected and achieved geometric-dynamic behavior requires some more detailed discussion as provided in the following subsection.

6.2.1 Precision

An *interpretation of a gd-plan* as defined in section 2.4 is a useful tool for error analysis of the plan-to-action mapper. When examining the precision of lateral control, for example, a displacement error is certainly more meaningful than errors in rate of curvature change. The implementation of the vehicle controller produces a file which contains a comparison between expected and achieved geometric-dynamic behavior. It contains a set of errors based on interpretations of gd-plans as well as some initial state parameters. Relating initial state parameters to errors allows one to identify situations in which the plan-to-action mapper performs with a low precision.

The principle of positional deviation is depicted in figure 6.4. The parameters Δx and Δy are calculated with respect to the coordinate system local to the vehicle at the time when the previous control impulse was set. These two displacements allow judgments about the precision of longitudinal and lateral plan-to-action mapping. Figure 6.5 gives an example of how the plan-to-action mapper behaves. In this case, it shows a very low deviation in lateral and longitudinal control.

Another useful plot is the curvature error with respect to lateral acceleration. It indicates the 'difficulty' of the situation. A result of an intermediate version

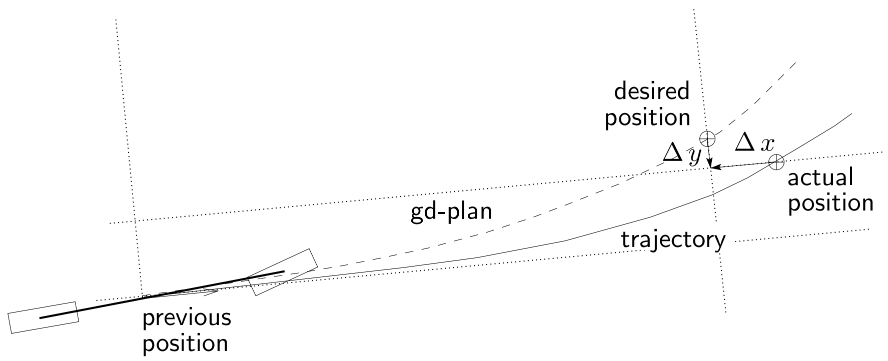


Figure 6.4: Longitudinal and lateral deviations.

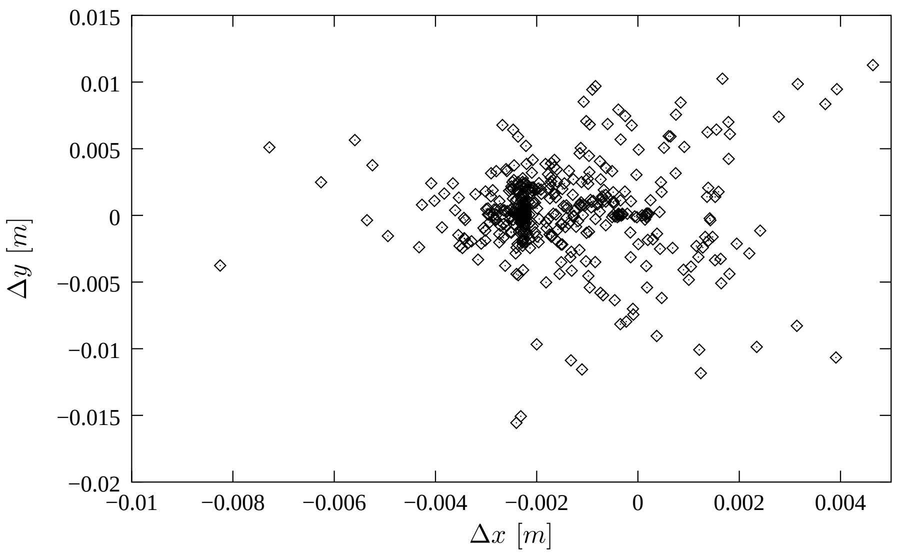


Figure 6.5: Longitudinal deviations Δx with respect to lateral deviations Δy between desired and real position recorded during simulation experiment.

of the vehicle controller is depicted figure 6.6. The linear dependence between lateral acceleration and the curvature error is indicated by a solid line.

As mentioned earlier, curvature increases with an increase of the front wheel angle. From the plot in figure 6.6, it can be concluded that the vehicle controller did not turn the front wheel angle enough at higher lateral accelerations. This shows one way how to express the 'learning'-process a driver model has to undergo in order to adapt itself to a new vehicle: The driver model has to learn the required increase of front wheel input corresponding to lateral acceleration. In other words, it has to learn the under- and oversteer behavior of the vehicle. With the current implementation of the plan-to-action mapper such errors do not occur any longer.

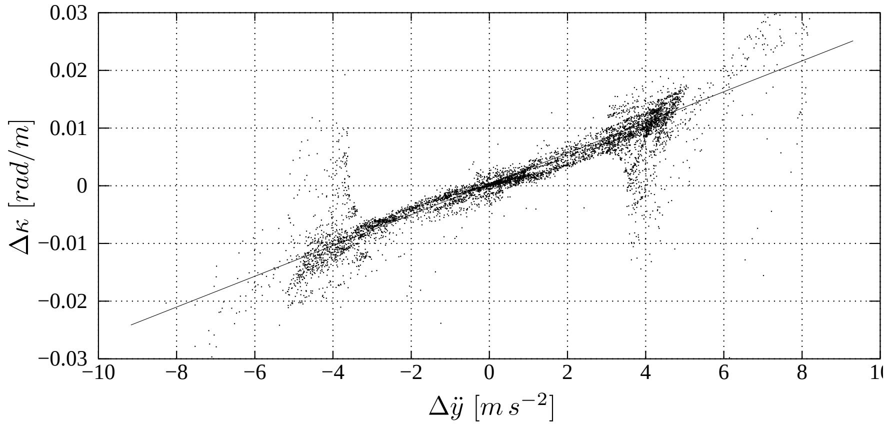


Figure 6.6: Curvature deviation with respect to lateral acceleration.

6.3 Geometric-Dynamic Plan Construction

In the previous section it was demonstrated how control behavior can be associated with modules of plan-to-action mapping. This section deals with the relation of geometric-dynamic behavior to modules of the gd-plan construction. This procedure is by far the most complex part of a Generalized Feedback Controller. Even with an almost perfect plan-to-action mapper the overall performance may be bad when the gd-plan constructed in a specific situation is inappropriate, i.e. leads the plant into undesired states. In order to trace the information processing inside the gd-plan construction unit several data is provided:

- The planned trajectory for the subsequent control interval.
- Information about the target pillar being cut by the limit maps or not.

- Target map parameter computation:
  - Parameters of the target map.
  - Lines from c.g. of the vehicle to the chosen target points.
  - The planned trajectory through the computed target point.
  - The curve chosen to nestle back to the nominal track.
- Limit map parameter computation:
  - Velocity profile of the course computed based on its curvature profile and a maximum lateral acceleration.
  - Arrows from current speed and position to speed target at the end of the subsequent control interval due to course geometry considerations (see section 5.5).
  - The most critical braking graphs due to course geometry considerations.

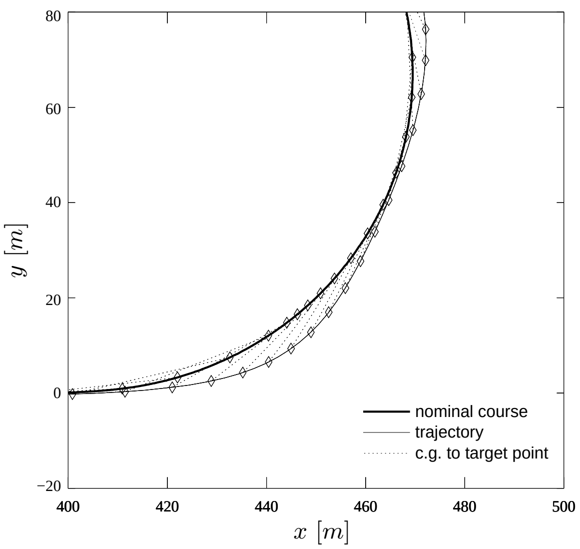
The following two subsections discuss two most important of those issues: trajectory and velocity profile planning.

6.3.1 Trajectory Planning

The search for a target point builds the core of the computation of a target map. Therefore, the vehicle controller provides some output allowing investigations on the behavior of different target point search methods. The discussion of section 5.2 already mentioned the complexity of this issue. Figure 6.7 shows two examples displaying the planned trajectory of the vehicle controller. In figure 6.7a line is drawn from the actual position of the vehicle to the target point that the driver model planned to drive through. This allows one to observe if the target point is chosen too far or to close to the actual position. Figure 6.7b shows the planned trajectory resulting from the rate of curvature change κ' that was calculated for the target point. Figure 6.7b shows a very bad deviation between planned and driven trajectory. It demonstrates that in this particular situation the plan-to-action mapper performs obviously not very precise. However, this plot was made with large control intervals $T_c = 0.8\,s$ to emphasize the effect.

Using nestle curve allows target points which are not necessarily located on the nominal course (see section 5.2.3). Potentially, this may guide the vehicle sequentially away from the track. By observation of the nestle curve plots this phenomena is illustrated in figure 6.8. Each *single one of the nestle curves* properly nestles back to the nominal course eventually. However, the *sequence of planned trajectories* that result from the sequence of nestle curves guides the vehicle away from the track. Fortunately, such phenomena are rare with the current version of the driver.

a)



b)

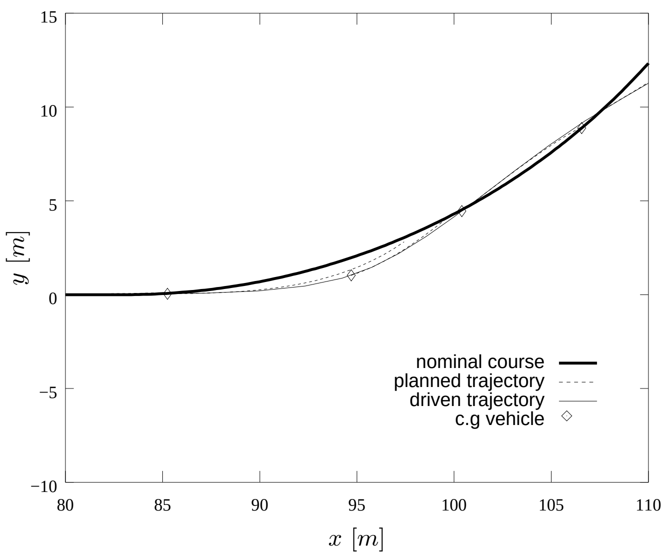


Figure 6.7: Investigations on the gd-plan. a) Lines from actual position to target point. b) Comparison between planned trajectory and driven trajectory.

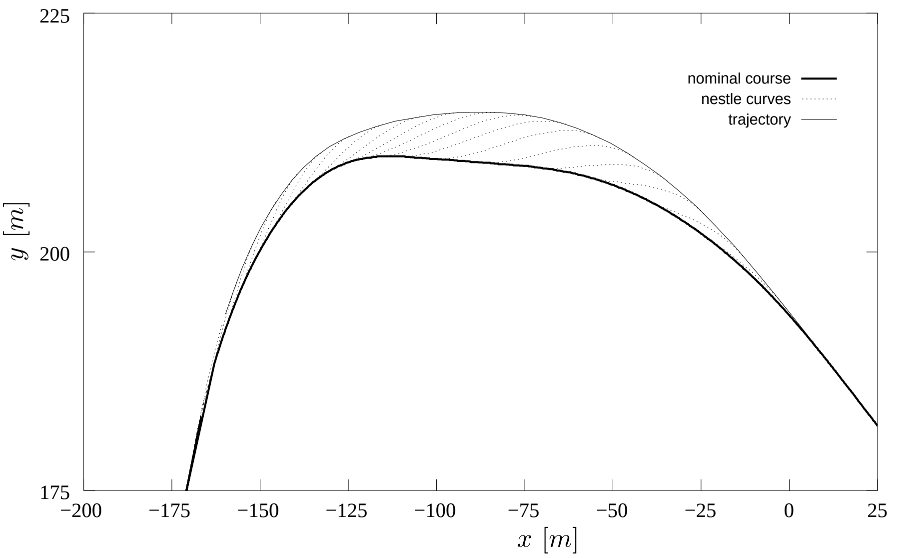


Figure 6.8: Nestle curves that guide back to the nominal course.

6.3.2 Velocity Profile Planning

Longitudinal planning is essential for the reliability of the system. This becomes obvious in lieu of the fact that a given curvature restricts the maximum velocity by the lateral acceleration where the vehicle begins to slide. In the developed driver model, there are two components that influence longitudinal control: the curvature profile of the geometric short term plan and the curvature profile of the nominal course.

Figure 6.9 shows an example. The shaded areas indicate the maximum velocity that should be respected. The lower solid line indicates the real velocity of the vehicle. Dashed lines indicate the desired acceleration by the velocity profile that would be reached if the acceleration is directly applied. The dotted lines finally represent the braking curves that ensure that it is possible to brake back to the nominal velocity $v_{nc}(p)$. As can be seen, the driver accelerates until $1535m$, because the velocity profile that results form the curvature profile can be easily respected. Then at $1550m$, amazingly the driver decelerates although there is no reason based on the velocity profile of the nominal course. The need to brake therefore results from the short term geometric plan. It can be concluded that the short term geometric-dynamic plan contains a higher curvature than the nominal course. This might be caused by an error in geometric-dynamic planning of the current situation or by a poor handling in previous control intervals, which brought the vehicle too far from the nominal course, so that strange geometric gd-plans were required.

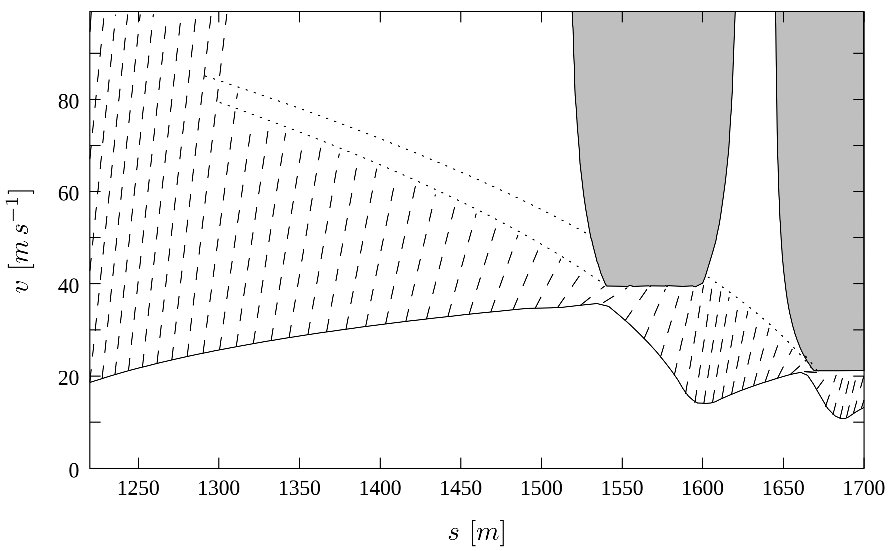


Figure 6.9: Investigating longitudinal planning: velocity profiles.

6.4 Conclusion

This chapter discussed the abilities to observe and analyze a Generalized Feedback Control unit using the example of a vehicle control unit as shown in figure 6.10. The module based, algorithmic structure facilitates an understanding of the whole system in terms of causalities, i.e. inputs and outputs in the time domain. Clearly defined interfaces allow discussions about responsibilities of submodules and detailed error tracing. Misbehavior of a module is identified by discrepancies in between nominal and actual outputs. The modular approach has further the advantage to enable the hierarchical tracing down of reasons for unsatisfactory behavior, rather than analyzing system functions as a whole.

The advantage of the algorithmic and modular approach comes with a major disadvantage: traditional concepts of stability analysis cannot be applied. In the following chapter, though, it is discussed a new methodology to determine and ensure the reliability of the controller.

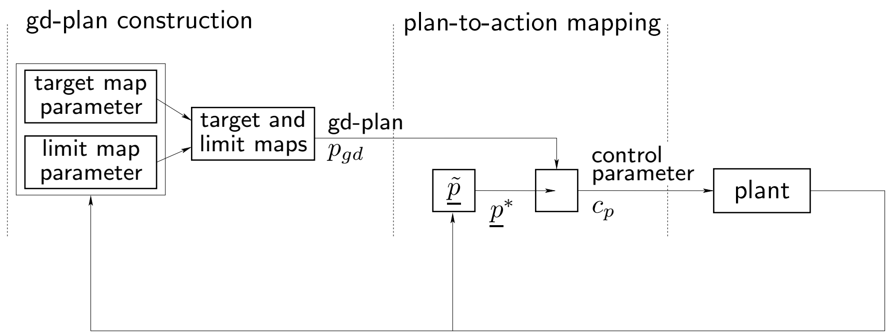


Figure 6.10: A Generalized Feedback Control system.

Chapter 7

Containability

Previous chapters discussed the construction of a controller based on Generalized Feedback Control and its application to vehicle control. This chapter focuses on the way how the reliability of the controller can be investigated. In classical control the issue of *stability* is of crucial importance. It is a measure for the reliability of a control system as it is defined:

> A dynamic system that reacts with a bounded output to a bounded input is a *stable system*.

Figure 7.1a shows an example of classical stability considerations. In this example, the bounded input, bounded output criteria is investigated by means of decaying attenuation (dotted lines) and ϵ/δ stability[1] (solid lines). Criteria to determine the stability of a system are usually based on the transfer function in the Laplace domain[2] or through Ljapunow's stability theory[3]. However, these kinds of investigations cannot be accomplished on Generalized Feedback Control, because it does not make restrictions on the algorithmic structure of its elements. Now, this chapter introduces a measurement for the reliability of a control system called *containability* through the following statement:

> If the state of a plant, controlled by a control system, never exits the domain of admissible systems states, then this control system possesses *containability*.

[1] ϵ/δ stability means that for any specified distance $\epsilon > 0$ around an equilibrium point, a δ can be determined, so that the value $x(t)$ with $t > \delta$ lies inside the ϵ environment.

[2] In linear systems, appropriate constraints on a controller can be formally expressed in the Laplace Domain. If the poles of the system transfer function have only negative real components, then it is sure that the plant's state does not diverge.

[3] Mathematically, the stability issue is related to the qualitative theory of differential equations. It is used to consider stability of state space models that are defined by a differential equation. Given a trajectory $\zeta(t)$ that is a solution of the differential equation, it is investigated if another solution that starts close to $\zeta(t)$ stays always close to $\zeta(t)$. If this condition is fulfilled, then the solution $\zeta(t)$ can be considered to be stable. Based on that idea, Ljapunow in 1892 developed the stability theory [Khalil et al., 1996].

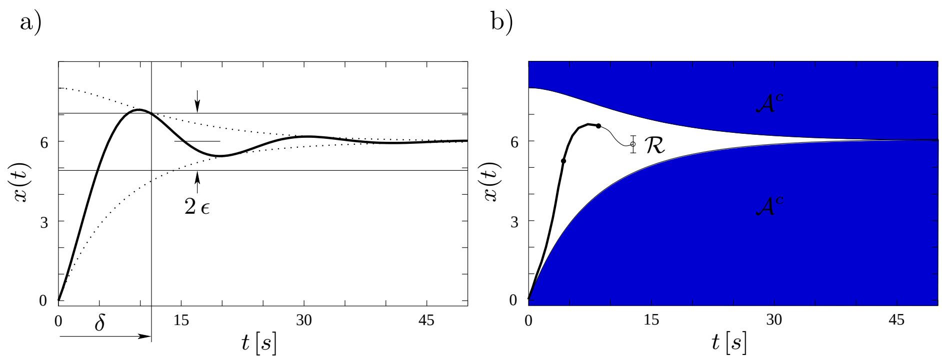


Figure 7.1: Classical stability and containability.

In figure 7.1b it is shown how to handle containability. First, an admissible $\mathcal{A}$ and an inadmissible domain $\mathcal{A}^c$ have to be defined. Second, it has to be made sure that the plan-to-action mapper is sufficiently precise. This is indicated by the range $\mathcal{R}$ to be possibly achieved in the subsequent control interval. Third, it has to be made sure, that the gd-plan does not target states that make it impossible to avoid the inadmissible domain $\mathcal{A}^c$. The following sections demonstrate how to derive stringent mathematical conditions for the criteria of containability. Its discussion is divided into two sections:

1. Avoiding to enter an inadmissible state during the subsequent control interval (figure 7.2a). The definition of a *manageable*, and a *safe target state domain* allow the specification of two criteria for containability on the short term.

2. Avoiding inadmissible states on the long term (figure 7.2b).

   Constraints are derived to prevent *dilemmas*, i.e. system states where the 'might' of the controller is not sufficient to prevent the plant from running into an inadmissible state.

The concept of containability, intuitively, provides a very strong criteria to judge the reliability of a control system. If the criteria for containability on the long term are fulfilled, the system can be considered to function well under the assumptions made about the admissible domain of system states.

7.1 The Subsequent Control Interval

In this section, it is discussed what criteria have to be met in order to avoid leaving the admissible domain of system states in the subsequent control interval. For this reasons, terms such as the *manageable domain of systems states*, *target range*, and the *domain of safe target states* are defined.

The bases for the whole discussing is the concept of an *admissible domain* of system states. It basically describes the criteria on a plant's state that have

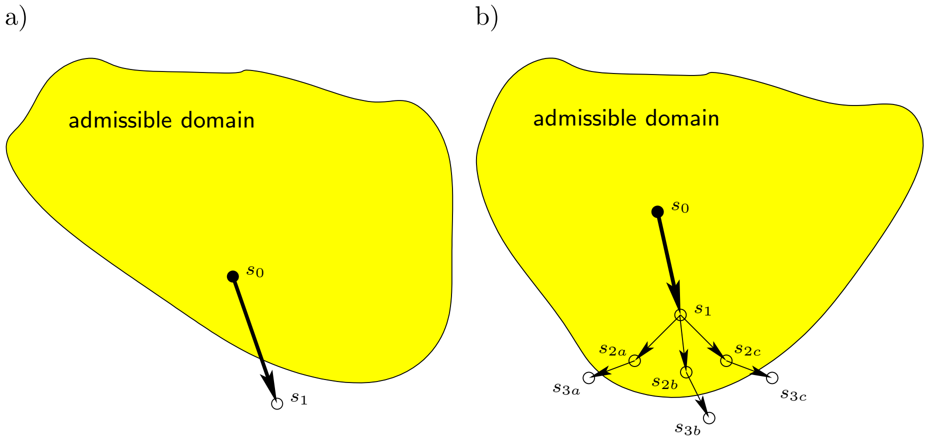


Figure 7.2: Leaving the admissible domain of plant states: a) due to error during the current control interval. b) due to a hopeless state where leaving the admissible domain is unavoidable.

to be met in order to talk about a well functioning system. For path following of a vehicle, the admissible domain $\mathcal{A}$ may be defined as the domain of states, where the absolute value of lateral acceleration is less than a certain maximum and the vehicle's position lies in a certain boundary around the nominal track, i.e.

$$\mathcal{A} \quad = \quad \left\{ s \; : \; \begin{array}{ccc} & |\ddot{y}| < \ddot{y}_{max} \\ \wedge & |\Delta y_n| < \Delta y_{n,max} \end{array} \right\} \tag{7.1}$$

For a perfect plan-to-action mapper, the desired state of the plant (determined by p_{gd}) and the achieved state of the plant are identical. In reality, however, the state at the end of the subsequent control interval can only be estimated as to lie inside a certain range. Let a target range $\mathcal{R}(p_{gd},\, s)$ be defined as the set of system states possibly reached when striving for a gd-plan p_{gd} in a given plant state s. It indicates the precision with which the plan-to-action mapper operates in a state s when a gd-plan p_{gd} is to be established.

If the plant is in a state where there is no gd-plan p_{gd} that can be fulfilled without risking to leave the admissible domain of system states, then this state cannot be considered to be manageable. Based on this idea, the *manageable domain* of system states can be defined as indicated in figure 7.3:

Definition: 18 (Manageable Domain of System States) *A set of system states $\mathcal{A}_m$ where for each element s there exists at least one gd-plan p_{gd} whose target range $\mathcal{R}(p_{gd},\, s)$ lies entirely inside the admissible domain $\mathcal{A}$ is called* Manageable Domain of System States. *This means*

$$s \in \mathcal{A}_m \quad \Leftrightarrow \quad \exists\, p_{gd} \in \mathcal{P}_{gd},\, with \;\; \mathcal{R}(p_{gd},\, s) \subseteq \mathcal{A}. \tag{7.2}$$

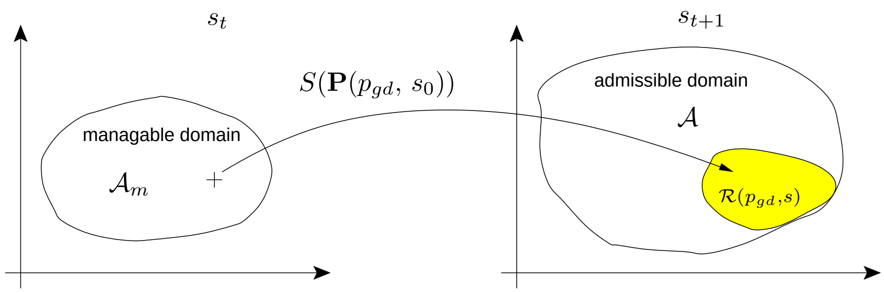


Figure 7.3: The manageable domain system states $\mathcal{A}_m$.

Practically, the domain $\mathcal{A}_m$ circumscribes the system states that are sufficiently 'known' by the plan-to-action mapper. When following the construction rules in section 3.1 it is determined by the set of initial states considered for the curve fitting process and the precision performed by the template functions. Assuming that for the given set of samples a sufficient precision is achieved, the manageable domain is equal to the domain that the samples were taken from. In section 3.2, where the example of vehicle control is discussed, the space of initial states is restricted by the initial speed and the initial front wheel angle, therefore the manageable domain $\mathcal{A}_m$ is given by

$$\mathcal{A}_m = \left\{ s \;:\; \begin{array}{rl} & |\delta| < \delta_{max} \\ \wedge & v_{min} < v < v_{max} \end{array} \right\} \tag{7.3}$$

For the plan-to-action mapper, any state in the admissible domain $\mathcal{A}$ has to be manageable in order for the controller to work properly. This is the first condition to be met for containability.

A second condition is that the gd-plan construction unit shall never target a system state where the plan-to-action mapper is not able to ensure that the plant's state will be admissible after applying a control action. Since the plan-to-action mapper operates with a certain error, a certain boundary along the inadmissible domain $\mathcal{A}^c$ cannot be targeted. This leads to the (temporary[4]) definition of a *domain of safe target states* $\mathcal{A}_s(s)$.

Definition: 19 (Domain of Safe Target States) *Given a system state s, an inadmissible domain $\mathcal{A}^c$, a target range $\mathcal{R}(p_{gd},\, s)$, and a state to be reached $S^D(p_{gd},\, s)$ by p_{gd}, the* DOMAIN OF SAFE TARGET STATES *$\mathcal{A}_s(s)$ is determined by*

$$\mathcal{A}_s(s) = \{ s_d(p_{gd},\, s)\, \forall\, p_{gd} \quad \textit{with} \quad \mathcal{R}(p_{gd},\, s) \cap \mathcal{A}^c = \emptyset \} \tag{7.4}$$

Imagine a plan-to-action mapper for vehicle control that achieves a system state with a precision $\mathcal{E}(\ddot{y})$ with respect to lateral acceleration $\ddot{y}$. Further, the

[4]In the next section, the safe target domain is redefined to meet the requirements of long term considerations.

lateral displacement never lies more than $\mathcal{E}(\Delta y_e)$ away from the nominal value Δy_e. Using equation (7.1), it is only safe to target states that lie inside the boundaries of

$$\mathcal{A}_s(s) \quad = \quad \left\{ s \ : \begin{array}{rl} & |\ddot{y}| < \ddot{y}_{max} - \mathcal{E}(\ddot{y}) \\ \wedge & |\Delta y_n| < \Delta y_{n,max} - \mathcal{E}(\Delta y_n) \end{array} \right\} \tag{7.5}$$

Summarizing the above results, there are two criteria that have to be fulfilled in order to keep the system state admissible during the subsequent control interval:

1. Any state s in the admissible domain $\mathcal{A}$ must be part of the manageable domain $\mathcal{A}_m$. This is a requirement to be met during the construction of the plan-to-action mapper.

