ΠP periodica polytechnica

Electrical Engineering
52/3-4 (2008) 137–144
doi: 10.3311/pp.ee.2008-3-4.02
web: http://www.pp.bme.hu/ee
© Periodica Polytechnica 2008

RESEARCH ARTICLE

Designing a tracking controller for passenger cars with steering input

Emese Szádeczky-Kardoss / *Bálint* Kiss

Received 2008-07-22

Abstract

The results presented in the paper were motivated by a practical application. The goal was to design a parking assist system for a passenger car such that the assist is realized by controlling the steering wheel while the human driver handles the pedals (throttle, clutch and brake pedals) of the vehicle and thus generates its velocity. The tracking control of the car with the single steering input is achieved by a novel time-scaling based controller. The solution was tested in simulations and also in a real vehicle.

Keywords

automatic vehicle control · parking assist system · tracking control · time-scaling · rapid prototyping

Acknowledgement

The authors gratefully acknowledge the comments and suggestions of the reviewer that helped to improve the quality of the paper.

The research was partially funded by the Hungarian Science Research Fund under Grant OTKA K71762 and by the Advanced Vehicles and Vehicle Control Knowledge Center under grant RET 04/2004. We thank the precious support of the research engineers of ThyssenKrupp Presta Hungary Kft.

Emese Szádeczky-Kardoss
Department of Control Engineering and Information Technology, BME, Magyar Tudósok krt. 2, H-1117 Budapest, Hungary
e-mail: szadeczky@iit.bme.hu

Bálint Kiss
Department of Control Engineering and Information Technology, BME, Magyar Tudósok krt. 2, H-1117 Budapest, Hungary
e-mail: bkiss@iit.bme.hu

1 Introduction

This paper presents novel results on the field of automatic vehicle control. The research was motivated by a practical application. The goal was to build a parking assist system (PAS) for a passenger car with a human driver.

An important goal of the automatic vehicle control is to improve safety and driver's comfort. PAS provides this for parking maneuvers which are usually performed at low velocities. PAS collects first information about the environment of the vehicle (position of the obstacles) which are necessary to find a parking lot and to complete a safe parking maneuver [10]. It detects the existence of an accessible parking place where the vehicle can park into. Once the parking place is identified, a feasible path geometry has to be designed and the tracking of this reference is realized by controlling the steering wheel angle.

One may distinguish fully or semi-automated PASs according to the driver's involvement during the maneuver. In the fully automatic case, PAS influences both the steering angle and the longitudinal velocity of the car. In the case of vehicles without automatic gear, the semi-automatic PAS is the only available option such that the driver needs to generate the longitudinal velocity of the car with an appropriate management of the pedals while PAS controls the steering wheel.

There exists today several PASs on the market. The Aisin Seiki Co. Ltd. has developed a parking system for Toyota [1] where a camera observes the environment of the car. The Evolve project resulted a fully automatic PAS for a Volvo type vehicle based on ultrasonic sensor measurements [6]. The Volkswagen Touran may be ordered with an option that also assists parking maneuvers [17]. This solution also uses ultrasonic sonars for the semi-automatic parking maneuvers. Some of these systems operate in open loop (e.g. [6]), or the aim of the control is not to track a reference path but to reach the parking position [11]. Our goal is to design a tracking controller for a semi-automated PAS.

To design a PAS, a mathematical model of the car has to be given. Several vehicle models are presented in the literature including kinematic and dynamic ones. Let us suppose that the kinematic models such as the ones reported in [2, 5, 13] describe in a satisfactory way the behavior of the vehicle at low velocities where parking maneuvers are executed.

If the model of the vehicle is known then the tracking controller algorithm can be designed based on the equations of motion.

In case of semi-automatic systems, the design of the tracking controller is more involved than in the fully automatic case. The tracking controller in the fully automatic case may influence the behavior of the car using two inputs (see e.g. [3]) whereas one loses one input, namely the longitudinal velocity in the semi-automated case where the car velocity is generated by the driver hence the velocity profile during the execution of the parking maneuver may be considerably different from the one used for the path planning.

Our solution for the single input control problem is based on a new time-scaling scheme [4, 8]. This means that the path planning method generates a reference path parameterized by a virtual time then a time-scaling function is used which maps this virtual time into the real time. This time-scaling function depends on the measured car velocity and on the tracking error along the path.

