UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4: Informatik

Universidad de Cádiz
E.S.I.

Entwicklung eines Regelungsverfahrens zur Pfadverfolgung für ein Modellfahrzeug mit Sattelanhänger

Masterarbeit

zur Erlangung des Grades eines Master of Science
im Studiengang Informatik

vorgelegt von

Francisco José Cortés Ramos

Betreuer:	Prof. Dr. Dieter Zöbel. Institut für Softwaretechnik. Fachbereich Informatik.
	Dipl.-Inform. Christian Weyand. Institut für Softwaretechnik. Fachbereich Informatik.
Erstgutachter:	Prof. Dr. Dieter Zöbel. Institut für Softwaretechnik. Fachbereich Informatik.
Zweitgutachter:	Dipl.-Inform. Christian Weyand. Institut für Softwaretechnik. Fachbereich Informatik.

Koblenz, im September 2009

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

. . . .

(Ort, Datum) (Unterschrift)

Zusammenfassung

Neben der fortschreitenden Automatisierung im innerbetrieblichen Warenverkehr ist auch die Automatisierung in ausgewählten Bereichen des ausserbetrieblichen Waren- und Güterverkehrs erstrebenswert. Durch den Einsatz von fahrerlosen Lkw-Gespannen auf Speditionshöfen kann die ökonomische Effizienz, der dort anfallenden Abläufe, erheblich erhört werden. Insbesondere werden dazu präzise Regelungsverfahren benötig, die auch für Sattelzüge ein exaktes Abfahren vorgegebener Wege gewährleisten. Das allgemeine Ziel dieser Arbeit ist die Adaption und Evaluation eines Regelverfahrens zur Pfadverfolgung für Sattelzuggespanne. Die Unterschiede im kinematischen Verhalten zwischen LKW mit einem einachsigen Starrdeichselanhänger und Sattelzügen herausgearbeitet werden. Im Weiteren werden die charakteristischen kinematischen Eigenschaften von Sattelzügen bei der Adaption eines Regelverfahrens berücksichtigt, das zunächst speziell für Fahrzeuge mit Starrdeichselanhänger konzipiert wurde. Das Regelungsverfahren zur Pfadverfolgung muss sowohl für vorwärts als auch rückwärtsgerichtete Fahrmanöver geeignet sein. Das Regelungsverfahren wird als abgeschlossene Komponente in die Steuersoftware eines Modellfahrzeugs integriert. Dazu werde für die Geometrie des Modellfahrzeugs spezifische mit dem Ziel, Grenzen möglicher Regelabweichungen zu bestimmen. Die Arbeit dokumentiert darüber hinaus die zentralen Softwarekomponenten des implementierten Regelungsverfahrens.

Abstract

Besides the progressive automation of internal goods traffic, there is an important area that should also be considered. This area is the carriage of goods in selected external areas. The use of driverless trucks in logistic centers can report economic efficiency. In particular, these precise control procedures require that trucks drive on predetermined paths. The general aim of this work is the adaption and evaluation of a path following control method for articulated vehicles. The differences in the kinematic behavior between trucks with one-axle trailer and semi-trailer vehicles will be emphasized. Additionally, the characteristic kinematic properties of semi-trailers for the adaptation of a control procedure will be considered. This control procedure was initially designed for trucks with one-axle trailer. It must work in forwards and backwards movements. This control process will be integrated as a closed component on the control software of the model vehicle. Thus, the geometry of the model vehicle will be specified, and the possible special cases of the control process will be discovered. The work also documents the most relevant software components of the implemented control process.

Preface

I'm a Spanish student from Cádiz, and with this Master's thesis I finish my higher education. I've studied Software Engineer at the University of Cádiz. This degree has a length of five years and it allows you to do a PhD thesis. After completion, I had the opportunity to complete my education with a double degree between my university and the University Koblenz-Landau in Germany.

This double degree allows you to obtain the higher education degree of your partner country. In my case, the Master of Science im Sudengang Informatik degree. This agreement is to realize a one year stay in the partner country. During this year, the student musts course some subjects and to realize the final thesis of his studies degree. The subjects provide to the student a better integration with the foreign university environment. That improves his knowledge of the customs and of the language.

In my case, during the first six months, I've studied subjects related to the Object-Oriented Programming and Automotive Systems in the Automation. This latter subject helped me to realize this Master's thesis. It was an introduction in detail of the ideas of this specific research area. During the next six months, I've been totally dedicated to this Master's thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis has been developed in the working group Real-time Systems at the Institute of Software Technology of the University Koblenz-Landau. This working group is concerned with a wide range of topics within the field of real-time systems and mobile systems. In the latter case, the working group concentrates on the development of new concepts in the fields of assisted driving and autonomous driving. Due to the increased challenge, especially articulated vehicles, e.g. commercial trucks with different types of trailer, are considered in this research area.

This thesis is developed in the context of autonomous driving. It will be another step to achieve autonomous driving, in this case for semi-trailer model vehicles, under a controlled environment. This step will consist in providing a path following system to a given path, following it in forwards or backwards motion.

1.1 Motivation

Nowadays, the autonomous driving research can be found in car manufacturers or in transportation companies. In the case of the car manufacturers, most effort is related to accomplish a concept of autonomous or semi autonomous parking, for commercial vehicles and passengers cars as well. In the case of logistics companies, all those efforts are related to afford autonomous driving or parking of trucks inside a logistic center area.

In relation to the particular basic conditions, different approaches are chosen to provide autonomous driving. One approach to driven in outdoor conditions can be the path recognition using a camera and a special recognition software. Basically the vehicles follow a line drawn on the ground. This system requires fixed infrastructure and this restriction causes a limited movement of vehicles along the prefixed paths. Another example could be a road with transponders and a vehicle that recognizes the position of them. This vehicle drives along those transponders. This case is also using fixed infrastructure, and an improvement of it supposes a creation of new paths adding transponders on the road.

Another idea is to use other techniques to determine the position of the vehicle, regarding to some position marks, using a laser scanner and a triangulation algorithm. This idea is applied in the laboratory of the working group and is only adequate to indoor conditions. This solution frees the vehicle of the restrictions of driving only along predefined paths. In contrast, it is necessary to plan a path for every movement of the vehicle and keep this vehicle on that path. So, it is needed a path planning and a path following algorithm.

Once the automation of a logistic center is feasible, most common maneuvers, e.g. parking in front of loading ramp, movements from one point to another one or combinations of them can be done driverless. The potential benefits of that situation are very important. It is possible to think in a logistic center, where all movements of the vehicles are planned and observed centrally. There is no need of drivers, only people to look after the vehicles inside the autonomous area. That means more efficiency, and of course less cost. Besides methods for planning and guaranteeing collision-free paths, a precise control algorithm is required to provide autonomous driving.

In a previous thesis [Schwarz(2009)], a control algorithm for trucks with a one-axle trailer was composed, implemented and evaluated. His work was to describe the kinematic properties of trucks with a one-axle trailer, as well as to create a path following algorithm for this vehicle. That algorithm provides forwards and backwards motion to the vehicle and it was integrated in an existing software. The results were tested using a model of a truck with a one-axle trailer. The objective of this thesis is to extend that previous work to another common type of truck, a semi-trailer.

Figure 1.1: Photo of the semi-trailer model vehicle.

1.2 Objective of the Thesis

As already mentioned in the previous section, this work will adapt the path following algorithm for a truck with a one-axle trailer to the related vehicle type called semi-trailer. This adaption is necessary. The initial algorithm can't be used directly for semi-trailers, because the kinematic properties of both vehicle types are different. Thereby, the formulas to achieve the correct values of the path following are not the same in all cases. The semi-trailer vehicle has got more possible states of their trailer.

The main tasks of this thesis are, first of all, to describe the kinematic properties of semi-trailer vehicles, representing the possible states and the formulas that describe their movements. The kinematic differences between both types of vehicles will be described and it will be explained in which way those differences are handled by the path following algorithm adaption.

Once all properties are described, the theory of the path following algorithm will be explained. Later it will be adapted and integrated on the actual software. This software provides not only the path following of a modeling vehicle, but also manual driving and more applications to drive modeling vehicles. The different elements of the modeling vehicle will be calibrated and the optimal values for some path following parameters will be calculated. Then, the algorithm will be implemented in the semi-trailer modeling vehicle, for testing the quality of this algorithm.

All the components of the truck and the actual software for modeling vehicles driving will be described. The errors of the path following process will be analyzed, to determinate the maximum bound of deviation from a given path. The errors will be measured doing several tests with the modeling truck.

In the end, the results of the adaption will be explained, and the conclusions of the whole work will be given, describing the differences in contrast to the previous work. It will be explained in which way those differences affect to the final result.

1.3 Structure of the Thesis

First, the basic theory of this work will be explained. A mathematical model which describes the kinematic behavior of articulated vehicles will be introduced in this context. Certain kinematic concepts which are relevant to the path following problem, e.g. the so called stable motion, will be considered in detail. Furthermore, the essential differences between semi-trailers and trucks with a one-axle trailer will be discussed. Continuing with the theory, in the second chapter, the path following method will be described.

The software environment of the work will be explained, distinguishing the previous existing software and the new developed software. In the next step, the model vehicle used to achieve the previous designed processes will be introduced. The hardware of that vehicle will be described in detail, specifying all its components and explaining the required calibration processes.

Then, the estimation of the real error provided by the model vehicle will be commented. To accomplish it, a lot of experiments have to be executed, to determine the optimal values of some configuration parameters. Finally, the path following algorithm is evaluated within a series of experiments. Thereby, the quality of the control algorithm should be analyzed and an upper bound of the estimated error should be determined for the model vehicle.

In the end, the conclusions will be commented, discussing the initial ideas and results. Additionally, some videos of the real working of the model vehicle, parking autonomously and path following motion, will be recorded and published on internet. Some ideas for future related works will be also commented.

Chapter 2

Basics

In this section, the modeling of articulated vehicles and their kinematical behavior will be described. The stable motion will also be described. In this case, the vehicle moves on constant circle arcs around a common circle. The parameters that affect the kinematical behavior of a vehicle will also be explained.

2.1 Kinematics and Modeling

Here the kinematics of the general-2-trailer vehicles is described. The parameters to describe the state of the complete vehicle will be defined. There are special articles which are very useful to describe the kinematics of vehicles, [Zöbel & Weyand(2008)] and [Altafini(2001)].

2.1.1 Description of an Articulated Vehicle

The general-n-trailer consists of a front vehicle with two axles and $n-1$ one-axle trailers, where n can be an arbitrary number. The configuration of the system is given by two position coordinates, one steering angle and n bending angles.

This configuration can be changed by two input parameters, the velocity and the steering angle.

The normal case of an n-trailer is a pulling vehicle part with one steering axle, and n trailers, with one single axle and two parallel wheels. Every trailer is hitched to the preceding one in a determinate point. The position of that hitch determines the type of the n-trailer system:

Standard-n-trailer: The systems of this group have got the hitch of every connection between trailers (and of course the connection of the first trailer with the pulling vehicle) in the middle of the axle of the preceding vehicle.

General-n-trailer: One or more trailers present off-hitching.

Both types of articulated vehicles are characterized by no-holonomic behavior. In practice, the standard-n-trailer is not a realistic configuration, due to variation of the hitch for a better performance of the truck movements. Most of the trailers we see every day are general-n-trailers.

2.1.2 The Bike Model

The bike model is a simplified model to describe the kinematical behavior, i.e. describes the movement of the vehicle depending on its actual configuration and steering angle. This model reduces the left and right wheel into a single wheel at the center of the front and rear axles. The wheels of every axle are assumed to have no lateral slip. In case of the front vehicle of an articulated vehicle, only the front wheels are steerable. The steering angle is given by the average value of the steering angles of the inside and outside front wheels. In this model, every axle of the articulated vehicle is a vehicle part.

The bike model for the specific truck of this thesis is shown in Fig. (2.1) .

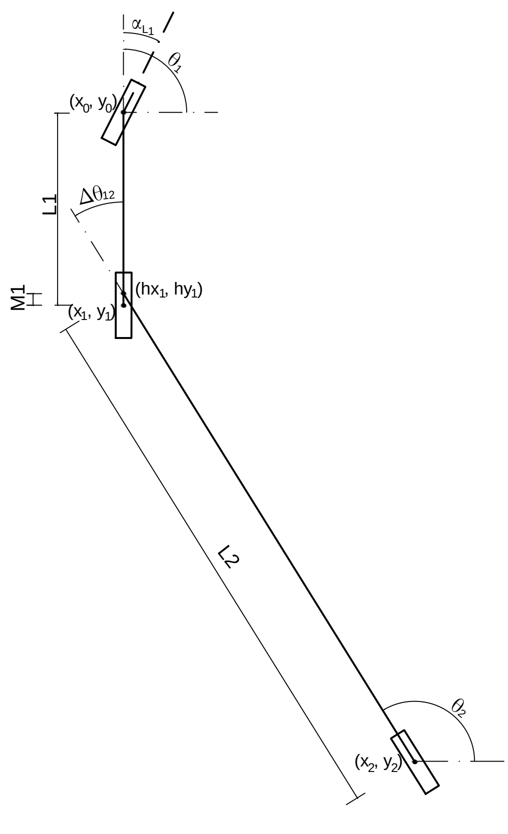


Figure 2.1: Bike model of the model vehicle.

Now every parameter of the bike model will be described:

- (x_i, y_i) describes the position of the axle midpoint of the i-th vehicle part of the articulated vehicle. As it is already mentioned, every axle of the articulated vehicle is assumed as a vehicle part. For a general-n-trailer there are $n + 1$ positions to describe.
- (hx_i, hy_i) describes the position of the hitch of the vehicle i.
- α_{L_1} represents the steering angle of the truck.
- $\Delta\theta_{i(i+1)}$ represents the bending angle between the vehicle i, and the next vehicle $i + 1$.
- L_i describes the length between the axle of the vehicle part i and the previous hitch. This length will be positive if the hitch is in front of the axle.
- M_i represents the length between the axle of the vehicle part i and the next hitch. This value will be positive if the hitch is behind the axle.

[Altafini(2001)] describes, with all those definitions of the parameters, the holonomic relations between two consecutive nodes i-th and i+1-th of a general-n-trailer:

$$x_{i+1} = x_i - L_{i+1} \cos\theta_{i+1} - M_i \cos\theta_i \tag{2.1}$$

$$y_{i+1} = y_i - L_{i+1} \sin\theta_{i+1} - M_i \sin\theta_i \tag{2.2}$$

Now, a new parameter v_i is introduced, it represents the speed of the middle point of the i-th vehicle part in the direction θ_i. The parameter v_i will be positive if the vehicle is driving forwards, and it will be negative if it is driving backwards. A formula to obtain a differential equation for the orientation angle θ_{i+1} is also defined. This formula calculates θ_{i+1} as a function of θ_i and the speed of the axle i, v_i, $i \in \{1, ..., n-1\}$:

$$\dot{\theta}_{i+1} = \frac{v_i \sin(\theta_i - \theta_{i+1})}{L_{i+1}} - \frac{M_i \cos(\theta_i - \theta_{i+1})\dot{\theta}_i}{L_{i+1}} \tag{2.3}$$

The next formula calculates v_{i+1} based on the speed of the previous axle v_i:

$$v_{i+1} = v_i \cos(\theta_i - \theta_{i+1}) + M_i \sin(\theta_i - \theta_{i+1})\dot{\theta}_i \tag{2.4}$$

It is important to note that all previous formulas can also be applied for standard-n-trailers. In this case the parameter M_i is set to zero.

To describe a movement of an articulated vehicle, further parameters are required in this kinematic model:

- α_{L_i} is the steering angle of the articulated vehicle, and it is defined as $\theta_0 - \theta_1$.
- $\Delta\theta_{i(i+1)}$ is the bending angle between two vehicle parts, i and $i+1$. It is defined as $\theta_{i+1} - \theta_i$. If the articulated vehicle consists of n vehicle parts, there will be $n-1$ bending angles.

In [Altafini(2001)], the formulas to calculate the orientation θ_0 and speed v_0 of the first vehicle part in an articulated vehicle are shown:

$$\theta_0 = \frac{v_0 \sin \alpha_{L_1} + L_1 \alpha_{L_1}}{L_1 + M_0 \cos \alpha_{L_1}} \tag{2.5}$$

$$v_1 = v_0 \cos \alpha_{L_1} + M_0 \sin \alpha_{L_1} \theta_0 \tag{2.6}$$

$$v_0 = \frac{v_1 - M_0 \sin \alpha_{L_1} \theta_0}{\cos \alpha_{L_1}} \tag{2.7}$$

$$v_0 = \frac{v_1}{\cos \alpha_{L_1}} - \tan \alpha_{L_1} M_0 \theta_0 \tag{2.8}$$

For this semi-trailer, this equation can be simplified since $M_0 = 0$:

$$v_0 = \frac{v_1}{\cos \alpha_{L_1}} \tag{2.9}$$

Now the formulas to describe the cartesian coordinates of the middle point of the i-th axle are shown:

$$\dot{x}_n = v_n \cos \theta_n \tag{2.10}$$

$$\dot{y}_n = v_n \sin \theta_n \tag{2.11}$$

2.1.3 Kinematics of the Model Vehicles

In this section, the previous formulas will be applied for our model vehicles. The model vehicle of this thesis is a semi-trailer and belongs to the category of general-2-trailers. The truck has got two single axles, the first one is steerable and the other one is fixed. The trailer has got a double axle, which is located almost at its end, as is typical for an American version of a semi-trailer. This particular situation needs one more step in the bike model simplification. Therefore, we suppose that there is only one single and fixed axle. This virtual axle is located between both real trailer axles. This simplification of the model is not perfect, but it leads very good results in practice.

Here are the lengths of the different vehicle parts of the resulting semi-trailer model:

$$\begin{aligned} L_0 = M_0 = M_2 &= 0 \\ L_1 &= 215\,mm \\ M_1 &= -13\,mm \\ L_2 &= 615\,mm \end{aligned}$$

In [Schwarz(2009)] the following equations are included:

$$\dot{\theta}_1 = \frac{v_1 \tan \alpha_{L_1}}{L_1} \tag{2.12}$$

$$\dot{\theta}_2 = v_1 \left(\frac{-\sin \Delta\theta_{12}}{L_2} - \frac{M_1 \cos \Delta\theta_{12} \tan \alpha_{L_1}}{L_1 L_2} \right) \tag{2.13}$$

$$\dot{\Delta}\theta_{12} = \dot{\theta}_2 - \dot{\theta}_1 = v_1 \left(-\frac{\tan \alpha_{L_1}}{L_1} - \frac{\sin \Delta\theta_{12}}{L_2} - \frac{M_1 \cos \Delta\theta_{12} \tan \alpha_{L_1}}{L_1 L_2} \right) \tag{2.14}$$

In summary, the introduced formulas form the following system of differential equations which describe the motion of a general-2-trailer in relation to the actual configuration and given control values:

$$\dot{\vec{q}} = \begin{pmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta}_1 \\ \Delta\dot{\theta}_{12} \\ \dot{\alpha}_{L_1} \end{pmatrix} = \begin{pmatrix} \cos\theta_1 & 0 \\ \sin\theta_1 & 0 \\ \tan\alpha_{L_1}/L_1 & 0 \\ \frac{-\tan\alpha_{L_1}}{L_1} - \frac{\sin\Delta\theta_{12}}{L_2} - \frac{M_1\cos\Delta\theta_{12}\tan\alpha_{L_1}}{L_1 L_2} & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ \dot{\alpha}_{L_1} \end{pmatrix}$$

More properties and parameters are explained now:

- r_i defines the distance between the i-th vehicle part and the center of the circle on which the axle of the vehicle part currently moves.
- r_{k_i} represents the distance between the i-th hitch and the center of the circle on which the axle of the vehicle part currently moves.
- (mx_i, my_i) defines the midpoint of the circle of the i-th vehicle part on which the axle of the vehicle part currently moves.
- α_{L_1} is the steering angle. It's also equal to the angle, at the midpoint of the circle, which is on the opposite side of the longitudinal axis L_1 of the front

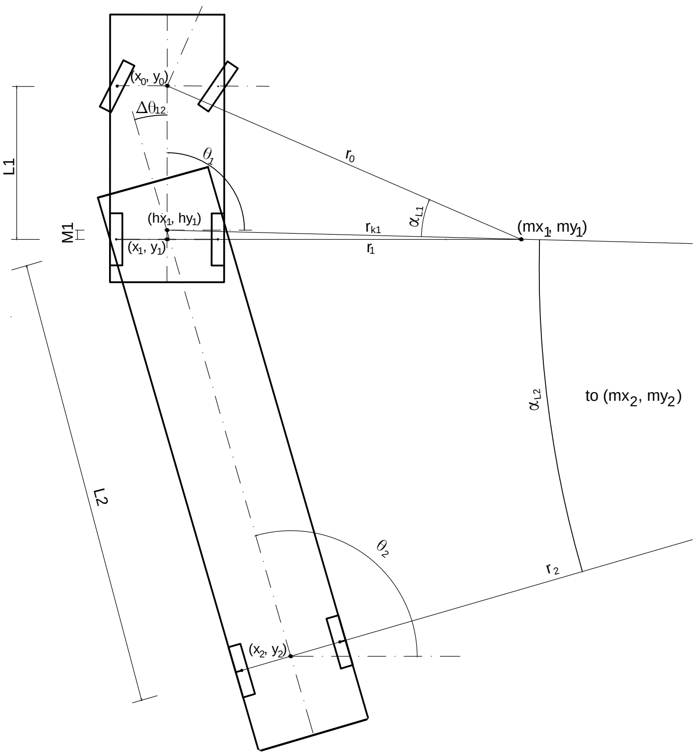


Figure 2.2: More properties of the model vehicle

vehicle.