2. For a given state s in the manageable domain $\mathcal{A}_m$ any gd-plan developed by the gd-plan construction unit has to target a safe target state.

In layman's terms, it can be said that the plan-to-action mapper has to possess enough accuracy and knowledge (expressed in $\mathcal{R}_m(p_{gd}, s)$ and $\mathcal{A}_m$) so that for any admissible state there exists at least one gd-plan p_{gd} so that the subsequent state is with certainty admissible. The gd-plan construction must always be able to find such an admissible gd-plan.

The above discussion deals with the issue of a single control interval without alluding to effects of current actions on situation further ahead in time. There are two scenarios where long term effects of actions can be ignored:

- The plant ends in a motionless state after the control action is performed. An example of such a system can be a light switch that is turned on or off.

- The controller possesses an unlimited 'might' to apply control inputs. An unlimited might to control a point mass' trajectory would be the ability to apply an arbitrarily large acceleration force in an arbitrary direction.

For these two scenarios the above mentioned criteria are enough to judge the ability to keep the plant in an admissible state. In the general case, however, the current action limits the possibilities of future states. For this reason, the following section discusses how containability can be investigated regarding the long term.

7.2 The Long Term

In general, parts of the state change of a system is caused by its own current state. For a moving particle in space the state variable velocity causes a change of the state variables of position. The amount of force that has to be applied from outside to effect a velocity change depends on the body's mass. This phenomenon is usually referred to as *inertia*. Let the intuitive term inertia indicate a system's inherent resistance against externally caused state changes. Inertia together with a restricted 'might' of the controller results in the following problem:

> The interaction of the plant's inertia with the limited 'might' of an agent results in a restricted ability to effect changes on the plant's states. It is possible that a certain system state is admissible but in this state the system may have a tendency towards the inadmissible domain that cannot be 'braked' sufficiently due to the limited might of the agent.

The investigation in the following paragraph results in a more detailed requirement on the gd-plan construction unit. Based on the concept of *dilemmas*, the concept of the *safe target state domain* $\mathcal{A}_s(s)$ is adapted including long term considerations. With the new safe target state domain, the criteria of the last section allow a general judgment on containability.

It is conceivable that the gd-plan construction unit may target a completely admissible system state from where, however, the plan-to-action mapper cannot target any admissible successor states. Consider, for example, driving a car as quickly as possible on a given track. Approaching a curve requires one to brake early enough to slow down to the velocity that is necessary to avoid losing control of the car. Before the vehicle reaches the curve, its state is always in the admissible domain independent on its velocity. Upon entering, however, the velocity may be too high to brake sufficiently enough to maintain control. Therefore, considerations about influences of current actions on future states are required. The following definition of a dilemma supports the discussion of the above mentioned problem:

Definition: 20 (Dilemma) *A* DILEMMA *is a system state s where there is no sequence of gd-plans $p_{gd,0}, p_{gd,1}, p_{gd,2}, \dots$ that allows the avoidance of the inadmissible domain of system states $\mathcal{A}^c$ with certainty.*

In case that the manageable domain covers the admissible domain, it never appears that there is no gd-plan allowing the avoidance of the inadmissible domain for the subsequent control interval. However, it is conceivable that there may be states where some of the reachable states are admissible, but all of the successor states of the reachable states are inadmissible. Let system states beyond the manageable domain be called *dilemmas of* 1^{st} *order*, and the second kind be called *dilemmas of* 2^{nd} *order*. The following definition determines the order of a dilemma

Definition: 21 (Dilemma of N^{th} Order) *A dilemma of order N is a system state, in which the longest sequence of actions avoiding the inadmissible domain $\mathcal{A}^c$ with certainty is of length $N-1$.*

Therefore, in a dilemma of first order, no state can be targeted without risking to leave the admissible domain during the subsequent control interval. In a dilemma of second order, the system is lost after two control intervals, etc. Figure 7.4 shows first, second and third order dilemmas assuming that the space of system states is discrete. It becomes more and more complicated to detect if a targeted state is a dilemma correspondent to the number of levels one desires to investigate.

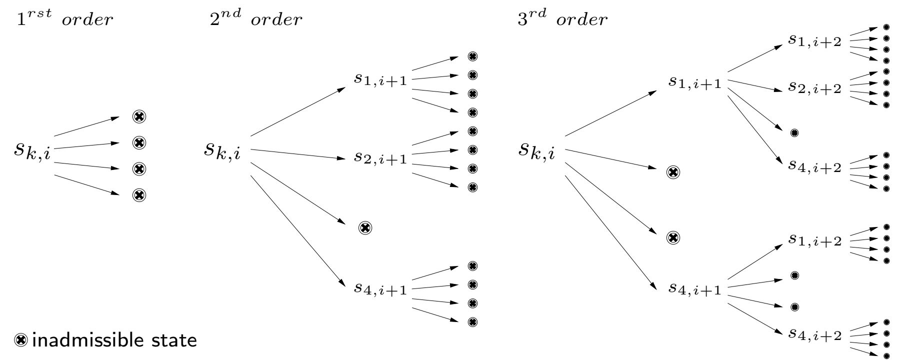


Figure 7.4: Dilemmas of different orders when actions taken by the controller are discrete.

For the general task of system control, the set of successor states is not discrete, but continuous, since the control parameters are also chosen from a continuous set of real variables $\Re$, e.g. the front wheel angle velocity and the propulsive force in a vehicle.

Figure 7.5 shows dilemmas in a continuous state space. $\mathcal{R}_1$ indicates the total domain of states that can be reached from the current state s_{act}. The case that a subset of $\mathcal{R}_1$ is inadmissible is covered by the discussion in the previous section that dealt with the short term considerations. Now, consider the case in figure 7.5a. The domain $\mathcal{R}_1$ lies completely inside the admissible domain. However, it contains a shaded subset of states that have only inadmissible successor states (the set $\mathcal{R}_2^s$). Obviously, this shaded area is part of a subset $\mathcal{D}_1$ where inadmissible states are unavoidably reached after the next control interval. Let the set $\mathcal{D}_1$ be the set of first order dilemmas. The gd-plan construction unit has to know this domain in order to avoid constructions of gd-plans that guides into a dilemma.

In figure 7.5b the shaded area in $\mathcal{R}_1$ again makes problems. Even that all its successor states $\mathcal{R}_2^s$ are admissible, they lie inside the dilemma domain $\mathcal{D}_1$ and therefore after two control intervals an inadmissible system state is unavoidable. The shaded area, in this case has to be considered as being part of the domain of second order dilemmas $\mathcal{D}_2$. In order to handle dilemma domains of N-th order the following definition is given.

Definition: 22 (Dilemma Domain) *A* Dilemma Domain *of* N^{th} *order* $\mathcal{D}_N$ *is a domain of system states, that contains all dilemmas of N-th order. This*

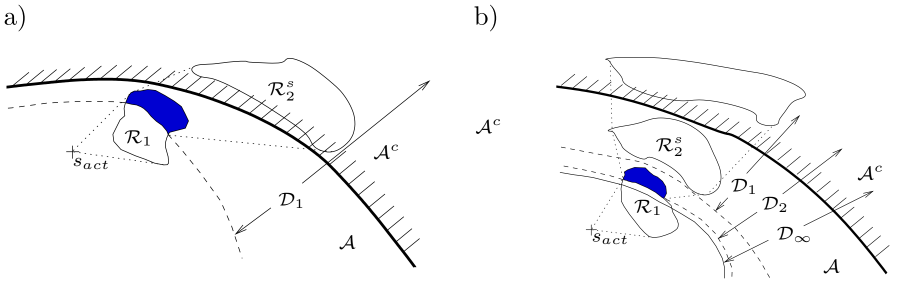


Figure 7.5: Dilemma in a continuous state space a) of first order and b) of higher order.

means, that if the system state is element of $\mathcal{D}_N$ it will be at latest after $N-1$ steps in an inadmissible state.

Logically, it follows that

$$\mathcal{D}_k \quad \supseteq \quad \mathcal{D}_i, \qquad \forall \qquad k > i. \tag{7.6}$$

Further, the complementary set $\mathcal{A}_m^c$ of the manageable domain can be considered as the dilemma domain of order zero $\mathcal{D}_0$:

$$\mathcal{D}_0 \quad \equiv \quad \mathcal{A}_m^c. \tag{7.7}$$

By statement (7.6), it can be concluded that the volume of the domains $\mathcal{D}_N$ never decreases with growing N. However, as indicated in figure 7.5b there is a possibility that the dilemma domains $\mathcal{D}_k$ converge against a dilemma domain $\mathcal{D}_\infty$ for increasing k. This way, it would be sufficient for the gd-plan construction unit to know the structure of $\mathcal{D}_\infty$ in order to avoid any possible dilemma.

Since it might be too difficult to describe precisely the domain $\mathcal{D}_\infty$ let us assume that it is possible to find a description for a set $\mathcal{D}'_\infty$ that contains all dilemmas, i.e.

$$\mathcal{D}'_\infty \quad \supseteq \quad \mathcal{D}_\infty = \lim_{n\to\infty} \bigcup_{i=1}^{n} \mathcal{D}_i. \tag{7.8}$$

If one is not able to find a domain $\mathcal{D}'_\infty$, then no assumptions about containability can be made, since it is never safe to say that a state is not a dilemma. An example from vehicle control shows how to specify a dilemma domain $\mathcal{D}'_\infty$.
Example

> Consider the task of driving as fast as possible under a given velocity profile[5] $v(l)$ with respect to the distance length l as described in figure 7.6. Let the acceleration and braking capabilities be restricted

[5]This task directly evolves from the task to drive around a course with a given curvature profile without exceeding a maximum lateral acceleration (see section 5.5).

to a_{max} and a_{min}. The velocity profile $v(l)$ directly defines the admissible domain $\mathcal{A}$ as the domain under the shaded graph of $v(l)$. Let us start from the current state s_{act} characterized by a velocity v_{act} at a certain distance on the track l_{act}. Let $\mathcal{R}_1$ be the range of possible states that can be reached in the next control interval. It results from applying all possible accelerations between a_{min} and a_{max}.

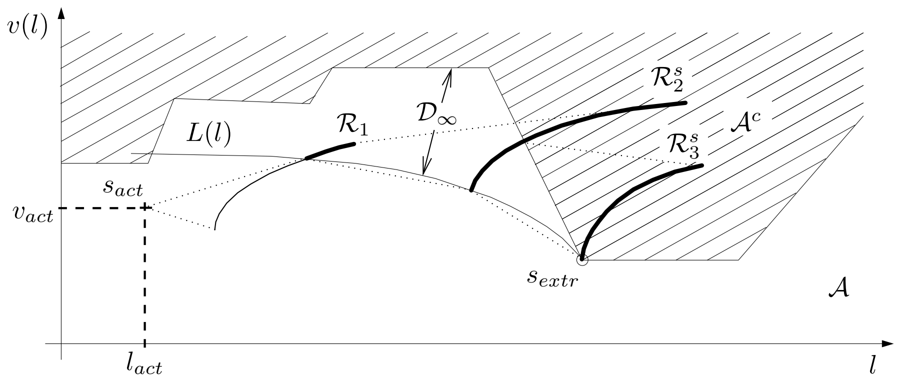


Figure 7.6: Example: domain of system states parameterized as velocity v and travel distance l. The admissible domain lies underneath the graph of $v(l)$.

How is it possible to avoid entering a state of a dilemma, i.e. a situation where it is not possible to brake enough? Assume, one is able to determine a point s_{extr} on the velocity profile $v(l)$ which demands the most braking effort. The velocity profile to brake with the maximum deceleration given by a_{min} to this point is given by the graph of $L(l)$. It will now be shown that any state $(v_s,\, l_s) \in \mathcal{R}_1$ where $v_s > L(l_s)$ is a dilemma, even if one does not know of which order.

Consider the dark subset of $\mathcal{R}_1$, where the states lie above $L(l)$. Since, it is not possible to brake harder than with a_{min} all actions that result from these states will produce a later state that lies above $L(l)$. The set of the states that can be reached from inside the dark subset of $\mathcal{R}_1$ is designated as $\mathcal{R}_2^s$. Some elements of $\mathcal{R}_2^s$ already enter the shaded area of the inadmissible domain $\mathcal{A}^c$. But even some of the states in $\mathcal{R}_2^s$ that are still admissible will later result in an inadmissible state. This is due to the limited deceleration that makes it impossible to fall below the line $L(l)$. Independent of how many steps there are until the vehicle reaches the point s_{extr} it will not be able to brake down enough and it will certainly run into the inadmissible domain $\mathcal{A}^c$. Thus, for this case there exists a dilemma domain $\mathcal{D}'_\infty$ as postulated in equation (7.8) that includes the dilemma domains of any order. It can be defined as

$$\mathcal{D}'_\infty \;\equiv\; \{(v,\, l):\; v \geq L(l)\}\,. \tag{7.9}$$

This example, which has a practical application in vehicle guidance, is treated in section 5.5 in detail. It showed that a description of the domain $\mathcal{D}'_\infty$ is practical and not something abstract.

At this point a sufficient condition can be stated that disproves containability for a given system when fulfilled:

Criterion: 1 (No Containability) *(sufficient condition) If there exists an N so that $\mathcal{D}_N$ covers the whole admissible domain $\mathcal{A}$, i.e.*

$$\mathcal{D}_N \supseteq \mathcal{A}, \tag{7.10}$$

or respectively no limit domain $\mathcal{D}'_\infty$ can be specified that does not cover the whole admissible domain, then the control task cannot be accomplished with certainty. The control unit CANNOT POSSESS CONTAINABILITY.

For the short term considerations it was enough to require from the gd-plan construction unit not to target a system state outside the *domain of safe target states* $\mathcal{A}_s(s)$. Together with the concept of the dilemma domain $\mathcal{D}'_\infty$ the promised redefinition of the safe target state domain can be accomplished:

Definition: 23 (Safe Target Domain of System States) *Given a system state s, an inadmissible domain $\mathcal{A}^c$, a target range $\mathcal{R}(p_{gd},\, s)$, and a state to be reached $s_d(p_{gd},\, s)$ by p_{gd}, the* DOMAIN OF SAFE TARGET STATES *$\mathcal{A}_s(s)$ is determined by*

$$\mathcal{A}_s(s) \;=\; \{s_d(p_{gd},\, s)\,\forall\, p_{gd} \quad \textit{with} \quad \mathcal{R}(p_{gd},\, s) \cap \mathcal{D}'_\infty \,=\, \emptyset\} \tag{7.11}$$

With this redefinition the criteria derived for short term considerations (section 7.1) are still valid and are sufficient to determine containability, i.e. the ability of the controller to keep the plant in an admissible state.

7.3 Conclusion

Due to the fact, that Generalized Feedback Control does not make any assumptions on the algorithmic structure of the gd-plan construction unit or the plan-to-action mapper, standard stability analysis tools cannot be applied. Therefore, a new means to judge the controller's reliability was introduced based on the idea of keeping the plant's state inside an admissible domain: containability.

The discussion was split up into two sections. In the first section a criteria for the plan-to-action mapper and the gd-plan construction unit was derived on the short term, i.e. for one single control interval. First, the *manageable domain* $\mathcal{A}_m$ of the plan-to-action mapper has to cover the admissible domain $\mathcal{A}$ as specified by the control task. Second, the gd-plan construction unit shall only compute gd-plans that target a state inside a so called *domain of safe target states* $\mathcal{A}_s$. The example of vehicle control provided intuitive examples for the derived concepts.

The avoidance of inadmissible states on the long term led to the definition of *dilemmas*, i.e. states in which it is impossible to avoid running into an inadmissible state sometime ahead in future. The notion of a dilemma domain including all dilemmas $\mathcal{D}'_\infty$ allowed a redefinition of the domain of safe target states $\mathcal{A}_s$. With this new understanding of safe target states, the same criteria as for short term considerations allows a judgment about the containability of a controller on the long term.

The examples treating vehicle control gave an impression how to proceed in order to determine the containability of a controller. The first criteria, i.e. that the manageable domain has to cover the admissible domain can be achieved through a sufficient precision when constructing the plan-to-action mapper. The second criteria, i.e. that only gd-plans shall be computed targeting safe target states can be achieved by transforming the safe target domain $\mathcal{A}_s(s)$ into a domain of safe gd-plans $\mathcal{A}_{gd}$ and using it as a limit map.

Chapter 8

Conclusion and Outlook

In this dissertation, the discrete control method *Generalized Feedback Control* was developed and applied to vehicle control. With this method feedback is divided into two categories: *motivation matching* and *circumstance cognition*. Whenever a control signal is computed both types of feedback are processed.

Interfering the current state of the plant with a motivation results in a nominal motion, i.e. a gd-plan, for the subsequent control interval. Knowing the current circumstances (the plant's state) control parameters can be found, in order to establish the gd-plan. The correspondent two units were named *gd-plan construction unit* and *plan-to-action mapper*.

A procedure was specified to construct a plan-to-action mapper based on a hierarchical curve fitting procedure. As a bases, a huge database was build storing the plant's reactions to different control inputs for varying initial states. Investigating the samples of this database, functions were determined to map from a given gd-plan to appropriate control parameters for one particular initial state. Then it was investigated how these functions change dependent on the initial state. The result was a universal formula allowing to find appropriate control parameters for the range of all possible initial states covered by the database. Through these formula the circumstance cognition feedback has been implemented.

In order to implement a unit for motivation matching a mathematical terminology has been developed called *target and limit maps*. This methodology separates motivations into motivations of 'fear', i.e. the avoidance of inadmissible states and 'desire', i.e. the targeting for a specific state. Using these concepts the utility of each gd-plan with respect to a motivation can be described through a scalar function. A combination procedure is described that results in one single utility function for all related motivations. An optimum search then allows to find one distinct gd-plan for a given situation that is consistent with all related motivations.

Using target and limit maps, a gd-plan construction unit was developed for vehicle control. The different physical aspects of driving, such as maximum lateral acceleration, a target point etc. resulted in different motivations modeled through target and limit maps. The discussion about the target point problem, in particular, unsealed the big advantage of being able to treat geometric-dynamic problems completely independent from problems related to the plant's system function. Together with the high performance of the plan-to-action mapper, a gd-plan construction unit was developed that was able to perform a very precise path following at high speeds. It was even possible to drive at lateral accelerations of 7 m/s^{-2} and higher. A similar performance was also achieved using non-linear decoupling [Freund, 1982]. Furthermore, the controller's deviation from the nominal track could be decreased to a few centimeters (see figure 5.18).

Having accomplished a functioning control system one question remains: How can the reliability of the system be investigated? Since traditional stability analysis cannot be applied, another means of showing the reliability of the controller using the idea of an admissible domain of system states was introduced. Two criteria for *containability* were identified that allowed to determine the ability of a controller to maintain the plant's state admissible on the long and the short term. The first criteria imposes precision constraints on plan-to-action mapping. The second criteria imposes requirements on gd-plan construction.

8.1 Future Work

The method of Generalized Feedback Control introduced in this dissertation is disruptive with respect to existing control methods. In order to fully exploit the potential of this method a variety of different control problems would have to be treated.

The aspect of adaption, mentioned in section 2.1.1, was not at all treated this dissertation. In order to do this the amount of parameters of the plan-to-action mapper has to be reduced, by determining their dependency on configuration parameters. It is however, important for systems 'on the fly' to be able to adapt. A vehicle may experience a different frictional coefficient when driving on ice, different air drag coefficients dependent on air pressure and so on.

The concept of containability provided a reliability measure based on two criteria for plan-to-action mapping and gd-plan construction. Even that, the examples mentioned in the correspondent chapter lead straightforward to the containability proof for the vehicle controller, it was not accomplished in every detail. Concerning the example application of a vehicle control unit the following improvements can be accomplished:

1. Plan-to-Action Mapping:
   - (a) Other physical concepts that affect the vehicle's motion could be included, like the rolling resistance and the phase shift in the steering system.
   - (b) Different vehicle models can be used and a formula for a generalized plan-to-action mapper could be derived for the class of ground vehicles.

(c) Methods have to be developed in order to capture relationships between the tilde parameters $\underline{\tilde{p}}$ and physical parameters of the plant. This is necessary, as mentioned in section 3.1, in order to be independent of one particular plant.

2. GD-Plan Construction:

(a) Limit maps could be created including calculations on the frictional ellipse. This could be done including the influence of aero-dynamics on wheel load. Figure 8.1a shows an example of the frictional ellipse with a slightly higher maximum on lateral acceleration than longitudinal acceleration. The frictional ellipse has a direct equivalent in the space of gd-plans. A qualitative plot on the domain of admissible gd-plans is depicted in figure 8.1b. Limit maps provide an excellent means to model the influence of the frictional ellipse.

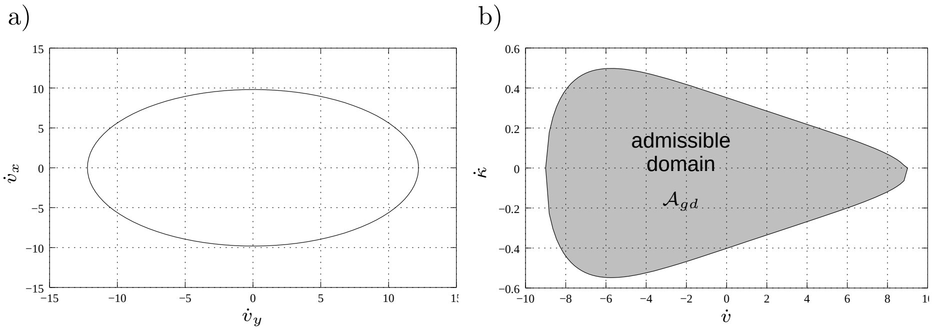


Figure 8.1: Frictional ellipse for a specific situation and influence on the domain of admissible gd-plans. a) Frictional ellipse. b) Admissible domain of gd-plans based on frictional ellipse.

(b) Maximization of velocity. This could be accomplished by creating a new type of nestle curves that include velocity considerations. However, this is difficult, since there are two goals that have to be considered at the same time. First, the nestling to the nominal course has to happen as early as possible. Second, the resulting curve should be as smooth as possible. These two goals are contradictory.

Another way to handle the optimization of velocity would be to use directly the mechanisms of target and limit maps. This means that a *second target map should be developed* for the optimization of speed.

The promising results using Generalized Feedback Control for vehicle control, insinuate the potentials of this approach that are waiting to be exploited. At no point its methods rely on a description of a plant in terms of linear systems. This makes it an interesting approach for any kind of non-linear control. Even that, no stability proof in the classical sense can be provided, the concept of containability allows a stringent measure of the reliability of the system. With these two basic features Generalized Feedback Control spans a wide spectrum of interesting future research topics.

Appendix A

Historical Background

When reviewing the historical background of Generalized Feedback Control it is good to understand that it evolved from the field of vehicle path following control, i.e. a control problem that is hard to handle with classical control techniques. In the following chapter examples of different vehicle control systems demonstrate advantages and shortcomings of present control methods.

The discussion starts of with a classical linear feedback control approach developed in the seventies of the last century. A more recent approach, the so called *non-linear decoupling* of differential equations is discussed and shows a decent performance. A brief discussion of Fuzzy Control an Neural Networks demonstrates how insights about human control influenced the evolution of control methods. An approach based on classification techniques is shown to work towards the concept of circumstance cognition. Finally, when discussing a controller for featuring non-holonmic path following, it becomes clear how Generalized Feedback Control was impending to find a mathematical concretization as established in this dissertation. Born out of an algorithmic necessity, the approach allows to quantify at the same time concepts such as *motivation* and *knowledge* that are very intuitive attributes to a 'natural' control system. The following sections provide an understanding of the control philosophies that inspired the development of Generalized Feedback Control. It is clearly demonstrated how it satisfies certain needs that lack all other approaches.

A.1 Linear Feedback Control

In linear feedback control, the calculation of control parameters is based on the weighted sum of several deviations between a nominal state and the current or the predicted state of the plant. In application to vehicle control, physical values are chosen that are related to the vehicle and the road geometries.

Two categories of approaches have to be mentioned representing linear feedback control concepts. The first category uses a nominal course that the driver wants to follow (figure A.1a). A second category uses the concept of road borders to draw conclusions about the steering action [Godthelp and Konings, 1981][1]. It is based on the distance between the current position of the vehicle and the point of an estimated crossing of road borders. The conclusions drawn from the nominal course related control approaches, however, can be extended to the road border related approaches. The following paragraphs, therefore, only discusses nominal course related controllers.

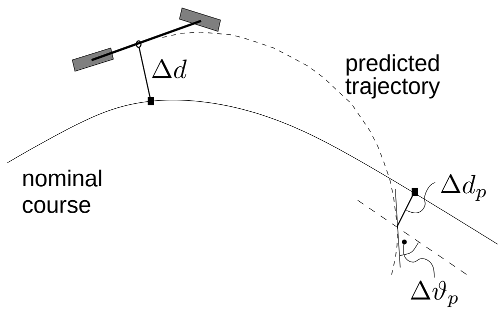


Figure A.1: Deriving the vehicle control action based on a nominal course. Lateral displacement Δd, the predicted displacement Δd_p and the predicted angular deviation $\Delta\vartheta_p$ are suitable candidates as inputs for the controller.

Research accomplished in the area of human driver modeling resulted in a variety of different approaches to handle vehicle control based on linear feedback control. In the seventies, a two layer model to describe human steering behavior has been introduced by McRuer [Weir and McRuer, 1973, McRuer and Krendl, 1974, McRuer et al., 1973]. This model consists of a subsystem for compensatory (closed-loop) and another for anticipatory (open-loop) steering control. The following section discusses a variant of it: the Donges [Donges, 1978] model.

Donges' driver model as shown in figure A.2 consists of an open-loop unit and a closed-loop unit. The open-loop unit uses an anticipated curvature of the nominal course $\kappa_n(t-\tau)$ to compute a part of the front wheel angle δ_a.

The closed-loop unit calculates the compensatory part based on differences between the nominal curvature $\kappa_n(t)$ and the actual curvature κ as well as the lateral displacement Δd and the angular displacement $\Delta\psi$ of the vehicle from the nominal course. Such a system can be transcribed mathematically as

$$\begin{aligned} \delta_a(t) &= c_{\kappa,n}(t) * \kappa_n(t), && \text{(A.1)} \\ \delta_c(t) &= c_\kappa(t) * (\kappa_n(t-\tau) - \kappa(t)) + c_y(t) * \Delta d(t) \\ &\quad + c_\psi(t) * \Delta\psi(t-\tau). && \text{(A.2)} \end{aligned}$$

with $c_\kappa(t)$, $c_y(t)$, $c_\psi(t)$ and $c_{\kappa,n}(t)$ are the impulse responses of the correspon-

[1]The TLC concept is widely applied in different driver models (e.g. [Kopf, 1992] and [Jürgensohn, 1997, pages 256–262]) to get a measure for the call for action of the driver.