To obtain the time-scaling function, we transform the one input model of the car into a two input model using again time-scaling such that a new (scaling) input is created for the system evolving according to the virtual time. Using this concept, the tracking controller has again two outputs. In other words, if we are not able to control the velocity of the car, we influence the time distribution of the reference path instead.

Our tracking control method with time-scaling is based on the flatness property of the two input kinematic car model [7]. We implemented the method using Matlab and Simulink. Several simulations were performed to verify the functioning of the closed loop system. We also tested our tracking controller in a real Ford Focus type car where we used the fast prototyping environment of dSPACE (AutoBox and ControlDesk) for the implementation and monitoring.

The remaining part of the paper is organized as follows. First, the components of PAS are presented. Then the problem of the single input vehicle model is discussed. The novel time-scaling scheme is given in Section 4. Section 5 presents our time-scaling based tracking control method with some results. The implementation and test results of the whole PAS are described in Section 6. Finally, a summary concludes the paper.

2 Components of PAS

To ensure autonomous behavior, several tasks have to be solved: the system should be able to detect obstacles in its environment; it has to measure or estimate its position and orientation; the reference motion has to be planned; and finally, this reference should be tracked as accurately as possible. These tasks are performed by separate interconnected subsystems which are depicted in Fig. 1.

ABS (Anti-lock Braking System) sensors of the vehicle can

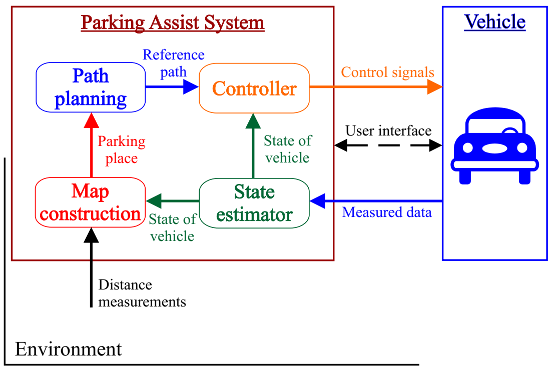


Fig. 1. Components of the parking assist system

detect the displacement of the wheels of the car. Based on these data, an estimator calculates the actual position and orientation of the car in a fixed world coordinate frame. This estimated state is used by the map construction and controller modules. To draw a map, additional data are also required about the environment. Ultrasonic sensors are used to measure the distances to the surrounding obstacles. Based on these distance measurements, a map can be created. One may then use simple algorithms to detect accessible parking places (if any) on the map [12].

During the motion planning, a reference path is calculated which connects the initial and the desired final configurations in one step (i.e. without changing the driving direction). In this planning phase, some constraints (e.g. the non-holonomic behavior described by the model, collision avoidance, maximal values of the actuator signals) have to be taken into consideration [15, 16]. Finally, the tracking control algorithm is used to track the reference path. This paper presents the tracking controller method in details.

Additionally, there is a graphical user interface between our system and the driver.

3 Single input and two input models

At the design phase of PAS, we had to face an interesting theoretical problem, namely the vehicle had to be controlled such that the longitudinal velocity is measured, but generated by the driver and it cannot be influenced by the controller. Hence the angle of the steering wheel remains the single output of the controller. The tracking control problem of such a system has not been addressed yet in the literature.

In the sequel, we discuss some fundamental differences between the two input and the single input kinematic models of the vehicle.

For the controller design, the kinematic model of the car was used. The state q of the vehicle is described by the position of its reference point R, which is the midpoint of the rear axle and by the orientation. These are denoted by x, y and θ (see Fig. 2). The car has usually two inputs, namely the steering angle of the front wheels φ and the longitudinal velocity v.

Suppose, that the Ackermann steering assumption holds true, which means that all wheels (i.e. both the non-steerable wheels

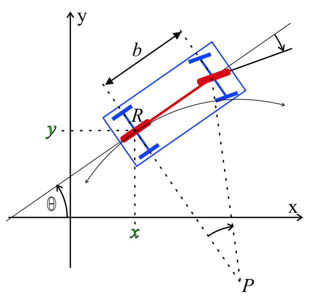


Fig. 2. Kinematic model of the vehicle in the x–y plane

on the rear axle and the steerable front wheels) turn around one point, which lies along the rear axle (see P in Fig. 2).