- α_{M_1} describes the angle, at the midpoint of the circle, which is on the opposite side of the longitudinal axis M_1 of the front vehicle.

- α_{L_2}: describes the angle, at the midpoint of the circle, which is on the opposite side of the longitudinal axis L_2 of the front vehicle. .

2.2 Stable Motion

2.2.1 Concept

As it is shown in [Zöbel & Weyand(2008)], if a semi-trailer drives forwards with a fixed steering angle $\alpha_{L_1} \neq 0$, the bending angle $\Delta\theta_{12}$ will converge to a fixed value. Then, there is a relation between those two values that keeps the truck parts moving on constant circle arcs around a common center. This situation is called stable motion. It does not matter if the vehicle is driving forwards or backwards. This case is described by a series of relations between some of the parameters shown in previous sections. Fig (2.3) represents a possible stable motion of the semi-trailer model vehicle.

In this case, the midpoint of the circle that describes the trajectory of every vehicle part must be the same, so, there is only one midpoint (mx, my) (see Fig. 2.2 of the model vehicle in a non-stable motion, with two circle midpoints). The parameters shown in Fig (2.3) are α_{L_1} for the steering angle and $\Delta\theta_{12}$ for the bending angle. If the values of those parameters provide a stable motion, they are called $\alpha_{L_1,circ}$ and $\Delta\theta_{12,circ}$.

The formula to calculate $\alpha_{L_1,circ}$ for a given $\Delta\theta_{12,circ}$ is shown in [Zöbel & Weyand(2008)]:

$$\alpha_{L_1,circ}(\Delta\theta_{12,circ}) = -\arctan\frac{L_1\sin\Delta\theta_{12,circ}}{L_2 + M_1\cos\Delta\theta_{12,circ}} \tag{2.15}$$

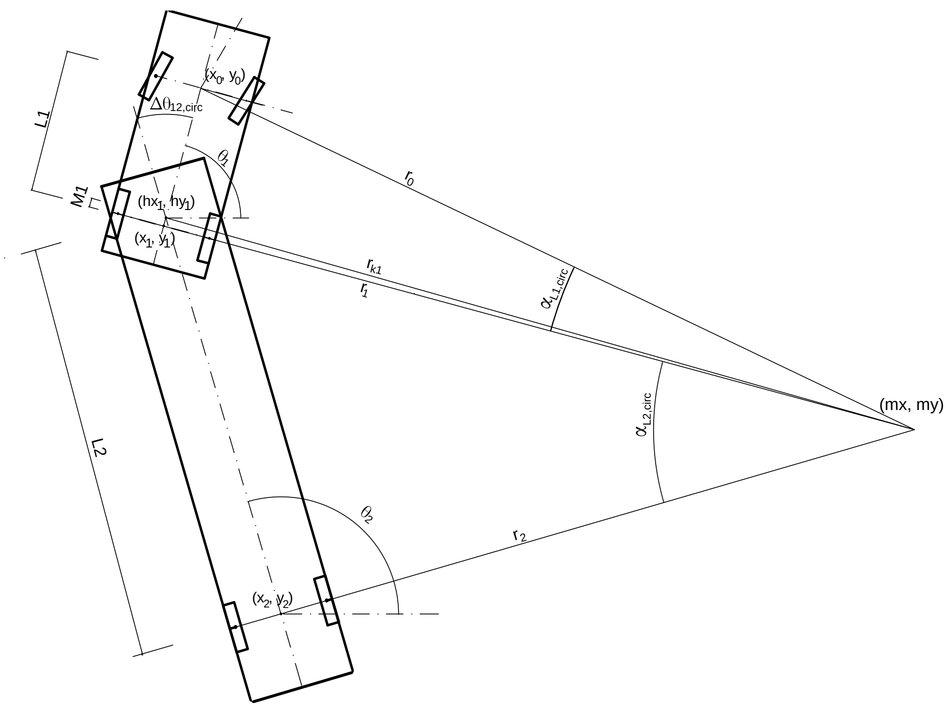


Figure 2.3: Stable motion of the model vehicle.

It is also explained that the previous formula cannot be inverted in a direct fashion, and a new formula for calculation of $\Delta\theta_{12,circ}$ is given:

$$\Delta\theta_{12,circ}(\alpha_{L_1,circ}) = \arcsin\left(\frac{L_2 \sin\left(\arctan\left(\frac{M_1}{L_1}\tan\alpha_{L_1,circ}\right)\right)}{M_1}\right) + \arctan\left(\frac{M_1}{L_1}\tan\alpha_{L_1,circ}\right) \tag{2.16}$$

Those equations describe the relations between $\alpha_{L_1,circ}$ and $\Delta\theta_{12,circ}$ for general-2-trailer vehicles. But those formulas are no valid for any value of $\alpha_{L_1,circ}$. So $\alpha_{L_1,circ}$ must be smaller than a limit value to achieve stable motion. This is explained the in next section.

2.2.2 Differences between Semi-Trailer and Truck with One-Axle Trailer in Stable Motion

According to [Zöbel & Weyand(2008)], the axle midpoints of the vehicle parts move on circular arcs with different radius and curvature around a common center. If r_1 is the radius which belongs to the axle of the tractor vehicle, r_2 is the radius of the trailer and r_{k1} is the radius of the coupling device, then the different values for the radius will decrease if the steering angle becomes bigger. Then, it is possible a situation of r_2 equal to zero, for a given value of the steering angle. That means the radius r_{k1} is equal to L_2. In this case, the whole articulated vehicle is moving around the axle midpoint of the trailer, and this midpoint is only rotating. In the previous subsection, the typical case of stable motion, which can be observed for every kind of truck with one-axle trailer, is explained. Now, a special case will be presented, for semi-trailer, and the relevant formulas will be shown. The following figures, Fig (2.4) and Fig. (2.5), are based on [Berg(2009)] and adapted for a semi-trailer. [1]

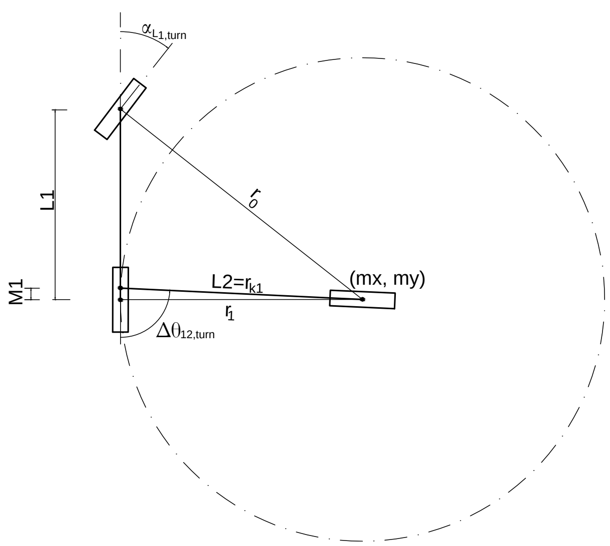


Figure 2.4: Bike model with $\alpha_{L_1} = \alpha_{L_1,turn}$

[1]The proportions in Fig. (2.2) and Fig (2.3) are based on the model vehicle. The next figures are modified in way that a shorter trailer is chosen.

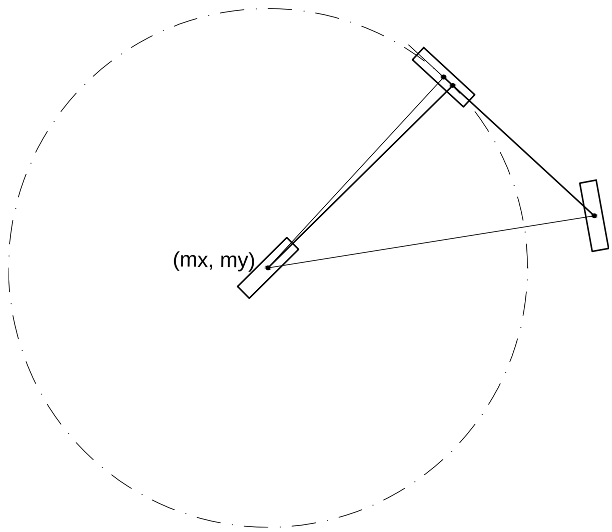


Figure 2.5: Rotational movement of the vehicle.

The value of α_{L_1} in this case is denoted as $\alpha_{L_1,turn}$, and $\Delta\theta_{12}$ is called $\Delta\theta_{12,turn}$. The angle of $\alpha_{L_1,turn}$ is the upper bound value to achieve a stable motion with a general-2-trailer. As it is shown in [Zöbel & Weyand(2008)] and explained in the previous section, driving forwards with $\alpha_{L_1} \neq 0$ leads to a convergence of $\Delta\theta_{12}$ with a fixed value, $\Delta\theta_{12,circ}(\alpha_{L_1})$. Now the new parameter is introduced in this mathematical relation, $\alpha_{L_1,turn}$. So if the value of $|\alpha_{L_1}| > |\alpha_{L_1,turn}|$ there is no possibility of stable motion, because the value of $\Delta\theta_{12}$ oscillates.

This rule can be used for any general-2-trailer. This means that it can also be applied to a truck with one-axle trailer as it were used in [Schwarz(2009)]. But in practice, this constraint is not relevant for the latter vehicle type because there are other restrictions that limit the number of possible states of the vehicle. In [Schwarz(2009)] it is explained that it is impossible to achieve the bending angle $\Delta\theta_{12,turn}$ because of the geometry of the truck and of the trailer. The width of the trailer will cause a collision between truck and trailer for a bending angle close to $90°$. In the case of an average truck with one-axle trailer vehicle the following bounds are found in [Zöbel & Weyand(2008)]:

$$0 \leq |\alpha_{L_1,circ}| \leq |\alpha_{L_1,max}| \leq |\alpha_{L_1,turn}| \tag{2.17}$$

In the case of our particular truck with one-axle trailer model, the next constraint is noticed:

$$|\alpha_{L_1,max}| < |\alpha_{L_1,turn}| \tag{2.18}$$

If we compare the motion of the vehicle parts of a semi trailer and a truck with one-axle trailer, then it will be shown that a steering angle bigger than $\alpha_{L_1,turn}$ can be achieved due to the relations between the lengths of the truck and of the trailer. The semi-trailer bears on the rear end of the truck. So the only possibility that a collision occurs between both vehicles is when the trailer touches the cabin of the truck. Now a bigger bending angle can be observed, and the following bounds are result of it:

$$0 \leq |\alpha_{L_1,circ}| \leq |\alpha_{L_1,max}| \tag{2.19}$$

$$0 \leq |\alpha_{L_1,turn}| \leq |\alpha_{L_1,max}| \tag{2.20}$$

So it is possible that the following situation occurs:

$$|\alpha_{L_1,turn}| \leq |\alpha_{L_1,circ}| \leq |\alpha_{L_1,max}| \tag{2.21}$$

For the case of our semi-trailer model vehicle, it was observed:

$$|\alpha_{L_1,turn}| < |\alpha_{L_1,max}| \tag{2.22}$$

Next, the formula to calculate $\alpha_{L_1,turn}$ is introduced. This new equation is consistent with the previous equations. The calculation of $\alpha_{L_1,turn}$ for any general-2-trailer vehicle only depends on the different lengths of the truck and of the trailer, and is shown in [Zöbel & Weyand(2008)]:

$$\alpha_{L_1,turn} = \arctan\left(\frac{L_1}{M_1}\tan\left(\arcsin\left(\frac{M_1}{L_2}\right)\right)\right) \tag{2.23}$$

In a case of a standard-2-trailer, the hitch is located at the midpoint of the second axle of the truck. The previous formula is not valid, because $M_1 = 0$. In [Berg(2009)] an alternate formula for this special case is presented:

$$\alpha_{L_1,turn} = \arctan\left(\frac{L_1}{L_2}\right) \tag{2.24}$$

The formula (2.24) is not relevant to our semi-trailer model, but it provides a solution for the addressed case.

Using the real lengths of our semi-trailer model, the value of $\alpha_{L_1,turn}$ is $19.27°$. The formula to calculate $\Delta\theta_{12,turn}$ for a given $\alpha_{L_1,turn}$ results in:

$$\Delta\theta_{12,turn} = \Delta\theta_{12,circ}(\alpha_{L_1,turn}) \tag{2.25}$$

So, using eq. (2.16), $\Delta\theta_{12,turn} = 87.64°$.

Particular case

Now, we examine the particular case in detail, based on the ideas and figures found in [Berg(2009)]. As it were mentioned before, there is no possible stable motion for $|\alpha_{L_1,circ}| > |\alpha_{L_1,turn}|$. But in the other case where $|\alpha_{L_1,circ}| < |\alpha_{L_1,turn}|$, it has to be noticed that there are two possible values of $\Delta\theta_{12,circ}$, one smaller than $|\Delta\theta_{12,turn}|$ and the other one bigger. With both values it is possible to drive in stable motion, Fig. (2.6) and Fig. (2.7):

The case of Fig (2.7) is curious, because the stable motion is possible with the truck driving forwards and the trailer driving backwards simultaneously, and vice

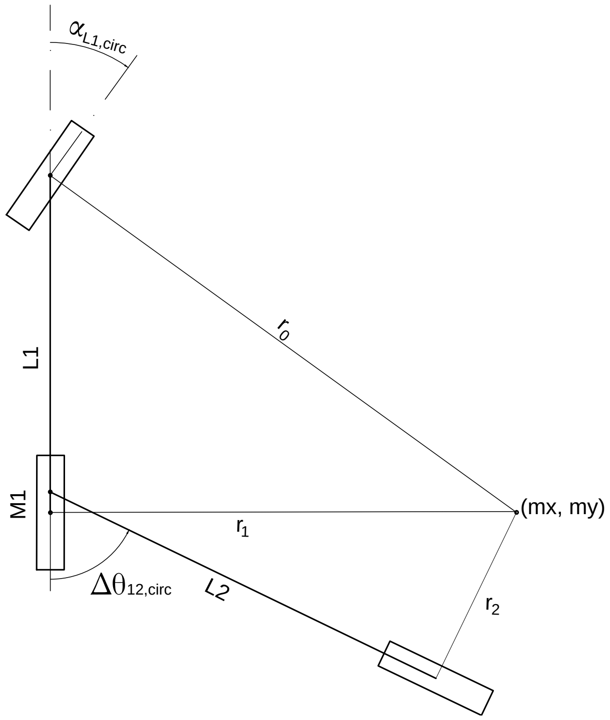


Figure 2.6: First case, stable motion for $\Delta\theta_{12,circ} < \Delta\theta_{12,turn}$.

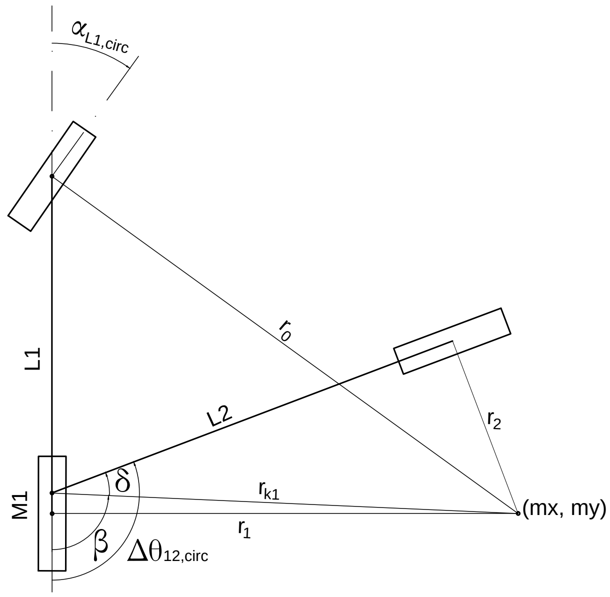


Figure 2.7: Second case, stable motion for $\Delta\theta_{12,circ} > \Delta\theta_{12,turn}$.

versa. For this case, the calculation of $\Delta\theta_{12,circ}$ is based on the next equations:

$$r_1 = L_1 \cot \alpha_{L_1} \tag{2.26}$$

$$r_{k1} = M_1^2 + r_1^2 \tag{2.27}$$

$$r_2 = \sqrt{L_2^2 - r_{k1}^2} \tag{2.28}$$

Using trigonometric rules, β and δ can be calculated:

$$\beta = \arctan\left(\frac{r_1}{M_1}\right) \tag{2.29}$$

$$\delta = \arctan\left(\frac{r_2}{r_{k1}}\right) \tag{2.30}$$

And the desired $\Delta\theta_{12,circ}$ value is calculated adding both previously calculated angles, see Fig (2.7):

$$\Delta\theta_{12,circ} = \begin{cases} \beta + \delta & \text{if } \alpha_{L_1,circ} < 0 \\ -(\beta + \delta) & \text{if } \alpha_{L_1,circ} > 0 \end{cases} \tag{2.31}$$

It is also possible to show a formula that calculates $\alpha_{L_1,circ}$ with a given $\Delta\theta_{12,circ}$. In this case, it is possible to use eq. (2.15).

In this section the important differences between truck with one-axle trailer and semi-trailer have been shown. In the next steps, all those differences and special cases will be added to the path following algorithm, trying to achieve a correct functionality for any type of general-2-trailer.

Chapter 3

Path Following Method

In this chapter, the path following method will be described. This method calculates all the necessary values and parameters to enable an autonomous vehicle to follow a given path precisely. This method consists of an iteration of steps. The first one is responsible to calculate a determinate target point to achieve. Next, an auxiliary path is calculated. This path goes from the actual position of the reference point of the vehicle to the target point. This auxiliary path can be a straight line or a circular arc with a certain radius.

The idea of the initial step of the method is to know which position the vehicle needs to reach. The path which the vehicle needs to follow, to achieve this position, is also important. In the latter step of the method, the required steering angle to follow the auxiliary path is calculated. There are many differences between forwards and backwards motion. For the backwards motion, more complex calculations are needed.

As it is already mentioned, the path following method iterates several times per second. That supposes a new calculation of a target point and an auxiliary path every time. The vehicle drives to a given target point for a short time. The steering angle will change for every new target point and auxiliary path calculated. Because of this iteration, the vehicle seems to follow the real path, and not the individual target points. The path following method is described in detail in [Schwarz(2009)], that is based on other texts, e.g. [Preissler(2001)] and [Pradalier & Usher(2008)].

The path following method introduced in [Schwarz(2009)] is valid for trucks with one-axle trailer. The method must also consider the special kinematic properties of semi-trailers and therefore the method must be appropriately adapted. Once the path following method is adapted, it can be implemented and integrated into the vehicle software.

3.1 Determination of the Target Radius

As it was already mentioned, the path following method consists of three steps. Once the original path is approximated, a target point is determined. The position of the target point depends of the length of the truck and of the trailer. It also depends of a new parameter c_v which can be configured.

In the next step, the method calculates an auxiliary path to adjust the actual deviation so that the vehicle returns again to the given path. This auxiliary path is either a circular arc or a straight line which starts in the actual position and ends in the target point. Furthermore, the tangent in its starting point corresponds with the orientation of the reference point. The reference point of the vehicle depends on the direction of the movement. If the vehicle is driving forwards, the reference point will be the midpoint of the rear axle of the truck. In the other case, the reference point will be the midpoint of the trailer axle.

3.1.1 Calculation of the Meeting Point

It was already explained that the pieces of an approximated path can be either straight lines or circles arcs. So it is necessary to distinguish between both cases to calculate the target point.

Straight line

If the relevant piece of path is a straight line, there will be two possibilities concerning the reference point P:

- The position of the reference point P has a deviation and is not on the path.
- The reference point is on the path.