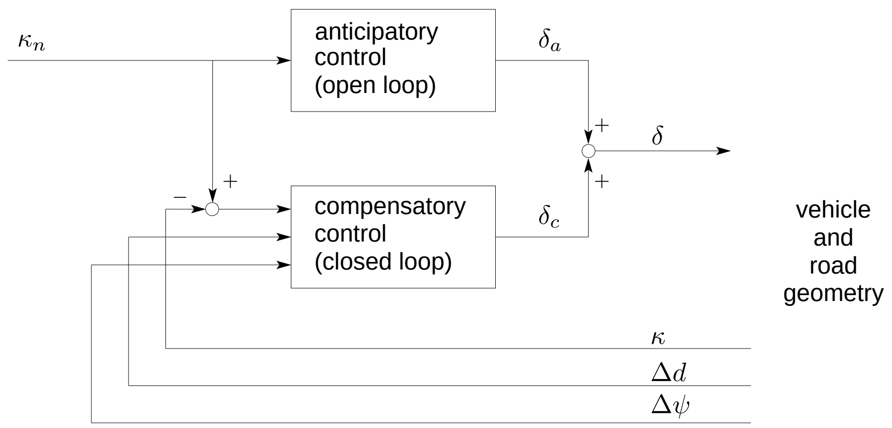


Figure A.2: A two layer model of driver steering behavior. Following: [Donges, 1978].

dent linear systems[2]. The separation of the front wheel input $\delta(t)$ into an anticipatory part $\delta_a(t)$ and a compensatory part $\delta_c(t)$ is very reminiscent of common experience in driving. It is intuitive that human drivers set the front wheel angle based on two components:

- One anticipatory component $\delta_a(t)$ corresponds to the 'learned behavior', i.e. the process of setting a distinct front wheel angle due to the curvature that has to be driven. This component is characteristically smooth, since it only depends on the smooth curvature profile of driveable tracks.
- The compensatory part $\delta_c(t)$ of the control signal based on differences between the desired state and the state that is reached by the 'learned behavior'. This part is characterized by a relatively rough shape, since compensation happens relative to errors of the smooth 'learned behavior'.

The front wheel input is a result of a concurrent activity of both kinds of learned and compensatory behavior. It is assumed that their interaction can be expressed by a plain sum

$$\delta(t) = \delta_a(t) + \delta_c(t). \tag{A.3}$$

Since this sum remains a sum in the frequency domain, it is possible to identify the learned behavior as the low frequency parts caused by the smooth signal $\delta_a(t)$. The higher frequency parts can be considered to be brought about by the rough compensation signal $\delta_c(t)$.

This model works satisfactorily in as long as the vehicle stays in a well defined operating point, i.e. at low lateral accelerations. However, implementations of

[2]The '$*$'-sign in equation (A.1) denotes the convolution. The effect of the convolution corresponds to a certain delay in reaction of the human driver.

this concept show an unsatisfactory performance at higher lateral accelerations[3]. At this point two structural deficiencies can be identified:

- A *linearization* of the plant about an operating point is required. These linearizations do not provide a sufficiently precise description of the system function, especially not for longer control intervals or in situations where the system is sensitive to control parameters.

- The control problem has two subtasks, lateral (curvature) and longitudinal (speed) control. These two problems are handled in *separate independent units.*

  However, in reality, there is a coupling between lateral and longitudinal control. Longitudinal control should consider the influence of the front wheel angle, since frictional forces on the front wheels decelerate in longitudinal direction. On the other hand, the lateral control unit should include considerations about the future profile of lateral acceleration[4]. Lateral acceleration depends on velocity providing a coupling between longitudinal and lateral control. A completely isolated treatment of both does not reflect the physical reality.

Nevertheless, further research has been accomplished based on linear feedback control. A recent contribution on driver models can be found in [Modjtahedzadeh and Hess, 1993]. A review on the subject of linear control for the use of driver models may be examined in [Guo and Guan, 1993].

The two concepts of anticipation and compensation have a close relationship to the two basic concepts of Generalized Feedback Control. Anticipation can be considered as a means to accomplish motivation matching, i.e. to determine a suitable transition of the plant state into a desired state. Compensation as discussed earlier (section 2.1.1, page 7) consists of circumstance cognition and parameter adaption (i.e. learning).

A.2 Non-Linear Decoupling

A concept called *non-linear decoupling* of state variables [Mayr, 1991, Freund and Mayr, 1989, Voegel, 1997] provides another means to compute control parameters for vehicle control. Non-linear decoupling of differential equations was originally used to control industrial robots, but has also proven to be effective in the automated control of vehicles. The following paragraphs describe only the fundamental concepts. For an explicit discussion of the mathematical background one may refer to the aforementioned literature.

[3]To overcome this problem, Post [Post et al., 1997] uses input-output, model reference adaptive control and was able to produce more stable results at lateral accelerations of about $6m\,s^{-2}$.

[4]This is because, lateral acceleration mainly determines the mapping from desired curvature to the appropriate front wheel angle.

The method of non-linear decoupling is based on the so called *state space formulation.* In the general non-linear case, it is defined the following way [Freund, 1982]:

$$\dot{\vec{x}}(t) = a(\vec{x}(t)) + B(\vec{x}(t))\,\vec{u}(t), \tag{A.4}$$
$$\vec{y}(t) = c(\vec{x}(t)) + D(\vec{x}(t))\,\vec{u}(t), \tag{A.5}$$

where the vector $\vec{x}(t)$ represents the current state of the system. $\vec{u}(t)$ contains the system inputs and $\vec{y}(t)$ the system outputs. The system vector $a(\vec{x}(t))$ describes the dynamic characteristics and $B(\vec{x}(t))$ describes the influence of the input $\vec{u}(t)$ on the system's state. $c(\vec{x}(t))$ defines how the output values can be derived from the system's current state and the matrix $D(\vec{x}(t))$ defines the direct influence of the input values $\vec{u}(t)$ on the output values $\vec{y}(t)$.

For vehicle control, the state vector $\vec{x}(t)$, the input vector $\vec{u}(t)$ and the output vector $\vec{y}(t)$ vectors are defined as:

$$\vec{x}(t) = \begin{pmatrix} \beta(t) \\ \psi(t) \\ \dot{\psi}(t) \\ v(t) \\ x_{cg}(t) \\ y_{cg}(t) \end{pmatrix}, \quad \vec{u}(t) = \begin{pmatrix} F_{s,f}(t) \\ F_p(t) \end{pmatrix}, \quad \vec{y}(t) = \begin{pmatrix} x_{cg}(t) \\ y_{cg}(t) \end{pmatrix}, \tag{A.6}$$

where $\beta(t)$ is the slip angle in the center of gravity of the vehicle, $\psi(t)$ is the yaw angle, $\dot{\psi}(t)$ is the yaw rate, v is the velocity, and $x_{cg}(t)$ and $y_{cg}(t)$ are the coordinates of the vehicle's center of gravity. The control input is specified here by the side force $F_{s,f}(t)$ on the front tire and the rear propulsive force $F_p(t)$. Based on the required forces, the front wheel angle and the propulsive force can be calculated. This last step, however, is neglected in the following paragraphs.

The aim is now to derive a control rule, i.e. a formula that allows to compute the control parameters $\vec{u}(t)$ as a function of the actual state $\vec{x}(t)$ and a desired output. The desired output is later defined in terms of a target point $(t_x(t), t_y(t))$.

According to equation (A.5) and (A.6) forces, i.e. the elements of the input vector $\vec{u}(t)$, are related to a position (x_{gd}, y_{cg}) as elements of the output of the system. Since a force directly causes acceleration it is preferable to describe the output in terms of accelerations. This means a new output vector $\vec{y}^*(t)$ has to be defined as

$$\vec{y}^*(t) \equiv \begin{pmatrix} \ddot{x}_{cg}(t) \\ \ddot{y}_{cg}(t) \end{pmatrix}. \tag{A.7}$$

In order to get an output equation, equation (A.5) can be reformulated as

$$\vec{y}^*(t) \equiv c^*(\vec{x}(t)) + D^*(\vec{x}(t))\,\vec{u}(t). \tag{A.8}$$

$D^*(\vec{x})$ and $c^*(\vec{x})$ have to be derived from $c(\vec{x})$ and $D(\vec{x})$. Let the desired output be defined as

$$\vec{y}^*(t) \equiv \Lambda \begin{pmatrix} t_x(t) \\ t_y(t) \end{pmatrix} - q^*(\vec{y}(t)). \tag{A.9}$$

where Λ is a weighting matrix defined as

$$\Lambda \equiv \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \tag{A.10}$$

That means that the desired acceleration in x- and y-direction is a function of the target point $(t_x(t),\, t_y(t))$ to drive through and the current position $\vec{y}(t) = (x_{cg}(t)\, y_{cg}(t))^T$. Therefore, equation (A.9) can be considered as a description of the dynamic behavior of the system. Assuming $D^*(\vec{x})$ to be invertible, the following control rule can be defined based on the last two equations:

$$\vec{u}(t) \equiv (D^*(\vec{x}(t)))^{-1}\left(-c^*(\vec{x}(t)) + \Lambda \begin{pmatrix} t_x(t) \\ t_y(t) \end{pmatrix} - q^*(\vec{y})\right), \tag{A.11}$$

Next, the vector $q^*(\vec{y})$ has to be specified. In order to relate real and nominal values directly, the first row has to contain the x-coordinate of the c.g. and the second row has to contain the y-coordinate of the c.g. According to equation (A.9), the difference of x-coordinates correspond to the x-acceleration and the difference of y-coordinates correspond to the acceleration. In order to gain another degree of freedom, the first derivative is also included. Therefore, the vector $q^*(\vec{y})$ can be specified as

$$q^*(\vec{y}(t)) \equiv \begin{pmatrix} \alpha_{1,1}\, x_{cg}(t) + \alpha_{1,2}\, \dot{x}_{cg}(t) \\ \alpha_{2,1}\, y_{cg}(t) + \alpha_{2,2}\, \dot{y}_{cg}(t) \end{pmatrix}. \tag{A.12}$$

The constants $\{\alpha_{i,k}\}_{i,\,k\in\{1,\,2\}}$ are arbitrary. Based on the equations (A.10) (A.12), the dynamic behavior of the decoupled system, given in equation (A.9), can be specified more precisely as

$$\ddot{x}_{cg} + \alpha_{1,2}\, \dot{x}_{cg}(t) + \alpha_{1,1}\, x_{cg}(t) = \lambda_1\, t_x(t), \tag{A.13}$$

$$\ddot{y}_{cg} + \alpha_{2,2}\, \dot{y}_{cg}(t) + \alpha_{2,1}\, y_{cg}(t) = \lambda_2\, t_y(t). \tag{A.14}$$

For the bicycle model as described by (B.14), (B.15) and (B.16), one gets finally the following control rule [Voegel, 1997]

$$\vec{u}(t) = \begin{pmatrix} -F_{s,r} \\ F_r \end{pmatrix} + \underline{Z} \begin{pmatrix} \lambda_1\, t_x(t) - \alpha_{1,1}\, x_{cg}(t) + \alpha_{1,2}\, \dot{x}_{cg}(t) \\ \lambda_2\, t_y(t) - \alpha_{2,1}\, y_{cg}(t) + \alpha_{2,2}\, \dot{y}_{cg}(t) \end{pmatrix}, \tag{A.15}$$

where F_r indicates the air-drag and $\underline{Z}$ is defined as

$$\underline{Z} \equiv m \begin{bmatrix} -\beta(t)\, f_1(t) + f_2(t) & -\beta(t)\, f_2(t) + f_1(t) \\ f_1(t) & f_2(t) \end{bmatrix}. \tag{A.16}$$

The abbreviations $f_1(t)$ and $f_2(t)$ are defined as

$$f_1(t) \equiv cos(\psi(t) + \beta(t)) \quad \text{and} \quad f_2(t) \equiv sin(\psi(t) + \beta(t)). \tag{A.17}$$

Even though these investigations are based on the relatively simple bicycle vehicle model, Voegel and Chucholowski showed a decent performance of the control algorithm for controlling a complex vehicle model [Chucholowski et al., 1999]. Their controller succeeded in driving with a lateral acceleration of about $7\,m\,s^{-2}$ with a lateral displacement less than $0.4\,m$ from the nominal course. Mayr [Mayr, 1991, chapter 10] showed that a more complicated model for the control algorithm does not have significant influence on performance. This approach, however, leaves an essential problem unanswered:

> The *placement of the target point.* When the target point is chosen too far ahead on the course, then the vehicle cuts the curves. When it comes too close, the vehicle crosses the nominal course with an angle too high and starts to oscillate.

The malfunctioning of this controller in certain situations insinuates the need for a short term path planning for each single control interval.

A.3 Non-holonomic Motion Planning

From the previous discussion, it became clear that for a complicated task such as vehicle control it is necessary for the controller to possess a unit for *dynamic path following.* Where the layer of navigation [Borenstein and Koren, 1991] and planning [Svestka and Overmars, 1995, Murray and Sastry, 1993, Koga et al., 1994, Fraichard, 1991] has been extensively investigated, the research on the topic of dynamic path following is very recent. Approaches dealing with this subject usually fall into the category of *non-holomic path following systems.*

The name 'non-holonomic' comes from the area of Lagrangian/Hamiltonian dynamics [Greenwood, 1988, Wells, 1967]. In this terminology, constraints on a mechanic system can be either holonomic, or non-holonomic. A holonomic constraint can be expressed in terms of the generalized coordinates, and is therefore geometric. A non-holonomic constraint restricts the first derivative of the generalized coordinates and is therefore kinematic, i.e. it restricts velocities. A vehicle is a typical example of a mechanical system with non-holonmic constraints. Instead of being able to move in any direction, the possible movements of a vehicle are restricted by its current orientation and the configuration of the front wheels.

Considering the constraints on possible motions, non-holonomic motion planning introduces the concept of 'feasible' trajectories. In detail, it tries to solve the task to follow a path as close as possible, while satisfying the kinematic and possibly dynamic constraints. In [Frezza et al., 1998, Sarkar et al., 1994] a controller is introduced that computes connecting contours towards the nominal course for each control interval through splines [DeBoor, 1978]. This approach is the closest one in its nature to the approach of Generalized Feedback Control. The mentioned literature shows a smoothening effect through the dynamic path planning in every control step. The following section describe approaches that are inspired by 'natural' control systems. This shows the motivation why Generalized Feedback Control generalized the concept of non-holonomic path following towards a control concept that is intuitively close to a human understanding of control.

A.4 Classification Approach

Probability theory is a powerful means whenever there is a lack of knowledge about causal relationships. In fact, every probability for an event is a conditional probability where the condition is the rest of unknown causal influences [Pearl, 1988].

With todays techniques of sensor devices it is not possible to measure quantitatively possible reasons inside a human operator, for example, why he did act in a situation in a specific way. The classification approach treats the intrinsic causalities as an unknown causal influence. The known influences, i.e. certain attributes of the plant, are used to specify conditional probabilities. As a result, an operator can be mimicked with respect to his input/output behavior described through probability distributions. The classification approach[5] considers the agent a priori as a black box that performs the mapping

$$situation \quad \longrightarrow \quad action. \tag{A.18}$$

Grashey introduces in his dissertation [Grashey, 1999] an approach for human driver modeling where the driving action consists of the longitudinal acceleration $\dot{v}$ and the change of the front wheel angle $\Delta\delta$, i.e.

$$action \equiv (\dot{v},\, \Delta\delta). \tag{A.19}$$

The control action results from an output of a classification unit. In term of classification methods, it has to be considered as a class in a finite set of classes, and the space of possible control actions has to be discrete. The redeemable sub-sampled space of actions can be indexed by an integer k

$$action(k) \equiv (\dot{v}_k,\, \Delta\delta_k), \tag{A.20}$$

where the pairs $(\dot{v}_k,\, \Delta\delta_k)$ should be distributed over the space of possible actions in a such a way that the errors between the actions $(\dot{v},\, \Delta\delta)$ that appear during a driving experiment and their 'nearest neighbors' $(\dot{v}_k,\, \Delta\delta_k)$ become minimal on average.

An attribute vector $\vec{x}_a$ provides a quantitative description of the situation. It contains essential environment information such as the slip and yaw angle of the vehicle, 'time-to-line-crossing' values[6] (so called TLCs) for the left and the

[5] Another probabilistic approach using Bayesian Networks in order to conclude for the driver's action can be found in [Forbes et al., 1995]

[6] The term 'time-to-line crossing' refers to the estimated trespassing of the road's borderlines when the trajectory is extrapolated with the instantaneous radius [Godthelp and Konings, 1981].

right border etc.

$$\vec{x}_a \equiv \begin{pmatrix} \beta \\ \psi \\ \dot{\psi} \\ TLC_{left} \\ TLC_{right} \\ \vdots \end{pmatrix}. \tag{A.21}$$

The exact definition of the attribute vector depends on the maneuver that has to be driven. Its constitution is decisive for the performance of the controller. $\vec{x}_a$ should exclusively contain such attributes of a driving situation that entail the most information to conclude to the action. Furthermore, they should not be conditionally dependent on each other. The mapping from a situation to an action in (A.18) can now be made concrete:

$$\vec{x}_a \quad \longrightarrow \quad k \in \{0,\, 1,\, 2,\, 3,\, \ldots N\}, \tag{A.22}$$

with k as the class number. The value $k = 0$ is usually considered to indicate the outcasts, i.e. the class of not classifiable vectors $\vec{x}_a$.

To produce a mapping from situation to action (A.18) that is able to steer a car, an exemplary driver has to be observed. His actions can be described as a stochastic process characterized by the probability density function

$$p(\vec{x}_a,\, k) \quad = \quad p(\vec{x}_a|\, k)\, p(k). \tag{A.23}$$

This function indicates the probability that a vector $\vec{x}_a$ and an action of class k appear at the same time. The class specific probability $p(\vec{x}_a|\, k)$ can be estimated by a probability distribution. For the complexity of problems of higher dimensions only the Gauß-Distribution is useful for practical applications [Patel and Read, 1996]. This distribution is defined as

$$p(\vec{x}_a|\, k) \quad \equiv \quad \frac{1}{\sqrt{(2\,\pi)^N\, |\underline{C}_k|}}\, e^{-\frac{1}{2}(\vec{x}_a - \vec{m}_k)^T\, \underline{C}_k^{-1}\, (\vec{x}_a - \vec{m}_k)}. \tag{A.24}$$

With $\underline{C}_k$ as the covariance matrix and $\vec{m}_k$ as the 'center of gravity' of class k, i.e. the average vector of all vectors that belong to class k. $\underline{C}_k$ and $\vec{m}_k$ distinctly define the shape of the distribution function for class k. Now, the *maximum-likelihood classifier* is used to determine if a situation results in an action of a specific class, as described as follows. So called *decision functions* assign to a vector $\vec{x}_a$ a value indicating the 'strength' that it should be believed to belong to class $\hat{k}$

$$d_{\hat{k}}(\vec{x}_a) \quad \equiv \quad p(\vec{x}_a|\, \hat{k}). \tag{A.25}$$

According to [Grashey, 1999], the strength to believe that $\vec{x}_a$ is rejected, i.e. an outcast is given by

$$d_0(\vec{x}_a) \quad \equiv \quad \sum_{k \neq 0} p(\vec{x}_a|\, k) - \frac{p(\vec{x}_a)}{p(k)} \frac{c_r}{c_f} \tag{A.26}$$

where c_r has to be specified as the *cost* of rejection, i.e. a constant that specifies a weight for classifying a situation as an outcast. The constant c_f has to be set as the cost for a false decision, i.e. when a class k_a is chosen where a class k_b should have been chosen. Then, it can be said that an attribute vector $\vec{x}_a$ belongs to class $\hat{k}$ if $d_{\hat{k}}(\vec{x}_a)$ is the maximum of all decision functions *and* this value is greater than the value of the decision of rejection $d_0(\vec{x}_a)$. This means that for $\vec{x}_a$ being of class $\hat{k}$ it has to hold

$$d_{k^*}(\vec{x}_a) \;=\; \max_{k=1,\,2,\,3,\,\dots} d_k(\vec{x}_a) \qquad \wedge \qquad d_{k^*}(\vec{x}_a) \;>\; d_0(\vec{x}_a). \tag{A.27}$$

The way to determine an action $(\dot{v}_k,\, \Delta\delta_k)$ for a given situation $\vec{x}_a$ is depicted in figure A.3. First, a unit for situation analysis has to describe the actual situation by means of the attribute vector $\vec{x}_a$. Second, for each class k the decision function $d_k(\vec{x}_a)$ is determined (equations (A.25) and (A.26)). Finally, the appropriate class k^* is chosen (equation (A.27)) and associated with the correspondent action $(\dot{v}_{k^*},\, \Delta\delta_{k^*})$ that is set as control input for the vehicle (equation (A.20)). For each class the covariance matrices $\underline{C}_k$ and the vectors

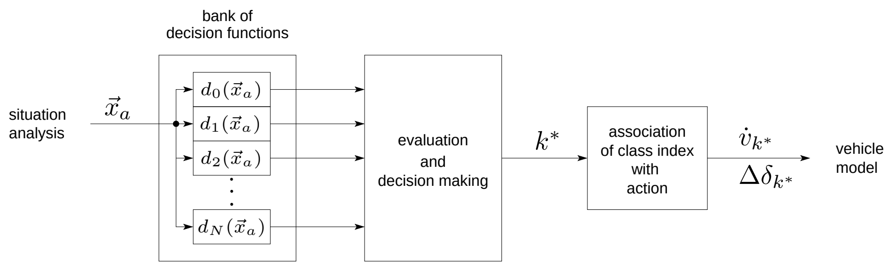


Figure A.3: Classification approach to model skill based driving behavior. Following [Grashey, 1999, figure 3.1].

$\vec{m}_k$ in equation (A.24) have to be determined during the learning process. This process would be extremely expensive with respect to calculation time if the dimensions of the attribute vector can not be reasonably reduced. To achieve this, Grashey uses the work of [Feraric, 1999] to divide the space of possible situations into subspaces related to *maneuvers*. For each maneuver the composition of the attribute vector $\vec{x}_a$ has to be determined and then the classificator (A.24) has to be parameterized. Aiming for stability the learning process has to provide enough different situations to the system. The amount of learned situations should cover the whole domain of possible situations. However presently, no system based on this approach exists that can guarantee stable driving with high precision under limited handling conditions.

It is important to note at this point that the term 'situation' in the context of classification approaches, is somehow identical to the 'initial state' in Generalized Feedback Control. The process described in figure A.3 is clearly not circumstance cognition. Instead, both the driver's motivation matching as well as circumstance cognition are captured together when sampling the input/output relationship. In this approach, circumstance cognition and motivation matching cannot be identified separately. The approach allows to imitate the input/output behavior of the driver. It does not allows, though, any greater insight into the process of dynamic planning or control.

In chapter 3 it is explained how to construct a unit for circumstance cognition, i.e. a plan-to-action mapper, based on empirical data. The classification approach tries to capture causal relationships considering the relationship between environment state and control parameters, targeting to mimick an exemplary operator. When constructing a unit for circumstance cognition one goes more into detail. Here causal relationships are to be discovered about the input/output relationship between the plant's input and the plant's output. Even not treated explicitly, the methodology using curve fitting introduced there is related the the subject of capturing causal relationships through conditional probabilities computed out of raw data [Pearl, 1999].

A.5 Neural Networks

Neural Networks provide an approach to mimick the information processing of 'natural' agents based on connectivity theory [Haykin, 1994]. In the style of the brain, a neural network consists of a number of neurons that are strongly interconnected. The output of each neuron is a weighted sum of its inputs coming from other neurons. It has been shown, that those networks can be trained to accomplish a large of tasks, such as pattern recognition [LeCun et al., 1995] or analog computation [Siegelmann and Sontag, 1994].

An approach in vehicle control is the system ALVINN[7], developed by Pomerleau [Pomerleau, 1993] at Carnegie Mellon University. Pomerleau's work focused on the application of Neural Networks for vision based systems in mobile robot control[8]. Visual information is used because of its ability to 'quickly provide a dense representation of the environment' [Pomerleau, 1993]. The direct mapping from visual information to control action fits the idea of unconscious reflex reaction, that a driver performs during vehicle control on the skill based layer.

Figure A.4 contains the structure of the architecture of the Neural Network: A multi-layered-perceptron. A two-dimensional input matrix represents the 'retina' of the visual system. Four hidden neurons are fully connected to all entries of that matrix. The output of this hidden layer is passed to the output layer consisting of 30 units performing motor commands from 'sharp left' to 'sharp right' that are applied on the front wheel angle.

[7] ALVINN is thought to be an abbreviation for 'Autonomous Land Vehicle In a Neural Network'.

[8] The significance of his work, however, exceeds vehicle control systems. It was used not only to drive two different kinds of cars but also to determine control commands for a walking robot [Ueno et al., 1990], designed for the Space Station Freedom.

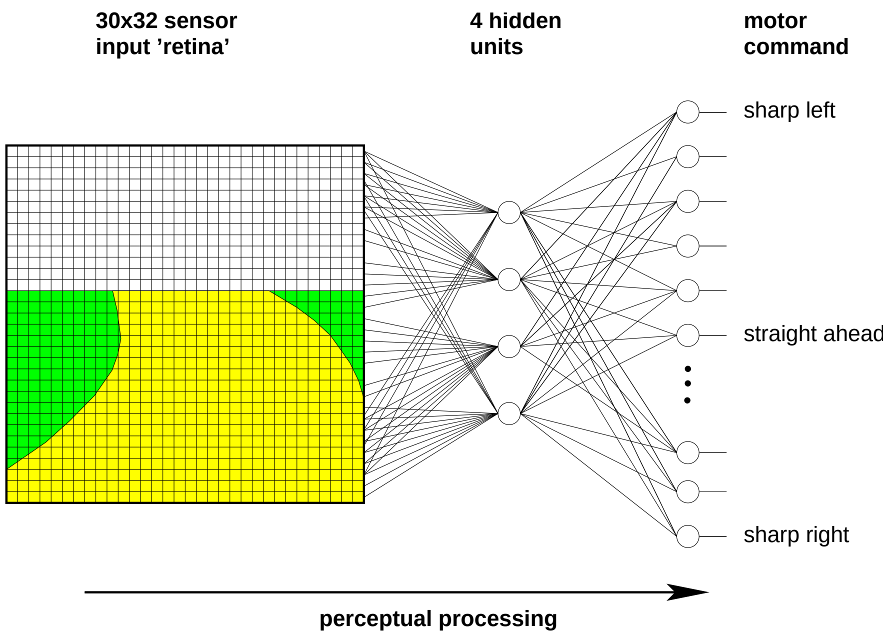


Figure A.4: Pomerleau's Neural Network System ALVINN for vehicle control. Following: [Pomerleau, 1993, figure 1.3].

To get an insight into the way environment information is mapped to steering commands the network is investigated in two ways:

1. *Weight analysis of the input retina*. The functioning of the network at this level is understood as a composition of filtering, feature detection, region finding and edge detection in the input matrix. In other words, the control process is understood in terms of *image processing*.

2. *Hidden unit sensitivity analysis*. It is based on a method adapted from neuro-science [Hubel, 1979]. Systematically, situations are identified that cause an individual neuron to respond. By this way situations can be identified, where the Neural Network is likely to fail.

Using characteristics of Neural Networks is a step towards the modeling of the intrinsic structure of the controller in a natural way. However, a precise analysis of the information processing inside the system in term of physical values is not possible. There is no way to determine tactics used to accomplish the control task. However, since the output of the network can be viewed as a weighted superposition of the neuron functions, the process of 'learning' is indeed a multi-dimensional curve fitting on the example data. The Neural Network acts like a template function for a curve fitting algorithm. Once the network has learned, it is an approximation to the inverse input/output function of the plant. This inverse is usually hard to determine in a closed mathematical form.

A.6 Fuzzy Logic

Fuzzy Logic describes information processing of a system by rulesets that are 'fuzzy' [Zadeh, 1965] avoiding the sharp edges of 'if-then' blocks as they appear in usual description languages for algorithms. In standard logic a statement is true or false but nothing in between. Fuzzy Logic allows to attribute to a statement a non-integer truth value. Respectively in standard (boolean) logic and element belongs to a set if it fulfills certain requirements or it does not. In Fuzzy Logic, an element can be assigned a non-integer membership value which is equivalent to the truth value of the statement that the element belongs to a certain set.

Multiple statements in Fuzzy Logic can be combined by logical operators working on the truth value of each single statement. A logical AND operation can, for example be expressed by a maximum operator, a logical OR can be expressed through a minimum operator[9].

When creating a control system, one applies expert knowledge expressed in linguistic terms rather than the differential equations of the system. It requires an operator that is able to explain the way he performs the control task in statements like

[9]Note, that such a choice is consistent with the logical AND and OR operators in case that the truth values a binary, i.e. 1 for true and 0 for false.