The movement of the car is described by the motion of a bicycle fitted to the longitudinal symmetry axis of the vehicle:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \frac{v}{b}\tan\varphi \end{bmatrix} \tag{1}$$

where b denotes the wheelbase of the vehicle.

In case of the semi-automatic system, the controller cannot consider the longitudinal velocity of the vehicle as a system input since it is generated by the driver and cannot be influenced by the controller. In this case, we denote this driver velocity by v_{car}. (We suppose, that this v_{car} velocity can be measured or well estimated.) Hence the kinematic model in the single input case reads

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_{car}\cos\theta \\ v_{car}\sin\theta \\ \frac{v_{car}}{b}\tan\varphi \end{bmatrix} \tag{2}$$

If the velocity is a control input as in (1), the system has some useful properties which can be used at the controller design. Some of these properties are not valid any more for the single input case (2). For example, such a property is the differentially flatness of the system (1) (see e.g. [7]) which implies that the model with two inputs can be linearized by a dynamic feedback, however the single input system is not differentially (but orbitally) flat.

There are several solutions in the literature for the tracking controller problem of the kinematic car with two inputs. However, the single input problem has not been addressed yet for arbitrary non-vanishing v_{car}.

4 Novel time-scaling scheme

During the path planning, a preliminary reference velocity profile (v_{ref}) is considered. Since the driver will generate another velocity profile ($v_{car} \neq v_{ref}$), it is enough if one is able to track the geometry of the reference of the designed path and the time distribution along the path will be adapted in real-time to the velocity profile generated by the driver. First, we introduce the time-scaling concept in general.

In the above equations, the states of the configuration $q \in \mathbb{R}^n$ were functions of time t, where $\dot{t} = 1$. In a more general form, we have a state equation

$$\dot{q}(t) = f(q(t), u(t), w(t)) \tag{3}$$

where $u \in \mathbb{R}^m$ is the vector of the control inputs and $w \in \mathbb{R}^k$ denotes the external signals. In the case of the kinematic car models, which are given by (1) and (2), we have $q = [x, y, \theta]^T$ and $n = 3$. For the car (1) with two inputs $u = [v, \varphi]^T$, $m = 2$ and $k = 0$. For the single input case (2) $u = \varphi$, $m = 1$ and $w = v_{car}$, $k = 1$.

The longitudinal velocity v is not a control input any more in the single input case. We suggest to use a novel time-scaling based scheme to compensate this lost control input.

We introduce a new scaled time, denoted by τ, such that τ is used to modify the time distribution along the path. We suggest that the relationship between t and τ should not only depend on the states of the car (as it is usually done in the literature [14]), but also on the external signal w and on a new input, denoted by $u_s \in \mathbb{R}$ which is the so-called scaling input. The instantaneous value of u_s can be calculated by a controller based on the closed loop behavior of the system. (Details are presented in the next section for PAS.)

Definition 4.1 *Let τ denote a new, virtual time, such that the relationship between the real time t and τ is given by the following time-scaling function:*

$$\frac{dt}{d\tau} = \left(\frac{d\tau}{dt}\right)^{-1} = \frac{1}{\dot{\tau}} = g(q, u_s, w) \tag{4}$$

Using this time-scaling, (3) can be expressed with respect to the time τ:

$$q' = \frac{dq}{d\tau} = \frac{dq}{dt}\frac{dt}{d\tau} = g(q, u_s, w) f(q, u, w) \tag{5}$$

(The prime denotes differentiation according to τ, hence $\tau' = 1$.) In this scaled model, the number of inputs is increased to $m + 1$ since both the input of the original system (i.e. u) and the scaling input u_s are input signals.

The time-scaling defined in (4) has to satisfy some conditions:

- $\tau(0) = t(0) = 0$, since the original and the scaled trajectories should start from the same initial configuration;
- $\dot{\tau} > 0$, since time cannot stop or rewind.