In the first case, the calculation of the meeting point is described in detail in [Schwarz(2009)]. The next figure can be found there:

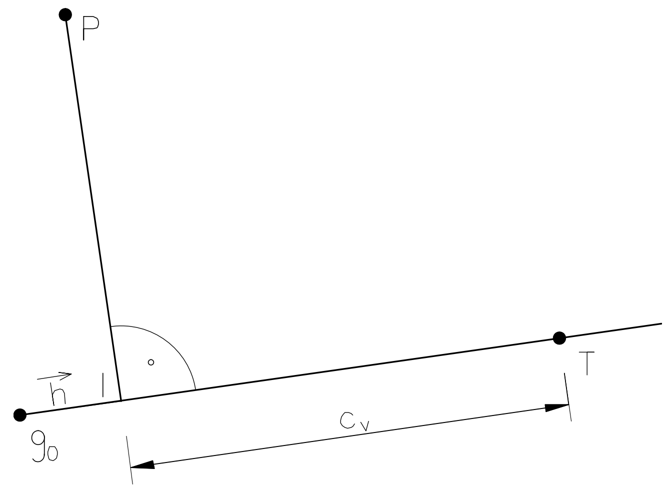


Figure 3.1: First case of target point calculation for a straight line (see [Schwarz(2009)].

The points P, I and T are actual position, ideal position and target point respectively. The parameter c_v is the projection of the path. The actual position P is far away from the ideal position I. This position is the point where a perpendicular line from the actual position P intersects the path. Since the ideal position is on the given path, there is no position error in this point. In [Schwarz(2009)] the some formulas are found shown. The first one describes the path:

$$g : \mathbb{R} \to \mathbb{R}^2, \lambda \longmapsto g(\lambda) := \vec{g_0} + \lambda \cdot \vec{h} \tag{3.1}$$

The parameter $\vec{h}$ describes the direction of the path. The length of $\vec{h}$ is not relevant, so it can be normalized as $\|\vec{h}\| = 1$.

The next formula describes the calculation of the target point T:

$$\vec{t} := \vec{i} + c_v \cdot \vec{h} = \Big(\langle \vec{p} - g_0, \vec{h}\rangle + c_v\Big) \cdot \vec{h} + \vec{g_0} \tag{3.2}$$

In this formula and in Fig. (3.1) the parameter c_v is introduced. This parameter is a value that determines the distance which the method looks ahead on the path to define the target point. This parameter is described in section 6.1. A high value causes that the target point is far from the actual position P and much path can be overlooked. This might lead to huge position errors. In contrast, a small value for this parameter causes that the target point is too near. So the vehicle will try to achieve the target point by driving extreme maneuvers. However this might result in an oscillating steering behavior. So an optimal value must be found, (see section 6.1.2).

In the latter case where the actual position P is on the path, the ideal point I and the actual point P are the same. This case is visualized in Fig. (3.2).

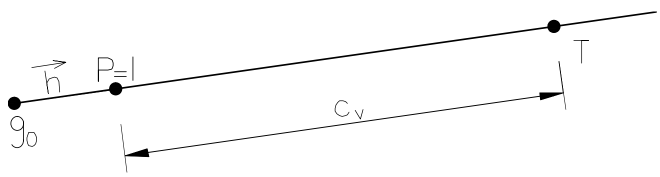


Figure 3.2: Second case of target point calculation for a straight line.

In this case, the previous formula to calculate the target point is:

$$\vec{t} := \vec{p} + c_v \cdot \vec{h} \tag{3.3}$$

Circle

In the other case the actual piece of path is a circle with a certain radius r. The calculation of the target point requires other formulas. As it is done in the straight line case, the actual position of the reference point could be on or next to the path.

In the case where the actual position is next to the path, [Schwarz(2009)] shows the next formula to calculate the target point:

$$\begin{aligned}\vec{t} &= \begin{pmatrix} \cos\frac{c_v}{r_s} & -\sin\frac{c_v}{r_s} \\ \sin\frac{c_v}{r_s} & -\cos\frac{c_v}{r_s} \end{pmatrix} \cdot (\vec{i} - \vec{m_s}) + \vec{m_s} \\ &= \frac{r_s}{\|\vec{p} - \vec{m_s}\|} \begin{pmatrix} \cos\frac{c_v}{r_s} & -\sin\frac{c_v}{r_s} \\ \sin\frac{c_v}{r_s} & -\cos\frac{c_v}{r_s} \end{pmatrix} \cdot (\vec{p} - \vec{m_s}) + \vec{m_s}\end{aligned}$$

The first matrix represents a rotation matrix. This matrix rotates the vector $(\vec{i} - \vec{m_s})$ in the xy-plane counterclockwise by an angle of $\frac{c_v}{r_s}$. Afterwards, the vector $(\vec{i} - \vec{m_s})$ is displaced by the vector $\vec{m_s}$.

In the case where the actual position P of the reference point is on the piece of path, the situation is represented in Fig. (3.4). The formula to calculate the target point is a variation of eq. (3.1.1), replacing $\vec{i}$ by $\vec{p}$:

$$\vec{t} = \begin{pmatrix} \cos\frac{c_v}{r_s} & -\sin\frac{c_v}{r_s} \\ \sin\frac{c_v}{r_s} & -\cos\frac{c_v}{r_s} \end{pmatrix} \cdot (\vec{p} - \vec{m_s}) + \vec{m_s} \tag{3.4}$$

3.1.2 Calculation of the Radius

In this section, the radius of the auxiliary circular path and its center will be calculated. As it is shown in [Schwarz(2009)], the beginning of the circular arc has to be tangential to the straight line which is defined by the reference point P and

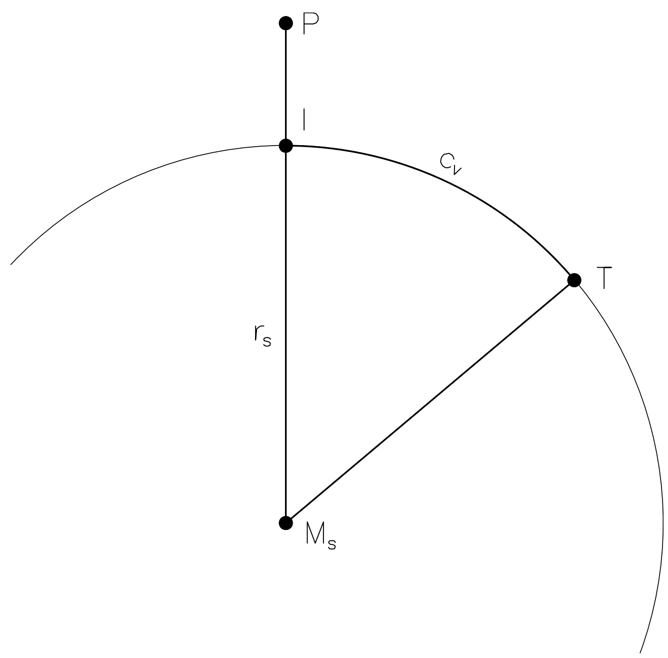


Figure 3.3: First case of target point calculation for a circle (see [Schwarz(2009)]).

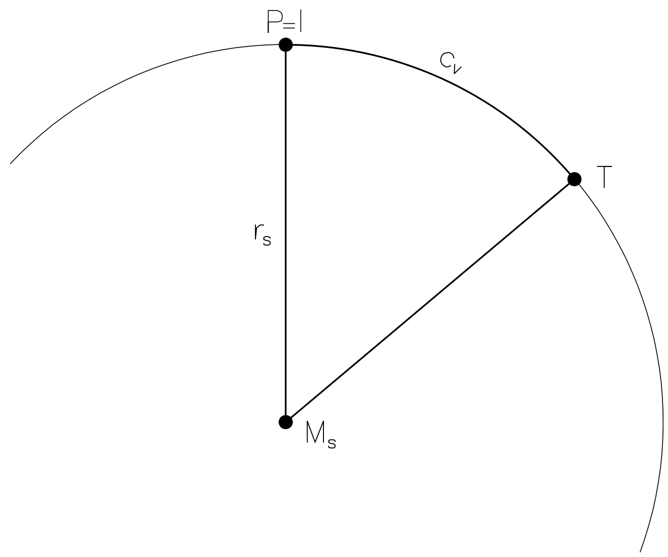


Figure 3.4: Second case of target point calculation for a circle.

the corresponding orientation θ. Furthermore, the center of the circle M_{kk} will be on the straight line r_{kk} which is orthogonal to the longitudinal axis of the vehicle part with the reference point. The circle center M_{kk} is as far away from the actual position of the reference point as from the target point P. So the solution is to connect both points by a line and add an orthogonal line just at the middle of the first one. The intersection point of the line r_{kk} and the orthogonal line identifies the circle center M_{kk}.

The thesis [Schwarz(2009)] includes the following figure, see Fig (3.5). It also includes the formulas to calculate the center of the circle M_{kk} and its radius r_{kk}:

$$M_{kk} := \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} \cdot \frac{\|\vec{p}-\vec{t}\|^2}{2 \cdot \left\langle \begin{pmatrix} -\theta_2 \\ \theta_1 \end{pmatrix}, \vec{p}-\vec{t} \right\rangle} + \vec{p} \tag{3.5}$$

$$r_{kk} = \frac{\|\vec{p}-\vec{t}\|^2}{2 \cdot \left\langle \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}, \vec{p}-\vec{t} \right\rangle} \tag{3.6}$$

3.2 Calculation of the Steering Angle

So far, the method calculates an auxiliary circle path from the actual position P to a target point T. The curvature of the auxiliary path at this point depends of those two relative positions P and T and the orientation respect to T. To follow the path, it is necessary to distinguish between forwards and backwards motion.

3.2.1 Forwards Motion

While driving forward the reference point is the midpoint of the rear axle of the truck. It moves on circular arc with the radius r_{kk}. By replacing r_1 with r_{kk}, the

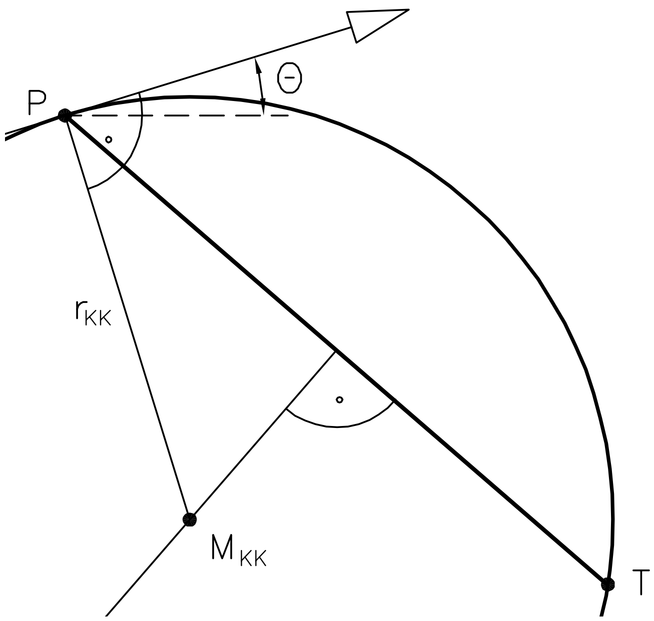


Figure 3.5: Calculation of the auxiliary path circle (see [Schwarz(2009)].)

radius is calculated by the following formulas which was already introduced in the previous section.

$$\alpha_{L_1,circ} = \arctan \frac{L_1}{r_{kk}} \tag{3.7}$$

3.2.2 Backwards Motion

In this case, the reference point is the midpoint of the trailer axle. The radius r_{kk} corresponds to r_2. The bending angle $\Delta\theta_{12,circ}$ is required to drive on the circular arc defined by r_{kk}. The calculation of the target bending angle is found in [Schwarz(2009)]:

$$\Delta\theta_{12,circ} = -\arctan \frac{M_1}{r_{kk}\sqrt{\frac{L_2^2 - M_1^2}{r_{kk}^2} + 1}} - \arctan \frac{L_2}{r_{kk}} \tag{3.8}$$

Now, we are able to calculate the steering angle α_{L_1} that provides a stable motion for the given target bending angle $\Delta\theta_{12,circ}$. This formula was already introduced in the previous chapter, eq. (2.15):

$$\alpha_{L_1,circ}(\Delta\theta_{12,circ}) = -\arctan\frac{L_1\sin\Delta\theta_{12,circ}}{L_2 + M_1\cos\Delta\theta_{12,circ}} \tag{3.9}$$

Based on the relationship between the steering angle α_{L_1} and the bending angle $\Delta\theta_{12}$, where $\alpha_{L_1} \neq \alpha_{L_1,circ}$ the following properties can be observed:

- In case of forward motion:

  If $\alpha_{L_1} > \alpha_{L_1,circ}$ then $\Delta\theta_{12}$ decreases.

  If $\alpha_{L_1} < \alpha_{L_1,circ}$ then $\Delta\theta_{12}$ increases.

- In case of backward motion:

  If $\alpha_{L_1} > \alpha_{L_1,circ}$ then $\Delta\theta_{12}$ increases.

  If $\alpha_{L_1} < \alpha_{L_1,circ}$ then $\Delta\theta_{12}$ decreases.

With these properties in mind, a higher or smaller value of α_{L_1} than $\alpha_{L_1,circ}$ is chosen to achieve a determinate bending angle $\Delta\theta_{12}$. In [Schwarz(2009)] a formula to obtain this correction of the bending angle is introduced:

$$\alpha_{L_1,target} = c_p \cdot (\Delta\theta_{12,target} - \Delta\theta_{12,is}) + \alpha_{L_1,circ}(\Delta\theta_{12,is}) \tag{3.10}$$

The parameter c_p is called proportional coefficient and it will be explained in detail in chapter 6. This parameter defines the speed to achieve $\Delta\theta_{12,target}$. On the one hand, the actual target bending angle $\Delta\theta_{12,target}$ shall be obtained as fast as possible, on the other hand huge changes of the steering angles shall also be avoided. Therefore, the calculations of the steering angle $\alpha_{L_1,target}$ is linearly related to the difference between $\Delta\theta_{12,is}$ and $\Delta\theta_{12,target}$. The higher is the correction factor which is added to $\alpha_{L_1,circ}$.

All the steps described in section 3.1 and 3.2 are repeated over the entire time in which the vehicle follows a given path. This is an iterative process, new target points are calculated continuously, depending of the parameters previously described. The final loop corresponds to a target point which is equal to the last point of the given path. Afterwards, the path following method finishes.

3.2.3 Alternative Case

In section 2.2.2 a particular situation for the semi-trailer is shown. This situation is the stable motion with an extreme bending angle, which means a bending angle higher than $\Delta\theta_{12,turn}$. Keeping this situation in mind, it would also be possible to adapt it to the path following. A calculated target point can be obtained with two alternative bending angles $\Delta\theta_{12,circ}$. The first one is to calculate the steering angle as previously done. This situation is represented in Fig. (3.6). This is called the normal case.

The alternative to achieve the target point is represented in Fig. (3.7). In this case, the bending angle is very big. This motion provides a backwards movement of the truck and a forwards movement of the trailer. The path is also followed, although the trailer axle is driving forwards. The vehicle will achieve the target point (see eq.(2.26) to (2.31).

This special situation can be used, but there are some drawbacks:

- It will need a lot of time to achieve this special situation.
- While the truck tries to reach it, the final target point could be passed.
- After reaching the target point, it will also take a lot of time to straight the vehicle again.

Because of those disadvantages, it has been decided not to use this special case for path following method. However it is possible to imagine a situation where the

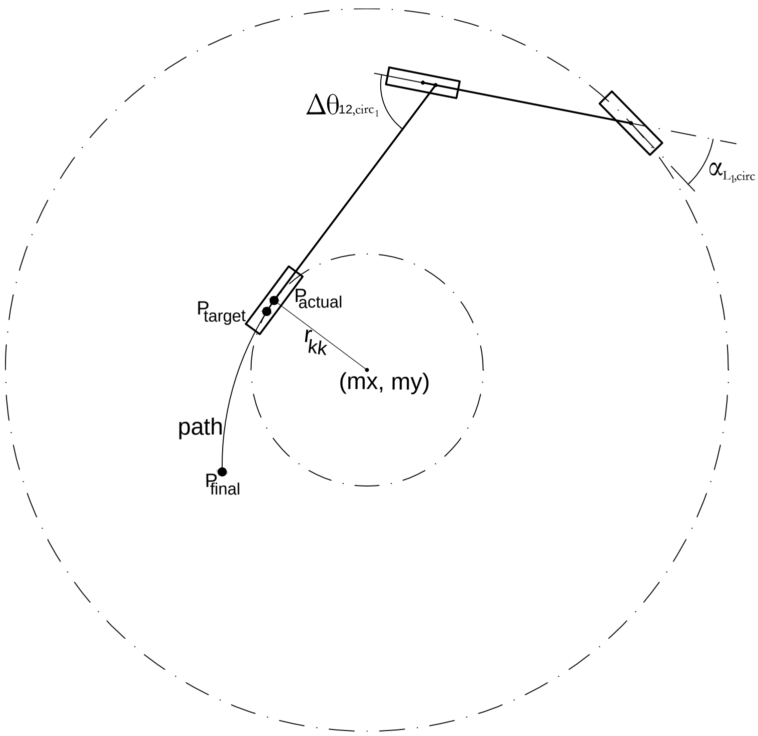


Figure 3.6: First case of path following: Normal case.

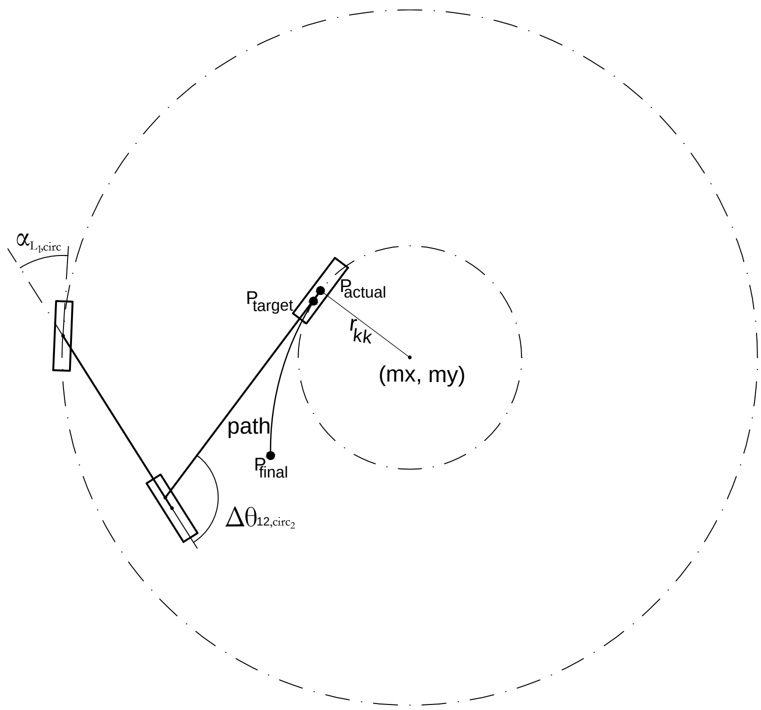


Figure 3.7: Second case of path following: Extreme case.

normal case is not possible, and the path can only be followed using the extreme case. In Fig (3.8) an example of this case is shown. There is an obstacle that prevents normal movements.

Some restrictions for this situation are:

- There is enough space between the vehicle and the beginning of the given path. So, the vehicle can achieve the extreme position before driving on the path.
- This piece of path is the last one, so there is no need to correct the position of the truck afterwards.

It is important to note that the actual system doesn't provide a way to notice obstacles in the environment. So the previous situation is theoretical, but this thesis

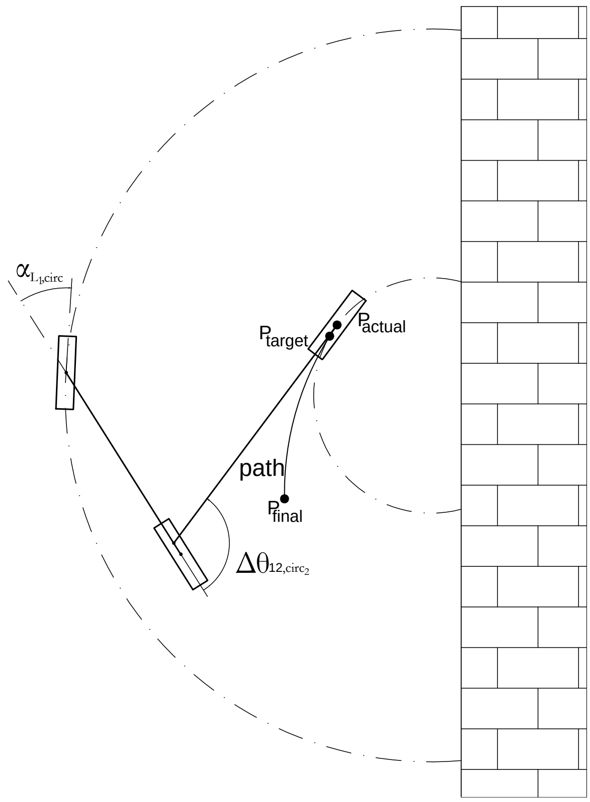


Figure 3.8: Possible optimal situation for the extreme case.

allows future improvements of the system or its implementations in other hardware systems, e.g. on real trucks.