> `If` the front wheel angle is large positive `and` lateral acceleration is high `then` front wheel angle velocity is large negative.
>
> `If` velocity is high `and` road curvature is very small `then` propulsive force is high positive.
>
> `If` velocity is low `or` road curvature is medium `then` propulsive force is very low.
>
> ... and so on.

Different methods were developed to determine a quantitative value for the output variable given the truth value of the statement that triggers it. Approaches for driver modeling may be reviewed in [Holve et al., 1995, Kageyama and Pacejka, 1991, Protzel et al., 1993, Xi, 1993, Kramer and Rohr, 1982].

Generalized Feedback Control was inspired by this approach with respect to the ability to express the interference of certain control rules in linguistic terms. Combining this idea with utility functions leads to the concept of target and limit maps as explained in chapter 4.

A.7 Conclusion

This chapter discussed the historical background in front of which Generalized Feedback Control has evolved. Two types of control approaches were discussed: methods based on mathematical/physical formulations of the control problem and methods that attempt to mimick natural (human) operators.

The first category treated the control problem on a mathematical level. Nonlinear decoupling of the differential equations could avoid a linearization around an operating point and a separate treatment of sub-units, e. g. lateral and longitudinal control, such as in linear feedback control. However, it became clear that in the presence of error it is necessary to provide a *dynamic path following*, such as discussed in section A.3.

The second category consists of approaches that are rather inspired by natural agents. Using probability theory, the classification approach tries to mimick an exemplary operator based on conditional probabilities ignoring its internal tactics to perform the control task. The idea of using conditional probabilities to identify causal relationships led towards the approach of building a plan-to-action mapper based on empirical data, rather than based on physical formulae.

Neural Networks inspired the idea to find a inverse input/output function based on curve-fitting techniques. This is very practical since in the large majority of cases no closed mathematical solution for this problem can be found. The curve fitting of sample data is an essential element when building a plan-to-action mapper (see chapter 3). Fuzzy Logic inspired the use of continuous functions to model the interference of different rules during reasoning. In Generalized Feedback Control utility functions are used (see chapter 4) to model different motivations and operations are described how they have to be interfered to produce one single parameter set for nominal motions.

Not mentioned in the previous sections is the approach of model predictive control [Qin and Badgewell, 1997] as being used in industrial engineering. Similar to plan-to-action mapping the control process for one control interval is considered open-loop. In model predictive control, though, the plant state is predicted and used for control parameter computation [Allgöwer and Zheng, 2000]. Also, the terminology of containability in GFC as a measure for reliability displays fundamental differences in the understanding of the control process.

The Generalized Feedback Control method consist of motivation matching and circumstance cognition. These two concepts correspond somehow to a dynamic path following system dealing with short term path generation and short term path following. Generalized Feedback Control, though, separates these two processes conceptually very precisely. As a result, the control parameter calculation to achieve a nominal motion can be treated independently of the problem to determine nominal motions appropriate for specific design specification. It provides a terminology to quantify the control motivation of an agent and to quantify the knowledge an agent has about the input/output relationship of the plant. In the following chapter the circumstance cognition type of feedback is described as it is performed by the plan-to-action mapper. Chapter 5 describes the subject of motivation matching.

Appendix B

Vehicle Model

B.1 Bicycle Model

The text refers multiple times to a vehicle model known as the bicycle model. The general idea is to model the behavior of the two tires in front and the two tires in the rear, with one single tire in front and one single tire in the rear. Such a vehicle model with three degrees of freedom and rear wheel drive is depicted in figure B.1. The variables used are displayed in table B.1. The following paragraphs derive the differential equations that were used to simulate the vehicle used for investigations of this dissertation. Applying Newton's Second Law for rotational and translatory systems results in a set of equations for the vehicle as depicted in figure B.1:

$$\begin{aligned}
\sum F_x : \quad & m\,\dot{v}_x \;=\; F_p - sin(\delta)\,F_{s,f}(\alpha_f) - cos(\beta)\,F_r + sin(\beta)\,F_{lat}. \\
\sum F_y : \quad & m\,\dot{v}_y \;=\; F_{s,r}(\alpha_r) + cos(\delta)\,F_{s,f}(\alpha_f) - sin(\beta)\,F_r - cos(\beta)\,F_{lat}. \\
\sum M_z^{cg} : \quad & \ddot{\psi}\,I_z \;=\; cos(\delta)\,L_f\,F_{s,f}(\alpha_f) - L_r\,F_{s,r}(\alpha_r).
\end{aligned} \tag{B.1}$$

The acceleration in x- and y-direction calculates to

$$\dot{v}_x \;=\; \frac{\partial}{\partial t}\big(v\,cos(\beta)\big) \;=\; \dot{v}\,cos(\beta) - v\,sin(\beta)\,\dot{\beta}, \tag{B.2}$$

$$\dot{v}_y \;=\; \dot{v}\,sin(\beta) + v\,cos(\beta)\,\dot{\beta}. \tag{B.3}$$

Describing the curvature as

$$\kappa \;=\; \frac{\partial}{\partial s}(\beta + \psi) = \frac{d\,t}{d\,s}\,\frac{\partial}{\partial t}(\beta + \psi), \;=\; \frac{1}{v}\left(\dot{\beta} + \dot{\psi}\right), \tag{B.4}$$

the lateral acceleration force becomes

$$F_{lat} \;=\; m\,v^2\,\kappa \;=\; m\,v\,\left(\dot{\beta} + \dot{\psi}\right). \tag{B.5}$$

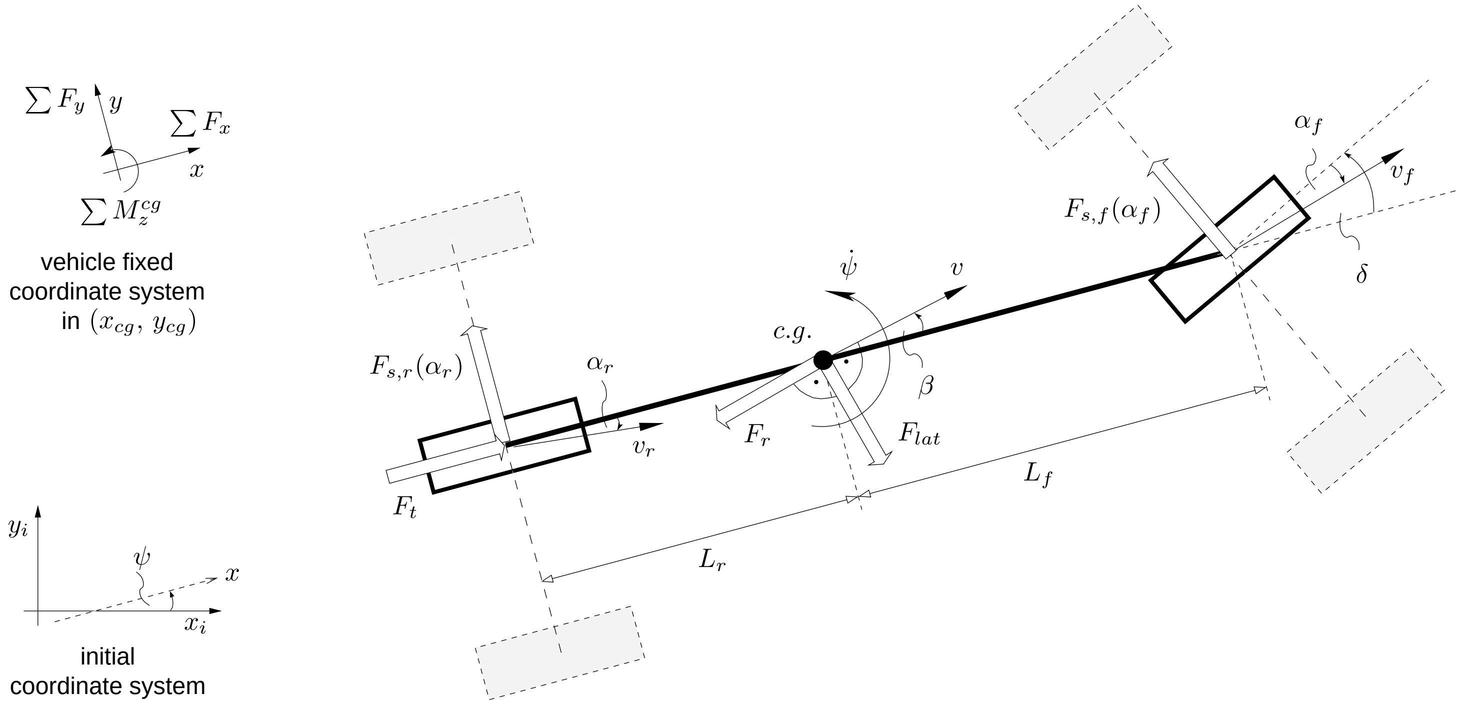


Figure B.1: Bicycle model with rear wheel drive reduced from a vehicle with four wheels.

Table B.1: Variables to describe the state of the bicycle model and their meaning.

Value	Meaning
$c.g.$	Center of Gravity. Coordinates are x_{cg} and y_{cg}.
δ	Front wheel angle.
β	Side slip angle in center of gravity (c.g.).
ψ	Yaw angle of vehicle with respect to the inertia system.
α_f, α_r	Side slip angle at front and rear tire.
F_p	Longitudinal propulsive force.
$F_{s,f}(\alpha_f)$, $F_{s,r}(\alpha_r)$	Side force at front and rear tire dependent on slip angles α_f and α_r.
F_r	Sum of resistance forces, e.g. air drag.
v	Velocity in center of gravity.
v_f, v_r	Velocity at center of front and rear tire.
F_{lat}	Lateral acceleration force, caused by current curvature $\kappa = (\dot{\beta} + \dot{\psi})/v$ and velocity.
L_f, L_r	Distance between front tire and c.g., respectively between c.g. and rear tire.
I_z	Moment of inertia, around z-axis.

Using (B.1), (B.2), (B.3) and applying (B.5) the vehicle's behavior can then be expressed in the state space equations

$$\begin{aligned}
\dot{x}_1 &= \frac{-sin(\delta - x_2)\, F_{s,f}(\alpha_f) - F_r + sin(x_2)\, F_{s,r}(\alpha_r) + cos(x_2)\, F_p}{m}, \\
\dot{x}_2 &= \frac{cos(\delta - x_2)\, F_{s,f}(\alpha_f) + cos(x_2)\, F_{s,r}(\alpha_r) - sin(x_2)\, F_p}{2\, m\, x_1} - \frac{x_3}{2}, \\
\dot{x}_3 &= \frac{cos(\delta)\, L_f\, F_{s,f}(\alpha_f) - L_r\, F_{s,r}(\alpha_r)}{I_z}.
\end{aligned} \tag{B.6}$$

where $x_1 = v$, $x_2 = \beta$ and $x_3 = \dot{\psi}$. The velocity at the front and the rear can be calculated using

$$\vec{v}_{f/r} = \omega \times \vec{r}_{f/r} + \vec{v}, \tag{B.7}$$

with $\omega = (0\ \ 0\ \ \dot{\psi})^T$. $\vec{v}_f$ and $\vec{v}_r$ are the velocity vectors in the front and the rear. $\vec{r}_f$ and $\vec{r}_r$ are the vectors from the center of gravity to center of the front and the rear axles. $\vec{v}$ indicates the velocity vector in the center of gravity. Then, the following equations can be derived for the front and rear side slip angle α_f and α_r:

$$tan(-\alpha_f) = \frac{-v\, cos(\beta)\, sin(\delta) + (v\, sin(\beta) + \dot{\psi}\, L_f) cos(\delta)}{v\, cos(\beta)\, cos(\delta) + (v\, sin(\beta) + \dot{\psi}\, L_f) sin(\delta)}, \tag{B.8}$$

$$tan(-\alpha_r) \;=\; \frac{v\,sin(\beta) - \dot{\psi}\,L_r}{v\,cos(\beta)}. \tag{B.9}$$

Pacejka [Bakker et al., 1987] suggests the so called 'magic formula' to describe the characteristics of a tire by the relation of the side force with respect to the slip angle. By applying his equation to the front and rear tire, one gets a formulation for front and rear side forces as

$$\begin{aligned} F_{s,f}(\alpha_f) &= D_f\,sin(C_f\,atan(B_f\,\alpha_f - E_f\,(B_f\,\alpha_f - atan(B_f\,\alpha_f)))), \\ F_{s,r}(\alpha_r) &= D_r\,sin(C_r\,atan(B_r\,\alpha_r - E_r\,(B_r\,\alpha_r - atan(B_r\,\alpha_r)))), \end{aligned} \tag{B.10}$$

where $D_{f/r}$, $C_{f/r}$, $B_{f/r}$ and $E_{f/r}$ have to be chosen to fit the characteristic behavior of the tires. This is the model that was used for the investigations made in the frame of this dissertation. The resulting forces at the front and rear tire are displayed in figure B.2a. Note, that the side force of the rear tire is always higher than the side force of the front tire. Consequently, the front tire has less resistance to lateral acceleration and slips, therefore, easier than the rear tire. This corresponds to a understeer-behavior, since for a fixed front wheel angle the curvature decreases with increasing lateral acceleration.

The equations in (B.6) describe a bicycle model as it was used for simulation runs in the frame of this dissertation. Other approaches usually use the *linearized bicycle model*. It is based on simplifications of the formulation in (B.6). It is assumed that the angles β and δ are very small, i.e. $|\beta|$, $|\delta| \ll 1$. Thus, it follows that

$$cos(Angle) = 1, \quad sin(Angle) = Angle, \tag{B.12}$$

$$Angle \cdot Force \ll OtherForces \;\Rightarrow\; Angle \cdot Force \approx 0. \tag{B.13}$$

Hence, a simplified set of equations which is often referred to as the *linear bicycle model* can be described as:

$$\dot{v} \;=\; \frac{1}{m}\,(F_p - F_r)\,, \tag{B.14}$$

$$\dot{\beta} \;=\; \frac{1}{2\,m\,v}\,(F_{s,r}(\alpha_r) + F_{s,f}(\alpha_f)) - \frac{\dot{\psi}}{2}, \tag{B.15}$$

$$\ddot{\psi} \;=\; \frac{1}{I_z}\,(L_f\,F_{s,f}(\alpha_f) - L_r\,F_{s,r}(\alpha_r))\,. \tag{B.16}$$

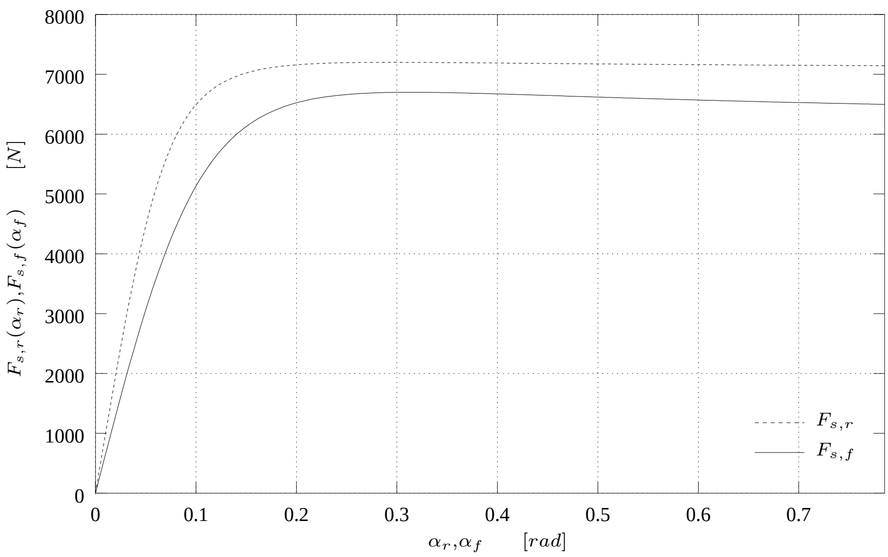
For small angles the tire slip angles from (B.8) and (B.9) can be approximated as

$$\alpha_f \;=\; -\beta - \frac{L_f}{v}\dot{\psi} + \delta \quad \text{and} \quad \alpha_r \;=\; -\beta + \frac{L_r}{v}\dot{\psi}. \tag{B.17}$$

A further linearization of the tire forces again simplifies the equations:

$$F_{s,f}(\alpha_f) \;=\; C_f^*\,\alpha_f \quad \text{and} \quad F_{s,r}(\alpha_r) \;=\; C_r^*\,\alpha_r, \tag{B.18}$$

a)



b)

$$\begin{array}{rclrcl} B_f &=& 7.7976\,\frac{1}{rad}, & B_r &=& 12.361\,\frac{1}{rad}, \\ C_f &=& 1.25\,\frac{1}{rad}, & C_r &=& 1.12\,\frac{1}{rad}, \\ D_f &=& 6700\ N, & D_r &=& 7200\ N, \\ E_f &=& -0.5\,\frac{1}{rad}, & E_r &=& -0.96\,\frac{1}{rad}. \end{array} \tag{B.11}$$

Figure B.2: The 'magic formula' for describing tire forces with respect to tire slip angle. a) Diagram of front and rear side forces. b) Coefficients of equations (B.10) to produce the depicted graph.

with C_f^* as the front tire stiffness and C_r^* as the tire stiffness in the rear. The resulting simplified state space equation is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{bmatrix} & 0 \\ -\frac{C_f^* + C_r^*}{2\,m\,x_1}\,x_2 & + \left(\frac{L_f\,C_f^* + L_r\,C_r^*}{2\,m\,x_1} - \frac{1}{2}\right)\,x_3 \\ -\frac{L_f\,C_f^* - L_r\,C_r^*}{I_z}\,x_2 & + \frac{L_f^2\,C_f^* - L_r^2\,C_r^*}{I_z}\,x_3 \end{bmatrix} + \begin{pmatrix} \frac{1}{m}\,(F_p - F_r) \\ \frac{C_f^*\,\delta}{2\,m\,x_1} \\ \frac{\delta}{I_z} \end{pmatrix}. \tag{B.19}$$

Note, that this equation is still not linear ($x_1 = v$ still appears in some denominators). It becomes linear, however, if the velocity can be assumed to be constant. Then the vehicle model would be reduced to a model of two degrees of freedom.

Appendix C

Control Impulses

For some applications of control it is necessary to model the frequency characteristic of control parameters[1]. This section discusses a method to influence the frequency characteristics of a control parameter by using impulses to model its derivative. This, however, only works under the following assumption:

> **Assumption**: It is assumed, that the system behavior does not significantly change if the *constant* time profile of the control parameter under investigation is replaced by a time profile that has the same *average* level over one control interval.

Conversely, this means that the derivative of this control parameter can be synthesized by a sequence of discrete impulses. In the following discourse the front wheel angle velocity is considered as an example and placeholder for any other possible control parameter. Let $\dot{\delta}_d$ be the desired front wheel angle velocity that has to appear at the output of the control unit over one control interval.

The impulses in the $\ddot{\delta}(t)$ time profile are generated using a general shaping function stretching it to a specific height. This way the average level of $\dot{\delta}(t)$ is equal to the constant term $\dot{\delta}_d$. In figure C.1, impulses in $\ddot{\delta}(t)$ of heights h_0, h_1, and h_2 are applied to reach for each control interval an average front wheel angles velocities $\dot{\delta}_0$, $\dot{\delta}_1$, and $\dot{\delta}_2$. Using impulse sequences in the derivative the time profile of the control parameter itself is, of course, not piecewise constant.

In the example of figure C.1b, the profile of $\dot{\delta}(t)$ of the segments of $\dot{\delta}_0$, $\dot{\delta}_1$, and $\dot{\delta}_2$ differs enormously from the profile of $\dot{\delta}(t)$ that results from integrating the impulses in $\ddot{\delta}(t)$. However, this might not have a significant influence on the total system behavior. The integration of $\dot{\delta}(t)$ results in profiles $\delta(t)$ as they are displayed in figure C.2. A bold line indicates the profile based on piecewise a constant $\dot{\delta}_d(t)$ the solid line indicates the profile based on the impulse chain in $\ddot{\delta}(t)$. Since the average of both profiles in the derivative must be identical, the profiles meet, after each control interval, at a common point. Although the front wheel angle velocities in the example of figure C.1b differ enormously, it can be seen that the resulting profiles of the front wheel angle are very similar.

[1]This is particulary true for appliciation in human driver modelling [Schuller et al., 1999].

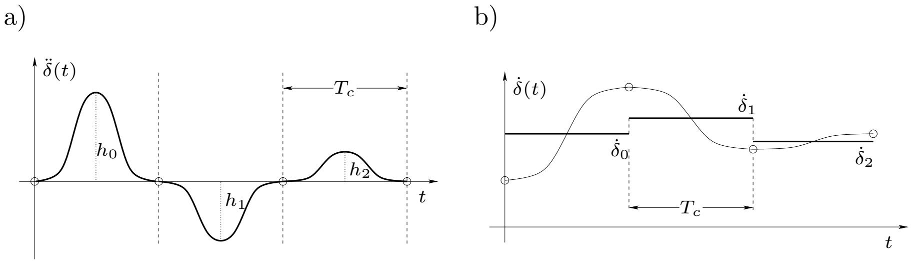


Figure C.1: Modeling control impulses in a control parameter's derivative: a) impulses in the derivative. b) resulting time profile of the control parameter itself.

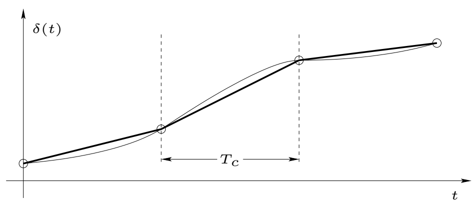


Figure C.2: Influence of impulse based synthesized control parameter and piece-wise constant control parameter on total system.

In the following discussion, impulses in the second derivative $\ddot{\delta}(t)$ is computed to achieve a desired average in front wheel angle velocity $\dot{\delta}(t)$. This is first done for the case those impulses do not overlap. Using the resulting equations, it is demonstrated that the spectrum of the front wheel angle velocity can be modeled by two parts: the spectrum of the control parameters sent by the control unit and the spectrum of the shaping function. To handle the general case, the use of overlapping control impulses to model $\ddot{\delta}(t)$ is demonstrated.

C.1 Non-Overlapping Control Impulses

The control unit determined a constant front wheel angle velocity $\dot{\delta}$ that must be affected during the next control interval. Impulse character in the second derivative is not directly compatible with this task. Instead, a formula is derived to model the impulses in such a way that a desired front wheel angle velocity $\dot{\delta}_d$ is performed *in average*.

The desired average $\langle\dot{\delta}(t)\rangle = \dot{\delta}_d$ is achieved by multiplying a generalized impulse shaping function in $\ddot{\delta}(t)$ with a factor $C_{\ddot{\delta}}$ that controls its 'height'. Then, a normalization on the shaping function is derived that allows one to change the shaping function without having to modify the mapping from the desired average $\dot{\delta}_d$ to the appropriate factor $C_{\ddot{\delta}}$.

For one control interval of length T_c, the second derivative $\ddot{\delta}(t)$ inside the actual control interval has a continuous shape that is described by a shaping function $f_s(\tau)$ multiplied by a control coefficient $C_{\ddot{\delta}}$

$$\ddot{\delta}(t) \quad = \quad C_{\ddot{\delta}}\, f_s(t/T_c), \qquad \text{where} \qquad f_s(\tau) = 0 \quad \forall \quad \tau \notin [0,\, 1], \tag{C.1}$$

with an $f_s(\tau)$ normalized to the interval $[0,\, 1]$ in order to be independent of the control interval size T_c. It is zero outside this interval to avoid overlapping between two control intervals. The constant factor $C_{\ddot{\delta}}$ allows one to model the height of the impulse. An example of a normalized impulse is shown in figure C.3a. The output signal consisting of a concatenation of such impulses is shown in figure C.3b. With definition (C.1), the velocity of the front wheel angle $\dot{\delta}(t)$

a) b)

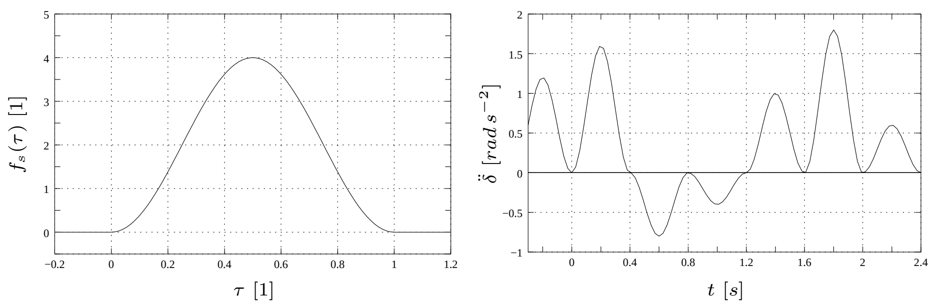


Figure C.3: Control impulses using the normalized lifted sine impulse $f_s(\tau) = 4\, sin^2(\pi\, \tau)$. a) Shaping function. b) Output signal.

becomes

$$\begin{aligned}
\dot{\delta}(t) \quad &= \quad \int_0^t \ddot{\delta}(\vartheta)\, d\vartheta + \dot{\delta}(0), && \text{(C.2)}\\
&= \quad \int_0^t C_{\ddot{\delta}}\, f_s(\vartheta/T_c)\, d\vartheta + \dot{\delta}(0) \;=\; C_{\ddot{\delta}}\, T_c \int_0^{t/T_c} f_s(\vartheta)\, d\vartheta + \dot{\delta}(0), && \text{(C.3)}\\
&= \quad C_{\ddot{\delta}}\, T_c \left(F_s(t/T_c) - F_s(0) \right) + \dot{\delta}(0), && \text{(C.4)}
\end{aligned}$$

with $F_s(\tau)$ denoting the antiderivative of $f_s(\tau)$. Since $f_s(\tau)$ is continuous and $f_s(t) = 0$ for all $t \notin [0, 1]$, it follows that $F_s(0) = 0$ and therefore

$$\dot{\delta}(t) \quad = \quad C_{\ddot{\delta}}\, T_c\, F_s(t/T_c) + \dot{\delta}(0). \tag{C.5}$$

By defining an arbitrary impulse character in $\ddot{\delta}(t)$, it is not admissible to assume a constant front wheel angle velocity $\dot{\delta}(t)$ during the next control interval. To approximate a constant $\dot{\delta}$, the continuous function $\dot{\delta}(t)$ must be designed in such a way that the average deviation between $\dot{\delta}(t)$ and the desired constant $\dot{\delta} = \dot{\delta}_d$ is minimized. To minimize the Euclidean distance[2] between $\dot{\delta}_d$ and $\dot{\delta}(t)$, the

[2]The Euclidean distance between two signals $g_1(t)$ and $g_2(t)$ calculates to the square root of $\int_{-\infty}^{\infty} |g_1(t) - g_2(t)|^2\, dt$.

average of $\dot{\delta}(t)$ has to be equal to the desired constant $\dot{\delta}_d$, thus

$$\dot{\delta}_d = \frac{1}{T_c}\int_0^{T_c} \dot{\delta}(t)\,dt = \frac{1}{T_c}\int_0^{T_c} \left(C_{\ddot{\delta}}\,T_c\,F_s(t/T_c) + \dot{\delta}(0)\right)\,dt, \tag{C.6}$$

$$= C_{\ddot{\delta}}\,T_c \int_0^1 F_s(\tau)\,d\tau + \dot{\delta}(0). \tag{C.7}$$

Therefore, the constant $C_{\ddot{\delta}}$ can be deliberately chosen so that the average of $\dot{\delta}(t)$ is as close as possible to the desired value $\dot{\delta}_d$.

$$C_{\ddot{\delta}} = \frac{\dot{\delta}_d - \dot{\delta}(0)}{T_c \int_0^1 F_s(\tau)\,d\tau}. \tag{C.8}$$

If the calculation of the control impulses can now be made independent of the shape of the function $f_s(\tau)$, then it is possible to change the shaping function without having to modify the way $C_{\ddot{\delta}}$ is calculated from $\dot{\delta}_d$. This results in making the integral over $F_s(\tau)$ from zero to one, independent of the shape of $f_s(\tau)$. It is now possible to define a module of non-overlapping control impulses as follows:

Definition: 24 (Module of Non-Overlapping Control Impulses) *Given a shaping function normalized to the interval* [0, 1] *with*

$$f_s(\tau) = 0 \quad \forall \quad \tau \notin [0,\,1] \tag{C.9}$$

and imposing on the antiderivative $F_s(\tau)$ of $f_s(\tau)$ that

$$\int_0^1 F_s(\tau)\,dt = \int_0^1 \int_0^{\vartheta} f_s(\tau)\,d\tau d\vartheta = 1, \tag{C.10}$$

then $f_s(\tau)$ defines an Module of Non-Overlapping Control Impulses. *A desired average of a time profile of a control parameter $\dot{\delta}$ can be achieved by defining its derivative $\ddot{\delta}$ as*

$$\ddot{\delta}(t) = C_{\ddot{\delta}}\,f_s(t/T_c), \tag{C.11}$$

where the coefficient $C_{\ddot{\delta}}$ is derived from the desired $\dot{\delta}_d$ by the formula

$$C_{\ddot{\delta}} = \frac{1}{T_c}\left(\dot{\delta}_d - \dot{\delta}(0)\right). \tag{C.12}$$

The freedom of modeling the shape of $f_s(\tau)$ enables one to modify the frequency spectrum of the front wheel angle. This can be done without any effect on the performance of the controller since the desired average front wheel angle velocity is always achieved. Describing the whole signal $\ddot{\delta}(t)$ as a concatenation of impulses $f_s(\tau)$ weighted by control stimuli $C_{\ddot{\delta}}(n)$ one gets

$$\ddot{\delta}(t) \equiv \sum_{n=-\infty}^{\infty} C_{\ddot{\delta}}(n)\,f_s\left(\frac{t - n\,T_c}{T_c}\right). \tag{C.13}$$

The Fourier Transform becomes

$$\tilde{\ddot{\delta}}(i\omega) = T_c \sum_{n=-\infty}^{\infty} \mathcal{F}\left\{C_{\ddot{\delta}}(n)\, f_s\left(\frac{t-n\,T_c}{T_c}\right)\right\}(i\omega), \tag{C.14}$$

$$= T_c\,\mathcal{F}\{f_s(\tau)\}(i\,\omega\,T_c) \sum_{n=-\infty}^{\infty} C_{\ddot{\delta}}(n)\, e^{-i\omega\, n\, T_c}, \tag{C.15}$$

$$= T_c\,\mathcal{F}\{f_s(\tau)\}(i\,\omega\,T_c)\,\mathcal{F}_D\left\{C_{\ddot{\delta}}(n)\right\}(i\,\omega\,T_c), \tag{C.16}$$

$$= T_c\,\tilde{F}_s(i\,\omega\,T_c)\;\tilde{C}_{\ddot{\delta}}(i\,\omega\,T_c). \tag{C.17}$$

The Time Discrete Fourier Transform $\tilde{C}_{\ddot{\delta}}(i\,\omega)$ produces a spectrum that is periodic and infinite. Its period is $\omega_T = \frac{2\pi}{T_c}$. The fact that the terms $\tilde{F}_s(i\,\omega\,T_c)$ and $\tilde{C}_{\ddot{\delta}}(i\,\omega\,T_c)$ appear independently facilitates the spectral forming of the front wheel angle. The spectrum consists of two multiplicative terms:

- The spectrum $\tilde{C}_{\ddot{\delta}}(i\,\omega\,T_c)$ of the control impulses that are sent to the control unit every time step T_c. In the ideal case of a perfect plan-to-action mapper, they are distinctly defined by the course geometry and the driving task.
- The spectrum $\tilde{F}_s(i\,\omega\,T_c)$ of the shape of control impulses which can be determined arbitrarily.