Suppose that a path planner has designed a reference path in the virtual time τ, i.e. the following mappings are given:

$$\tau \rightarrow \{x_{ref}(\tau), x'_{ref}(\tau), x''_{ref}(\tau), x'''_{ref}(\tau)\} \tag{6}$$
$$\tau \rightarrow \{y_{ref}(\tau), y'_{ref}(\tau), y''_{ref}(\tau), y'''_{ref}(\tau)\} \tag{7}$$

During the time-scaling, we modify the time distribution along this reference path to get the reference in the real time t:

$$x_{ref}(t) = x_{ref}(\tau) \tag{8}$$
$$\dot{x}_{ref}(t) = x'_{ref}(\tau)\dot{\tau} \tag{9}$$
$$\ddot{x}_{ref}(t) = x''_{ref}(\tau)\dot{\tau}^2 + x'_{ref}(\tau)\ddot{\tau} \tag{10}$$
$$x^{(3)}_{ref}(t) = x'''_{ref}(\tau)\dot{\tau}^3 + 3x''_{ref}(\tau)\dot{\tau}\ddot{\tau} + x'_{ref}(\tau)\tau^{(3)} \tag{11}$$

The further derivatives and the other state variables can be scaled in a similar way. It can be seen from (8)–(11) that the time-scaling does not change the geometry of the reference path, only the velocity and the higher derivatives are modified if $\dot{\tau} \neq 1$.

5 Time-scaling based tracking controller

In this section, we present first the key idea of our controller method (see Subsection 5.1), and then we give a flatness based solution (see Subsection 5.2). In [9] we also describe another tracking controller using the same idea of the time-scaling based control. (The controller in [9] is a state feedback, designed for the linearized error dynamics.)

5.1 The idea of the time-scaling based solution

The literature suggests several solutions [3, 7] to control the two input kinematic car given in (1). These methods ensure exponential tracking of the reference path. In our one input case, these algorithms cannot be used without modification since our controller cannot influence the velocity of the vehicle. Our idea is to complement the lost velocity input by the time-scaling input as in (5).

For, the following time-scaling function can be used:

$$\frac{dt}{d\tau} = \frac{u_s}{v_{car}} \tag{12}$$

In this case, the model equation given in (2) which evolves according to t can be transformed using the time-scaling, and we get

$$q' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} u_s \cos\theta \\ u_s \sin\theta \\ \frac{u_s}{b}\tan\varphi \end{bmatrix} \tag{13}$$

This scaled model has now two inputs (u_s and φ), and it is again differentially flat. Hence one of the controllers described in the literature can be used for tracking. The selected method will compute φ and u_s according to the tracking error and its time derivatives between the real and the scaled reference trajectories. This φ input is used to control the steering system while the scaling input u_s influences the time-scaling.

The time-scaling function and its derivatives, which are required for (8)–(11), can be calculated using the following relationships, which are based on (12):

$$\begin{aligned} \tau(t) &= \int_0^t \frac{v_{car}}{u_s} d\vartheta\ , \quad \tau(0) = t(0) = 0 && (14)\\ v_{car} &= \dot{\tau} u_s && (15)\\ \dot{v}_{car} &= \ddot{\tau} u_s + \dot{\tau}\dot{u}_s && (16)\\ \ddot{v}_{car} &= \tau^{(3)} u_s + 2\ddot{\tau}\dot{u}_s + \dot{\tau}\ddot{u}_s && (17) \end{aligned}$$

If the signs of u_s and v_{car} are the same then the time-scaling function satisfies the $\dot{\tau} > 0$ condition. If one of the two signals equals 0, the car is not controllable. This occurs at the very beginning and at the end of the motion, if it connects two idle configurations.

The scheme of the closed loop control is depicted in Fig. 3. First, the path planning module calculates the reference path in τ. In the next step, this reference is scaled based on the longitudinal velocity of the car v_{car}, which is generated by the driver, and on the scaling input u_s, which is calculated by the controller. After the time-scaling, we have the scaled reference in t. The controller determines its outputs using the difference between the real and the scaled reference trajectories. So the inputs of the vehicle are the longitudinal velocity v_{car} generated by the driver, and φ, which is calculated by the controller. The output of the car is the position of the reference point R and the orientation.