Chapter 4

Software Environment

The software environment that provides the control of the truck and all its functionality will be explained in this chapter. The working group has developed its own software library to realize various applications in the field of autonomous and assisted driving with truck-trailer systems. This driving functionality can be used in simulation mode, using model vehicles or even a real vehicle. In this chapter, there is a focus on the software which is executed on the model vehicle and on the software components which are used on the host system.

4.1 Actual Software

In the actual software it is important to distinguish between two systems. The software executing on the host system and on the target system. The host system is a standard PC. The target system is the vehicle. Both systems are always communicating with each other. The host system sends and receives information of the target system. The information sent can be information which describes the path that the vehicle shall follow, the working mode, etc. In contrast, the target system sends information related to its state, i.e. positions, orientation, etc. The target system is a customized PC and further hardware components which will be introduced in detail in chapter 5. Now the most relevant components of both systems are explained.

4.1.1 Software on the Host

The software that executes on the host is a Windows application that provides the instructions to the software on the target, and receives data of it. These instructions can be related to the configuration of the truck, path information, etc. This software module is called EZLeitstand.

EZLeitstand

This module controls the software on the host system. It allows selecting the functionality of the target. It also receives information related to the target system and it displays those information. In Fig. (4.1) an UML class diagram of the most important components of EZLeitstand are shown.

Now the tasks of every component will be described, and its relations with other classes as well:

- **Leitstand**: This class manages the whole PC software. It controls the behavior of the PC software and the communication with the software on the truck as well, (represented by the module Steuer Software in Fig. 4.1). This class can be composed by several vehicle classes (Fahrzeug class) that represent a model of a true vehicle. See Fig. (4.2).
- **Steuer Software**: In the UML diagram, it represents the software executing on the target. It will be explained in the next sections.
- **Fahrzeug**: This vehicle class receives all the possible events and sends them to the correct receivers. For every vehicle, there is an individual connection and specific properties to load, e.g. number of vehicle parts, lengths of those

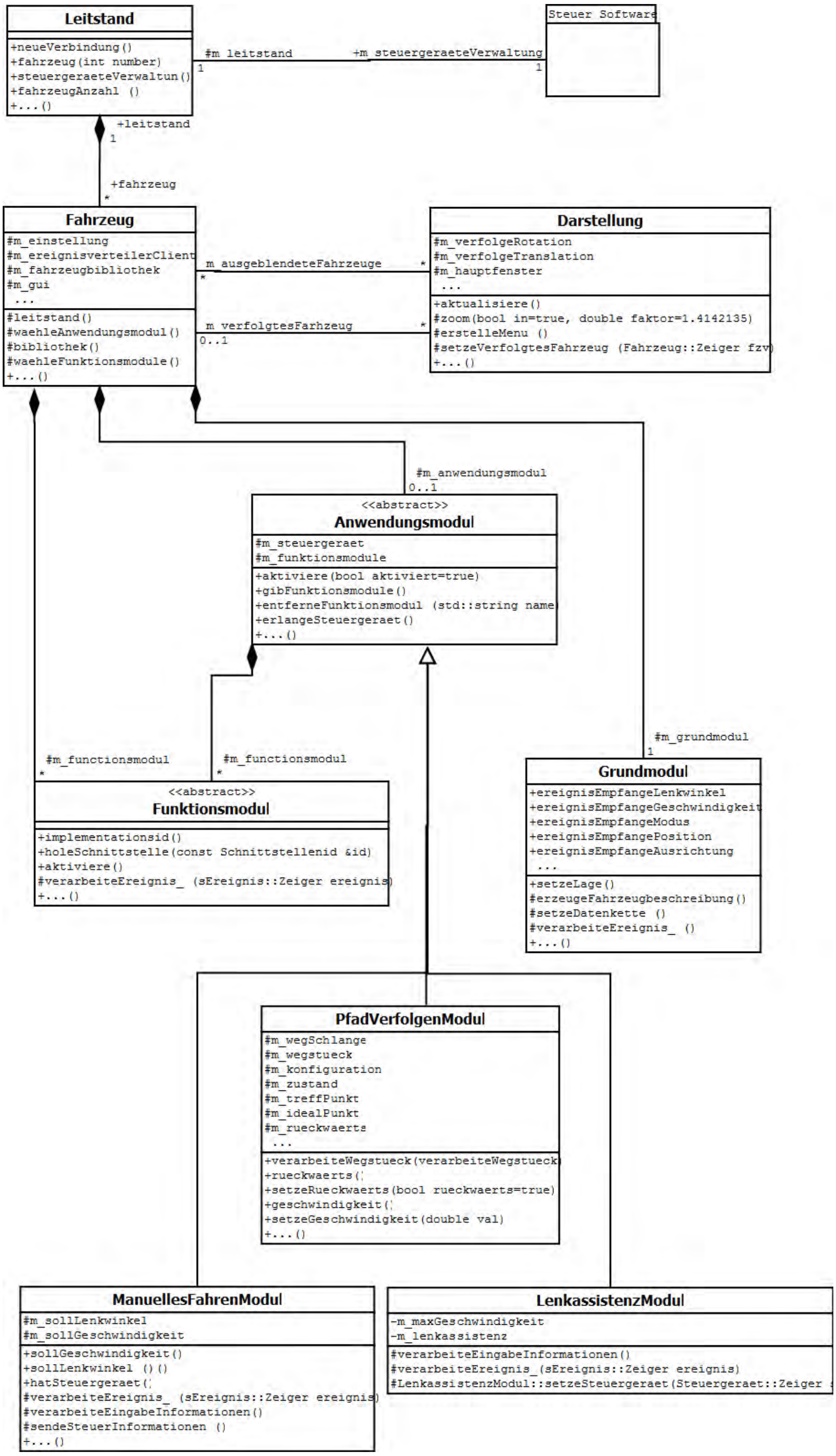


Figure 4.1: Most relevant classes of EZLeitstand UML diagram.

parts, etc. This class will be composed by different modules, e.g. the basic module (Grundmodul), the application module (Anwendungsmodul) and the function modules (FunktionsModul). This class has got also a relation with the presentation class (Darstellung).

- **Darstellung**: This class will show the truck on the GUI, and it has got some functions to visualize it. Some of those functions are the zoom functions, the vehicle follow function, etc.
- **Grundmodul**: This class provides a visualization of the basic information of the state of the vehicle, e.g. position, orientation, bending angle, etc. See Fig. (4.3).
- **Funktionsmodul**: This function class represents an abstract class for further potential extensions of the software. It provides a reference class to develop new classes. An example of this class is the recording of the real tracked path, and its comparison with the original path. It provides a way to obtain errors of the path following method.
- **Anwendungsmodul**: This application class is an abstract class to represent the different utilities of the truck. This class provides a reference class to develop function modules, e.g. path following (Pfadverfolgung), manual motion control (Manuelles Fahren), etc. See Fig. (4.4).
- **ManuellesFahrenModul**: It is a subclass of Anwendungsmodul. This module provides the functionality to drive manually with the truck. It has a simple control panel, and the truck can be controlled by sliders. See Fig. (4.5).
- **PfadVerfolgenModul**: It is a subclass of Anwendungsmodul. This module provides the autonomous driving functionality. It is possible to select different paths, circle or straight lines. It is also possible to drive forwards or backwards. The created paths, or combination of them, can be saved and loaded. See Fig. (4.6).
- **LenkassistenzModul**: It is a subclass of Anwendungsmodul. This module is not yet finished. It will provide assistance with a truck with one-axle trailer.

GUI

In this subsection, some parts of the EZLeitstand GUI will be shown.

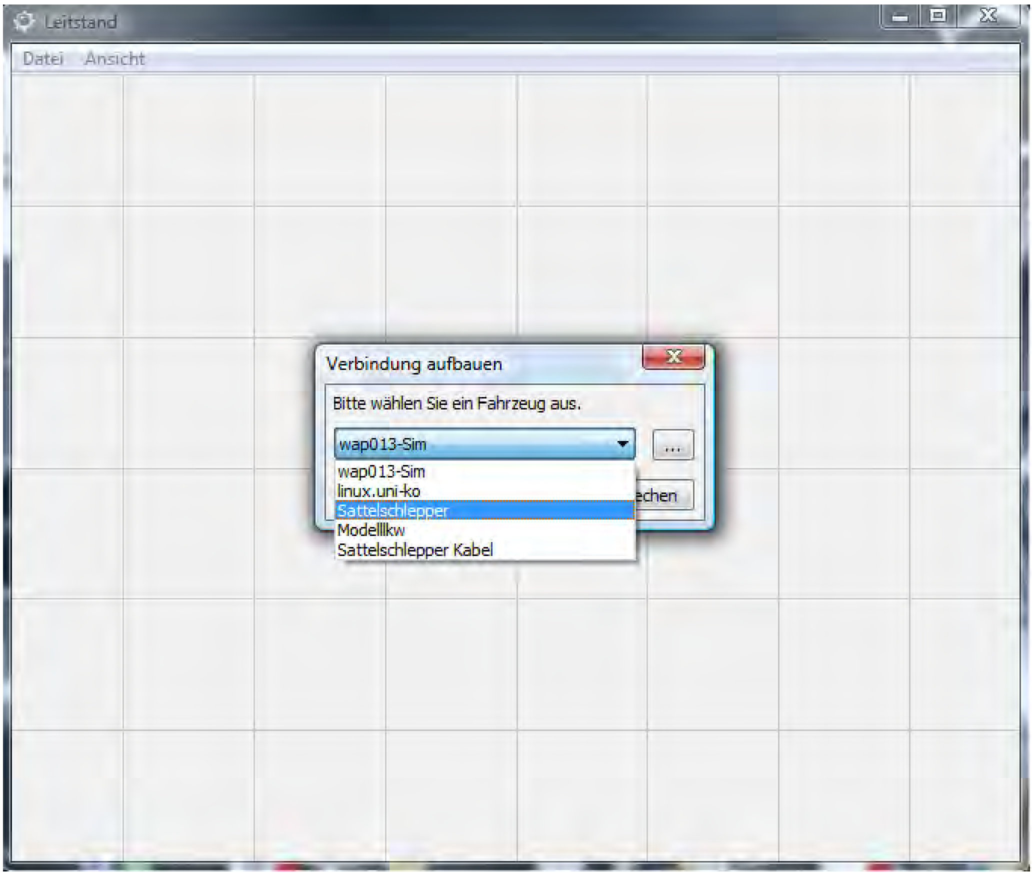


Figure 4.2: Selection of vehicle connection. It is also possible to work in simulation mode.

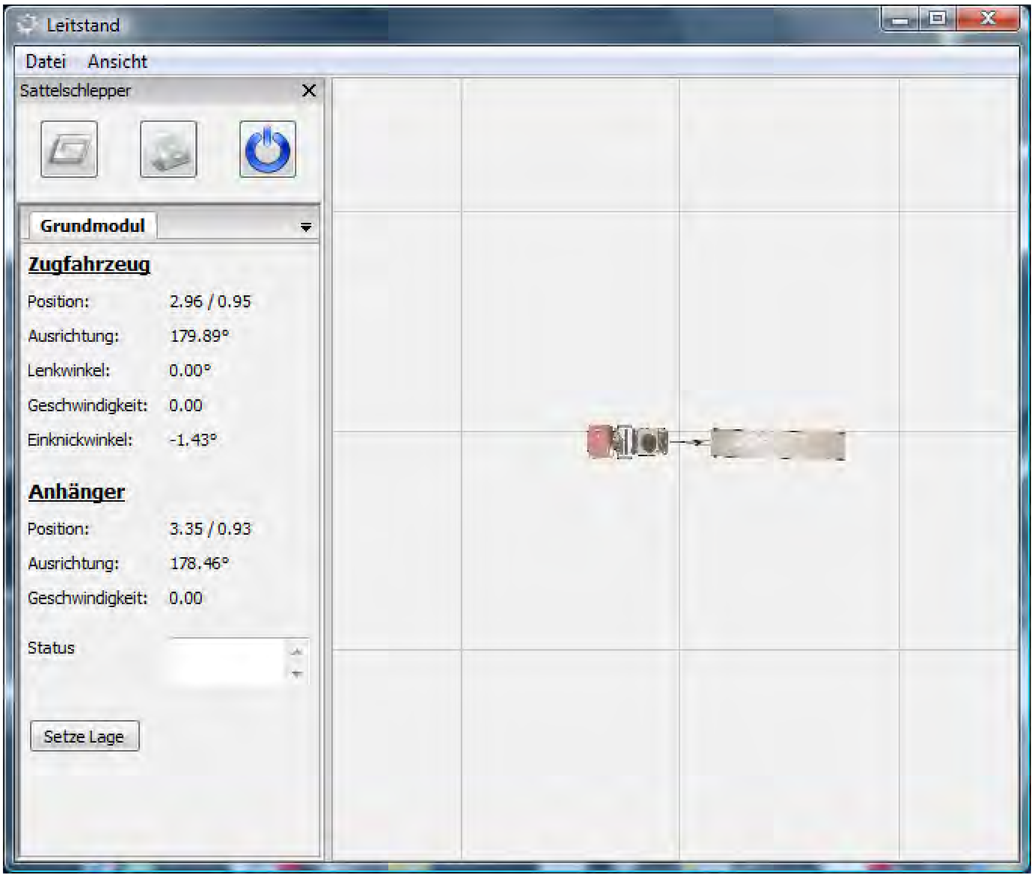


Figure 4.3: Basic module: All the basic information of the vehicles.

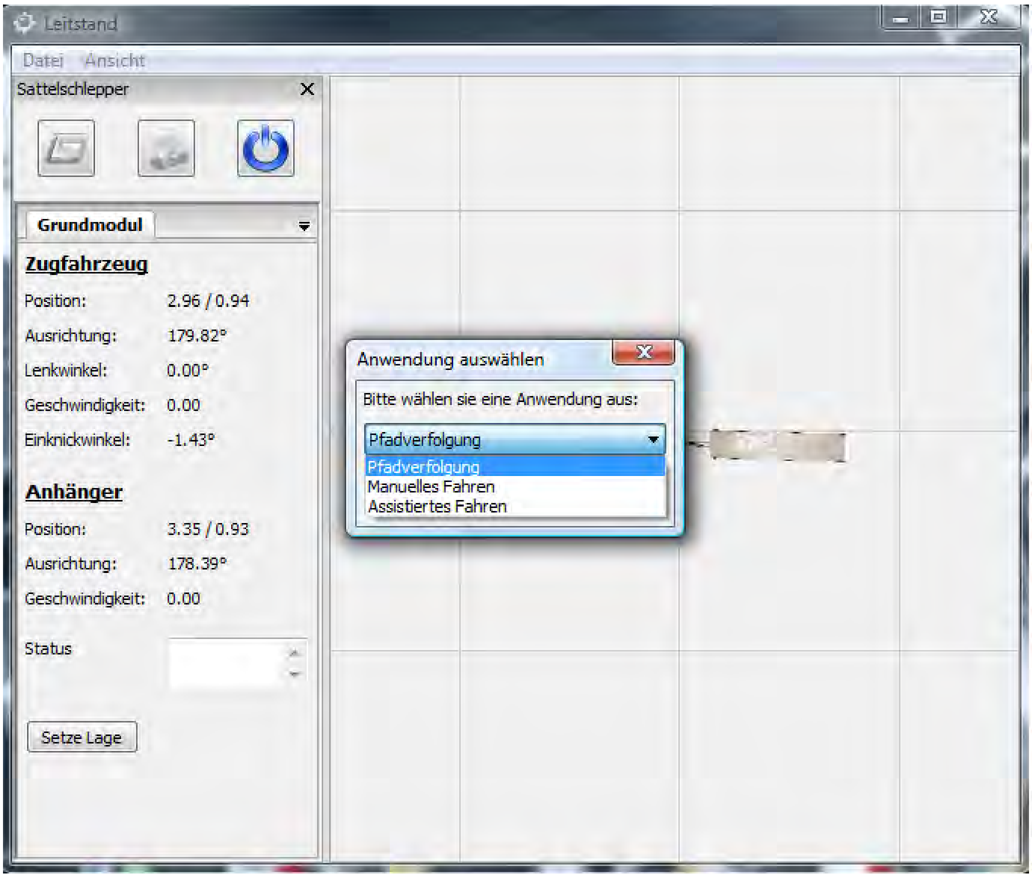


Figure 4.4: Application module: Selection of an utility.

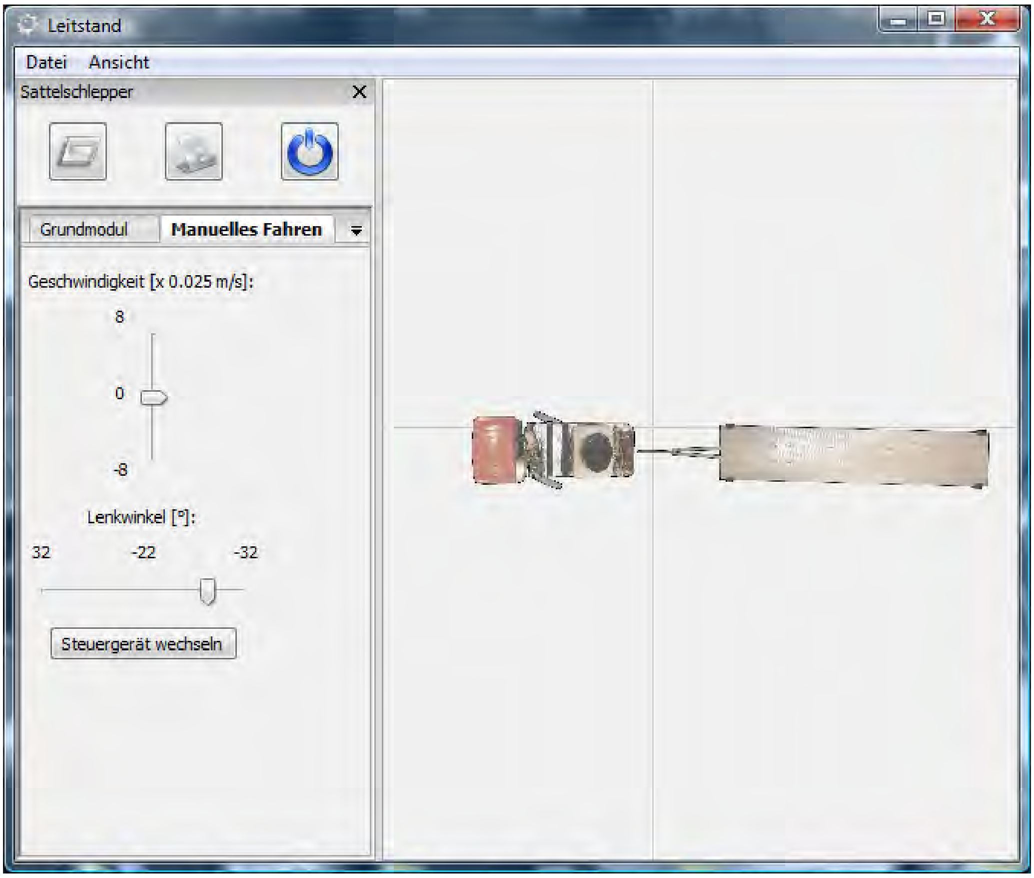


Figure 4.5: Manual module: Speed in value multiplied by $0.0025 m/s$ and steering angle in degrees (°).