This creates the opportunity to model the spectrum of $\dot{\delta}(t)$ in such a way that special types of effectors may be modeled. If the approximation of $\dot{\delta}(t)$ is close enough to the constant front wheel angle velocity $\dot{\delta}_d$, then the control unit does not perceive any difference between different shapes of control impulses. Furthermore, the spectrum $\tilde{C}_{\ddot{\delta}}(i\,\omega\,T_c)$ is, in fact, distinctly defined by the course that has to be driven. Under the previous assumptions, it can be concluded that the spectrum of $\ddot{\delta}(t)$ can be modeled arbitrarily. As shown in section C.3, the current implementation does not provide an independence of the computations of the plan-to-action mapper from the control impulse shape. Nevertheless, this way of modeling represents a first step towards implementing spectral characteristics that are associated with different classes of actuators.

C.2 Overlapping Control Impulses

Having defined how to model non-overlapping control impulses, we are ready to investigate a generalization of that concept. The following paragraphs treat the case where the control impulses overlap. This appears, for example, in cases where the actuators are expected to show an integrating behavior. An example of a control signal consisting of superposed overlapping impulses may be considered in figure C.4b.

In this case it is not possible to provide a convenient normalization for the shaping function as it was achieved in definition 24. Therefore, the control constants $C_{\ddot{\delta}}(n)$ can no longer be chosen independently from the shaping function

$f_s(\tau)$. It is shown how the factor $C_{\ddot{\delta}}(0)$ for the current control impulse can then be derived using the previous factors $\{C_{\ddot{\delta}}(1), C_{\ddot{\delta}}(2), \ldots\}$ and the set of coefficients $\{\zeta_1, \zeta_2, \ldots\}$ that sufficiently describes the shaping function.

The shaping function is normalized in time to the interval [0, 1]. To achieve a certain length T_s, the shaping function is stretched. The impulses appear in distances of T_c at times $-k\,T_c$. With these constraints, the influence of an impulse set at time $-k\,T_c$ on the total second derivative of the front wheel angle becomes (analogous to equation (C.1))

$$\ddot{\delta}_k(t) \quad \equiv \quad C_{\ddot{\delta}}(k)\, f_s\left(\frac{t + k\,T_c}{T_s}\right), \tag{C.18}$$

with

$$f_s(\tau) = 0 \quad \forall \quad \tau \notin [0,\, 1]. \tag{C.19}$$

The superposition of all those impulses results in the signal of the second derivative of the front wheel angle

$$\ddot{\delta}(t) \quad \equiv \quad \sum_{k \geq 0} \ddot{\delta}_k(t). \tag{C.20}$$

Here again, $t = 0$ represents the actual moment. The restriction of $k \geq 0$ is deliberately chosen. It means, that no impulses can be set in the future. Impulses are only set in the past, i.e. at times $-k\,T_c < 0$, and at the actual moment $t = 0$. The following paragraphs explain a formula describing the influence of the control impulses on the average of the front wheel angle velocity $\langle\dot{\delta}(t)\rangle_{[0,\,T_c]}$. Finally, an expression is found that allows one to set the height of the actual control impulse to achieve a desired average front wheel angle velocity $\dot{\delta}_d$. The front wheel angle velocity $\dot{\delta}(t)$ as the integral of $\ddot{\delta}(t)$ results in

$$\dot{\delta}(t) \;=\; \int_{-\infty}^{t} \sum_{k \geq 0} \ddot{\delta}_k(\tau) d\tau \quad = \quad \sum_{k \geq 0} \int_{-\infty}^{t} \ddot{\delta}_k(\tau) d\tau. \tag{C.21}$$

This expression can be split up as follows

$$\dot{\delta}(t) \quad = \quad \sum_{k \geq \frac{T_s}{T_c}} \int_{-\infty}^{t} \ddot{\delta}_k(\tau) d\tau \tag{C.22}$$

$$+ \sum_{\frac{T_s}{T_c} > k \geq 0} \left(\int_{-\infty}^{0} \ddot{\delta}_k(\tau) d\tau + \int_{0}^{t} \ddot{\delta}_k(\tau) d\tau \right). \tag{C.23}$$

$\ddot{\delta}_k(t)$ is only non-zero in case that $t \in [-k\,T_c,\, T_s - k\,T_c]$. This can be used to cut out a part of the domain of integration. It follows

$$\dot{\delta}(t) \quad = \quad \underbrace{\sum_{k \geq \frac{T_s}{T_c}} \int_{-k\,T_c}^{T_s - k\,T_c} \ddot{\delta}_k(\tau) d\tau \; + \sum_{\frac{T_s}{T_c} > k \geq 0} \int_{-k\,T_c}^{0} \ddot{\delta}_k(\tau) d\tau}_{=const.} \tag{C.24}$$

$$+ \sum_{\frac{T_s}{T_c} > k \geq 0} \int_0^t \ddot{\delta}_k(\tau) d\tau. \tag{C.25}$$

Regarding the case where $t = 0$ revolves that the constant term is equal to $\dot{\delta}(0)$ so that $\dot{\delta}(t)$ can be expressed as

$$\dot{\delta}(t) = \dot{\delta}(0) + \sum_{\frac{T_s}{T_c} > k \geq 0} \int_0^t \ddot{\delta}_k(\tau) d\tau. \tag{C.26}$$

This way of writing $\dot{\delta}(t)$ carries a great advantage. It allows one to consider the influence of all previous control impulses $k > T_s/T_c$ by one single constant term $\dot{\delta}(0)$. Otherwise, all previous impulses would have to be stored somewhere. In order to compute the average front wheel angle velocity, it is preferable to calculate the front wheel angle.

$$\delta(t) = \int_{-\infty}^t \dot{\delta}(0) + \sum_{\frac{T_s}{T_c} \geq k \geq 0} \int_0^\vartheta \ddot{\delta}_k(\tau) d\tau \; d\vartheta, \tag{C.27}$$

$$= \delta(0) + \dot{\delta}(0)\, t + \sum_{\frac{T_s}{T_c} \geq k \geq 0} \int_0^t \int_0^\vartheta \ddot{\delta}_k(\tau)\, d\tau\, d\vartheta, \tag{C.28}$$

where the integral evaluates to

$$\int_0^t \int_0^\vartheta \ddot{\delta}_k(\tau)\, d\tau\, d\vartheta = C_{\ddot{\delta}}(k)\, T_s \int_0^t \int_{\frac{k\,T_c}{T_s}}^{\frac{\vartheta + k\,T_c}{T_s}} f_s(u)\, du\, d\vartheta, \tag{C.29}$$

$$= C_{\ddot{\delta}}(k)\, T_s^2 \int_{\frac{k\,T_c}{T_s}}^{\frac{t+k\,T_c}{T_s}} F_s(v) - F_s\left(\frac{k\,T_c}{T_s}\right) dv, \tag{C.30}$$

$$= C_{\ddot{\delta}}(k)\, T_s^2 \left[F_s^{[2]}\left(\frac{t + k\,T_c}{T_s}\right) - F_s^{[2]}\left(\frac{k\,T_c}{T_s}\right) - t\, F_s\left(\frac{k\,T_c}{T_s}\right)\right]. \tag{C.31}$$

The aim is to set the intensity of the impulses in the second derivative so that the front wheel angle velocity over the next control interval is in average equal to the desired $\dot{\delta}_d$. The average of $\dot{\delta}(t)$ can be expressed as

$$\langle \dot{\delta}(t) \rangle_{[0,\,T_c]} = \frac{\delta(T_c) - \delta(0)}{T_c}, \tag{C.32}$$

$$= \dot{\delta}(0) + \sum_{\frac{T_s}{T_c} > k \geq 0} C_{\ddot{\delta}}(k)\, \frac{T_s^2}{T_c} \left[F_s^{[2]}\left(\frac{(1+k)\,T_c}{T_s}\right) - F_s^{[2]}\left(\frac{k\,T_c}{T_s}\right) - T_c\, F_s\left(\frac{k\,T_c}{T_s}\right)\right]. \tag{C.33}$$

It is useful to define a shorthand

$$\zeta_k \equiv \frac{T_s^2}{T_c}\left[F_s^{[2]}\left(\frac{(1+k)\,T_c}{T_s}\right) - F_s^{[2]}\left(\frac{k\,T_c}{T_s}\right) - \frac{T_c}{T_s}\,F_s\left(\frac{k\,T_c}{T_s}\right)\right]. \tag{C.34}$$

Thus, the factor $C_{\ddot\delta}(0)$ of the actual impulse can be determined. For a desired average front wheel angle velocity $\langle\dot\delta(t)\rangle_{[0,\,T_c]} = \dot\delta_d$ this factor becomes by equation (C.34) and (C.33)

$$C_{\ddot\delta}(0) = \frac{1}{\zeta_0}\left[\dot\delta_d - \dot\delta(0) - \underbrace{\sum_{\frac{T_s}{T_c}>k\geq 1} C_{\ddot\delta}(k)\,\zeta_k}_{X}\right]. \tag{C.35}$$

Term X is the accumulated influence from previous control impulses. Since the impulses last only for a period of T_s it is clear that only control impulses hang over where $k < T_s/T_c$

This is reminiscent of equation (C.8). In the case that $T_s = T_c$, the results are identical. In this case, however, it is not possible to introduce a convenient normalization without restricting the shape of the function. It suffices, however, to provide a set of constants $\{\zeta_k\}_{k=0,1\ldots K}$ as in the following definition of an module of overlapping control impulses

Definition: 25 (Module of Overlapping Control Impulses) *Given a shaping function $f_s(\tau)$ normalized to the interval* $[0,\,1]$ *with*

$$f_s(\tau) = 0 \quad \forall \quad \tau \notin [0,\,1]. \tag{C.36}$$

and the corresponding constants

$$\zeta_k \equiv \frac{T_s^2}{T_c}\left[F_s^{[2]}\left(\frac{(1+k)\,T_c}{T_s}\right) - F_s^{[2]}\left(\frac{k\,T_c}{T_s}\right) - \frac{T_c}{T_s}\,F_s\left(\frac{k\,T_c}{T_s}\right)\right], \tag{C.37}$$

then $f_s(\tau)$ defines an MODULE OF OVERLAPPING CONTROL IMPULSES. *A desired average of a control parameter $\dot\delta$ can be achieved by defining its derivative $\ddot\delta$ as*

$$\sum_{k=0}^{K} C_{\ddot\delta}(k) f_s\left(\frac{t+T_c\,k}{T_s}\right), \tag{C.38}$$

where the coefficient $C_{\ddot\delta}(0)$ to achieve the desired $\dot\delta$ is calculated based on the previous control coefficients $\{C_{\ddot\delta}(1),\, C_{\ddot\delta}(2), \ldots\}$ by

$$C_{\ddot\delta}(0) = \frac{1}{\zeta_0}\left(\dot\delta_d - \dot\delta(0) - \sum_{k=1}^{N} C_{\ddot\delta}(k)\,\zeta_k\right). \tag{C.39}$$

Example

$$f_s(\tau) = \begin{cases} \tau\, e^{-9\,\tau} & \forall \quad \tau \in [0,\, 1] \\ 0 & else \end{cases} \quad . \tag{C.40}$$

This shaping function decays fast, is approximately zero beyond $t = 1$ and has a Fourier transform that is easy to calculate.

$$\mathcal{F}\{f_s(t/T_s)\} \quad = \quad \frac{T_s}{(9 + i\,\omega\, T_s)^2}. \tag{C.41}$$

The higher frequencies are therefore damped with the term ω^{-2}, so it behaves like a 'strong' low pass filter.

The maximum of $f_s(\tau)$ happens to appear approximately in the middle of the first quarter of [0, 1], so let $T_s = 4\,T_c$. Thus, an overlapping occurs with the last three control impulses. Figure C.4 shows how a set of preceeding impulses are superimposed to the actual front wheel angle acceleration $\ddot{\delta}(t)$ with an interval size $T_c = 0.4\,s$. The dashed line represents the profile that would appear, if the actual added signal is zero. Obviously, the influence of impulses that lie more than $3\,T_c$ backwards is absolutely negligible. Using equation (C.37), the ζ_k's are calculated as

$$\zeta_k \quad = \quad T_c \left(\frac{4281}{159898}\, k + \frac{2382}{388747} \right) e^{-\frac{9}{4}\,k}. \tag{C.42}$$

Note that these coefficients are constant as long as the basic shape of the control impulses stays the same. To calculate the factor $C_{\ddot{\delta}}(0)$ that is required to achieve a desired average $\dot{\delta}_d$ the only information that is needed about the shape of the impulses are the four fixed coefficients ζ_0, ζ_1, ζ_2 and ζ_3.

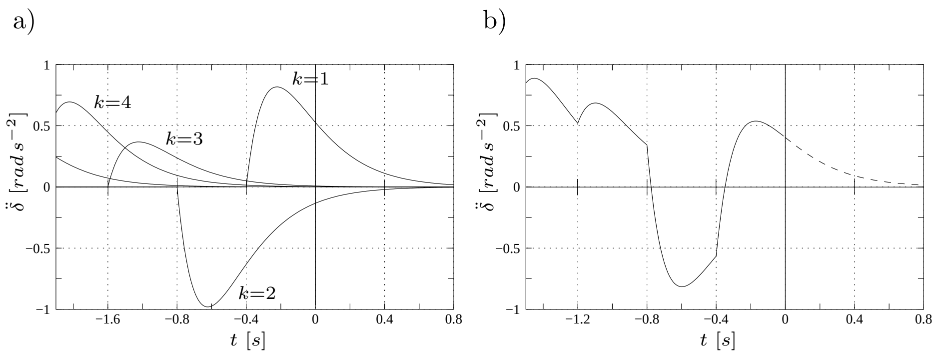


Figure C.4: Overlapping control impulses. a) Parts of impulses initiated in previous control intervals. b) Superimposed signal.

C.3 Spectra

This section describes how the shapes of control impulses influence the spectrum of the control signal. Equation (C.17) showed that the spectrum of the second derivative of the front wheel angle is the product of the spectrum of the control impulses and the spectrum of the shaping function. Signal theory indicates that a time window of Δt only allows one to make precise conclusions about frequency parts in the spectrum that lie beyond $1/\Delta t$ [Gabor, 1946]. This means that the shape of the control impulses primarily influences the frequency parts that are greater than $1/T_c$. For frequencies less than $1/T_c$, the spectrum of the control impulses dominates.

The following experiments were done with control interval sizes of 0.4 seconds. To show the potential of spectral forming, three different shaping functions are used. First, consider the smooth 'lifted sine'-function

$$f_{rc}(\tau) \quad \equiv \quad 4\, sin^2(\pi\, \tau). \tag{C.43}$$

The results of a simulation with $T_c = 0.4\, s$ are shown in figure C.5. Typically, the frequency components until $1/T_c$ are relatively constant and decay quickly. An extremely rough shaping function is the 'descending sawtooth'-function

$$f_{ds}(\tau) \quad \equiv \quad 3 - 3\, \tau. \tag{C.44}$$

Figure C.6 illustrates the results of a simulation using this impulse shape. The signal energy decays much slower with respect to the frequency than in case of the lifted sine impulse. A 'quadratic'-function may be defined as

$$f_q(\tau) \quad \equiv \quad -12\, \tau^2 + 12\, \tau. \tag{C.45}$$

A simulation result is depicted in figure C.7. The signal energy does not decay as quickly as with the lifted sine function, but much faster than the descending sawtooth function.

Equation (C.8) defines the height of the control impulses, expressed by the coefficient $C_{\ddot{\delta}}$, as being proportional to the desired average front wheel angle velocity $\langle\dot{\delta}\rangle$. Regarding the time signals in figure C.5a, figure C.6a and figure C.7 it becomes obvious that the heights, and therefore the desired $\langle\dot{\delta}\rangle$s, differ dependent on the shape of the control impulses. Since the control unit calculates the $\langle\dot{\delta}\rangle$ in a deterministic way, it can be concluded that the environment situations that cause the $\langle\dot{\delta}\rangle$s are not the same. Although the desired average front wheel angle velocity is always achieved, the vehicle behavior changes depending upon the profile of the impulses in $\ddot{\delta}(t)$. In other words, the requirement for performing an average front wheel angle velocity $\langle\dot{\delta}\rangle$ is not enough to precisely determine the vehicle's trajectory. The ideal situation of a complete independence between control parameters and the shape of the control impulses cannot be achieved.

a) b)

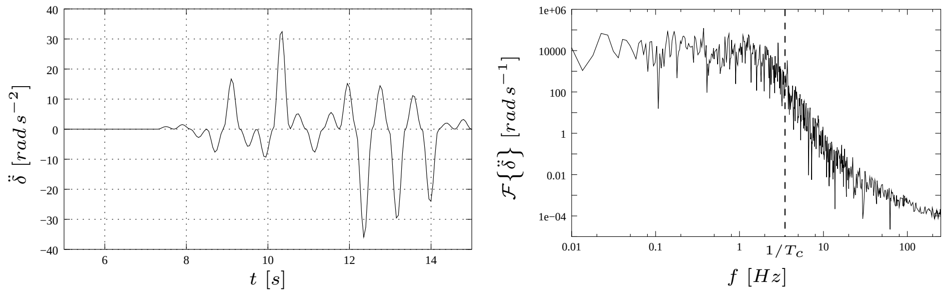


Figure C.5: Lifted sine impulses. a) 10 seconds of the time signal. b) Fourier transformation over the whole time signal of 200 seconds.

a) b)

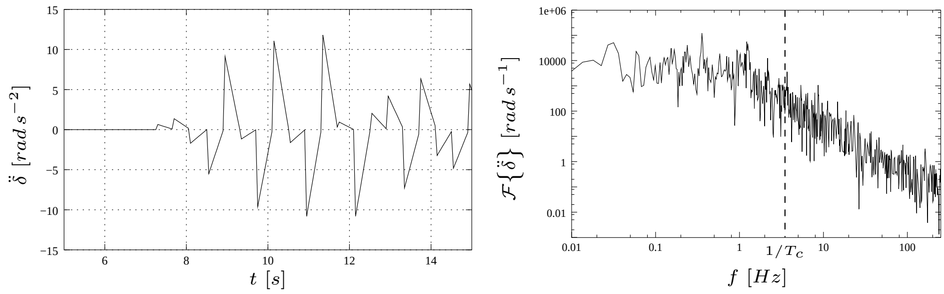


Figure C.6: Descending Sawtooth impulses. a) 10 seconds of the time signal. b) Fourier transformation over the whole time signal of 200 seconds.

a) b)

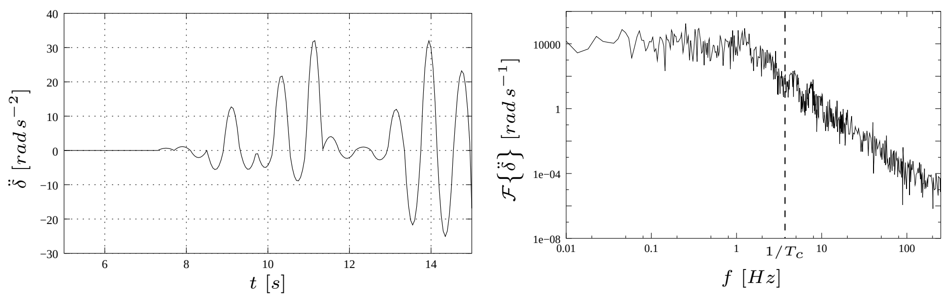


Figure C.7: Quadratic impulses. a) 10 seconds of the time signal. b) Fourier transformation over the whole time signal of 200 seconds.

C.4 Conclusion

In order model frequency domain characteristics of a control parameter $\dot{\delta}$ the profile of its derivative $\ddot{\delta}$ was described as a chain of impulses. The basic assumption, hereby, was that the system behavior does not significantly change when the constant time profile of a control parameter is replaced by a time profile of the same average level. In a first step, impulses were chosen that do not overlap. Based on a normalized shape of impulses, a formula was derived that allows calculation of an amplification factor corresponding to the desired constant front wheel angle velocity $\dot{\delta}$, as it was specified by the plan-to-action mapper.

In the second step, a formula was derived for impulses that overlap. This formula was more general than the previous one. In fact, in the case that the impulse length was chosen to be equal to the control interval length, the formula became the same as the one for non-overlapping impulses.

Some simulation experiments showed how the spectrum of the control values can be influenced by the choice of control impulse shapes. The impulse shapes basically allowed one to model the frequency parts higher than $1/T_c$. The frequency parts lower than $1/T_c$ were basically determined by the geometry of the nominal course and the velocity at which it was driven.

Appendix D

Mathematics of Second Order Nestle Curves

Section 5.2.3 elaborated on nestle curves of second order. This chapter explains the mathematical background and shows how the solution mentioned in equation (5.24) was derived. Remember, that the considerations are made with respect to a coordinate system, which has the origin in the vehicle's c.g. and the x-axes along the velocity vector. The constraints for a second order nestle curve $\vec{c}(q)$ parameterized by q are the following:

1. Starting point $\vec{c}(q = 0) = \vec{0}$.
2. End point $\vec{c}(q = 1) = (t_x \, t_y)^T$.
3. Angle at the origin $(q = 0)$ is zero, i.e. $\alpha_c(0) = 0$. That means, that the tangent at $q = 0$ is parallel to the x-axes.
4. Angle at the end of the curve $(q = 1)$ is fixed, i.e. $\alpha_c(1) = \alpha_t$. This specifies the tangent at the end of the nestle curve.
5. Curvature of the nestle curve at $q = 0$ is equal the specified κ_0, i.e. $\kappa(\vec{c}(0)) = \kappa_0$.
6. The curve has to be smooth. The curve must have a reasonable shape in all possible circumstances. Especially, loops shall never occur.

To fulfill these constraints the following idea was developed: The basis is a straight line from the origin $\vec{0}$ to the target point $(t_x \, t_y)^T$. This line has then to be warped in a way that the lasting angular and curvature constraints are fulfilled. A first rotational field warps the line around the origin and results in a curve $\vec{c}_0(q)$. A second field rotates the line around the target point and the result is the curve $\vec{c}_1(q)$. Each field is supposed to influence the final curve in a way so that the constraints are fulfilled that concern the correspondent center of rotation.

Finally, both fields are superposed and the nestle curve is the line that results from averaging the points $\vec{c}_0(q)$ and $\vec{c}_1(q)$ for each q. The end points at $q = 0$ and $q = 1$ have to be fixed points for both rotational fields. By this way, the conditions 1) and 2) on the starting point and end point are automatically fulfilled. The first rotational field warps the initial line from $(0\,0)^T$ to $(t_x\, t_y)^T$ around the origin by rotating each vector $(q\, t_x \;\, q\, t_y)^T$

$$\vec{c}_0(q) \;\; \equiv \;\; \underline{\varrho}(\lambda_0(q)) \begin{pmatrix} q\, t_x \\ q\, t_y \end{pmatrix}, \tag{D.1}$$

with $\underline{\varrho}(\gamma)$ as a matrix that rotates a point with an angle of γ around the origin. An example is to be considered in figure D.1a. The second rotational field warps the line around the target point; mainly to achieve the target angle α_t.

$$\vec{c}_1(q) \;\; \equiv \;\; \underline{\varrho}(\lambda_1(q)) \left[\begin{pmatrix} q\, t_x \\ q\, t_y \end{pmatrix} - \begin{pmatrix} t_x \\ t_y \end{pmatrix} \right] + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \tag{D.2}$$

This field should be mainly active close to the target point and its rotational influence for $q = 0$ should vanish as depicted in the example of figure D.1b. The superposition is supposed to give a curve that is able to satisfy all constraints. This superposition is achieved by averaging the vectors $\vec{c}_0(q)$ and $\vec{c}_0(q)$ for all $q \in [0,\, 1]$. Thus,

$$\vec{c}(q) \;\; \equiv \;\; \begin{pmatrix} x_c(q) \\ y_c(q) \end{pmatrix} = \frac{1}{2}\Big(\vec{c}_0(q) + \vec{c}_1(q)\Big), \tag{D.3}$$

$$= \;\; \frac{1}{2}\left[\underline{\varrho}(\lambda_0(q)) \begin{pmatrix} q\, t_x \\ q\, t_y \end{pmatrix} + \underline{\varrho}(\lambda_1(q)) \begin{pmatrix} (q-1)\, t_x \\ (q-1)\, t_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}\right] \tag{D.4}$$

The effect of such a superposition may be considered in figure D.2. It depicts the superposition of the curves from figure D.1a and figure D.1b. From geometric considerations the following simplification can be derived:

$$\vec{c}(q) \;\; = \;\; \frac{1}{2} \begin{pmatrix} R\, q\, \cos(\xi_0(q)) - R\,(1-q)\cos(\xi_1(q)) + t_x \\ R\, q\, \sin(\xi_0(q)) - R\,(1-q)\sin(\xi_1(q)) + t_y \end{pmatrix}, \tag{D.5}$$

with

$$\begin{aligned} &\xi_0(q) \;\equiv\; A + \lambda_0(q), \quad \xi_1(q) \;\equiv\; A + \lambda_1(q), \\ &A = arctan(t_y, t_x) \quad \text{ and } \quad R = \sqrt{t_x^2 + t_y^2}. \end{aligned} \tag{D.6}$$

The precise form of the two functions $\xi_0(q)$ and $\xi_1(q)$ that indicate the angle of rotation for a given q are left open. Later this allows one to simplify equations by adding constraints on these functions. Finally, it carries the advantage to fulfill condition 6, the smoothness issue, a posteriori - when the solution for all other constraints is found.