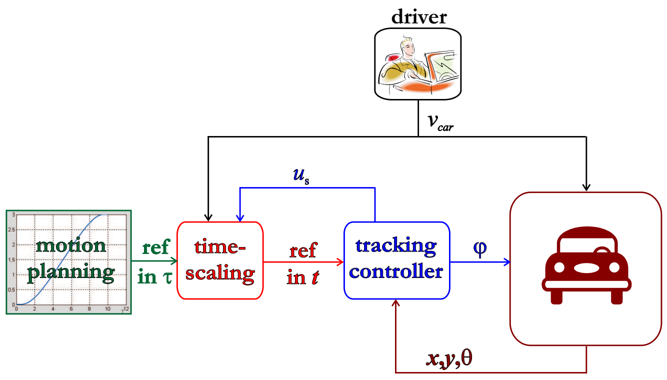


Fig. 3. Scheme of the tracking controller with time-scaling

5.2 Flatness based controller using time-scaling

This tracking controller is designed using the differentially flatness property of the time-scaled model (13). By virtue of this property, one can apply a linearizing feedback such that the resulting system is two chains of integrators

$$x^{(3)} = \omega_x, \qquad y^{(3)} = \omega_y \tag{18}$$

Suppose moreover that one specifies the tracking behavior in terms of the tracking errors $e_x = x - x_{ref}(t)$ and $e_y = y - y_{ref}(t)$ such that the differential equations

$$\begin{aligned} e_x^{(3)} + k_{x,2}\ddot{e}_x + k_{x,1}\dot{e}_x + k_{x,0}e_x &= 0 && (19)\\ e_y^{(3)} + k_{y,2}\ddot{e}_y + k_{y,1}\dot{e}_y + k_{y,0}e_y &= 0 && (20) \end{aligned}$$

hold true. The coefficients $k_{a,i}$ ($a \in \{x, y\}$, $i = 0, 1, 2$) are design parameters and have to be chosen such that the corresponding characteristic polynomials have all their roots in the left half of the complex plane.

Define first the dynamics of the feedback as

$$\dot{\zeta}_1 = \zeta_2 = \dot{u}_s, \qquad \dot{\zeta}_2 = v_1 = \ddot{u}_s, \qquad \dot{\zeta}_3 = v_2 \tag{21}$$

$$u_s = \zeta_1, \qquad \varphi = \zeta_3 \tag{22}$$

where ζ_1, ζ_2, and ζ_3 are the inner states of the feedback and u_s, φ denote the outputs of the controller. Observe that ζ_2 and v_1 give precisely the derivatives of u_s which need to realize the time-scaling in (14)–(17), hence no numerical differentiation is needed.

The inputs v_1 and v_2 of the feedback dynamics must be determined such that the tracking errors e_x and e_y satisfy (19) and (20), respectively.

For, one needs to determine first $\dot{x}$, $\ddot{x}$, $x^{(3)}$, and $\dot{y}$, $\ddot{y}$, and $y^{(3)}$ as functions of v_1, v_2, and x, y, θ, ζ_1, ζ_2, ζ_3 which are the states of the closed loop system including the measured states of the kinematic car model, and the states of the feedback (21)–(22). For the calculations, the scaled two input model (13) is used. After some cumbersome but elementary differentiations, one obtains

$$\dot{x} = x'\dot{\tau} = \dot{\tau}\zeta_1\cos\theta \tag{23}$$

$$\ddot{x} = \dot{\tau}\zeta_2\cos\theta - \frac{\dot{\tau}^2\zeta_1^2\sin\theta\tan\zeta_3}{b} + \ddot{\tau}\zeta_1\cos\theta \tag{24}$$

$$x^{(3)} = \begin{bmatrix} \dot{\tau}\cos\theta & -\frac{\dot{\tau}^2\zeta_1^2\sin\theta}{b\cos^2\zeta_3} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + A \tag{25}$$

with

$$A = -\frac{3\dot{\tau}^2\zeta_1\zeta_2\sin\theta\tan\zeta_3}{b} - \frac{\dot{\tau}^3\zeta_1^3\cos\theta\tan^2\zeta_3}{b^2} - \frac{3\dot{\tau}\ddot{\tau}\zeta_1^2\sin\theta\tan\zeta_3}{b} + 2\ddot{\tau}\zeta_2\cos\theta + \tau^{(3)}\zeta_1\cos\theta \tag{26}$$