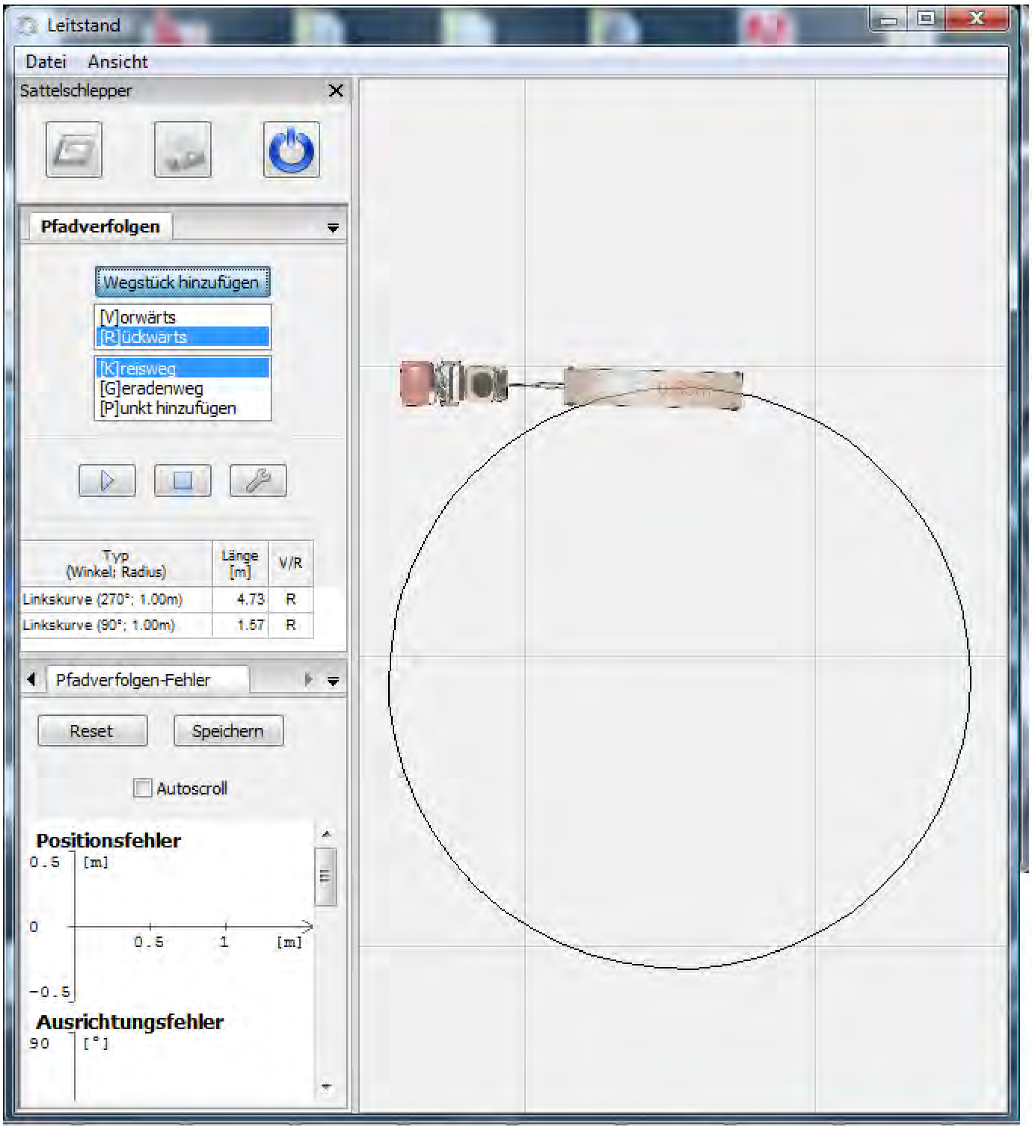


Figure 4.6: Path following module: Example with a 1 meter radius circle driving backwards and the visualization of the errors.

4.1.2 Software on the Target

The model truck was equipped with a customized PC board and a Wireless LAN bridge. This PC is running a Puppy Linux version. It is customized with only the necessary components which are required for working with the truck. So it is a very compact version, and it doesn't need so much space on the hard disk or in the main memory. For a complete description of the hardware of the truck see chapter 5.

The software on the target is composed of six programs that interact with each other. In Fig. (4.7) the relations between those programs are described.

All the components will be described now:

- **Ereignisverteilerserver**: The event handler server provides a control of every event. It receives the events of any program and sends it to the correct receivers. The programs that can send and receive events are called clients. Every client can register to receive events of certain types. Then, all the events of those type will be forwarded by the event handler server.
- **Hardwarestub**: This class provides the control of the hardware of the truck. It is responsible to send and receive events which are related to physical actions, e.g. change of velocity, steering angle, etc. It is also responsible to provide the actual state of the truck to the management process, and finally to the PC software. It is possible to think about this class as an interface between software and hardware. If the software wants to execute some action with the truck, it needs to communicate it to the Hardwarestub process.
- **Kinematikaufbereitung**: The kinematic preparation class is responsible to receive different sensor data and to merge it into the vehicle data chain. This data chain represents the data of the vehicle at the actual moment, so it includes parameters like speed, actual position, actual orientation, etc.
- **Laser**: This process provides the calculation of the actual position of the vehicles within the environment. It is responsible to receive data from the

laser scanner and to apply a triangulation algorithm on this data. The objective is to calculate the relative position in relation to the reflector marks which are located in the environment of the vehicle.

- **Verwaltung**: This management process is responsible to handle and to monitor all the actions of the other processes. It also controls the actual modus (manual driving, path following mode, etc).
- **Pfadverfolgung**: This process receives data of the vehicle and calculates the actual speed and steering angle to follow a given path. Then those values will be sent to the Hardwarestub process to set them in the vehicle actuators. In the next section this process will be explained in detail.

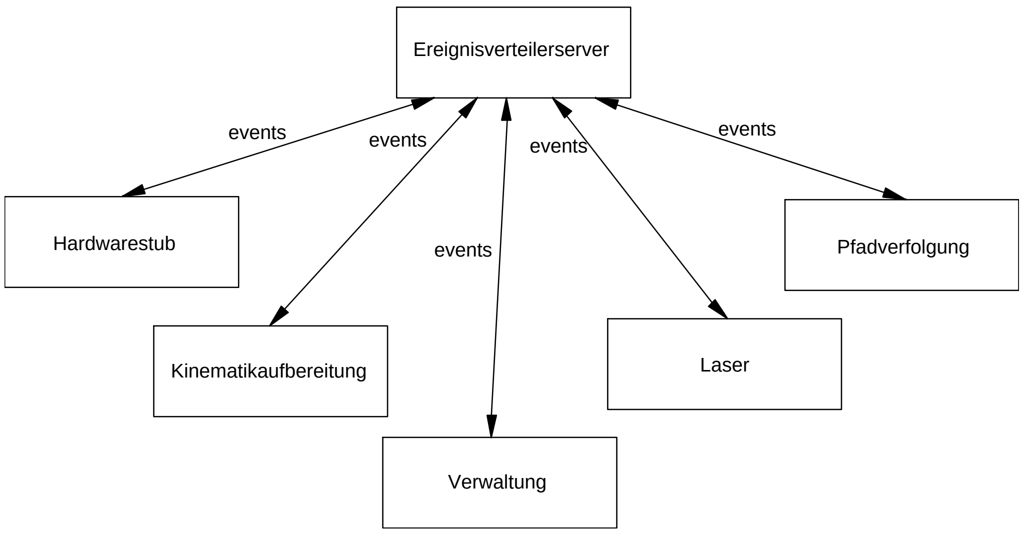


Figure 4.7: Schema of the software on the target.

4.2 New Software

In this section, the new developed software components and the adapted software parts will be described. The new software is based on two basic classes that provide the possibility to extend the path following functionality to a semi-trailer model vehicle.

4.2.1 EZErweiterteStabileFahrt

This class is an extension of the existing class called stable motion (StabileFahrt). This extension is necessary to handle all the possibilities of the new truck. The stable motion situation was discussed in section 2.2. It explains the so called normal case of a stable motion of a truck with one-axle trailer. In contrast, a second case was introduced in which the bending angle between truck and trailer becomes quite large and which can especially observed for semi-trailers. At the end of chapter 2, a special case of semi-trailer stable motion is shown. In this case, a given steering angle $\alpha_{L_1,circ}$ may have two possible values for the bending angle $\Delta\theta_{12,circ}$. One of those values is smaller than $\Delta\theta_{12,turn}$ and the other one is bigger. This special case must also be handled.

This advanced stable motion class is responsible to describe also the special case of the stable motion. This new class is a specialization of the stable motion class (StabileFahrt). This already realized class provides the functions to calculate the necessary values to achieve a stable motion. The parameters related to the stable motion are steering angle α_{L_1}, bending angle $\Delta\theta_{12}$, and radius of the reference point of the vehicle r_i. The variable i corresponds to the vehicle part. In the case of the semi-trailer, there are two vehicle parts, the truck and the trailer. So it will be two available parameters, r_1 and r_2. If the reference point of the vehicle belongs to the truck, the radius corresponds to r_1. If the reference point belongs to the trailer, the radius corresponds to r_2.

All the parameters already mentioned are interrelated. We only need one of them to calculate the others. So the functions of this class needs an input value of a parameter, and the rest of the parameters are calculated and stored. Then it is possible to get those parameters by calling the corresponding functions.

The problem with the original class is that it only provides the functionality for trucks with one-axle trailer vehicles. Sometimes this functionality can be used for semi-trailer vehicles, but is not appropriate for all cases. As it is already mentioned, for a given $\alpha_{L_1,circ}$ there are two possible values of $\Delta\theta_{12,circ}$ that provides stable motion. The original stable motion class can handle the case of

$\Delta\theta_{12,circ} \leq \Delta\theta_{12,turn}$. But it cannot handle the case of $\Delta\theta_{12,circ} > \Delta\theta_{12,turn}$.

Now the functionality of the new class will be described, depending of the parameter set:

1. **Bending angle** $\Delta\theta_{12,circ}$: In this case, there is only one possible value for α_{L_1}, and for r_1 and r_2 as well. The members of the superclass can be used, according to the kinematical properties of general-n-trailers.

2. **Steering angle** α_{L_1}: If the input is a steering angle, two different values can be calculated for $\Delta\theta_{12,circ}$. But there is only one possible value to r_1, and another one for r_2. A new parameter is introduced as a flag to decide which values shall be calculated. The methods of the new class can calculate the values for both cases. If the flag determines the case of $\Delta\theta_{12,circ} > \Delta\theta_{12,turn}$, then r_2 will change its sign.

3. **Radius of a vehicle** r_i: If a radius is set, e.g. r_1, the member functions of the superclass will be used. So $\alpha_{L_1,circ}$, $\Delta\theta_{12,circ}$ and r_2 are calculated. Next, it is necessary to know the desired final state of the truck, so the flag input value is considered. This final state can correspond to the case of $\Delta\theta_{12,circ} > \Delta\theta_{12,turn}$ or not. If it is the case, the value of $\Delta\theta_{12,circ}$ and r_2 are changed. So the final state of the vehicle corresponds to Fig.(2.7).

In Fig. (4.8) the two classes with their most relevant member functions and attributes are shown. The extensions of the new class are explained now:

- **New attributes**: This class is a specific class for semi-trailer vehicles. So it is assumed that the complete vehicle is composed of a truck and a trailer with one axle. Hence, new attributes related to the length of truck and trailer are added. This simplifies the calculation of $\alpha_{L_1,turn}$ and $\Delta\theta_{12,turn}$ that are also new attributes of the class. The value of $\alpha_{L_1,turn}$ and $\Delta\theta_{1,2,turn}$ never change. They are calculated once by the constructor and they are stored in special member variables.

- **Changes on member functions**: For some functions, like "einknickwinkel()" (getter function for bending angle) or "setzeEinknickwinkel()" (setter function for bending angle), there is no need to indicate which bending angle to get or to set. In the case of a semi-trailer there is only one. The other overwritten functions keep their input parameters and add a flag parameter, to differentiate the cases previously explained.

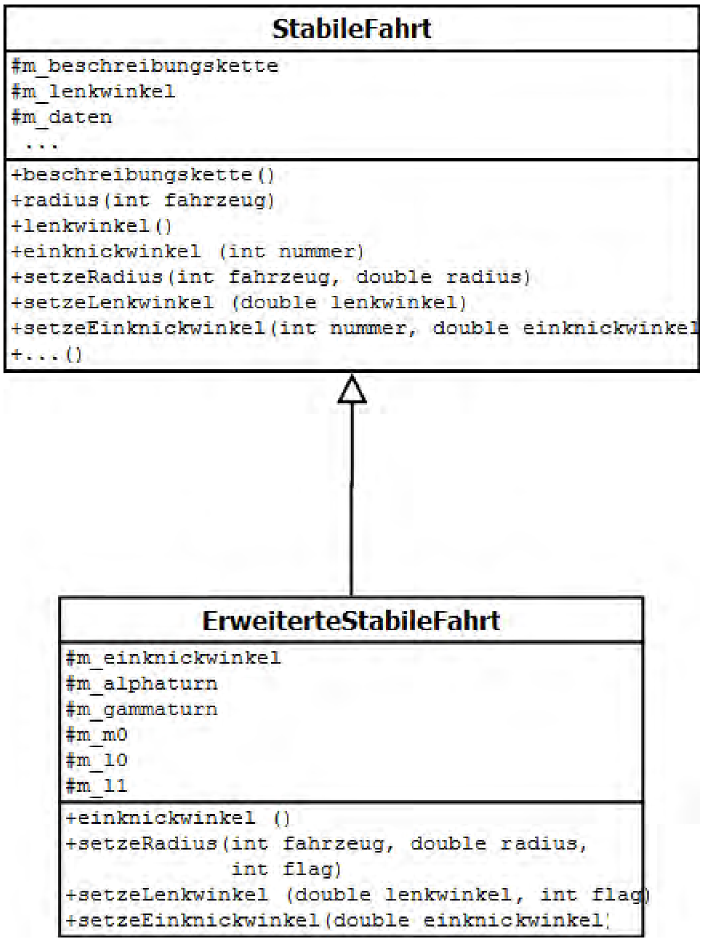


Figure 4.8: ErweiterteStabileFahrt UML diagram.

The stable motion class, and the advanced stable motion class as well, belong to a library called EZKern (the core of EZ). This library is composed by several classes. These classes have different kinds of responsibilities, e.g. they provide stable motion, they define the data structure to define paths, etc. The data structure of this library is used to model the state of the vehicle.

4.2.2 EZPfadverfolgung

Now, the previously realized component which implements the introduced path following method will be explained. Especially, the necessary modifications will be discussed in detail. In [Schwarz(2009)], different components related to the path following method were developed. Those components are deployed on both software parts, host and target. The different components developed and shown in [Schwarz(2009)] are explained next:

Software components of the path following method on the host system

The modules running on PC are:

- **PfadVerfolgenModul**: The path following module is also commented in section 4.1.1. In this module it is possible to create manually paths as combinations of circular arcs and straight lines. It is also possible to visualize that path, to save it or to load it.

- **PfadverfolgenFehlerFunktionsmodul**: This function module can be executed while a vehicle is in autonomous state and follows a given path. It receives continuously the current vehicle data, e.g. position, orientation and steering angle. Based on these data, it calculates and visualizes error values for selected parameters in relation to the given path. The errors are visualized in diagrams which can be saved afterwards.

- **PfadAufzeichnenFunktionsmodul**: The path recording function module allows to record and to save the real path driven by the vehicle. So it is possible to load those paths again later.

- **KarteZeichnenFunktionsmodul**: This module is called map drawing function module. It represents on the screen a map of a road. There is a scaled model of that road in the laboratory.

Software components of the path following method on the target system

In the already mentioned thesis [Schwarz(2009)], some path following components were also developed and programmed to interact with the components which were previously presented:

- **Pfadverfolgung**: This is an abstract class which provides a basic class to create new specialized subclasses. Those new classes implement the required operations of the path following method. Those classes can also use its methods to communicate with other modules of the software on the host, by sending and receiving events.
- **KreiseUndGeradenPfadverfolgung**: This class is a specialization of the previous class Pfadverfolgung. It is responsible for the preprocessing of the previous selected path, which was either defined manually or loaded from a file. The path following process can only manage path pieces as circular arcs or straight lines. This class divides the real path to several segments of straight lines or circles with different radiuses.
- **KorrekturKreisPfadverfolgung**: This class is a subclass of KreiseUndGeradenPfadverfolgung. This class provides the actual path following method. It takes the vehicle parameter values, e.g. position, orientation, etc, and the given path as well and it responds with specific values for the steering angle. In this class the path following formulas seen in chapter 3 are implemented.

In [Schwarz(2009)] it is shown a UML class diagram that represents the relations between the previous classes, Fig. (4.9).

Connections between path following components

Once commented the components of the path following functionality, it is possible to describe the connections between them. As it is shown is Fig. (4.10), the most relevant elements for a path following explanation are modeled. The host

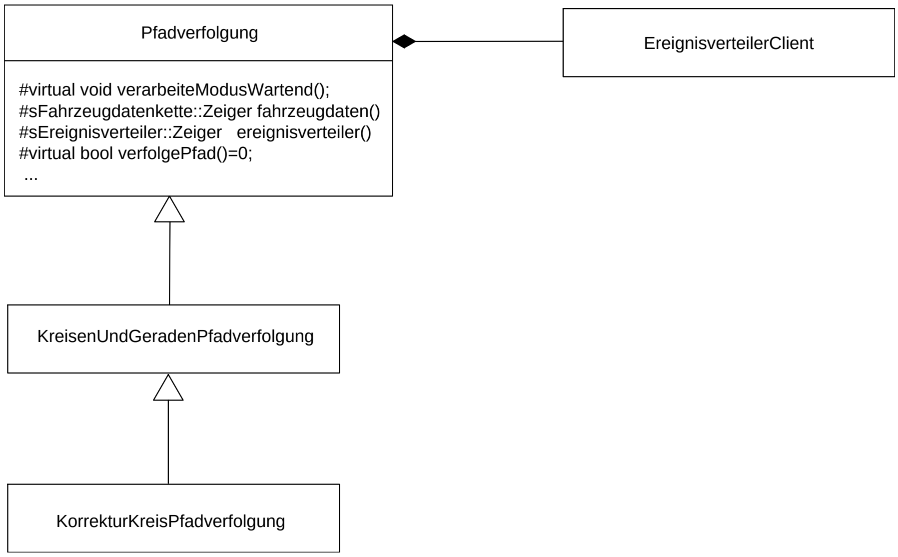


Figure 4.9: UML diagram of path following classes.

software part consists of the Leitstand (control) module which uses the Kern (core) library. The Leitstand module and its components were previously described. This module communicates with the host path following component sending orders, and receiving information. The information sent is related to the path, the working modus of the truck, etc. The information received is related to the state of the vehicle.

The Kern library was also commented previously. This library offers many functions related to various aspects of the system, e.g. information recovery, description of the vehicle, etc. In this particular case, it is represented because the stable motion class described in section 4.2.1 is part of this library. As it is shown in Fig. (4.10), the class KorrekturKreisPfadverfolgung directly uses this library. The main calculations of the path following method are done in this class KorrekturKreisPfadverfolgung. The stable motion class provides the functionality to calculate a steering angle to obtain a stable motion.

The original path following structure has been modified. Before the extension, the class KorrekturKreisPfadverfolgung could only communicate with the class StabileFahrt. Now, it can also communicate with the new class ErweiterteStabileFahrt (see Fig. 4.10). This class has been added and some parts of the class KorrekturKreisPfadverfolgung have been adapted, in a way that the new class could be created and used on it. The path following algorithm seen in chapter 3 is implemented in the class KorrekturKreisPfadverfolgung. So, this class is adapted to those formulas and it is also adapted to the new class ErweiterteStabileFahrt.

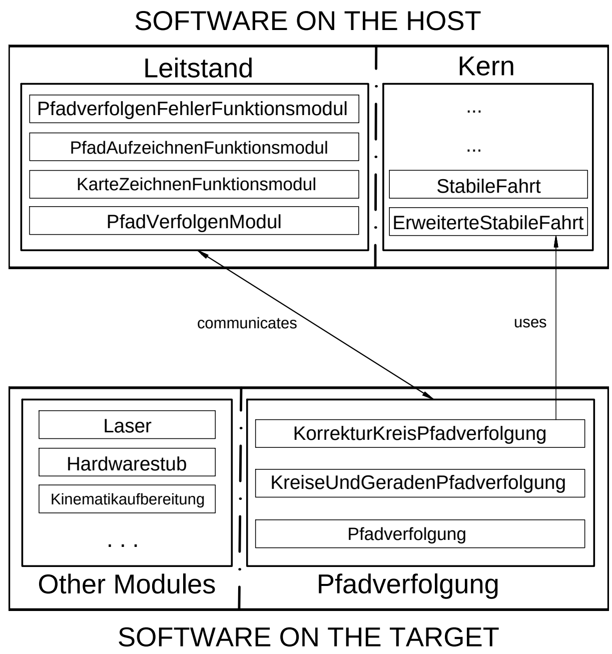


Figure 4.10: Schema of the extension on path following modules.

The new class uses the same specification of its superclass and only its methods were rewritten. Now the final path following module can be applied to semi-trailer vehicles, keeping it previous functionality. Furthermore, the final software supports general-2-trailer vehicles with two components, a truck and a trailer with a non-steerable axle. It is also able, in theory, to work with standard-2-trailer vehicles, because all the restrictions that they suppose, for example see eq. (2.24), are handled with the new modifications. It is supported only in theory, because there is no standard-2-trailer model vehicle in the working group to realize the tests. It is not usual to find those types of vehicles.