The fixed points of the two rotational fields introduce a border condition for the rotational angles $\xi_0(q)$ and $\xi_1(q)$.

$$\lambda_0(1) = 0 \quad \Rightarrow \quad \xi_0(1) = A, \tag{D.7}$$

$$\lambda_1(0) = 0 \quad \Rightarrow \quad \xi_1(0) = A. \tag{D.8}$$

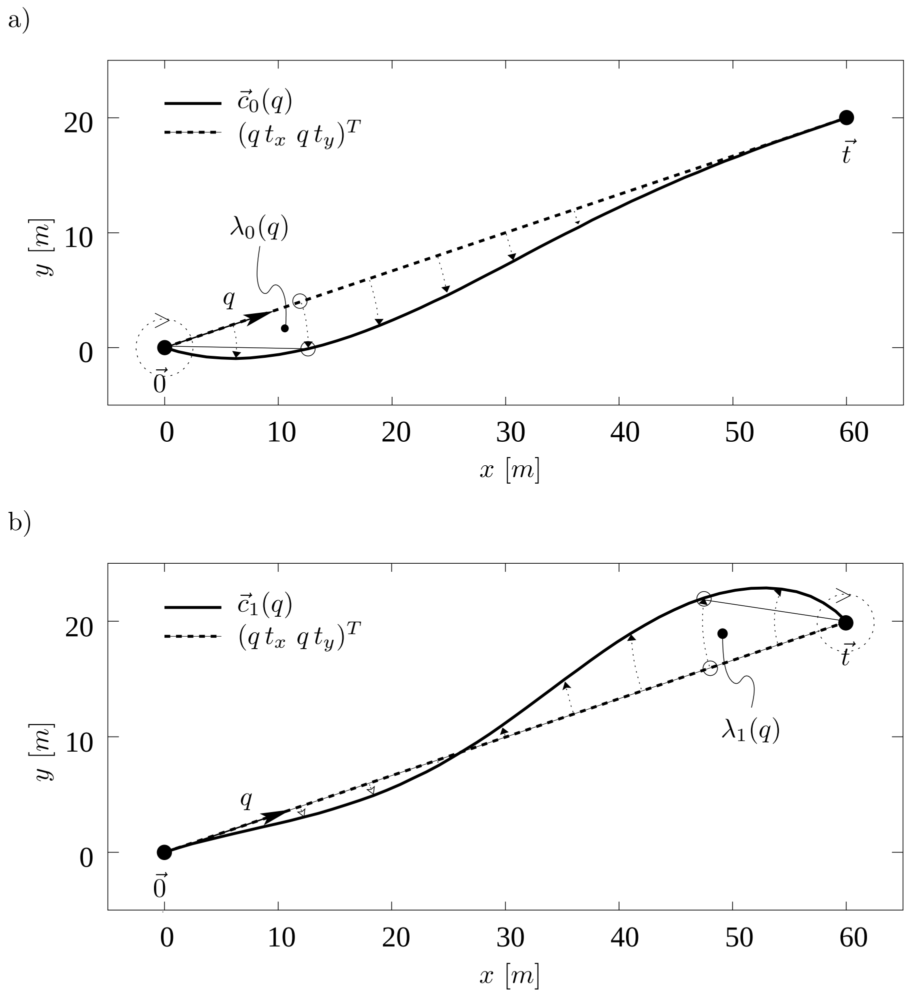


Figure D.1: Two rotational fields that warp the direct connection from the origin to the target point. They produce formulae that can be easily parameterized to fulfill angular and curvature constraints.

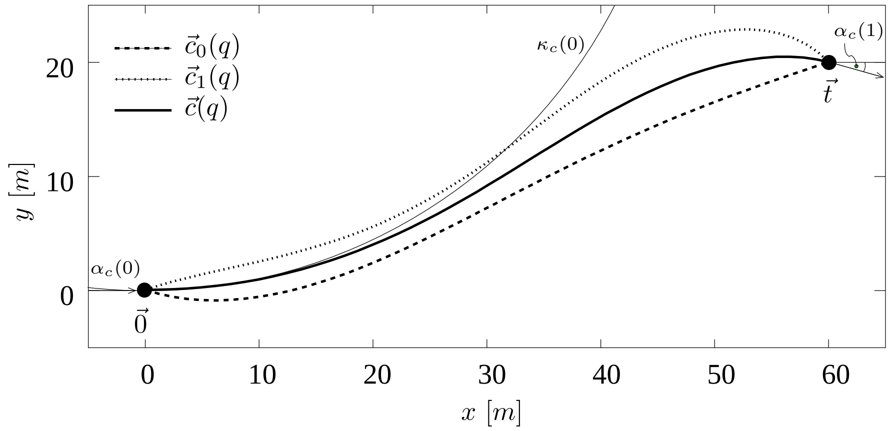


Figure D.2: Superposition of two warped curves $\vec{c}_0(q)$ and $\vec{c}_1(q)$ to one single nestle curve $\vec{c}(q)$ that obeys all constraints.

This parameterization already fulfills the constraints 1 and 2. Now, the functions $\xi_0(q)$ and $\xi_1(q)$ have to be designed in a way so that the mathematical expressions do not become too complicated and the lasting constraints 3,4,5, and 6 can still be fulfilled.

D.1 Angular Constraints

Conditions 3 and 4 introduced constraints on the angles at the starting point and the end point of the nestle curve. Using the parameterization (D.5) of the nestle curve the tangent of $\alpha_c(q)$ becomes

$$\begin{aligned} tan(\alpha_c(q)) &\equiv \frac{y_c'(q)}{x_c'(q)} \\ &= \frac{\sin(\xi_0(q)) + q\cos(\xi_0(q))\xi_0'(q) + \sin(\xi_1(q)) - (1-q)\cos(\xi_1(q))\xi_1'(q)}{\cos(\xi_0(q)) - q\sin(\xi_0(q))\xi_0'(q) + \cos(\xi_1(q)) + (1-s)\sin(\xi_1(q))\xi_1'(q)}. \end{aligned}$$

The tangent angle at $q = 0$ and $q = 1$ therefore becomes

$$tan(\alpha_c(0)) = \frac{\sin(\xi_0(0)) - \cos(A)\xi_1'(0) + \sin(A)}{\cos(\xi_0(0)) + \sin(A)\xi_1'(0) + \cos(A)}, \tag{D.9}$$

$$tan(\alpha_c(1)) = \frac{\sin(A) + \cos(A)\xi_0'(1) + \sin(\xi_1(1))}{\cos(A) - \sin(A)\xi_0'(1) + \cos(\xi_1(1))}. \tag{D.10}$$

The condition $\alpha_c(1) = \alpha_t$ and $\alpha(0) = 0$ results in

$$\xi_0'(1) = \frac{-\sin(A - \alpha_t) + \sin(-\xi_1(1) + \alpha_t)}{\cos(A - \alpha_t)}, \tag{D.11}$$

$$\xi_1'(0) \;=\; \frac{\sin(\xi_0(0)) + \sin(A)}{\cos(A)}. \tag{D.12}$$

In order to make the conditions more handleable new constraints on $\xi_0(q)$ and $\xi_1(q)$ are introduced.

$$\xi_1(1) \;=\; 2\,\alpha_t - A \quad\Rightarrow\quad \xi_0'(1) \;=\; 0, \tag{D.13}$$

$$\xi_0(0) \;=\; -A \quad\Rightarrow\quad \xi_1'(0) \;=\; 0. \tag{D.14}$$

In a later step, the constraints on $\xi_0'(1)$ and $\xi_1'(0)$ are easier to handle than the general form in equations (D.9) and (D.10).

D.2 Curvature Constraints

In order to accomplish condition 5, constraints on curvature at $q = 0$ have to be considered. The curvature of the nestle curve as a function of q calculates to

$$\kappa_c(q) \;\equiv\; \frac{\partial}{\partial\, q}\,\alpha_c(q) \;=\; \frac{y_c''(q)\,x_c'(q) - y_c'(q)\,x_c''(q)}{((x_c'(q))^2 + (y_c'(q))^2)^{\frac{3}{2}}}. \tag{D.15}$$

With a symbolic algebra tool the following expression for $\kappa_c(q)$ can be derived

$$\kappa_c(q) \;=\; \frac{2}{R}\,\frac{\kappa_{nom}(q)}{(\kappa_{denom}(q))^{\frac{3}{2}}}, \tag{D.16}$$

with

$$\begin{aligned}
\kappa_{nom}(q) \;\equiv\; & \Big[(1-q)\sigma_1(q)\xi_1'^2(q) - q\,\sigma_0(q)\xi_0'^2(q) \\
& + \big(2\,\xi_0'(q) + q\,\xi_0''(q)\big)\gamma_0(q) + \big(2\,\xi_1'(q) - (1-q)\xi_1''(q)\big)\gamma_1(q)\Big] \\
& \cdot\Big[\gamma_0(q) + \gamma_1(q) - q\,\sigma_0(q)\xi_0'(q) + (1-q)\,\sigma_1(q)\xi_1'(q)\Big] \\
& + \Big[\,(1-q)\,\gamma_1(q)\xi_1'^2(q) - q\;\gamma_0(q)\xi_0'^2(q) \\
& + \big(2\,\xi_0'(q) + q\;\xi_0''(q)\big)\sigma_0(q) + \big(2\,\xi_1'(q) - (1-q)\xi_1''(q)\big)\sigma_1(q)\Big] \\
& \cdot\Big[\sigma_0(q) + \sigma_1(q) + q\,\gamma_0(q)\xi_0'(q) - (1-q)\,\gamma_1(q)\xi_1'(q)\Big].
\end{aligned} \tag{D.17}$$

$$\begin{aligned}
\kappa_{denom}(q) \;\equiv\; & \Big[\gamma_0(q) + \gamma_1(q) - q\;\sigma_0(q)\xi_0'(q) + (1-q)\,\sigma_1(q)\xi_1'(q)\Big]^2 \\
& + \Big[\sigma_0(q) + \sigma_1(q) + q\;\gamma_0(q)\xi_0'(q) - (1-q)\,\gamma_1(q)\xi_1'(q)\Big]^2
\end{aligned} \tag{D.18}$$

with

$$\sigma_0(q) \;=\; sin(\xi_0(q)), \quad \gamma_1(q) \;=\; cos(\xi_1(q)), \tag{D.19}$$

$$\gamma_0(q) \;=\; cos(\xi_0(q)), \quad \sigma_1(q) \;=\; sin(\xi_1(q)). \tag{D.20}$$

Using all previous constraints (D.7), (D.8), (D.13) and (D.14) the curvature for $s = 0$ becomes

$$\kappa_c(0) \quad = \quad \frac{1}{2}\,\frac{2\,\xi_0'(0) - 2\,\xi_1''(0)}{|\cos(A)|\,R} \;=\; \kappa_0. \tag{D.21}$$

This condition on $\xi_0'(0)$ and $\xi_1''(0)$ is easy to treat for further solution finding.

D.3 Solution

Now, appropriate functions $\xi_0(q)$ and $\xi_1(q)$ have to be chosen, so that all constraints are fulfilled. The sections D.1 and D.2 derived the following constraints from the initial conditions:

1. Fixed points of rotational fields at start ($q = 0$) and end ($q = 1$):

$$\xi_0(1) = A, \quad \xi_1(0) = A. \tag{D.22}$$

2. Angular constraints at the beginning $q = 0$ and at the end $q = 1$:

$$\xi_0(0) = -A, \quad \xi_1(1) = 2\,\alpha_t - A, \quad \xi_1'(0) = 0, \quad \xi_0'(1) = 0. \tag{D.23}$$

3. Curvature constraint for $q = 0$:

$$\frac{1}{2}\,\frac{2\,\xi_0'(0) - 2\,\xi_1''(0)}{|\cos(A)|\,R} \;=\; \kappa_0. \tag{D.24}$$

As long as the above constraints on $\xi_0(q)$ and $\xi_1(q)$ are fulfilled the initial conditions 1-5 will be fulfilled also. The exact form of $\xi_0(q)$ and $\xi_1(q)$, is not fixed. Condition 6, however, forces to choose such functions so that the resulting nestle curve is sufficiently smooth.

The following paragraphs derive a composition of $\xi_0(q)$ and $\xi_1(q)$ that is 'polynomial like'. $\xi_0(q)$ requires a constant term in order to accomplish $\xi_0(0) = -A$. The derivative $\xi_0'(q)$ cannot be constant. Instead, it changes from $\xi_0'(1) = 0$ to something that rises from the curvature constraint (D.24). Therefore, at least a quadratic term is required. The condition $\xi_0'(1) = 0$ can directly be included by using a term $a_x\,(q-1)^N$, with $N \geq 2$. N allows to model the slope of decay of the influence of the first rotational field. Finally, $\xi_0(q)$ is chosen to

$$\xi_0(q) \quad \equiv \quad a_x\,(q-1)^3 + c_x. \tag{D.25}$$

The condition $\xi_1'(0) = 0$ requires that the polynomial that describes $\xi_1(q)$ does not have a linear term. There are still three equations that require three unknowns in order to perform a system that has a unique solution. Thus, $\xi_1(q)$ is chosen as

$$\xi_1(q) \quad \equiv \quad a_y\,q^3 + b_y\,q^2 + c_y. \tag{D.26}$$

With the definitions (D.25) and (D.26) used in the parameterization (D.5) and applying the conditions that were collected in the front of this section, the solution for the parameter set $\{a_x,\, c_x,\, a_y,\, b_y,\, c_y\}$ becomes

$$\begin{aligned} &a_x = 2\,A, \quad c_x = A. \\ &a_y = 2\,\alpha_t - 8\,A + \kappa_0\,|\cos(A)|\,R, \quad b_y = -\kappa_0\,|\cos(A)|\,R + 6\,A, \\ &c_y = A. \end{aligned} \tag{D.27}$$

Example

Given a current position (0, 0), a current angle $\alpha_c(0) = 0$, and a current curvature $\kappa_0 = 0.01\,rad\,m^{-1}$ it has to be nestled to the point $\vec{t} = (100\,m,\, -10\,m)$ with the angle $\alpha_c(1) = 5^o$. For this case, the coefficients a_x, c_x, a_y, b_y and c_y compute to

$$a_x = -0.199337\,rad, \tag{D.28}$$

$$c_x = c_y = -0.0996687\,rad, \tag{D.29}$$

$$a_y = 1.97188\,rad, \tag{D.30}$$

$$b_y = -1.59801\,rad. \tag{D.31}$$

The graph of the resulting nestle curve is shown in figure D.3. Its angular profile with respect to the parameterization index q is shown in figure D.4a. The correspondent curvature profile can be seen in figure D.4b.

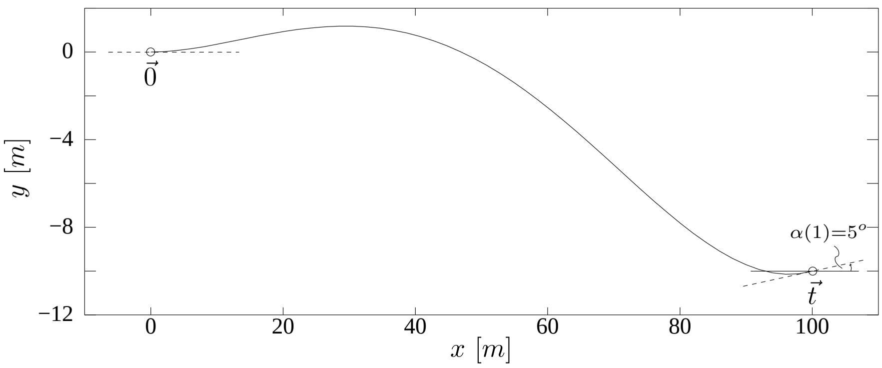


Figure D.3: Example of a nestle curve of second order. X- and y-coordinates of the trajectory $\vec{c}(q)$ for $q = 0$ until $q = 1$.

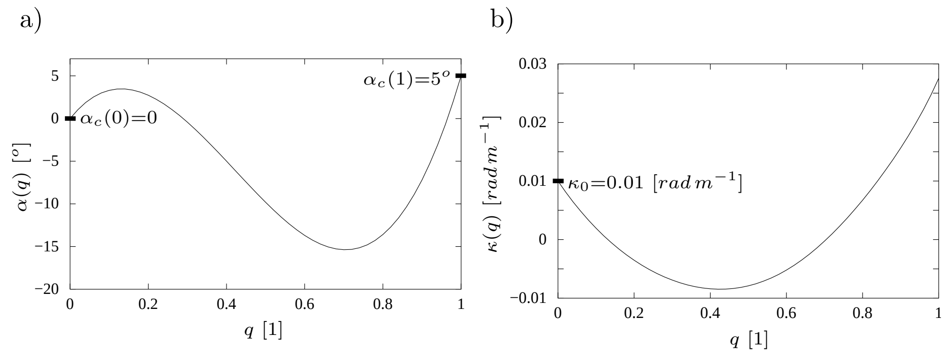


Figure D.4: Example of a nestle curve of second order. a) Angular profile. b) Curvature profile.

Appendix E

Rate of Curvature Change Determination

In order to find a rate of curvature change κ' section 5.3 defined a two layered procedure. It is based on two domains. The domain $D_{\kappa'}$ defines a suitable boundary for the rates of curvature change that have to be investigated. Further, for a given κ' a domain D_s defines boundaries for s, in order to bracket the point of minimum distance between the spiral and the target point. These two issues are handled in the following sections.

E.1 Distance between Spiral and Target Point

The initial curvature κ and the constant rate of curvature change κ' define a spiral given by equation (5.39). The distance of this spiral to the target was defined by equation (5.41) as the minimum distance of the points on the spiral to the target point. Therefore, one has to search the s where $\vec{X}_{\kappa'}(s)$ is as close as possible to the target point. An example of such a minimum distance finding process is illustrated in figure E.1. The following paragraphs determine an appropriate interval $D_s = [s_{low}, s_{up}]$ around the position s_{min}. The domain D_s is derived such that it does not contain a second local minimum.

To avoid loops one has to make restrictions on the maximum angle change. For the driver model, it does not make sense to take a target point where the vehicle model changes the angle to its actual position more than 90 degree. Thus, one can state based on equation (5.30)

$$\begin{aligned} 0 \;\leq\; |\alpha(s)| \;&\leq\; \tfrac{\pi}{2}, \\ 0 \;\leq\; \left|\kappa\, s + \tfrac{1}{2}\,\kappa'\, s^2\right| \;&\leq\; \tfrac{\pi}{2}. \end{aligned} \tag{E.1}$$

Furthermore, the nearest point is not allowed to lie backwards so $s > 0$. It will now be searched for the s_{up} where $|\alpha(s_{up})| = \frac{\pi}{2}$. From this point, different cases need to be investigated. Without any loss of generality one can assume

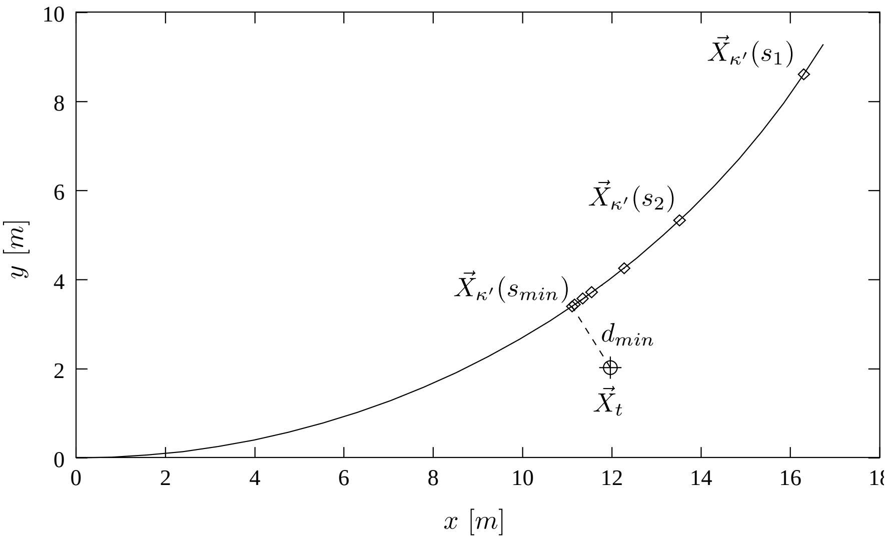


Figure E.1: Iterative search for the place $\vec{X}_{\kappa'}(s_{min})$ on the spiral that lies closest to the target point $\vec{X}_t$. A minimization algorithm calculates repeatedly candidates for s_{min} until the distance converges to d_{min}.

that $\kappa \geq 0$, because the case $\kappa < 0$ can be treated as if $\kappa > 0$ by inverting the y-coordinate of the target point and the sign of the rate of curvature change. The following cases remain for investigation:

1. $\kappa' > 0$:

   The angle $\alpha(s)$ increases, so one has to search for a s to satisfy

   $$\alpha(s) \;=\; \kappa\, s + \tfrac{1}{2}\,\kappa'\, s^2 \;=\; \tfrac{\pi}{2}. \tag{E.2}$$

   The two possible solutions are

   $$s_{1/2} \quad = \quad \tfrac{1}{\kappa'}\left(-\kappa \pm \sqrt{\kappa^2 + \kappa'\,\pi}\right). \tag{E.3}$$

   Note that for $\kappa' > 0$ it can always be stated that $s_2 < 0$ and therefore only s_1 can be a solution, i.e.

   $$s_{up} \;=\; \tfrac{1}{\kappa'}\left(-\kappa + \sqrt{\kappa^2 + \kappa'\,\pi}\right). \tag{E.4}$$

2. $\kappa' < 0$:

   In this case, one must distinguish between two sub cases. Figure E.2 may be given to illustrate the problem. If the angle becomes $\frac{\pi}{2}$ before the curvature changes sign, then one has to find the place where the angle

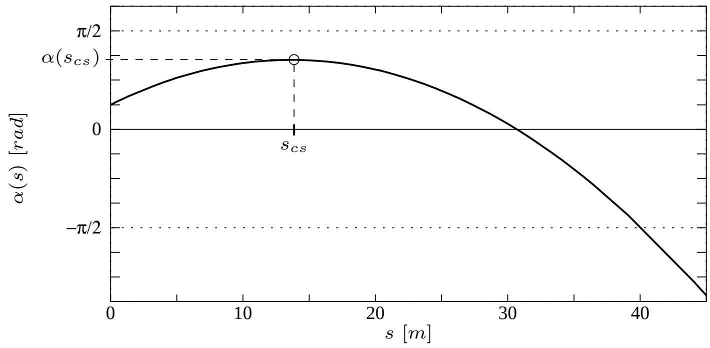


Figure E.2: Angle profile with respect to way length s for a segment of constant rate of curvature change.

becomes $\frac{\pi}{2}$ and make it the upper bound s_{up}. If curvature switches sign before the angle becomes $\frac{\pi}{2}$, then one has to consider an intersection with $-\frac{\pi}{2}$, i.e. a negative angle. Since the angle is a continuous function over the distance s, it suffices to calculate the distance s_{cs} where the curvature switches sign and then the angle that arises at this point. If the angle at s_{cs} is less than $\frac{\pi}{2}$, then it is not greater than $\frac{\pi}{2}$ in the whole interval $[0, s_{cs}]$. From $\kappa(s) = \kappa + \kappa' s$, the position s_{cs} where the curvature changes sign simply calculates to

$$s_{cs} \quad = \quad -\frac{\kappa}{\kappa'}. \tag{E.5}$$

By equation (5.30) the angle change at this point becomes

$$\alpha(s_{cs}) \quad = \quad -\frac{\kappa^2}{\kappa'} + \frac{1}{2}\frac{\kappa^2}{\kappa'} = -\frac{1}{2}\frac{\kappa^2}{\kappa'}. \tag{E.6}$$

The distinction between $\alpha(s_{cs})$ greater or less than $\frac{\pi}{2}$ leads to the following cases

(a) $\alpha(s_{cs}) > \frac{\pi}{2}$:

This is equivalent to $\kappa' > -\frac{\kappa^2}{\pi}$. One has therefore to search for the intersection with $\frac{\pi}{2}$ before s_{cs}. The same equation (E.2) and its solutions s_1 and s_2 in (E.3) have to be considered. Obviously, the square root always exists. Since, $\kappa' < 0$ it follows directly that both, s_1 and s_2, are always positive. s_2 is always greater than s_1 so that s_{up} becomes

$$s_{up} \;=\; \frac{1}{\kappa'}\left(-\kappa - \sqrt{\kappa^2 + \kappa'\,\pi}\right). \tag{E.7}$$

(b) $\alpha(s_{cs}) < \frac{\pi}{2}$:

This is equivalent to $\kappa' < -\frac{\kappa^2}{\pi}$. Here it is searched for the intersection with $\frac{\pi}{2}$. Respectively to equation (E.2) the two solutions are

$$s_{1/2} = \frac{1}{\kappa'}\left(-\kappa \pm \sqrt{\kappa^2 - \kappa'\,\pi}\right). \tag{E.8}$$

s_1 and s_2 are always defined, but only s_2 is greater than zero. Thus s_{up} becomes

$$s_{up} = -\frac{1}{\kappa'}\left(\kappa + \sqrt{\kappa^2 - \kappa'\,\pi}\right). \tag{E.9}$$

(c) $\alpha(s_{cs}) = \frac{\pi}{2}$:
This is equivalent to $\kappa' = -\frac{\kappa^2}{\pi}$. The solution, of course, is simple.

$$s_{up} = -\frac{\kappa}{\kappa'} = \frac{\pi}{\kappa}. \tag{E.10}$$

Collecting the results from theses cases one can determine s_{up} by

$$s_{up} = \begin{cases} -\frac{1}{\kappa'}\left(\kappa - \sqrt{\kappa^2 + \kappa'\,\pi}\right) & \forall \quad \kappa' > 0 \\ -\frac{1}{\kappa'}\left(\kappa + \sqrt{\kappa^2 + \kappa'\,\pi}\right) & \forall \quad -\frac{\kappa^2}{\pi} < \kappa' \leq 0 \\ \frac{\pi}{\kappa} & \forall \quad \kappa' = -\frac{\kappa^2}{\pi} \\ -\frac{1}{\kappa'}\left(\kappa + \sqrt{\kappa^2 - \kappa'\,\pi}\right) & \forall \quad \kappa' < -\frac{\kappa^2}{\pi} \end{cases}. \tag{E.11}$$

The minimum distance length s_{low} is without any prove estimated as the x-coordinate of the target point, i.e.

$$s_{low} = t_x. \tag{E.12}$$

A rate of curvature change κ' for which $s_{low} > s_{up}$ would be plausibly senseless. This would mean that the spiral turns around before even coming close to the target point. The following hypothesis shall be the basis for choosing the interval $[s_{low}, s_{up}]$ as the desired domain D_s in equation (5.41).

Hypothesis: 1 *Let $\vec{X}_{\kappa'}(s)$ be a parameterization of a spiral index by the distance s. It starts from the origin $(0,0)$ with an angle of zero and is defined by a starting curvature κ and constant rate of curvature change κ'. Let the distance between any point on the spiral and a specific point $\vec{X}_t$ be*

$$d(s) = |\vec{X}_{\kappa'}(s) - \vec{X}_t|. \tag{E.13}$$

Then there is only one distinct $s_0 \in [s_{low}, s_{up}]$ for which

$$\left.\frac{d}{ds}d(s)\right|_{s=s_0} = 0, \text{ and } \left.\frac{d^2}{ds^2}d(s)\right|_{s=s_0} > 0. \tag{E.14}$$

That means that there is only one point in $[s_{low}, s_{up}]$ on the spiral where the distance gets minimal.