Similarly

$$\dot{y} = y'\dot{\tau} = \dot{\tau}\zeta_1\sin\theta \tag{27}$$

$$\ddot{y} = \dot{\tau}\zeta_2\sin\theta + \frac{\dot{\tau}^2\zeta_1^2\cos\theta\tan\zeta_3}{b} + \ddot{\tau}\zeta_1\sin\theta \tag{28}$$

$$y^{(3)} = \begin{bmatrix} \dot{\tau}\sin\theta & \frac{\dot{\tau}^2\zeta_1^2\cos\theta}{b\cos^2\zeta_3} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + B \tag{29}$$

with

$$B = \frac{3\dot{\tau}^2\zeta_1\zeta_2\cos\theta\tan\zeta_3}{b} - \frac{\dot{\tau}^3\zeta_1^3\sin\theta\tan^2\zeta_3}{b^2} + c\frac{3\dot{\tau}\ddot{\tau}\zeta_1^2\cos\theta\tan\zeta_3}{b} + 2\ddot{\tau}\zeta_2\sin\theta + \tau^{(3)}\zeta_1\sin\theta \tag{30}$$

These expressions allow to calculate e_x, $\dot{e}_x$, $\ddot{e}_x$, e_y, $\dot{e}_y$, and $\ddot{e}_y$ using the reference trajectory and the states of the closed loop system. Plugging in these expressions into (19) and (20), and using (18) one gets

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \begin{bmatrix} \dot{\tau}\cos\theta & -\frac{\dot{\tau}^2\zeta_1^2\sin\theta}{b\cos^2\zeta_3} \\ \dot{\tau}\sin\theta & \frac{\dot{\tau}^2\zeta_1^2\cos\theta}{b\cos^2\zeta_3} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} A \\ B \end{bmatrix} \tag{31}$$

where the inverse of the coefficient matrix can be calculated symbolically. One obtains

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{\dot{\tau}} & \frac{\sin\theta}{\dot{\tau}} \\ -\frac{b\sin\theta\cos^2\zeta_3}{\dot{\tau}^2\zeta_1^2} & \frac{b\cos\theta\cos^2\zeta_3}{\dot{\tau}^2\zeta_1^2} \end{bmatrix} \begin{bmatrix} \omega_x - A \\ \omega_y - B \end{bmatrix} \tag{32}$$

The tracking feedback law is defined by (21)–(22) and by (32). First, the inputs of the feedback dynamics v_1 and v_2 can be calculated using (32); then using the feedback dynamics (21)–(22), one gets the outputs of the controller, namely the steering input u_s and the angle of the steered wheels φ.

A singularity occurs if $\zeta_1^2 = u_s^2 = 0$ which corresponds to zero longitudinal velocity. Another singular situation corresponds to $\zeta_3 = \varphi = \pm\pi/2$ which may occur if the steered wheels are perpendicular to the longitudinal axis of the car. ($b > 0$ and $\dot{\tau} > 0$ are both supposed.) Singularities imply the loss of controllability of the kinematic car model.

5.3 Implementation and verification

The control method presented in Subsection 5.2 was implemented using Matlab and Simulink.

First, the behavior of the car was only simulated using the equations of motion. The closed loop behavior was tested in several simulations with different initial conditions, with several controller parameters. An example is presented in Figs. 4–6.

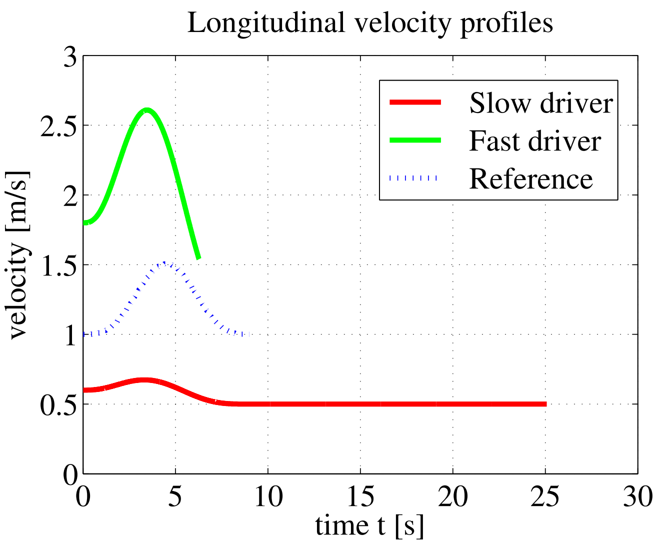
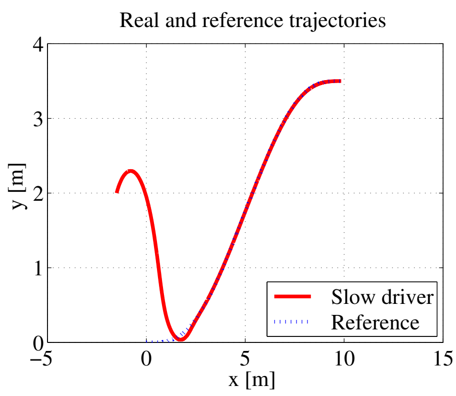