In this chapter, the most relevant software components of the system have been described, and how they are integrated into the main system. Once all the formulas presented in chapter 2 are implemented as algorithms, the software is finished. The next step is to configure the semi-trailer model vehicle. This configuration tries to achieve a better performance between software and hardware.

Chapter 5

Hardware of the Truck

5.1 The Semi-Trailer Model Vehicle

In this section, the hardware of the semi-trailer model vehicle will be described. All its important components will be presented and the interaction and communication between those components will be also explained.

The original model vehicle is a radio-controlled truck, from the german company Wedico. The model vehicle is designed as a Mercedes Actros semi-trailer, see Fig. (1.1). The model truck is scaled to 1:16, and its measures are shown in Table (5.1).

Since this model is a semi-trailer and the truck is quite short, all the additional components to enable autonomous driving are located on the trailer. A schema of

	Length	**Width**	**Weight**
Truck	163	167	197
Trailer	821	163	163

Table 5.1: Measures of the vehicle (millimeters).

the model is presented in Fig. (5.1) based on a schema included in [Eckert(2007)].

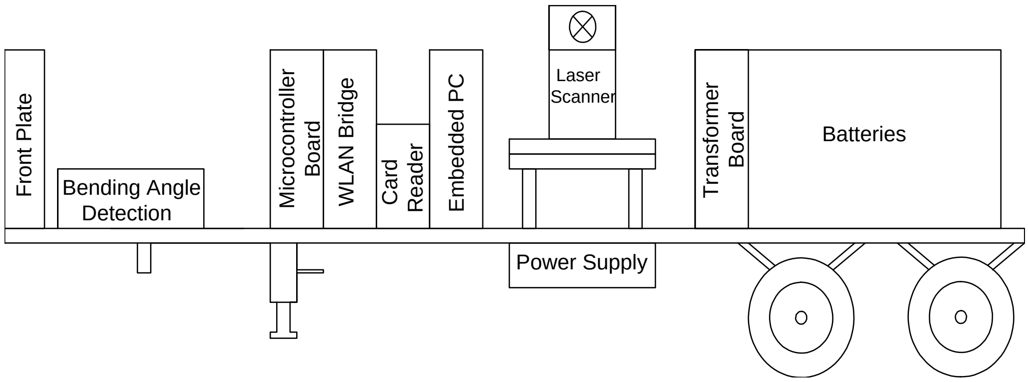


Figure 5.1: Schema of the trailer model vehicle.

Now, the different parts of the whole system will be explained:

- **Power supply**: The energy can be provided by cable, or by batteries. The cable connection is located at the middle of the trailer, and the batteries should be mounted on the rear end of the trailer.
- **Laser scanner**: It is located at the middle of the trailer. This laser uses three landmarks, composed by well aligned reflectors which are placed in the vehicles environment. They define a coordinate system within the vehicles are moving. Using a triangulation method, it is possible to determine exactly the position and orientation of the truck in relation to the coordinate system which is defined by the landmarks.
- **Logic**: The logic part of the truck is located between the laser scanner and the sensor for the bending angle detection. It is composed by:

  Embedded pc: It is located next to the laser scanner. It is possible to connect with the standard PC devices, e.g. monitor, keyboard, etc. All the processes are executed on this embedded PC.

  Flash card reader: It is located next to the embedded PC and directly connected to it. It stores the operating system and all required the files.

  WLAN bridge: It is located next to the flash card reader. It connects the embedded PC to the network. It has got a Wi-Fi antenna that is attached on the front plate, and connected with the bridge via cable. It provides the whole communication with the PC where the host software is executing.

Microcontroller board: It is located next to the sensor for the bending angle detection. It is connected to the sensors and actuators of the truck. It is also connected with the PC board.

Sensor for the bending angle detection: It measures the angles between the longitudinal axis of the truck and the longitudinal axis of the trailer. The angle is determined by a certain voltage.

Front plate: It connects the engine and the steering angle actuator of the truck with the power supply. The Wi-Fi antenna is attached on it.

5.2 Calibration of the Truck

A calibration of the truck is required because the source data of the vehicle cannot be directly used. The initial data generated by the vehicle is a certain voltage, measured by the actuators. Then an analog-to-digital converter transforms the voltage data into an integer value. But this integer values needs to be converted to an angular value. So a mapping function is required to convert those integer values into angular values in radians. This mapping function is identified by the calibration process. For the bending and steering angle, the calibration tools, processes and results will be described.

5.2.1 Steering Angle

Calibration tools

The calibration tool for the steering angle is a self-made protractor. This protractor has a pointer that allows to measure the real steering angle of both steerable wheels. This tool is composed by:

- Two sheets of paper, one of them of size A3, and the other of size A4. A semicircle is drawn on each of them. This semicircle is divided in $180°$. It is also divided in $10°$ and $5°$ ranges. See Fig. (5.2).

- Two plastic pieces, of 4x23cm and 6x26cm, with a line on it. These pieces will act as pointers.

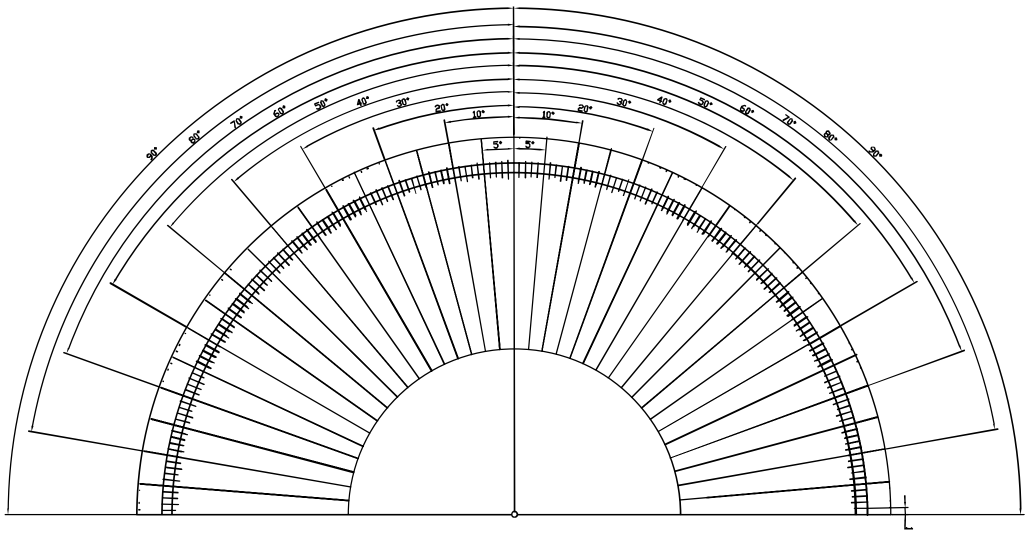


Figure 5.2: Protractor.

Calibration process

First both protractor papers are placed and fixed on the ground. Then, the two pieces of plastic are stuck to the wheels, in a way that the line drawn on the pointer corresponds with the projection of the extension of the wheel. See Fig (A.1) and Fig. (A.2).

On the one hand, the actual software EZLeitstand provides a way to move the wheels of the truck in certain angles in degrees. But now we cannot trust to those values, the real angular values can differ. On the other hand, the software on the truck provides a way to visualize the corresponding integer values of those angular values. So the calibration process is based on moving the wheels, reading this integer value and watching the corresponding angle shown by the measuring tool. This angle must be observed for both steerable wheels.

Since the steering actuator is not perfect, it takes different values for a given integer value. This value depends in which direction the wheels are moving. The process was designed to take care of this restriction. The whole process is described sequentially. In every case, all the data were stored:

1. From $\alpha_{L_1} = 0$ to its maximum, turning right. The same process also turning left.
2. From $\alpha_{L_1} = \alpha_{L_1,max}$ to $\alpha_{L_1} = -\alpha_{L_1,max}$, and vice versa.

Finally, for every integer value there are six different values in degrees. Three of them are related with the right wheel and the others are related to the left wheel. For every wheel, the final value is:

$$\alpha_{L_1} = \frac{\alpha_{L_1,test1} + \alpha_{L_1,test2} + \alpha_{L_1,test3}}{3} \tag{5.1}$$

Now, we have got two final values, one of them corresponds to the right wheel and other one corresponds to the left wheel. The value of the virtual wheel is the average of both values:

$$\alpha_{L_1,virtual} = \frac{\alpha_{L_1,Rightwheel} + \alpha_{L_1,Leftwheel}}{2} \tag{5.2}$$

Calibration results

Once the truck is calibrated, it is possible to check if our calibration is good enough. We can repeat the process but not in the same way. For testing, the integer values of the steering angle are not used. In the software EZLeitstand, some values for the bending angle will be sequentially set. Then the real angle of those values can be seen in the protractor. The error will be the difference between both values. The cases are described next:

1. $\alpha_{L_1,virtual} = 0°$ until $\alpha_{L_1,virtual} = \alpha_{L_1,virtual,max}$ with steps of $1°$.

2. $\alpha_{L_1,virtual} = 0°$ until $\alpha_{L_1,virtual} = -\alpha_{L_1,virtual,max}$ with steps of $1°$.

The result shows that, in average, the error of the steering angle is $0.6°$ for the first case, and $0.27°$ for the latter case. These results are very good error values, according to the interaction between software and hardware. In the case of the steering actuator, this interaction is never perfect.

Some restrictions of the steering angle were also detected. The maximal value of the steering angle can be $35°$ in both directions. But in this case, to achieve this extreme angle, the cabin of the truck moves a little bit. This is caused by the pressure of the steering actuator on the wheels. So it is necessary to set a smaller maximal value for the steering angle:

$$-32° \leq \alpha_{L_1,virtual} \leq 32° \tag{5.3}$$

5.2.2 Bending Angle

Calibration tools

The tool to calibrate the bending angle is similar to the previous one. It is also a self-made protractor. In this case, there are only one protractor and one pointer. The materials used to calibrate the bending angle are:

- One sheet of paper of size A3, with a semi-circle drawn on it. It is the same as in Fig.(5.2).

- One piece of plastic, of 6x26cm, with a line on it. This piece will act as pointer, and it has got a hole in one extreme. This hole has the same figure as the hitch of the trailer. It is located just in the middle of the line.

Calibration process

First, the protractor paper is located over some piece of wood. The idea of this is to achieve, more or less, the height of the truck in its hitch. The next step is to fit the pointer to the hitch of the trailer. This is done in a way that the line of the pointer corresponds with the projection of the bending angle. See Fig (A.3).

The software on the truck provides a way to obtain the integer values of the bending angle. The calibration process is based on the manual movement of the pointer to a certain angle. Then, the corresponding integer value is read, (see Fig A.4).

The whole process is described sequentially. The data were stored in every case:

1. From $\Delta\theta_{12} = 0$ to $120°$, moving right in $5°$ ranges. The same process also moving left.
2. From $\Delta\theta_{12} = 120°$ to $\Delta\theta_{12} = -120°$, in $5°$ ranges and vice versa.

Finally, for every observed value there are three different integer values. All of them are very similar. The final integer value for every observed angle is:

$$\alpha_{L_1} = \frac{value_1 + value_2 + value_3}{3} \tag{5.4}$$

Calibration results

In the end, it is possible to check if the calibration is good. Then, it is necessary to repeat the process. But now it does not matter the integer values previously mentioned. The important values are the real bending angle shown by the protractor and the bending angle shown on the screen. The testing of the error has got the next steps:

- From $\Delta\theta_{12} = 0$ to $120°$ moving left in $5°$ ranges.

- From $\Delta\theta_{12} = 0$ to -120° moving right in 5° ranges.

The average error is 0.112° for the first case and 0.095° for the second one. That means the results are really good, at least, using this calibration process. It would be interesting to test the error using another calibration process. However it is very difficult to create a new calibration process that does not use similar tools to those used in this process.

Chapter 6

Error Estimation of the Path Following Method

In the chapter 3, the path following method has been described. Now, this method will be tested to obtain the real errors of it. The errors of these tests will help to determine which values we can consider as acceptable. Furthermore, it will determine if a recalibration of the model vehicle is necessary.

For the measurement of errors, the actual software provides information for three types of error:

- The position error of the reference point ξ_y is the distance between the position of the reference point of the vehicle and the projection of the actual piece of path.
- The orientation error of the reference vehicle ξ_{θ_1} is the difference between the actual orientation of the reference vehicle and the target orientation.
- The bending angle error $\xi_{\Delta\theta_{12}}$ is the difference between the bending angle and the curvature of the actual piece of path.

These errors can be observed for every path followed by the vehicle, and they can be saved as graphics.

6.1 Determination of the Setting Parameters

In this section, the target is to obtain an optimal value of two parameters that regulate the behavior of the vehicle. These two parameters are:

- c_p is called the proportional coefficient. The objective of this parameter is to set a value that provides a way to achieve a determinate bending angle $\Delta\theta_{12,fixed}$ from an actual bending angle $\Delta\theta_{12,is}$.
- c_v is the projection on the path. This value is an angular value. This parameter provides an acceptable deviation of the meet point regarding to the ideal point. The ideal point is the point where there is no position error in path following motion. The idea is to look ahead to a target point with an acceptable distance between the actual position of the reference point and this target point.

6.1.1 Structure of the Experiment

For the parameter c_p it is necessary to set a value that provides a fast achievement of the fixed bending angle $\Delta\theta_{12,fixed}$. But without so much oscillation after reaching it. In this case, it does not really matter if the path is a straight line or a circular arc. If a fixed value for the bending angle $\Delta\theta_{12,fixed}$ is set, the vehicle will forget the path and it will try to reach this fixed value. The structure of the experiment for c_p is based on two tests:

- First test: $\Delta\theta_{12,fixed}$ is set to $50°$, and $\Delta\theta_{12,is}$ is set to $0°$.
- Second test: $\Delta\theta_{12,fixed}$ is set to $0°$, and $\Delta\theta_{12,is}$ is set to $30°$.

For every test there are several subtests. Each of them with a different value for c_p. Specifically, the test starts with an integer value of $c_p = 2$. Then, this value is increased in one unit repetitively, until the next value provides worse results than the previous one. That means that there is a value between those two integer values that is the optimal one for c_p. In this case, the test will be repeated with different decimal values in that range.

For the parameter c_v, the appropriate value is the value which provides a fast reduction of the position error. But without much oscillation of the position when the position error $\xi_y = 0$ is reached. The structure of the experiment is based, first, on the movement of the truck to a certain position, with $\alpha_{L_1} = 0°$ and $\Delta\theta_{12} = 0°$. Then a straight line path is defined, in backwards motion. Later, manually, the truck is moved to a distance of 0.5 meters in the y coordinate from its original position. And finally, the path following method is started.

6.1.2 Results of the Experiment

The results of the previous experiment will be commented in this section. For the parameter c_p, in the first test, there is not much difference between values in the range of $(3, 5)$. It seems that, more or less, all those values take similar results. Almost all values has got the similar deviation when the value of $\Delta\theta_{12} = 50°$ is achieved. Furthermore, it happens only after 0.3 meters from its initial position. See Fig.(B.1).

So it is necessary to consider the second test. In that test, it is shown that for values of $c_p = 2$ and $c_p = 3$ there is more oscillation of the bending angle error than for $c_p = 4$. See Fig. (B.3). Then, the values of $c_p = 4$ and $c_p = 5$ are tested. Both values take really good results, with only a bit of oscillation. See Fig. (B.4). The value of $c_p = 4$ takes better results if the general results for both tests are considered. See Fig.(B.2). In both cases, it is only necessary 0.07 meters to reach the value of $\Delta\theta_{12,target} = \Delta\theta_{12,is}$.

According to the results, it is shown that the convergence to $\Delta\theta_{12,target}$ is really fast. These results are consequence of the correct calibration of the steering angle α_{L_1} and bending angle $\Delta\theta_{12}$. The particular structure of the articulated vehicle also influences. In a semi-trailer, a variation of the steering angle α_{L_1} supposes a big and fast influence on the bending angle $\Delta\theta_{12}$. This is due to the steerable axle of the truck and the hitch are relatively close.

For the parameter c_v, the test starts with $c_v = 0.3$. For this value, the position error zero is achieved quickly. In contrast, there is a lot of oscillation afterwards. So the bending angle error is very high for this value. There are a lot of problems to stabilize the vehicle, see Fig. (B.5). That suggests a higher value for c_v.

Another possible value is $c_v = 0.6$. With this value, the oscillation after reaching the position error zero is quite smaller than for the value of $c_v = 0.3$, see Fig. (B.5). In contrast, this value supposes a lot of time to achieve the position error zero. So the next step is to discover a new value in the range of $[0.3, 0.6]$. This value must provide small oscillation and a quick achievement of the position error zero.

The next attempts are with values of $c_v = 0.4$ and $c_v = 0.5$, see Fig. (B.6). For the value of $c_v = 0.4$ the position error zero is reached after one meter, more or less, with a significant oscillation afterwards. The value of $c_v = 0.5$ achieves the position error zero a little bit later, and the oscillation is a bit smaller. The final solution is the value of $c_v = 0.425$, that reaches the position error zero almost as fast as with the value of $c_v = 4$, but with less oscillation. See Fig. (B.7).

It is noticed that there is oscillation after achieving position error zero, small or big, for every value tested. Even for high values, e.g. $c_v = 0.6$. This suggests that there is a minimal oscillation required. I think this is because of the small length of the truck and the long length of the trailer. That means that the position error zero can be reached quickly. But afterwards, it is necessary to stabilize the whole articulated vehicle. Then, the truck needs to move the wheels many times, to stabilize its long trailer. In Fig. (B.7) it is shown that the steering angle changes a lot after reaching the position error zero.

6.2 Calculation of the Error Bounds

In the previous section, the optimal values of c_v and c_p have been calculated. Now we will try to know the real performance of the vehicle. Several tests to know the real error of the vehicle will be done. The tests will be based on three steps. The first one is to set a path to follow in backwards motion. The paths will be straight lines and circles with different radiuses. The next step is to provide an initial error to the vehicle. That means that the values of position, orientation or bending angle will be modified. The latter step is to start the path following method and to observe the behavior.

There will be two kinds of test, the first one will set an initial error for a given parameter. It will test the influence of this parameter on the final result on its own. The second one is to set a combination of initial errors in two parameters. It will test if this situation increases the final result of the error in a significant way. All the tests will be done using the real model vehicle.

The objective is to obtain an estimate of the real errors of the vehicle. The main idea is to set an upper bound of those errors. With these bounds it is possible to confirm maximum error values. For a given initial error ξ less or equal to the initial error tested ξ_{tested} in some of the previous parameters, the maximum error of it ξ_{max} will be less or equal to the maximum error bound $\xi_{tested.bound}$.

6.2.1 Behavior of the Error with an Initial Error

This first test will evaluate the behavior of the vehicle with only one initial error. This error will be added to the position error ξ_y, orientation error ξ_{θ_1} or bending angle error $\xi_{\Delta\theta_{12}}$.

Structure of the experiment

The design of the test includes two trajectories to follow, for every initial error:

- A straight line.
- A circle with one meter radius.

For every trajectory, the next initial errors will be tested:

- Position error ξ_y = ± 10 cm and ± 20 cm.
- Orientation error ξ_{θ_1} = $\pm 5°$ and $\pm 10°$.
- Bending angle error $\xi_{\Delta\theta_{12}}$ = $\pm 5°$ and $\pm 10°$.

Results of the experiment

The test results of adding an initial position error are:

- Position error ξ_y in a range of $[-0.1, 0.1]$ meters, in a circular path:

$$|\xi_y| \leq 0.1m,\ |\xi_{\theta_1}| \leq 11°,\ |\xi_{\Delta\theta_{12}}| \leq 14°.$$

- Position error ξ_y in a range of $[-0.1, 0.1]$ meters, in a straight line path:

$$|\xi_y| \leq 0.1m,\ |\xi_{\theta_1}| \leq 11°,\ |\xi_{\Delta\theta_{12}}| \leq 21°.$$

- Position error ξ_y in a range of $[-0.2, 0.2]$ meters, in a circular path:

$$|\xi_y| \leq 0.2m,\ |\xi_{\theta_1}| \leq 19°,\ |\xi_{\Delta\theta_{12}}| \leq 30°.$$

- Position error ξ_y in a range of $[-0.2, 0.2]$ meters, in a straight line path:

$$|\xi_y| \leq 0.2m,\ |\xi_{\theta_1}| \leq 19°,\ |\xi_{\Delta\theta_{12}}| \leq 38°.$$

The highest initial errors are used to set an upper bound ($\xi_y = \pm 0.2m$). It is possible to confirm that for any value of $|\xi_y| \leq 0.2m$, the upper error bound for any type of path is:

$$|\xi_y| \leq 0.2m,\ |\xi_{\theta_1}| \leq 19°,\ |\xi_{\Delta\theta_{12}}| \leq 38°.$$

This value corresponds to the case of a straight line path. That means that an initial position deviation affects in a higher way to a straight line path. In this case the bending angle error $\xi_{\Delta\theta_{12}}$ is significantly high.