With this assumption it can be concluded that the domain D_s only contains the absolute minimum and no other local ones. An ordinary minimum search algorithm can be applied to search for the minimal distance in the specified interval.

E.2 Spiral to hit Target Point

The previous section showed that it is possible to define a distance between a point $\vec{X}_t$ and a spiral as the minimum distance between both. With the fixed initial curvature the spiral's shape is distinctly defined by the rate of curvature change κ'. With the distance measure between the target point $\vec{X}_t$ and the spiral, it is possible to search systematically in the set of spirals the one that is closest to the target point. Again, an interval of rates of curvature change $[\kappa'_{low}, \kappa'_{up}]$ has to be determined that includes the spiral with the minimal distance to the target point. The closer the borders can be determined in which the absolute minimum lies, the faster a minimization algorithm finds a solution.

This section, therefore, focuses on appropriate borders for the rate of curvature change κ' that are assumed to contain only one local minimum that is at the same time the absolute minimum. This assumption is not proven. However, it is proven that the absolute minimum has to lie in the former calculated borders. For the same reason as in the previous section, the initial curvature κ is supposed to be greater or equal zero. As mentioned before, the case $\kappa < 0$ can be transformed into a identical situation with $\kappa > 0$ (see page 161). Therefore, this constraint does not introduce any restrictions on the general validity of the following discussion.

Let us assume, that we already found the solution κ' and the s, where the spiral hits the target point. By convention, the angle $\alpha(s)$ at this point has to be less than $\frac{\pi}{2}$. Remembering equation (E.1) it can be stated for this κ' that

$$\begin{array}{lcrcl} \Delta\alpha \geq 0 & \Rightarrow & \frac{2\,\kappa}{s} & \leq \kappa' \leq & \frac{\pi}{s^2} - \frac{2\,\kappa}{s}, \\ \Delta\alpha < 0 & \Rightarrow & -\frac{\pi}{s^2} - \frac{2\,\kappa}{s} & < \kappa' < & -\frac{2\,\kappa}{s}. \end{array} \tag{E.15}$$

It is not trivial in the general case to determine if the angle $\alpha(s)$ is positive or negative when the spiral hits the target. There are, however, two sufficient conditions that can be stated. If the target lies inside the circle of the constant initial curvature, than the curvature $\kappa(s)$ has to increase and the angle is positive (since it can not be greater than $\pi/2$) when the spiral hits the target point. With the point $(0, \frac{1}{\kappa})$ as the center of the extrapolated circle it follows that

$$t_x + (t_y - \tfrac{1}{\kappa})^2 \, < \, \tfrac{1}{\kappa^2} \quad \Rightarrow \quad \alpha \, \geq \, 0. \tag{E.16}$$

On the other hand, the convention that κ is positive implies that if the target point lies in the lower half plane, then the trajectory has to change direction from upwards to downwards. Correspondingly, the angle at the end will be negative, i.e.

$$t_y \leq 0 \quad \Rightarrow \quad \alpha \, < \, 0. \tag{E.17}$$

Equation (E.15) predicates for any case that

$$-\frac{\pi}{s^2} - \frac{2\,\kappa}{s} \quad \leq \kappa' \leq \quad \frac{\pi}{s^2} - \frac{2\,\kappa}{s}. \tag{E.18}$$

This relation will be used if it cannot be determined if α is negative or positive when the spiral hits the target point. The absolute minimum distance that one has to travel on the spiral to reach the target point is the direct line from the origin to the target point, i.e.

$$s_{min} = \sqrt{t_x^2 + t_y^2}. \tag{E.19}$$

Assuming that the spiral that does not turn more than $\frac{\pi}{2}$, the longest possible distance from the origin to the target point along the spiral would be the addition of the x- and y-coordinates of the target point, i.e.

$$s_{max} = |\, t_x \,| + |\, t_y \,| \tag{E.20}$$

Thus, one can define the domain $D_{\kappa'} = [\kappa'_{low},\, \kappa'_{up}]$ that is needed for equation (5.42) as

$$\kappa'_{low} \equiv \begin{cases} \frac{2\,\kappa}{|\,t_x\,|+|\,t_y\,|} & \forall \quad x_{t,1}^2 + (t_y - \frac{1}{\kappa})^2 \,<\, \frac{1}{\kappa^2} \\ -\frac{\pi}{x_{t,1}^2+x_{t,2}^2} - \frac{2\,\kappa}{\sqrt{x_{t,1}^2+x_{t,2}^2}} & else \end{cases} \tag{E.21}$$

and

$$\kappa'_{up} \equiv \begin{cases} -\frac{2\,\kappa}{|\,t_x\,|+|\,t_y\,|} & \forall \quad t_y < 0 \\ \frac{\pi}{x_{t,1}^2+x_{t,2}^2} - \frac{2\,\kappa}{|\,t_x\,|+|\,t_y\,|} & else \end{cases} . \tag{E.22}$$

These boundaries frame the best κ' that is required to pass through or as close as possible to the target point $\vec{X}_t$. Although it was not proven in this section, practical experience shows that the distance to the target point as a function of κ' does not contain a second local minimum besides the absolute minimum. For this reason, it is possible to use $[\kappa'_{low},\, \kappa'_{up}]$ as the domain $D_{\kappa'}$. The minimization algorithm is then used to find the numerical solution for the constant rate of curvature change κ' that has to be applied to pass through the target point.

Appendix F

Derivation of Admissible Domains of Acceleration

Section 5.4 pointed out the general aspects of how to calculate the domains of admissible accelerations for a given curvature profile with a constant rate of curvature change. The following sections derive analytical formula for these domains. It is advisable to use a symbolic algebra tool such as *Maple*^{TM} or *MuPAD*^{TM} in order to trace the reasoning.

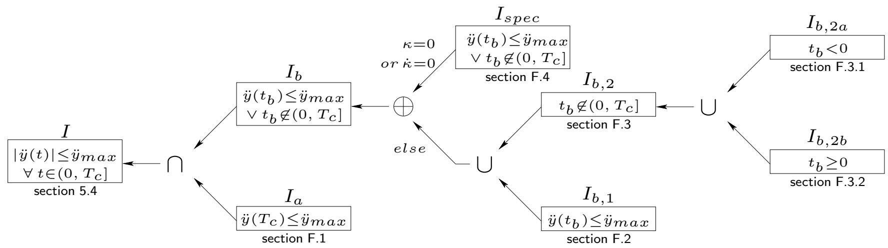


Figure F.1: Overview over structure of the admissible domain of accelerations I. Each box contains the condition that is fulfilled for all $\dot{v}$ in the specific sub-domain. The name of the sub-domain is given on top of each box. The section where the domain is discussed is specified under the box.

F.1 Calculation of I_a

I_a is defined as the domain of $\dot{v}$ where the lateral acceleration at the end of the control interval is less than $\ddot{y}_{max}$. Recalling equation (5.50) this can be expressed as

$$|\,\ddot{y}(T_c)\,| \;=\; \left|(v + \,\dot{v}\,T_c)^2 \left(\kappa + \kappa' \left(v + \frac{1}{2}\,\dot{v}\,T_c\right)\,T_c\right)\right| \;\leq\; \ddot{y}_{max}. \tag{F.1}$$

To examine the domain where this condition holds it is useful to study the behavior of the function $\ddot{y}(T_c)$ with respect to $\dot{v}$. The roots of $\ddot{y}(T_c)$ with respect to $\dot{v}$ are

$$\dot{v}_{a,r,1/2} \;=\; -\frac{v}{T_c} \quad \text{and} \quad \dot{v}_{a,r,3} \;=\; -2\,\frac{v\,\kappa'\,T_c + \kappa}{\kappa'\,{T_c}^2}. \tag{F.2}$$

Deriving $\ddot{y}(T_c)$ with respect to $\dot{v}$ from equation (F.1) one gets two possible extrema

$$\dot{v}_{a,1} \;=\; -\frac{v}{T_c} \quad \text{and} \quad \dot{v}_{a,2} \;=\; -\frac{4\,\kappa + 5\,v\,\kappa'\,T_c}{3\,\kappa'\,{T_c}^2}. \tag{F.3}$$

The position of the first extremum is independent of κ'. It is identical to the left border of I_0 (see (5.49), page 81). The second one always lies on the right hand side of $-\frac{v}{T_c}$ for a certain range of κ'. It can directly be derived that

$$\dot{v}_{a,2} \;\in\; I_0 \quad \Leftrightarrow \quad \kappa' \;\in\; (\tfrac{-2\,\kappa}{v\,T_c},\, 0). \tag{F.4}$$

The second derivatives at $\dot{v}_{a,1}$ and $\dot{v}_{a,2}$ are

$$\left.\frac{\partial^2}{\partial\,\dot{v}^2}\,\ddot{y}(T_c)\right|_{\dot{v}=\dot{v}_{a,1}} \;=\; (v\,\kappa'\,T_c + 2\,\kappa)\;{T_c}^2, \tag{F.5}$$

$$\left.\frac{\partial^2}{\partial\,\dot{v}^2}\,\ddot{y}(T_c)\right|_{\dot{v}=\dot{v}_{a,2}} \;=\; -\,(v\,\kappa'\,T_c + 2\,\kappa)\;{T_c}^2. \tag{F.6}$$

So it can be stated that

$$\dot{v}_{a,1}\ \text{produces a minimum} \quad \Leftrightarrow \quad \kappa' \;>\; -\frac{2\,\kappa}{v\,T_c}, \tag{F.7}$$

$$\dot{v}_{a,2}\ \text{produces a maximum} \quad \Leftrightarrow \quad \kappa' \;\in\; (\tfrac{-2\,\kappa}{v\,T_c},\, 0). \tag{F.8}$$

Referring to condition (F.4) it is obvious that if $\dot{v}$ lies in I_0, then $\dot{v}_{a,1}$ will always produce a minimum and $\dot{v}_{a,2}$ will always produce a maximum. The values of $\ddot{y}(T_c)$ at $\dot{v}_{a,1}$ and $\dot{v}_{a,2}$ are

$$\ddot{y}(T_c)|_{\dot{v}=\dot{v}_{a,1}} = 0 \quad \text{and} \quad \ddot{y}(T_c)|_{\dot{v}=\dot{v}_{a,2}} = \frac{2}{27}\,\frac{(v\,\kappa'\,T_c + 2\,\kappa)^3}{\kappa'^2\,{T_c}^2}. \tag{F.9}$$

By these results it is possible to draw a graph of $\ddot{y}(T_c)$ with respect to $\dot{v}$. This can be seen in figure F.2. The shaded area indicates the forbidden domain, since there is $\dot{v} \;<\; -v/T_c$. If $\kappa' \notin (-\frac{2\,\kappa}{v\,T_c},\, 0]$ as illustrated in figure F.2a it is relatively easy to define the domain, where $\ddot{y}(T_c) \;<\; \ddot{y}_{max}$. Since there is no extremum at the right hand side (see (F.4)) there can be only one intersection

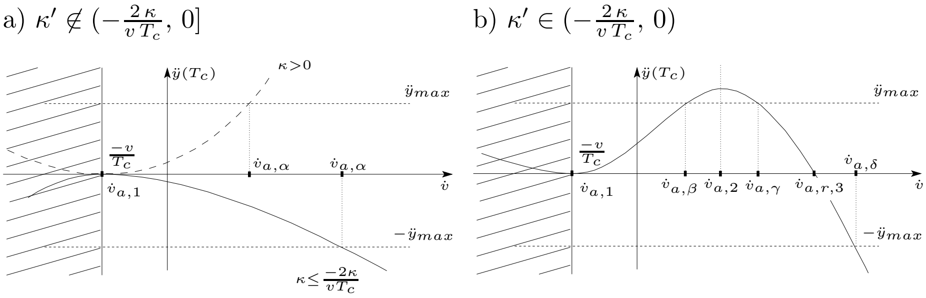


Figure F.2: Graph of $\ddot{y}(T_c)$ with respect to $\dot{v}$ depending on different domains of κ'.

with $\pm\ddot{y}_{max}$ dependent on $\dot{v}_{a,1}$ being a minimum or a maximum. The admissible interval for $\dot{v}$ ranges from $-\frac{v}{T_c}$ to $\dot{v}_{a,\alpha}$. $\dot{v}_{a,\alpha}$ can now be defined as

$$\dot{v}_{a,\alpha} > -\frac{v}{T_c} \qquad \wedge \qquad \ddot{y}(T_c)|_{\dot{v}=\dot{v}_{a,\alpha}} = \begin{cases} \ddot{y}_{max} & \forall \quad \kappa' > 0 \\ -\ddot{y}_{max} & \forall \quad \kappa' < -\frac{2\kappa}{v\,T_c} \end{cases} . \tag{F.10}$$

In case of $\kappa' \in (-\frac{2\kappa}{v\,T_c},\,0)$ the extremum lies as a maximum in I_0. Here, one has to distinguish between the case when the extremum lies under the desired maximum lateral acceleration $\ddot{y}_{max}$ and the case when it lies above it. For the second case, one can define $\dot{v}_{a,\beta}$ and $\dot{v}_{a,\gamma}$

$$\dot{v}_{a,\beta} \in (-\tfrac{v}{T_c},\,\dot{v}_{a,2}] \qquad \wedge \qquad \ddot{y}(T_c)|_{\dot{v}} = \ddot{y}_{max}, \tag{F.11}$$
$$\dot{v}_{a,\gamma} \in [\dot{v}_{a,2},\,\dot{v}_{a,r,3}] \qquad \wedge \qquad \ddot{y}(T_c)|_{\dot{v}} = \ddot{y}_{max}. \tag{F.12}$$

The third point $\dot{v}_\delta$ is determined by the intersection with $-\ddot{y}_{max}$, i.e.

$$\dot{v}_{a,\delta} \in (\dot{v}_{a,r,3},\,\infty) \qquad \wedge \qquad \ddot{y}(T_c)|_{\dot{v}} = -\ddot{y}_{max}. \tag{F.13}$$

If the maximum at $\dot{v}_{a,2}$ lies under $\ddot{y}_{max}$ then one only has to take into account the point $\dot{v}_{a,\delta}$ where the lateral acceleration intersects with $-\ddot{y}_{max}$. The concrete values for $\dot{v}_{a,\alpha}$, $\dot{v}_{a,\beta}$, $\dot{v}_{a,\gamma}$ and $\dot{v}_{a,\delta}$ are calculated by a root search algorithm based on the equation $\ddot{y}(T_c) \pm \ddot{y}_{max} = 0$.

In case that $\kappa \neq 0$ and $\kappa' = 0$ the solution is special. Then the only restriction on $\dot{v}$ evolves to

$$-\frac{v}{T_c} < \dot{v} \leq \frac{1}{T_c}\left(\sqrt{\frac{\ddot{y}_{max}}{\kappa}} - v\right). \tag{F.14}$$

By means of all these results, we are now able to define the domain I_a, where condition (F.1) holds.

$$I_a \equiv \begin{cases} (-\frac{v}{T_c},\, \dot{v}_{a,\alpha}] & \forall \quad \kappa' \notin (-\frac{2\,\kappa}{v\,T_c},\, 0] \\ (-\frac{v}{T_c},\, \dot{v}_{a,\beta}] \cup [\dot{v}_{a,\gamma}, \dot{v}_{a,\delta}] & \forall \quad \kappa' \in (-\frac{2\,\kappa}{v\,T_c},\, 0) \\ & \quad \wedge \;\; \ddot{y}(T_c)|_{\dot{v}=\dot{v}_{a,2}} > \ddot{y}_{max} \\ (-\frac{v}{T_c},\, \dot{v}_{a,\delta}] & \forall \quad \kappa' \in (-\frac{2\,\kappa}{v\,T_c},\, 0) \\ & \quad \wedge \;\; \ddot{y}(T_c)|_{\dot{v}=\dot{v}_{a,2}} \leq \ddot{y}_{max} \\ \left(-\frac{v}{T_c},\, \frac{1}{T_c}\left(\sqrt{\frac{\ddot{y}_{max}}{\kappa}} - v\right)\right] & \forall \quad \kappa \neq 0 \,\wedge\, \kappa' = 0 \end{cases} \quad . \tag{F.15}$$

F.2 Calculation of $I_{b,1}$

The subset I_b considers the cases where condition (5.52) holds. This means that it determines the domain of $\dot{v}$ where the absolute value $\ddot{y}(t_b)$ is less or equal $\ddot{y}_{max}$.

Root search in the derivative of $\ddot{y}(t_b)$ with respect to $\dot{v}$ results in

$$\dot{v}_{b,1} = -\frac{2\,\kappa'\,v^2}{v\,\kappa'\,T_c + 4\,\kappa} \quad \text{and} \quad \dot{v}_{b,2/3} = -\frac{2\,\kappa'\,v^2}{v\,\kappa'\,T_c - 2\,\kappa}. \tag{F.16}$$

The second derivatives show that $\dot{v}_{b,1}$ is always a minimum and $\dot{v}_{b,2/3}$ is a double point of inflection:

$$\frac{\partial^2}{\partial\,\dot{v}^2}\,\ddot{y}(t_b)\bigg|_{\dot{v}=\dot{v}_{b,1}} = \frac{(v\,\kappa'\,T_c + 4\,\kappa)^4}{96\,\kappa\,v^2\kappa'^2} > 0, \tag{F.17}$$

$$\frac{\partial^2}{\partial\,\dot{v}^2}\,\ddot{y}(t_b)\bigg|_{\dot{v}=\dot{v}_{b,2/3}} = 0. \tag{F.18}$$

The value of $\ddot{y}(t_b)$ for $\dot{v} = \dot{v}_{b,1}$ calculates to

$$\ddot{y}(t_b)|_{\dot{v}=\dot{v}_{b,1}} = v^2\,\kappa. \tag{F.19}$$

Solving $\ddot{y}(t_b) = 0$ results in a triple root at the same point as the point of inflection $\dot{v}_{b,2/3}$, i.e.

$$\dot{v}_{b,r,1/2/3} = -\frac{2\,\kappa'\,v^2}{v\,\kappa'\,T_c - 2\,\kappa}. \tag{F.20}$$

$\ddot{y}(t_b)$ has two poles at $\dot{v} = -2\,\frac{v}{T_c}$ and $\dot{v} = 0$. The correspondent left and right hand limits are

$$\lim_{\dot{v}\to -\frac{2\,v}{Tc}^{\pm}} \ddot{y}(t_b) = sgn(\kappa)\,\infty \quad \text{and} \quad \lim_{\dot{v}\to 0^{\pm}} \ddot{y}(t_b) = \mp sgn(\kappa')\,\infty. \tag{F.21}$$

Further, $\ddot{y}(t_b)$ strives to a constant value for $\dot{v} \to \infty$

$$\ddot{y}_{lim} = \lim_{\dot{v}\to\infty} \ddot{y}(t_b) = -\frac{2}{27}\,\frac{(v\,\kappa'\,T_c - 2\,\kappa)^3}{\kappa'^2 {T_c}^2}. \tag{F.22}$$

Table F.1: Domains of the extrema $\dot{v}_{b,1}$ and $\dot{v}_{b,2}$ of the function $\ddot{y}(t_b)$ dependent on intervals where κ' is located.

	Domains of κ'			
	$(-\infty,\, -\frac{4\,\kappa}{v\,T_c})$	$(-\frac{4\,\kappa}{v\,T_c},\, 0)$	$(0,\, \frac{2\,\kappa}{v\,T_c})$	$(\frac{2\,\kappa}{v\,T_c},\, \infty)$
$\dot{v}_{b,1}$	$(-\infty,\, -\frac{2\,v}{T_c})$	$(0,\, \infty)$	$(-\frac{2\,v}{T_c},\, 0)$	
$\dot{v}_{b,2}$	$(-\frac{2\,v}{T_c},\, 0)$		$(0,\, \infty)$	$(-\infty,\, -\frac{2\,v}{T_c})$

In order to search intersections with $\pm\ddot{y}_{max}$ by a root search algorithm in a reliable manner one has to divide up the domain into intervals, that do not contain any local minima. The first separation comes at the poles at $-\frac{2\,v}{T_c}$ and 0. The second separation happens at the minimum $\dot{v}_{b,1}$. Dependent on κ' it lies in $(-2\,\frac{v}{T_c},\, 0)$, $(0,\, \infty)$ or $(-\infty,\, -2\,\frac{v}{T_c}]$. κ' also directly defines the domain, where $\dot{v}_{b,2}$ lies in. The dependency can be seen in table F.1.

With the information of the former equations one is now able to draw a schematic picture of the function $\ddot{y}(t_b)$ for $\kappa' > 0$ and $\kappa' < 0$ as depicted in Figure F.3. The two domains in that $\ddot{y}(t_b) \leq \ddot{y}_{max}$ are $[\dot{v}_{b,\alpha},\, \dot{v}_{b,\beta}]$ and $[\dot{v}_{b,\gamma},\, \dot{v}_{b,\delta}]$. The second interval does not always exist and $\dot{v}_{b,\delta}$ may also lie in infinity. $\dot{v}_{b,\alpha}$, $\dot{v}_{b,\beta}$, $\dot{v}_{b,\gamma}$, and $\dot{v}_{b,\delta}$ can be found by means of an ordinary root search algorithm since I_0 is now separated into domains where the function behaves monotonously.

a) $\kappa' > 0$ b) $\kappa' < 0$

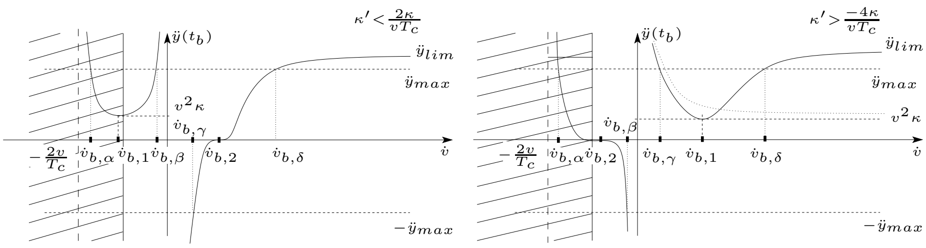


Figure F.3: Dependence of $\ddot{y}(t_b)$ on $\dot{v}$ for a) $\kappa' > 0$ (For $\kappa' > \frac{2\,\kappa}{v\,T_c}$ there is no root in the allowed domain) and b) $\kappa' < 0$. The nearer κ' comes to $-\frac{4\,\kappa}{v\,T_c}$ the more the minimum moves to $+\infty$. The dotted curve indicates the limit where $\kappa' \to -\frac{4\,\kappa}{v\,T_c}$. Then the minimum lies in infinity and $\ddot{y}(t_b)$ converges to $v^2\,\kappa$. For $\kappa' < -\frac{4\,\kappa}{v\,T_c}$ there is no minimum in the admissible domain.

$I_{b,1}$ has the following form:

$$I_{b,1} \equiv \begin{cases} [\dot{v}_{b,\alpha},\, \dot{v}_{b,\beta}] & \forall \quad \dot{v}_{b,\alpha} > -\frac{v}{T_c} \;\wedge\; v^2\,\kappa \leq \ddot{y}_{max} \\ (-\frac{v}{T_c},\, \dot{v}_{b,\beta}] & \forall \quad \dot{v}_{b,\alpha} < -\frac{v}{T_c} \;\wedge\; \dot{v}_{b,\beta} > -\frac{v}{T_c} \\ & \quad\;\; \wedge\; v^2\,\kappa \leq \ddot{y}_{max} \\ \emptyset & else \end{cases}$$
$$\cup \begin{cases} [\dot{v}_{b,\gamma},\, \dot{v}_{b,\delta}] & \forall \quad \kappa' > 0 \;\wedge\; \ddot{y}_{lim} > \ddot{y}_{max} \\ [\dot{v}_{b,\gamma},\, \infty] & \forall \quad \kappa' > 0 \;\wedge\; \ddot{y}_{lim} \in (-\ddot{y}_{max},\, \ddot{y}_{max}] \\ [\dot{v}_{b,\gamma},\, \dot{v}_{b,\delta}] & \forall \quad \kappa' \in [-\frac{4\,\kappa}{v\,T_c},\, 0) \;\wedge\; \ddot{y}_{lim} > \ddot{y}_{max} \\ & \quad\;\; \wedge\; v^2\,\kappa \leq \ddot{y}_{max} \\ [\dot{v}_{b,\gamma},\, \infty] & \forall \quad \kappa' < 0 \;\wedge\; \ddot{y}_{lim} < \ddot{y}_{max} \\ \emptyset & else \end{cases} . \tag{F.23}$$

F.3 Calculation of $I_{b,2}$

Equation (5.47) defined the time of the extremum of lateral acceleration as a function of $\dot{v}$. The following sections discuss the search for the subset $I_{b,2} \subset I_0$ where $t_b \notin [0,\, T_c)$. The discussion is divided into two parts, which correspond to two sub-domains. These sub-domains are defined as

$$I_{b,2a} \equiv \{\dot{v} \in I_0 : t_b < 0\} \quad \text{and} \quad I_{b,2b} \equiv \{\dot{v} \in I_0 : t_b \geq T_c\}. \tag{F.24}$$

Thus, the domain $I_{b,2}$ becomes then

$$I_{b,2} \equiv I_{b,2a} \cup I_{b,2b}. \tag{F.25}$$

F.3.1 Calculation of $I_{b,2a}$

First, the $\dot{v}$ where $t_b < 0$ is calculated, namely

$$t_b = -\frac{2\,\kappa'\,v^2 + v\kappa'\,\dot{v}\,T_c + 4\,\dot{v}\,\kappa}{3\,\dot{v}\,\kappa'\,(2\,v + \dot{v}\,T_c)} \quad < \quad 0\,. \tag{F.26}$$

Regarding the denominator of t_b in equation (F.26) leads to two poles but only one of them lies inside the allowed interval I_0. Considering the right and left hand limits to the poles we get

$$\lim_{\dot{v}\to 0^{\pm}} t_b = \mp\infty, \tag{F.27}$$
$$\lim_{\dot{v}\to -\frac{2\,v}{T_c}^{\pm}} t_b = \mp sign(\kappa')\infty. \tag{F.28}$$

Further, t_b has one single root at

$$\dot{v}_{t,r} = -2\,\frac{\kappa'\,v^2}{v\,\kappa'\,T_c + 4\,\kappa}. \tag{F.29}$$

The two poles, together with the root, enable assumptions to be made about the conditions for which t_b is negative. There are three cases corresponding to the diagrams in figure F.4:

1. $\kappa' \geq 0$ (figure F.4a):

   The right hand limit at the pole at $-\frac{2\,v}{T_c}$ is negative, where else the left hand limit at 0 is alway positive. $v_{t,r}$ has therefore to lie in between $(-\frac{2\,v}{T_c},\,0)$. If $\dot{v} > 0$, then t_b is always negative. The second domain where it could be negative is within $-\frac{v}{T_c}$ and $v_{t,r}$ as long as $v_{t,r}$ lies in $(-\frac{v}{T_c},\,0)$. It can now be shown that

   $$\dot{v}_{t,r} \in (-\tfrac{v}{T_c},\,0) \quad \Leftrightarrow \quad \kappa' \in (0,\,\tfrac{4\,\kappa}{v\,T_c}). \tag{F.30}$$

2. $\kappa' < 0$:

   The right hand limit at the pole $-\frac{2\,v}{T_c}$ is positive. From (F.29) it follows that the root never lies in the interval $(-\frac{2\,v}{T_c},\,0)$. Here, two sub cases have to be distinguished:

   - $\dot{v}_{t,r} > 0$ (figure F.4b):
     The appropriate condition for κ' can be formulated as

     $$\dot{v}_{t,r} > 0 \quad \Leftrightarrow \quad \kappa' \in (-\tfrac{4\,\kappa}{v\,T_c},\,0). \tag{F.31}$$

     In this case, t_b is never negative in the allowed domain I_0, except if $\dot{v} \in (0,\,\dot{v}_{t,r})$.

   - $\dot{v}_{t,r} < -\frac{2\,v}{T_c}$ (figure F.4c):
     The condition for κ' computes to

     $$\dot{v}_{t,r} < 0 \quad \Leftrightarrow \quad \kappa' < -\frac{4\,\kappa}{v\,T_c}. \tag{F.32}$$

     where t_b is only negative if $\dot{v} > 0$.

a) $\kappa' > 0$ b) $\kappa' < 0 \;\wedge\; \dot{v}_{t,r} \geq \frac{-2\,v}{T_c}$ c) $\kappa' < 0 \;\wedge\; \dot{v}_{t,r} < \frac{-2\,v}{T_c}$

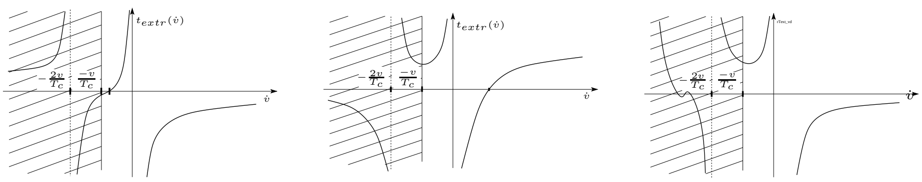


Figure F.4: Shapes of the function $t_b(\dot{v})$ dependent on different cases for κ'.

With the above considerations in mind it is possible to state

$$I_{b,2a} \quad \equiv \quad \begin{cases} (0,\,\dot{v}_{t,r}) & \forall \quad \kappa' \in (-\frac{4\,\kappa}{v\,T_c},\,0) \\ (-\frac{v}{T_c},\,\dot{v}_{t,r}) \cup (0,\,\infty) & \forall \quad \kappa' \in (0,\,\frac{4\,\kappa}{v\,T_c}) \\ (0,\,\infty) & \forall \quad \kappa' \notin (-\frac{4\,\kappa}{v\,T_c},\,\frac{4\,\kappa}{v\,T_c}) \end{cases} \quad . \tag{F.33}$$

F.3.2 Calculation of $I_{b,2b}$

Next, to calculate the $\dot{v}$ where $t_b \geq T_c$ one has to consider different cases. Starting from the inequality

$$t_b \;=\; -\frac{2\,\kappa'\,v^2 + v\kappa'\,\dot{v}\,T_c + 4\,\dot{v}\,\kappa}{3\,\dot{v}\,\kappa'\,\left(2\,v + \dot{v}\,T_c\right)} \quad \geq \quad T_c. \tag{F.34}$$

In the allowed domain, where $\dot{v} > -\frac{v}{T_c}$, it is equivalent to

$$\dot{v} + \frac{7}{3}\,\frac{v}{T_c} + \frac{4}{3}\,\frac{\kappa}{T_c^2\,\kappa'} + \frac{2}{3}\,\frac{v^2}{T_c^2}\,\frac{1}{\dot{v}} \quad \leq \quad 0. \tag{F.35}$$

To simplify further arguments, let us define

$$p = \frac{7}{3}\,\frac{v}{T_c} + \frac{4}{3}\,\frac{\kappa}{T_c^2\,\kappa'}, \quad q = \frac{2}{3}\,\frac{v^2}{T_c^2}. \tag{F.36}$$

So that one writes

$$\dot{v} + p + \frac{q}{\dot{v}} \,\leq\, 0. \tag{F.37}$$

The roots of the left hand side of the inequality are

$$\dot{v}_{0,1/2} \;\equiv\; -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}. \tag{F.38}$$

For simplicity of further considerations, let us define

$$\dot{v}_{0,min} \;\equiv \min\{\dot{v}_{0,1},\, \dot{v}_{0,2}\}, \quad \dot{v}_{0,max} \;\equiv \max\{\dot{v}_{0,1},\, \dot{v}_{0,2}\}, \tag{F.39}$$

Depending on $\dot{v}$ one has to distinguish two separate cases:

1. $\dot{v} < 0$:

   Then (F.37) is equivalent to

   $$\dot{v}^2 + p\,\dot{v} + q \,\geq\, 0, \tag{F.40}$$

   where the left hand of this expression strives to positive infinity for $\dot{v} \to \pm\infty$. If the roots from (F.38) do not exist, then the left hand side is entirely positive for all $\dot{v}$. If they exist then the left hand side is only negative in the domain $(\dot{v}_{0,min},\, \dot{v}_{0,max})$. Regarding the term under the root in (F.38) and following the argumentation chain back the equations (F.37), (F.35), and (F.34), it can be stated

   $$t_b \geq T_c \quad \Leftrightarrow \quad \begin{cases} \quad true & \forall \;\; \frac{p^2}{4} - q \,<\, 0 \\ \dot{v} \notin (\dot{v}_{0,min}, \dot{v}_{0,max}) & \forall \;\; \frac{p^2}{4} - q \,\geq\, 0 \end{cases} \tag{F.41}$$

2. $\dot{v} > 0$: Then (F.37) is equivalent to

$$\dot{v}^2 + p\,\dot{v} + q \,\leq\, 0. \tag{F.42}$$

The above argumentation applied to this equation leads directly to the assumption

$$t_b \geq T_c \quad \Leftrightarrow \quad \dot{v} \in [\dot{v}_{0,min}, \dot{v}_{0,max}] \wedge \frac{p^2}{4} - q \,\geq\, 0. \tag{F.43}$$

3. $\dot{v} = 0$:

This case is not covered by (F.35). Equation (F.34) but shows that t_b is infinite. So, in this case it is always true that $t_b \geq T_c$.

The conditions in (F.41) and (F.43) can therefore be reformulated. Basing on that one can now define the domain I_{b2b}, where $t_b \,\geq\, T_c$:

$$I_{b,2b} \;\equiv\; \begin{cases} I_0 & \forall \;\; \dot{v} < 0 \wedge \frac{p^2}{4} < q \\ (\dot{v}_{0,min},\, \dot{v}_{0,max})^c & \forall \;\; \dot{v} < 0 \wedge \frac{p^2}{4} \geq q \\ [\dot{v}_{0,min},\, \dot{v}_{0,max}] & \forall \;\; \dot{v} > 0 \wedge \frac{p^2}{4} \geq q \\ \{0\} & \forall \;\; \dot{v} = 0 \end{cases}, \tag{F.44}$$

$$= \begin{cases} (-\frac{v}{T_c},\, 0] & \forall \;\; \frac{p^2}{4} < q \\ (\dot{v}_{0,min},\, \dot{v}_{0,max})^c \cap (-\infty,\, 0] & \\ \cup\, [\dot{v}_{0,min},\, \dot{v}_{0,max}] \cap [0,\, \infty) & \forall \;\; \frac{p^2}{4} \geq q \end{cases}. \tag{F.45}$$

Thus,

$$I_{b,2b} \;=\; \begin{cases} (-\frac{v}{T_c},\, 0] & \forall \;\; \frac{p^2}{4} < q \\ (\dot{v}_{0,min},\, \dot{v}_{0,max})^c \cap (-\infty,\, 0] & \forall \;\; \frac{p^2}{4} \geq q \\ & \quad \wedge\, \dot{v}_{min} < \dot{v}_{max} \leq 0 \\ [\dot{v}_{0,min},\, \dot{v}_{0,max}] \cap (-\infty,\, 0] & \forall \;\; \frac{p^2}{4} \geq q \\ & \quad \wedge\, \dot{v}_{max} > \dot{v}_{min} \geq 0 \\ (-\infty,\, \dot{v}_{0,min}) \cap [0,\, \dot{v}_{0,max}] & \forall \;\; \frac{p^2}{4} \geq q \\ & \quad \wedge\, \dot{v}_{max} > 0 > \dot{v}_{min} \end{cases}. \tag{F.46}$$

Recall that $\dot{v}_{min}$ and $\dot{v}_{max}$ can never be the same, since by $v \neq 0$ it follows that q from (F.36) can never be zero. Thus, the term under the root in (F.38) is never equal to zero and therefore $\dot{v}_{min} \neq \dot{v}_{max}$.

F.4 Calculation of $I_{b,spec}$

For the cases where either $\kappa = 0$ or $\kappa' = 0$ the conditions for t_b and the lateral acceleration are not covered by previous considerations on I_b. The following paragraphs construct the domain $I_{b,spec}$ of admissible $\dot{v}$ for this cases.

1. $\kappa \neq 0 \wedge \kappa' = 0$:
   In this case $\ddot{y}(t_b) = (v + \dot{v}\,t_b)^2\,\kappa$, with $t_b = -\frac{v}{\dot{v}}$. Thus, $\ddot{y}(t_b) = 0$ and therefore $|\ddot{y}(t_b)| \leq \ddot{y}_{max}$ for all possible values of $\dot{v}$. It follows, that

$$I_{b,spec} \equiv I_0. \tag{F.47}$$

2. $\kappa = 0 \wedge \kappa' \neq 0$:
   Here, the lateral acceleration at t_b becomes:

$$\ddot{y}(t_b) = -\frac{2}{27}\frac{\kappa'\,v^3\,(2\,v + \dot{v}\,t_b)}{\dot{v}}. \tag{F.48}$$

   The intersections with $\pm\,\ddot{y}_{max}$ are therefore

$$\dot{v}_+ = -\frac{4\,\kappa'\,v^4}{2\,\kappa'\,v^3 + 27\,\ddot{y}_{max}} \quad \text{and} \quad \dot{v}_- = -\frac{4\,\kappa'\,v^4}{2\,\kappa'\,v^3 - 27\,\ddot{y}_{max}}. \tag{F.49}$$

   The monotony of the function, respectively the sign of the derivative, only depends on κ' since

$$\frac{\partial}{\partial\,\dot{v}}\,\ddot{y}(t_{t_b}) = \frac{4}{27}\,\frac{\kappa'\,v^4}{\dot{v}^2}. \tag{F.50}$$

   If there was no pole at $\dot{v} = 0$ (see (F.48)) this monotony would allow to say that $\ddot{y}(t_b) \geq \ddot{y}_{max}$ is equivalent to $\kappa' > 0 \wedge \dot{v} \geq \dot{v}_+$. Fortunately this function is injective, i.e.

$$\ddot{y}(t_b)|_{\dot{v}=\dot{v}_1} = \ddot{y}(t_b)|_{\dot{v}=\dot{v}_2} \quad \Leftrightarrow \quad \dot{v}_1 = \dot{v}_2. \tag{F.51}$$

   This allows a couple of statements:

   (a) $\kappa' > 0$: It follows that $\dot{v}_+ < 0$. Thus one gets

$$\ddot{y}(t_b) \leq \ddot{y}_{max} \quad \forall \quad \dot{v} \notin (\dot{v}_+,\, 0], \tag{F.52}$$
$$\ddot{y}(t_b) \geq -\ddot{y}_{max} \quad \forall \quad \dot{v}_- < 0 \wedge \dot{v} \in [\dot{v}_-,\, 0), \tag{F.53}$$
$$\ddot{y}(t_b) \geq -\ddot{y}_{max} \quad \forall \quad \dot{v}_- > 0 \wedge \dot{v} \notin [0,\, \dot{v}_-). \tag{F.54}$$

   By assuming $\kappa' > 0$, it follows that $\ddot{y}(t_b)$ is increasing with respect to $\dot{v}$. Combining this with the fact that $\ddot{y}(t_b)|_{\dot{v}=\dot{v}_-} = -\ddot{y}_{max} < 0 < \ddot{y}_{max} = \ddot{y}(t_b)|_{\dot{v}=\dot{v}_+}$ makes it clear that in this case $\dot{v}_- < \dot{v}_+$. The former statements can now be transformed into statements about $\dot{\delta}(t_b)$ lying in $[-\ddot{y}_{max},\, \ddot{y}_{max}]$.

$$\dot{v} \in [\dot{v}_-,\, \dot{v}_+] \quad \forall \quad \dot{v}_- < 0, \tag{F.55}$$
$$\dot{v} \notin (\dot{v}_+,\, \dot{v}_-) \quad \forall \quad \dot{v}_- > 0. \tag{F.56}$$

   (b) $\kappa' < 0$: It follows that $\dot{v}_- < 0$. Thus, it follows

$$\ddot{y}(t_b) \leq \ddot{y}_{max} \quad \forall \quad \dot{v}_+ > 0 \wedge \dot{v} \notin [0, \dot{v}_+), \tag{F.57}$$
$$\ddot{y}(t_b) \leq \ddot{y}_{max} \quad \forall \quad \dot{v}_+ < 0 \wedge \dot{v} \in [\dot{v}_+,\, 0), \tag{F.58}$$
$$\ddot{y}(t_b) \geq -\ddot{y}_{max} \quad \forall \quad \dot{v} \notin (\dot{v}_-,\, 0]. \tag{F.59}$$

Respective to the above case, it can be concluded from $\kappa' > 0$ that $\dot{v}_- > \dot{v}_+$. It is now possible to make statements about $\dot{\delta}(t_b)$ lying in $[-\ddot{y}_{max},\, \ddot{y}_{max}]$.

$$\dot{v} \in [\dot{v}_+,\, \dot{v}_-] \quad \forall \quad \dot{v}_+ < 0, \tag{F.60}$$

$$\dot{v} \notin (\dot{v}_+,\, \dot{v}_-) \quad \forall \quad \dot{v}_+ > 0. \tag{F.61}$$

Note that $\dot{v}_+$ as well as $\dot{v}_-$ can never be zero since it was assumed that $\kappa' \neq 0$. The interval $I_{b,spec}$ can be defined by collecting the different cases.

$$I_{b,spec} \quad \equiv \quad \begin{cases} [\dot{v}_-,\, \dot{v}_+] & \forall \quad \dot{v}_- < 0 \wedge \kappa' > 0 \\ (\dot{v}_+,\, \dot{v}_-)^c & \forall \quad \dot{v}_- > 0 \wedge \kappa' > 0 \\ [\dot{v}_+,\, \dot{v}_-] & \forall \quad \dot{v}_+ < 0 \wedge \kappa' < 0 \\ (\dot{v}_+,\, \dot{v}_-)^c & \forall \quad \dot{v}_+ > 0 \wedge \kappa' < 0 \end{cases} . \tag{F.62}$$

Index

References

[Allgöwer and Zheng, 2000] Allgöwer, F. and Zheng, A. (2000). *Nonlinear Predictive Control.* Birkhäuser.

[Bakker et al., 1987] Bakker, E., Nyborg, L., and Pacejka, H. (1987). Tyre modelling for use in vehicle dynamics studies. *SAE Paper 870421.*

[Black, 1977] Black, H. (1977). Inventing the negative feedback amplifier. *IEEE Spectrum*, pages 55–60.

[Bode, 1964] Bode, H. (1964). Feedback, the history of an idea. In *Selected Papers on Mathematical Trends in Control Theory*, pages 106–123. Dover, New York.

[Borenstein and Koren, 1991] Borenstein, J. and Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3).

[Brent, 1973] Brent, R. (1973). *Algorithms for Minimization without Derivatives.* PrenticeHall.

[Buss et al., 2003] Buss, M., Hardt, M., Kiener, J., Sobotka, M., Stelzer, M., von Stryk, O., and Wollherr, D. (2003). Towards an autonomous, humanoid, and dynamically walking robot: Modelling, optimal trajectory planning, hardware architecture, and experiments. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, Karlsruhe, Germany.

[Chucholowski et al., 1999] Chucholowski, C., Vögel, M., Stryk, O., and Wolter, T.-M. (1999). Real time simulation and online control for virtual test drives of cars. *Scientificand Engineering Computing*, 8:157–166.

[Close and Frederick, 1993] Close, C. and Frederick, D. (1993). *Modeling and Analysis of Dynamic Systems.* Houghton Mifflin, Boston, 2 edition.

[DeBoor, 1978] DeBoor, C. (1978). *A Practical Guide to Splines.* Springer Verlag.

[Donges, 1978] Donges, E. (1978). A two level model of driver steering behavior. *Human Factors*, 20:151–165.

[Dorf, 1988] Dorf, R. (1988). *Encyclopedia of Robotics*. John Wiley & Sons, NewYork.

[Feraric, 1999] Feraric, J.-P. (1999). *Echtzeitfähige Modellierung des individuellen Fahrverhaltens zur Realisierung adaptiver Unterstützungsfunktionen in einem Monitor Warnsystem*. PhD thesis, University of German Armed Forces, Munich.

[Forbes et al., 1995] Forbes, J., Huang, T., and Kanzawa, K. (1995). The bat mobile: Towards a bayesian automated taxi. In *Proc. of the 14th International Conference on Artificial Intelligence*, Montreal, Canada.

[Fraichard, 1991] Fraichard, T. (1991). Smooth trajectory planning for a car in a structured world. In *Proc. of the 1991 IEEE Int. Conf. on R&A*, pages 318–323, Sacramento, CA.

[Freund, 1982] Freund, E. (1982). Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators. *International Journal of Robotics Research*, 1(6).

[Freund and Mayr, 1989] Freund, E. and Mayr, R. (1989). A control system for automatically guided vehicles. In *Proceedings of the 20th Conference ISATA*, Florenz, Italy.

[Frezza et al., 1998] Frezza, R., Picci, G., and Soatto, S. (1998). A LaGrangian formulation of nonholonomic path following. In *Conference of Vision and Control*.

[Gabor, 1946] Gabor, D. (1946). Theory of communication. *J. Inst. Elect. Eng.*, pages 429–457.

[Godthelp and Konings, 1981] Godthelp, J. and Konings, H. (1981). Levels of steering control; some notes on the time-to-line crossing concept as related to driving strategy. In *Proc. of the Third European Annual Conference on Human Decsion Making and Manual Control*, pages 389–409, Delft, The Netherlands. Technical University.

[Graham and Lathrop, 1953] Graham, D. and Lathrop, R. (1953). The synthesis of optimum response: Criteria and standard forms. *In Transactions of the AIEE*, pages 273–288.

[Grashey, 1999] Grashey, S. (1999). *Ein Klassifikationsansatz zur Fertigkeits-basierten Verhaltensmodellierung*. PhD thesis, University of German Armed Forces, Munich.

[Greenwood, 1988] Greenwood, D. (1988). *Principles of Dynamics*. Prentice Hall, 2 edition.

[Grupen and Coelho, 2000] Grupen, R. and Coelho, C. (2000). Structure and growth: A model of development for grasping with robot hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2000*, Takamatsu, Japan.

[Guo and Guan, 1993] Guo, K. and Guan, H. (1993). Modelling of driver/vehicle directional dontrol systems. *Vehicle System Dynamics*, pages 141–184. 22.

[Hanski and Cambefort, 1991] Hanski, I. and Cambefort, Y. (1991). *Dung Beetle Ecology.* Princeton University Press, Princeton, NewJersey.

[Haykin, 1994] Haykin, S. (1994). *Neural Networks - A Comprehensive Foundation.* MacMillan and IEE Press.

[Hilborn, 1993] Hilborn, R. (1993). *Chaos and Nonlinear Dynamics. An Introduction for Scientists and Engineers.* Oxford University Press.

[Holve et al., 1995] Holve, R., Protzel, P., Bernasch, J., and Naab, K. (1995). Adaptive fuzzy control for driver assistance in car-following. In *Proc. of the 3rd European Congresson Intelligent Techniques and Soft Computing*, pages 1149–1153.

[Hubel, 1979] Hubel, D.and Wiesel, T. (1979). Brain mechanisms of vision. *Scientific American.* 241 (3).

[Jürgensohn, 1997] Jürgensohn, T. (1997). *Hybride Fahrermodelle.* PhD thesis, ZMMS, TU-Berlin.

[Kadiyala, 1993] Kadiyala, R. (1993). A toolbox for approximate linearization of nonlinear systems. *IEEEControlSystems*, pages 47–56.

[Kageyama and Pacejka, 1991] Kageyama, I. and Pacejka, H. (1991). On a new driver model with fuzzy control. *Vehicle System Dynamics*, pages 314–324.

[Kalman, 1960] Kalman, R. (1960). Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana*, 9(42–61):102–109.

[Keeney and Raiffa, 1976] Keeney, R. and Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoff.* John Wiley and Sons, New York.

[Khalil et al., 1996] Khalil, H., Teel, A., Georgiou, T., Praly, L., and Sontag, E. (1996). Lyapunov stability. In *IEEE Handbook of Control*, pages 889–895. IEEE Press.

[King et al., 2002] King, R., Leifheit, J., and Freyer, S. (2002). Automatic identification of mathematical models of chemical and biochemical reaction systems. In *CHISA 2002*, Prag, Czechoslovakia.

[Koga et al., 1994] Koga, Y., Kondo, K., Kuffner, J., and Latombe, J.-C. (1994). Planning motions with intentions. In *Computer Graphics 28th Annual Conference*.

[Kondo and Ajimine, 1968] Kondo, M. and Ajimine, A. (1968). Drivers sight point and dynamics of the driver-vehicle-system related to it. *SAE Technical Paper Series*, pages 1–14. 680104.

[Kopf, 1992] Kopf, M. (1992). *Ein Beitrag zur Modellbasierten, adaptiven Fahrerunterstützung für das Fahren auf deutschen Autobahnen*. PhD thesis, University of Armed Forces, Munich.

[Kramer and Rohr, 1982] Kramer, U. and Rohr, G. (1982). A fuzzy model of driver behaviour: Computer simulation and experimental results. In *Proceedings of the IFAC Congress and Analysis*, pages 31–35, Baden-Baden, Germany. Design and Evaluation of Man-Machine-Systems.

[Lawson and Hanson, 1974] Lawson, C. and Hanson, R. (1974). *Solving Least Square Problems*. Prentice-Hall, N.J.

[LeCun et al., 1995] LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., and Vapnik, V. (1995). Comparison of learning algorithms for handwritten digit recognition. In *International Conference on Artificial Neural Networks*.

[Mandelbrot, 1988] Mandelbrot, B. (1988). *The Fractal Geometry of Nature*. W.H. Freeman & Co.

[Mason and Salisbury, 1985] Mason, M. and Salisbury, J. (1985). *Robot Hands and the Mechanics of Manipulation*. MIT Press.

[Maxwell, 1868] Maxwell, J. (1868). On governors. In *In Proceedings of the Royal Society of London, 16, 1868. Selected Papers on Mathematical Trends in Control Theory*, pages 270–283. Dover, New York.

[Mayr, 1970] Mayr, O. (1970). *The Origins of Feedback Control*. MIT Press, Cambridge, Mass.

[Mayr, 1971] Mayr, O. (1971). *Feedback Mechanisms in the Historical Colections of the National Museum of History and Technology*. Smithsonian Institution Press, Washington, DC.

[Mayr, 1991] Mayr, R. (1991). *Varfahren zur Bahnfolgeregelung für ein automatisch gefürtes Fahrzeug*. PhD thesis, TU-Dortmund.

[McRuer et al., 1973] McRuer, D., Allen, R., Weir, D., and Klein, H. (1973). New results in driver steering control models. *HumanFactors*, 19:381–397.

[McRuer and Krendl, 1974] McRuer, D. and Krendl, E. (1974). Mathematical models of human pilot behavior. Technical report, NATO, AGARDO-graph. Technical Report (188).

[Menzel and D'Aluiso, 2000] Menzel, P. and D'Aluiso, F. (2000). *Robo Sapiens, Evolution of a New Species.* MIT Press.

[Modjtahedzadeh and Hess, 1993] Modjtahedzadeh, A. and Hess, E. (1993). A model of driver steering dynamics for use in assessing vehicle handling qualities. *Journal of Dynamic Systems Measurement and Control*, pages 456–464. Transactions from ASME (115).

[Murray and Sastry, 1993] Murray, R. and Sastry, S. (1993). Non-holonomic motion planning: Steering using sinusoids. *IEEE T. Automatic Control*, pages 700–716.

[Newton et al., 1957] Newton, G., Gould, L., and Kaiser, J. (1957). *Analytical Design of Linear Feedback Control.* John Wiley & Sons, New York.

[Patel and Read, 1996] Patel, J. and Read, C. (1996). *Handbook of the Normal Distribution.* Dekker Verlag, 2 edition.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, Los Altos, California.

[Pearl, 1999] Pearl, J. (1999). *Causality: Models Reasoning and Inference.* Cambridge University Press.

[Pomerleau, 1993] Pomerleau, D. (1993). *Neural Network Perception for Mobile Robot Guidance.* Kluwer Academic Publisher.

[Post et al., 1997] Post, J., Burke, R., Drexel, M., and Robertson, J. (1997). An adaptive control model for lateral path following with closed loop handling simulations. *SAE Technical Paper Series*, pages 217–231. 971061.

[Press et al., 1992] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipies in C.* Cambridge Press, 2 edition.

[Protzel et al., 1993] Protzel, P., Holve, R., Bernasch, J., and Naab, K. (1993). Fuzzy distance control for intelligent vehicle guidance. In *Proc. of the Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS.*

[Qin and Badgewell, 1997] Qin, S. and Badgewell, T. (1997). An overview of industrial model predictive control technology.

[Ramsey, 1931] Ramsey, F. (1931). *Truth and Propability.* Harcourt Brace Jovanovich, New York, braithwaite, r.b. edition. The Foundations of Mathematics and Other Logical Essays.

[Rawlings et al., 1994] Rawlings, J., Meadows, E., and Muske, K. (1994). Nonlinear model predictive control: A tutorial and survey. In *In ADCHEM'94*, pages 185–197, Kyoto, Japan.

[Russel and Norvig, 1995] Russel, S. and Norvig, P. (1995). *Artificial Intelligence.* Prentice Hall.

[Sarachik, 1997] Sarachik, P. (1997). *Principles of Linear Systems.* Cambridge University Press, NewYork.

[Sarkar et al., 1994] Sarkar, N., Yun, X., and Kumar, V. (1994). Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots. *International Journal of Robotics Research.*

[Schaefer et al., 2000] Schaefer, F., Ramsey, M., Haque, and Schuller, J. (2000). Development of target point search methods for course following systems: Treating vehicle guidance. *Advanced Vehicle Technologies*, pages 47–59. ASME Conference 2000 Orlando, Florida.

[Schuller et al., 1999] Schuller, J., Schaefer, F., Neukum, A., and Krueger, H. (1999). *A Driver Model as a Development Tool for Vehicle Handling Design.* IAVSD Conference, South Africa.

[Siegelmann and Sontag, 1994] Siegelmann, H. and Sontag, E. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 2(131):331–260.

[Svestka and Overmars, 1995] Svestka, P. and Overmars, M. (1995). Motion planning for car-like robots using a probabilistic learning approach. *Journal of Robotics Research*, 8:119–143.

[Ueno et al., 1990] Ueno, H., Xu, Y., and Brown, H. (1990). On control and planning of a spacestation robot walker. In *Proc. of IEEE International Conference on Systems Engineering*, Pittsburgh, PA.

[van der Torre and Weydert, 1998] van der Torre, L. and Weydert, E. (1998). Goals, desires, utilities and preferences. In *Proceedings of the ECAI-98 Workshop: Decision Theory meets Artificial Intelligence - qualitative and quantitative approaches.*, Brighton.

[Voegel, 1997] Voegel, M. (1997). Fahrbahnmodellierung und Kursreglung. Technical report, Mathematisches Institut, TU-Munich.

[Vyshnegradskii, 1877] Vyshnegradskii, I. (1877). On controllers of directaction. Technical report, Izv. SPBTekhnolog. Inst.

[Weir and McRuer, 1973] Weir, D. and McRuer, D. (1973). Measurement and interpretation of driver steering behavior and performance. *Human Factors*, pages 367–378. 15.

[Wellman and Doyle, 1991] Wellman, M. and Doyle, J. (1991). Preferential semantics for goals. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91).*

[Wells, 1967] Wells, D. (1967). *Lagrangian Dynamics (Schaum's Outline)*. McGraw-Hill.

[Willems and Polderman, 1997] Willems, J. and Polderman, J. (1997). *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Texts in Applied Mathematics, Band 26. Springer.

[Xi, 1993] Xi, G.andQun, Y. (1993). Driver-vehicle-environment closed-loop simulation of handling and stability using fuzzy control theory. *Vehicle System Dynamics*, pages 172–183. 23.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, pages 338–353. 8.