The test results of adding an initial orientation error are:

- Orientation error ξ_{θ_1} in a range of $[-5°, 5°]$, in a circular path:

$$|\xi_y| \leq 0.078m,\ |\xi_{\theta_1}| \leq 7°,\ |\xi_{\Delta\theta_{12}}| \leq 20°.$$

- Orientation error ξ_{θ_1} in a range of $[-5°, 5°]$, in a straight line path:

$$|\xi_y| \leq 0.09m,\ |\xi_{\theta_1}| \leq 8°,\ |\xi_{\Delta\theta_{12}}| \leq 31°.$$

- Orientation error ξ_{θ_1} in a range of $[-10°, 10°]$, in a circular path:

$$|\xi_y| \leq 0.07m,\ |\xi_{\theta_1}| \leq 10°,\ |\xi_{\Delta\theta_{12}}| \leq 33°.$$

- Orientation error ξ_{θ_1} in a range of $[-10°, 10°]$, in a straight line path:

$$|\xi_y| \leq 0.14m,\ |\xi_{\theta_1}| \leq 10°,\ |\xi_{\Delta\theta_{12}}| \leq 42°.$$

The highest initial errors are used to set an upper bound ($\xi_{\theta_1} = \pm 10°$). It is possible to confirm that for any value of $|\xi_{\theta_1}| \leq 10°$ the upper error bound for any type of path is:

$$|\xi_y| \leq 0.14m,\ |\xi_{\theta_1}| \leq 10°,\ |\xi_{\Delta\theta_{12}}| \leq 42°.$$

It is shown that for an initial orientation deviation, the motion on the circular path obtains smaller errors than on the straight line path. On a straight line path the value of $\Delta\theta_{12}$ must be zero. Then, an initial orientation deviation causes the increase of the bending angle, to reach the actual path position. Thus, the bending angle error $\xi_{\Delta\theta_{12}}$ becomes high.

The test results adding an initial bending angle error are:

- Bending angle error $\xi_{\Delta\theta_{12}}$ in a range of $[-5°, 5°]$, in a circular path:

$$|\xi_y| \leq 0.02m, \; |\xi_{\theta_1}| \leq 5°, \; |\xi_{\Delta\theta_{12}}| \leq 6°.$$

- Bending angle error $\xi_{\Delta\theta_{12}}$ in a range of $[-5°, 5°]$, in a straight line path:

$$|\xi_y| \leq 0m, \; |\xi_{\theta_1}| \leq 1°, \; |\xi_{\Delta\theta_{12}}| \leq 5°.$$

- Bending angle error $\xi_{\Delta\theta_{12}}$ in a range of $[-10°, 10°]$, in a circular path:

$$|\xi_y| \leq 0.02m, \; |\xi_{\theta_1}| \leq 3°, \; |\xi_{\Delta\theta_{12}}| \leq 10°.$$

- Bending angle error $\xi_{\Delta\theta_{12}}$ in a range of $[-10°, 10°]$, in a straight line path:

$$|\xi_y| \leq 0.012m, \; |\xi_{\theta_1}| \leq 1°, \; |\xi_{\Delta\theta_{12}}| \leq 10°.$$

The highest initial errors are used to set an upper bound ($\xi_{\Delta\theta_{12}} = \pm 10°$). It is possible to confirm that for any value of $|\xi_{\Delta\theta_{12}}| \leq 10°$ the upper error bound for any type of path is:

$$|\xi_y| \leq 0.02m, \; |\xi_{\theta_1}| \leq 3°, \; |\xi_{\Delta\theta_{12}}| \leq 10°.$$

We can notice that the initial bending angle error $\xi_{\Delta\theta_{12}}$ only affects a bit on the other errors. The orientation error $|\xi_{\theta_1}|$ can be maximum $3°$ and the position error $|\xi_y|$ is insignificant. The geometry of the truck is the responsible of those error values. The semi-trailer vehicles provide a quick bending angle correction. This is due to its small length between the rear axle of the truck and the hitch. It is also due to the trailer axle is located on the rear end of it.

In [Schwarz(2009)] similar error tests are shown. It is possible to see how the geometric differences between both trucks affect to the final error. It is shown that the results with an initial bending angle error $\xi_{\Delta\theta_{12}}$ are better for the semi-trailer. In contrast, we can notice that for an initial position error ξ_y the truck with one-axle trailer provides better results.

6.2.2 Behavior of the Error with a Combination of Initial Errors

In this section, the results of the combination of two initial errors will be described. The different combinations will be commented. Some combinations will affect negatively to the result, and others will affect positively to it. Some of them even improve the results of the case with one single initial error.

Structure of the experiment

The tests will be done for the combination of the next values:

- $\xi_y = \pm 0.1m$ and $\pm 0.2m$.
- $\xi_{\theta_1} = \pm 10°$.

Results of the experiment

According to the previous structure of the experiment, there are eight different combinations. All the results are shown in Table (B.3).

At first sight, extreme cases with high error values are shown. An example of them are the values of $\xi_y = -20m$ and of $\xi_{\theta_1} = +10°$, or the values of $\xi_y = -10m$ and of $\xi_{\theta_1} = +10°$. Those values are the result of a deviation of the position toward the center of the circle. In those cases, there is also a rotation of the vehicle in a way that the circle path is more difficult to reach. In general, we can notice that for combinations of positive values of ξ_y and negative values of ξ_{θ_1}, and vice versa, the results are quite worse than in the single initial error tests. In contrast, there are some combinations that improve the results. The values of $\xi_y = +20m$ and of $\xi_{\theta_1} = +10°$, or the values of $\xi_y = -20m$ and of $\xi_{\theta_1} = -10°$ are proof of that. Those cases take better results than the previous single initial error tests. This improvement is quite important for the bending angle error. It supposes a decrease of this error in $16°$. Similarly, the combination of values of both errors with the same sign takes better results.

Chapter 7

Conclusion

7.1 Summary

According to the initial ideas of the thesis, all the required objectives have been accomplished. First, the theory behind the kinematical behavior of the vehicles, especially of semi-trailers, was described. The differences between two types of vehicles, the semi-trailer and the truck with one-axle trailer, have been emphasized. It was commented that most of those differences are related to the possible situations in stable motion. The necessity of an adaptation of the previous path following method was explained in detail.

The new version of the path following method was also described. This adaptation provides support to all the new situations that the semi-trailer vehicle can afford. The previous path following method was explained in detail and the new extensions as well. Next, the software is described. Initially, the actual software is commented and the most important software components related to this thesis was emphasized. Then, the new software components are introduced. The integration of the new software was also explained in detail. The extensions of the software are focused on the implementation of the new path following method.

Furthermore, a complete description of the hardware of the semi-trailer model vehicle was done. The main components of it were described. The necessity of a calibration of the steering angle and bending angle was commented, and the process to calibrate them as well. In the end, once the vehicle is working properly, the experiments to test the quality of the method were done. The results in comparison with the previous model vehicle were highlighted, and the main evaluation was done.

7.2 Appraisal

The best way to assess this thesis is to compare it with the previous one, [Schwarz(2009)]. In both thesis, there are some similar experiments to determine the quality of the path following method. The final result of the complete work is better if it provides a small error in the path following motion. But a high error does not necessary mean that the path following method is not optimal. There are a lot of causes that affect to the final result. Some causes can be the calibration of steering angle or bending angle, the laser scanner quality, etc. So, it is necessary to consider all those causes to evaluate the final result.

In a first approach, if we compare the results of the error tests between the truck with one-axle trailer and the semi-trailer, it is shown that the results of the new vehicle are not worse. Furthermore, sometimes they are better than the same result of the truck with one-axle trailer test, despite the higher length of the semi-trailer. The calibration method takes an important part in these results. In [Schwarz(2009)], the calibration method was different, with little similarities with the method used in this thesis. It is also mentioned that there was the necessity to recalibrate several times. In this thesis the calibration of all elements of the hardware was an important task. So, the processes and components to achieve it were studied in detail.

Another element that affects to the final result is the laser scanner quality. The semi-trailer model vehicle has got a newer laser scanner model than the version installed on the truck with one-axle trailer model vehicle. This affects to the calculation of the position. In the previous model vehicle, this calculation is a little bit slower. However, there are cases where the truck with one-axle trailer provides better results. These cases correspond to an initial position error regarding to a given path. This is a normal result, due to the geometry of the semi-trailer vehicle. This semi-trailer has a long trailer and a short truck. In contrast, the semi-trailer corrects quicker the bending angle errors.

The final behavior of the vehicle in path following motion depends on many factors. The path following method determines the movements that the vehicle needs to do. The software components are responsible to implement correctly this method. It is also responsible to provide the proper communication between host and target. Finally, the hardware needs drive according to the orders received by the software. So, it applies changes on its actuators to provide the required motion.

Due to those factors, a problem in any of them supposes the wrong working of the vehicle. In this thesis, all these factors have been handled properly. That means that the final precision of the vehicle movements are really impressive. To prove it, this thesis has been completed with a video that shows the real working of the vehicle. In this video, the vehicle does different maneuvers. The most important are the parallel parking and the perpendicular parking driving backwards.

We can think about the next projects of the working group. This projects could be based on this thesis, on [Schwarz(2009)], or on another works of the group. Some future works could be focused on extending the path following method to other type of vehicles, e.g. trucks with one-steerable-axle trailer. Another possibility is to work on a path planning method, to reach a position from another one. Thus, the initial step to achieve autonomous motion is to apply the path planning method. Afterwards, the path following method can be used to track this path.

Bibliography

[Altafini(2001)] ALTAFINI, W. (2001). Some properties of the general n-trailer. In: *International Journal of Control*.

[Berg(2009)] BERG, U. (2009). *Entwicklung von Rückfahrassistenzsystemen für Fahrzeuge mit Starrdeichselanhänger oder Sattelanhänger*. Ph.D. thesis.

[Bolzern *et al.*(1998)Bolzern, DeSantis, Locatelli & Masciocchi] BOLZERN, P., DESANTIS, R., LOCATELLI, A. & MASCIOCCHI, D. (1998). Path-tracking for articulated vehicles with off-axle hitching. In: *IEEE Transactions on Control Systems Technology*, vol. 6.

[Bolzern & Locatelli(2002)] BOLZERN, P. & LOCATELLI, A. (2002). A comparative study of different solutions to the path-tracking problem for an articulated vehicle. In: *IEEE International Conference of Control Applications*.

[Eckert(2007)] ECKERT, J. (2007). Konstruktive auslegung des aufbaus eines lkw-modells im maßstab 1:16 für untersuchungen im bereich des teilautonomen fahrens. University Koblenz-Landau, Student work.

[Pradalier & Usher(2007)] PRADALIER, C. & USHER, K. (2007). A simple and efficient control scheme to reverse a tractor-trailer system on a trajectory. In: *2007 IEEE International Conference on Robotics and Automation*.

[Pradalier & Usher(2008)] PRADALIER, C. & USHER, K. (2008). Robust trajectory tracking for a reversing tractor trailer. In: *Journal of Field Robotics*, vol. 25.

[Preissler(2001)] PREISSLER, A. (2001). Heuristische ansätze zur regelung eines fahrzeugs mit anhänger. University Koblenz-Landau, Diplom-Thesis.

[Schwarz(2009)] SCHWARZ, C. (2009). Entwicklung eines regelungsverfahrens zur pfadverfolgung für ein modellfahrzeug mit einachsigem anhänger. University Koblenz-Landau, Diplom-Thesis. URL `http://kola.opus.hbz-nrw.de/volltexte/2009/395/pdf/diplomarbeit_schwarz.pdf`.

[Wojke(2005)] WOJKE, P. (2005). Ezauto - softwarearchitektur für anwendungen des assistierten und autonomen fahrens **10**.

[Zöbel(2001)] ZÖBEL, D. (2001). Mathematical modeling of the kinematics of vehicles. In: *Mathematical Modeling of Technical Processes* (HRUBINA, K., ed.). Socrates/Erasmus Summer School, Prevsov, Slovak Republic.

[Zöbel(2003)] ZÖBEL, D. (2003). Trajectory segmentation for the autonomous control of backward motion for truck and trailer **4**(2), 59–66.

[Zöbel & Weyand(2008)] ZÖBEL, D. & WEYAND, C. (2008). On the maneuverability of heavy goods vehicles. In: *2008 IEEE International Conference on Systems, Man, and Cynernetics (SMC 2008)*. Singapore.

Appendix A

Calibration Results

A.1 Calibration of the Steering Angle

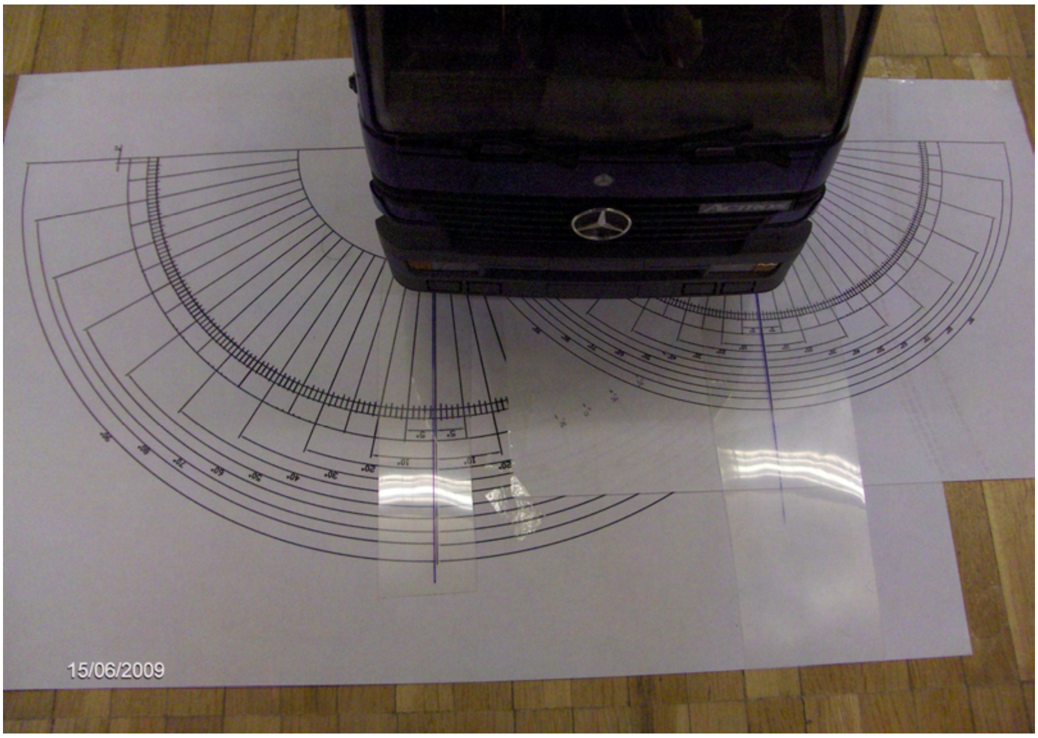


Figure A.1: Protractor for the steering angle calibration.

Figure A.2: Pointer on the wheels.

A.2 Calibration of the Bending Angle

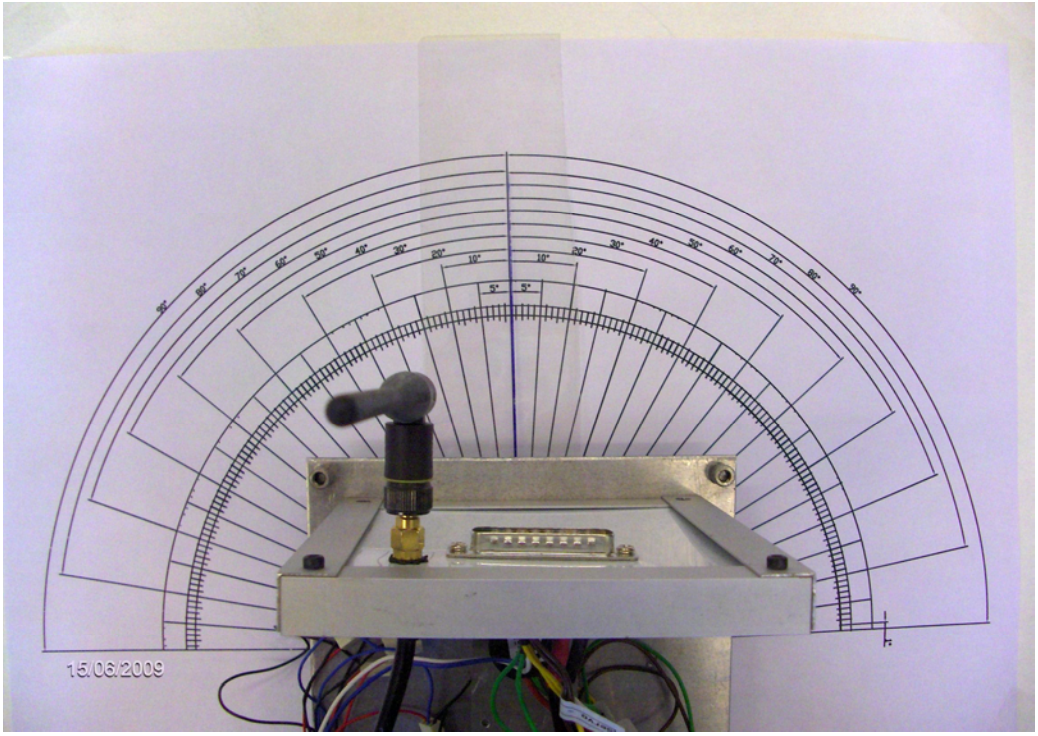


Figure A.3: Start position of the bending angle measuring process.

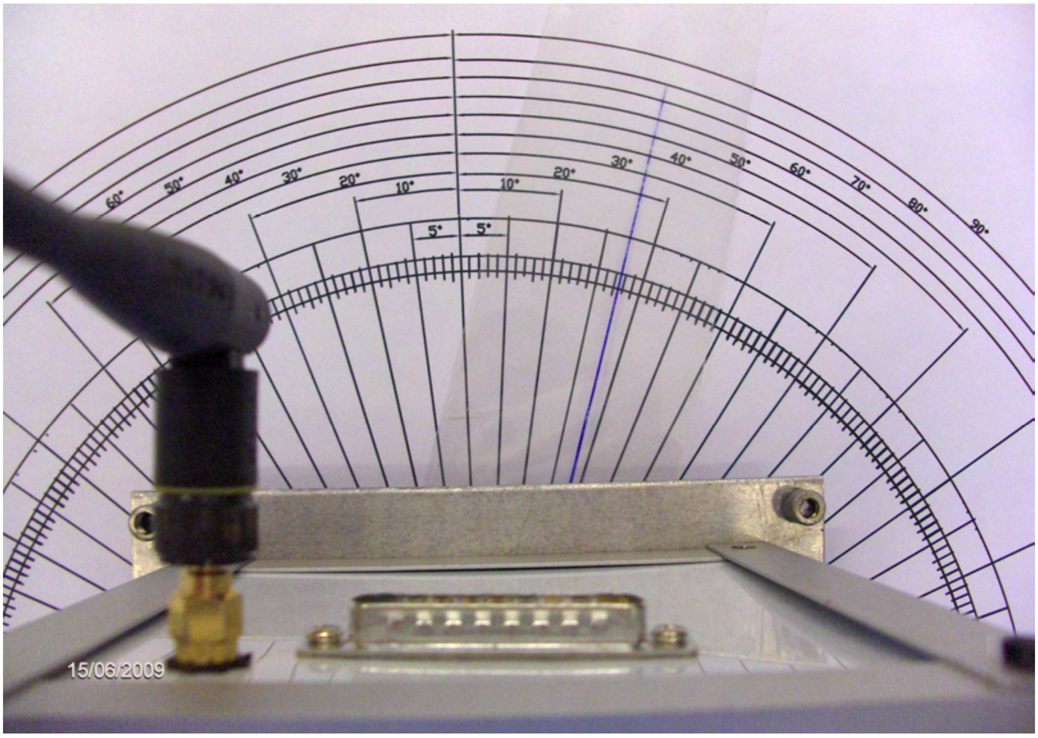
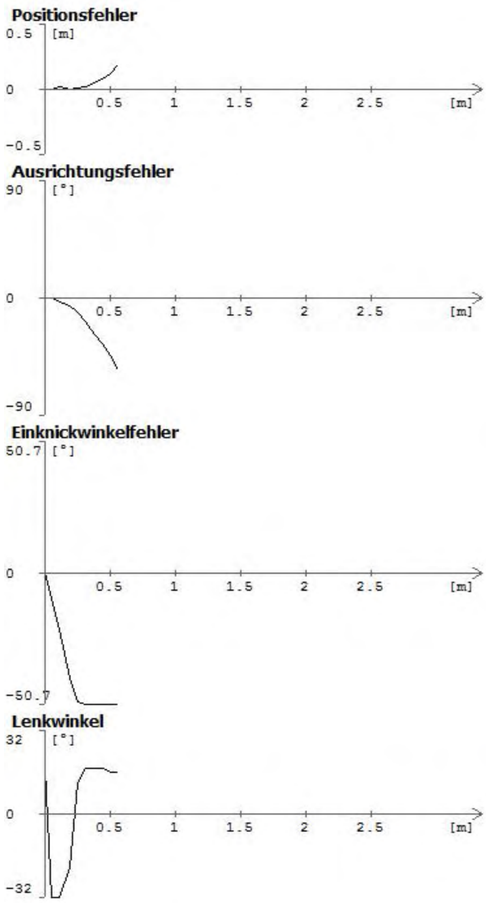


Figure A.4: Measuring process.

Appendix B

Results of the Driving Experiments

B.1 Determination of c_p



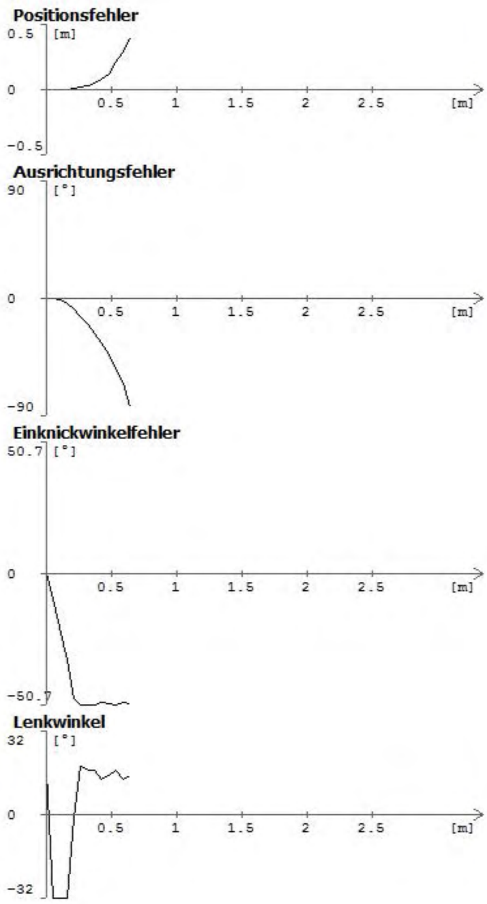


Figure B.1: First test, with $cp = 3$ and $cp = 5$

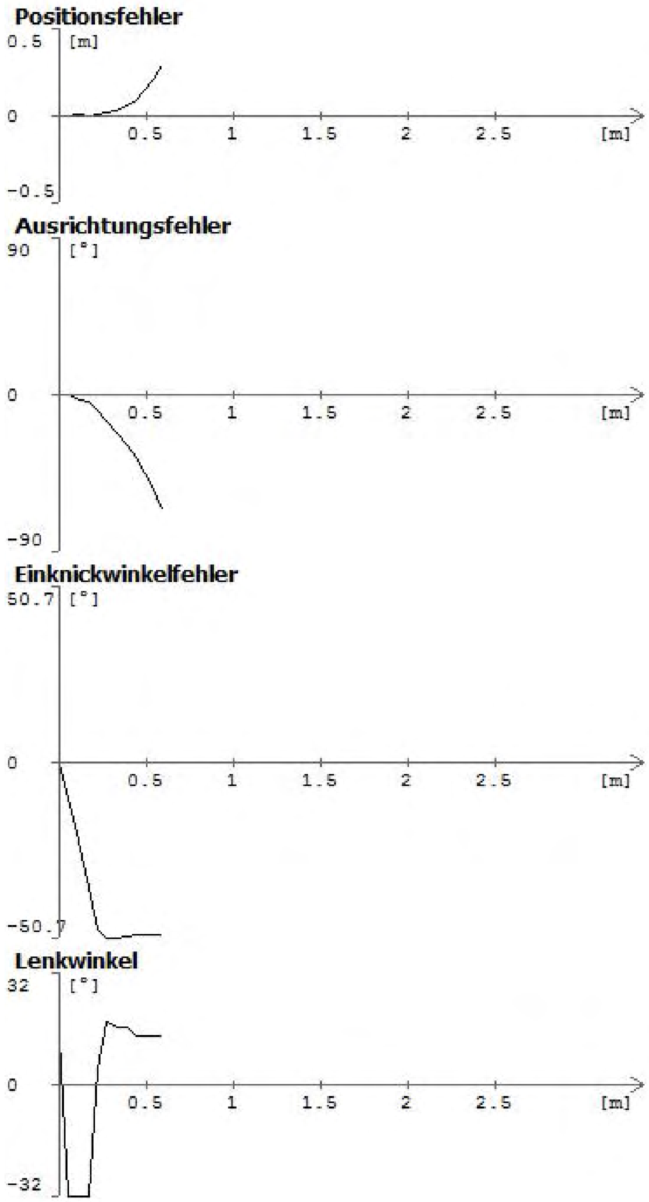
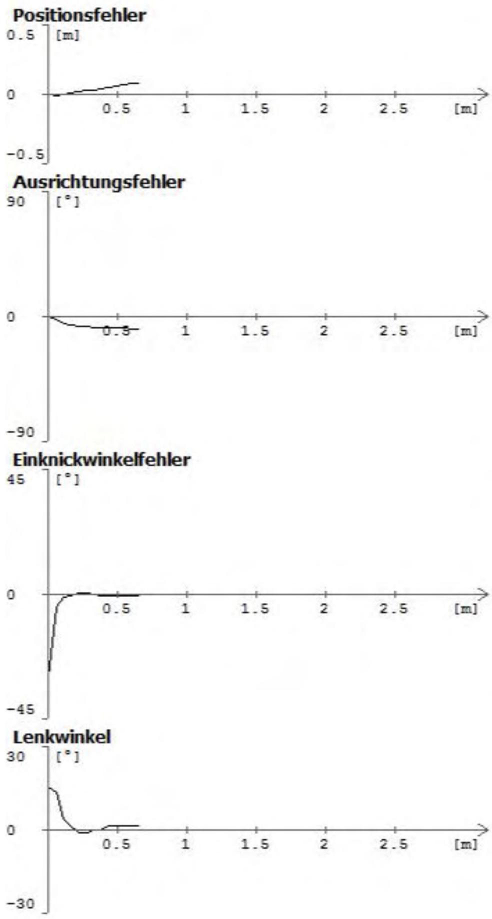


Figure B.2: First test, with $cp = 4$



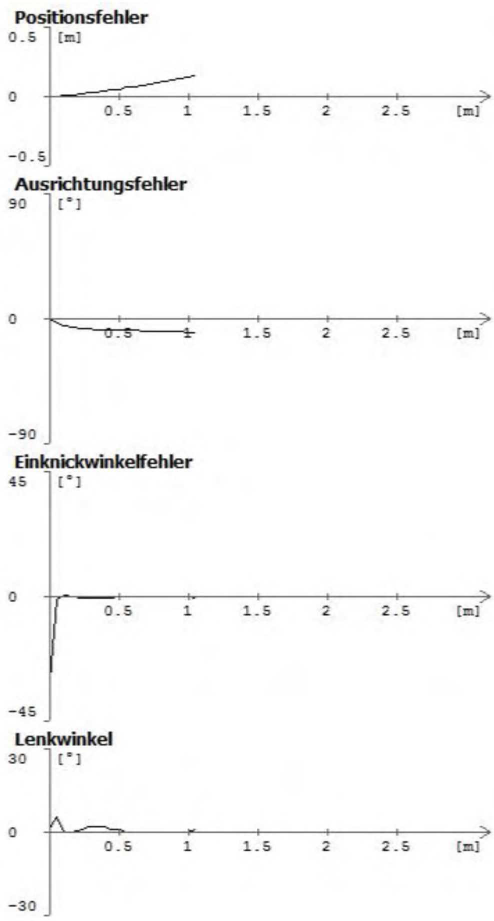


Figure B.3: Second test, with $cp = 2$ and $cp = 3$

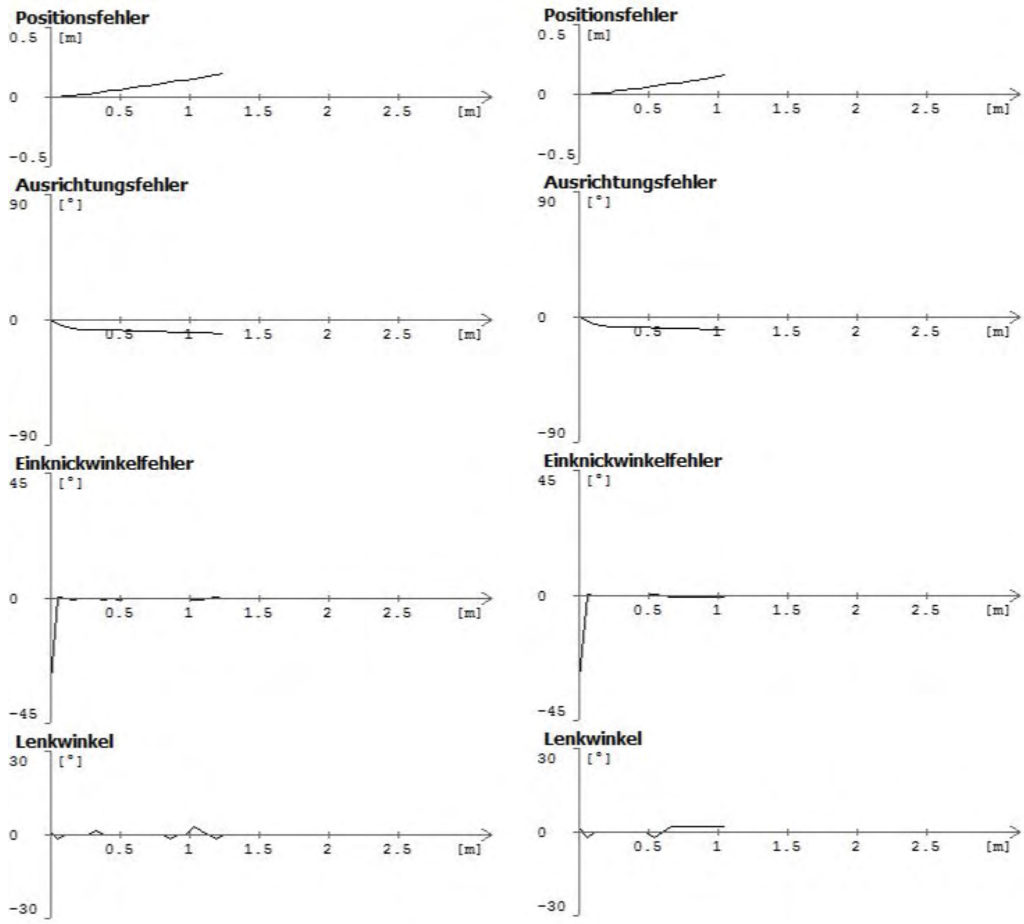


Figure B.4: Second test, with $cp = 4$ and $cp = 5$

B.2 Determination of c_v

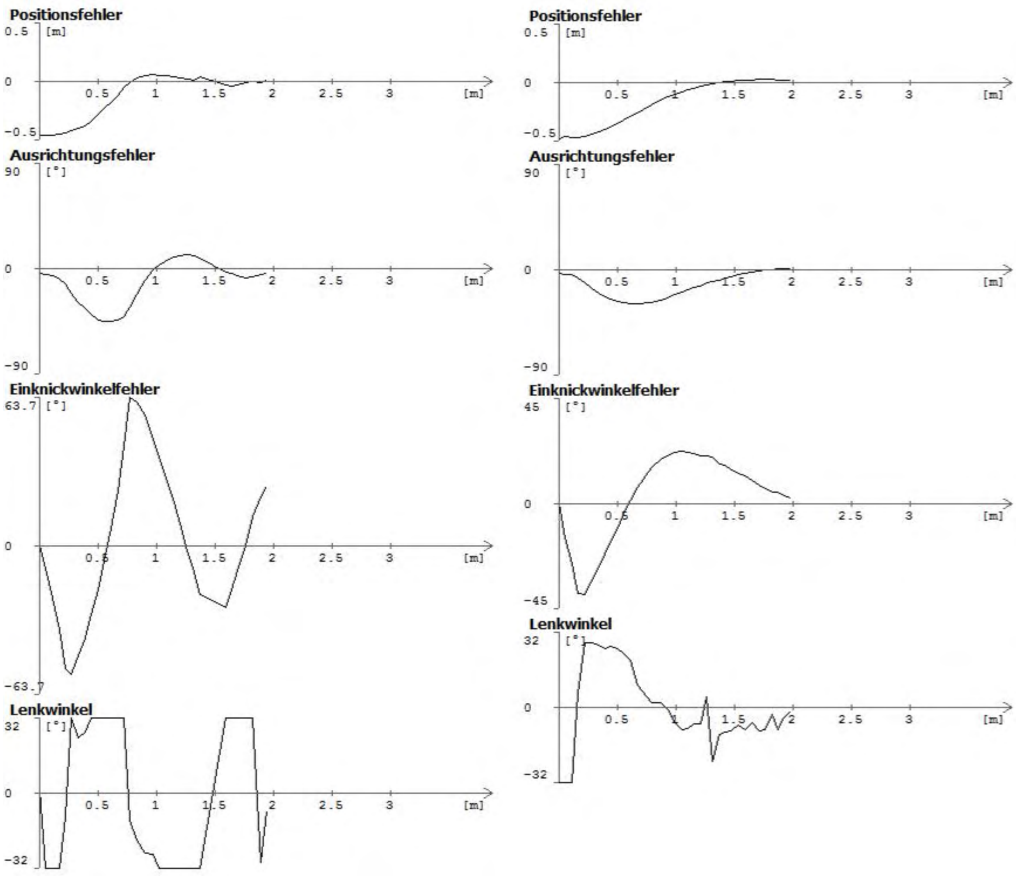


Figure B.5: $c_v = 0.3$ and $c_v = 0.6$

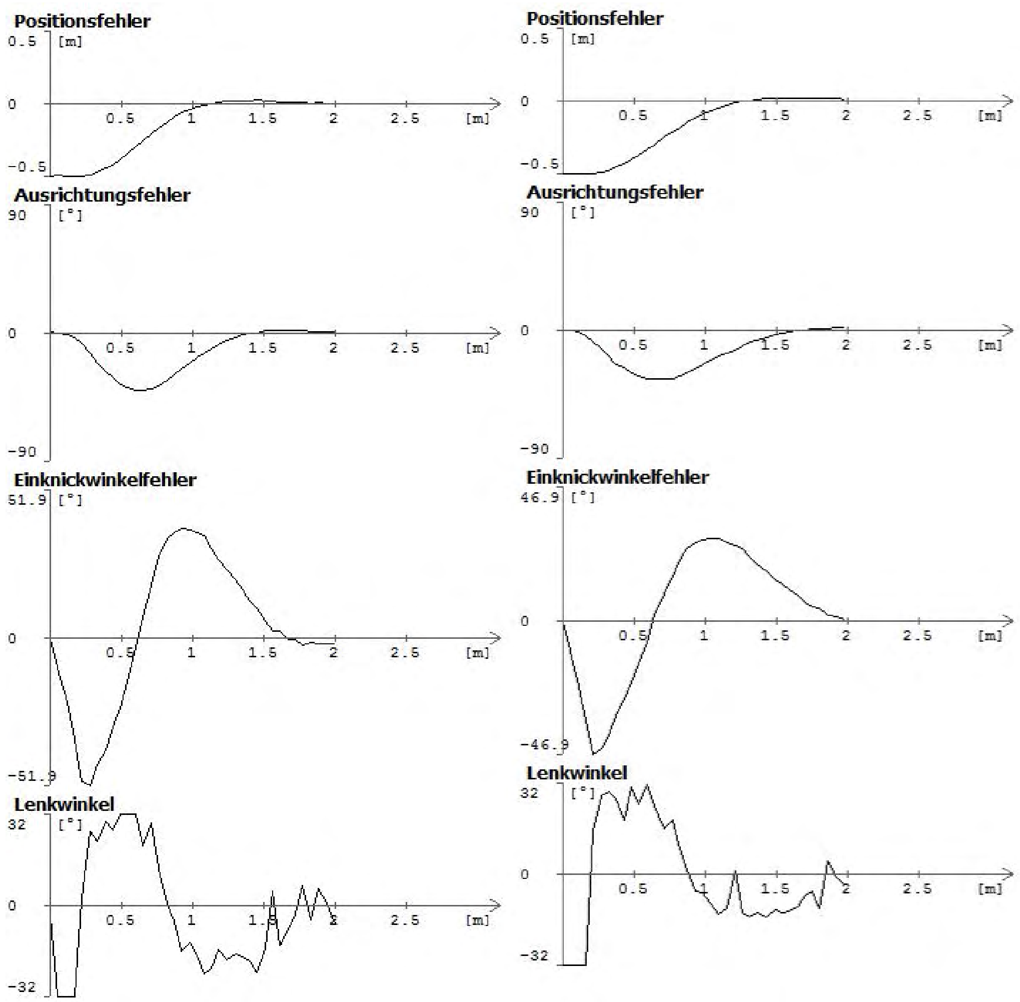


Figure B.6: $c_v = 0.4$ and $c_v = 0.5$

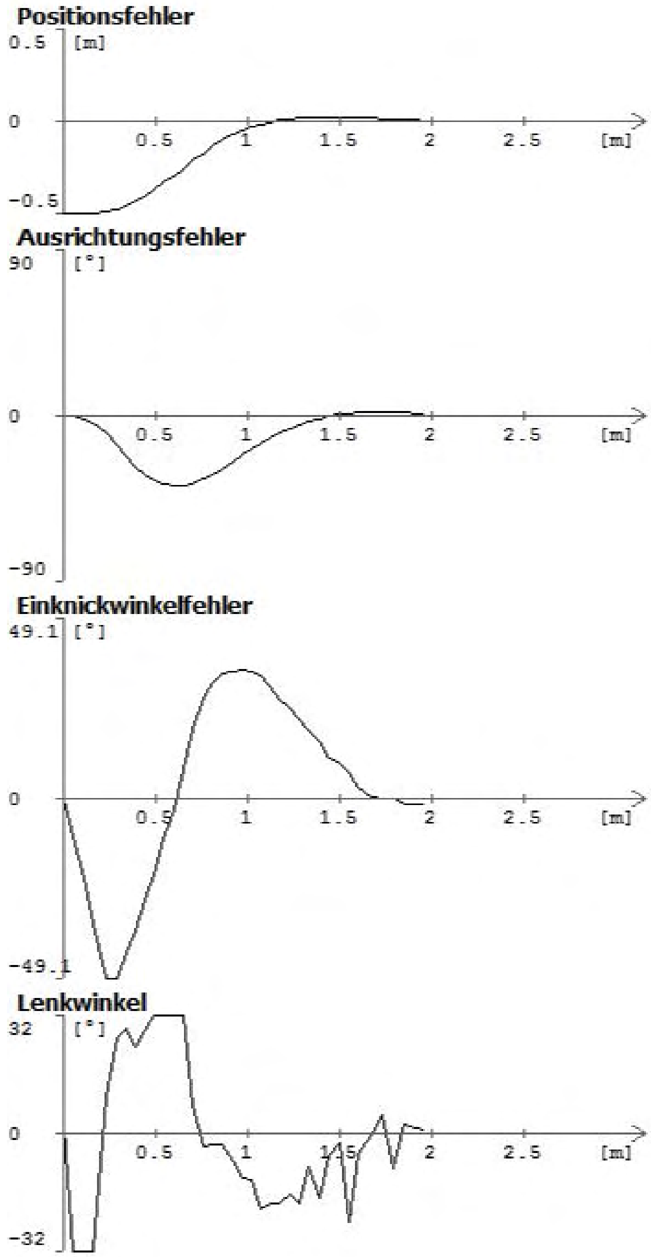


Figure B.7: The best value, $c_v = 0.425$

B.3 Results of a Combiantion of Initial Errors

ξ_y	ξ_{θ_1}	$\xi_{y,max}$	$\xi_{\theta_1,max}$	$\Delta\theta_{12,max}$
+0.1	+10	0.042	14	31
-0.1	+10	0.2	25	49
+0.2	+10	0.14	14	14
-0.2	+10	0.26	29	50
+0.1	-10	0.16	16	18
-0.1	-10	0.1	10	14
+0.2	-10	0.28	22	24
-0.2	-10	0.2	17	18

Table B.1: Combination of initial errors. Length in meters and angles in degrees ($^\circ$).