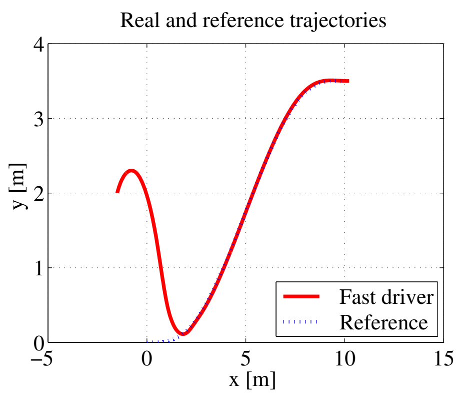
Fig. 4. Velocity profiles in the simulation

Two different cases were simulated. There is a quick driver, whose velocity profile is higher than the reference velocity, and a slow driver generates slower velocity (see Fig. 4). Both for the slow and quick driver, we generated the same reference path with some initial error (see Fig. 5). Using our time-scaling based method, the vehicle was able to eliminate the initial error and to track the reference path.

The value of the virtual time and its derivative are presented in Fig. 6. If the derivative is less than 1, the time-scaling slows down the reference motion, if it is higher than 1, the motion is accelerated. Both for the quick and slow drivers, the reference motion was slowed down at the beginning of the simulation. This was done due to the initial error. After the initial error was eliminated, thanks to the slower motion, the motion was accelerated in the case of the quick driver. Hence he was able to finish the motion in 6 seconds, while the traveling time was 25 seconds for the slow driver.

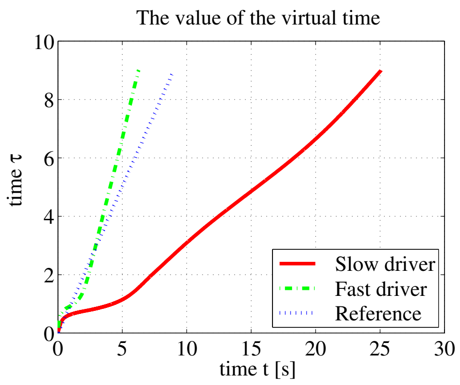


(a) Path in case of the slow driver

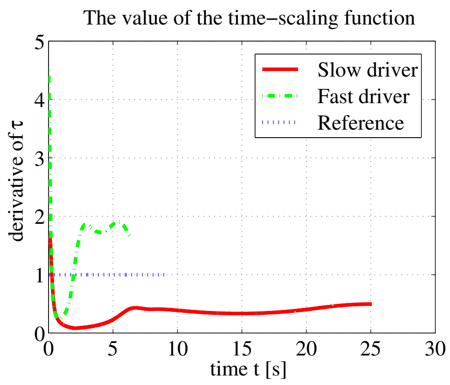


(b) Path in case of the quick driver

Fig. 5. The paths in the x–y plane



(a) The value of τ



(b) The value of $\dot{\tau}$

Fig. 6. Time-scaling in the simulation

The results of the simulations were promising, hence we tested our algorithm in the real car as well. The real test results were similar to the simulations, we achieved exponential tracking. We were ready to integrate our tracking controller into the whole PAS.

6 Implementation of the whole system

We had to use some special hardware and software environments for the development. We implemented a system which has to work in real-time, such that the inputs are received online from real measurements and the output is connected to the steering system of the vehicle. We used a rapid prototyping environment for the implementation.

We had to face some technical difficulties as well. For example: There is an upper bound on the steering input signal and on its derivative. We cannot measure too low velocities (less than 0.23 m/s). We cannot measure directly the state of the car; the position and the orientation has to be estimated from the velocities of the wheels.

The hardware components of the whole system are depicted in Fig. 7. The Ford Focus type vehicle is equipped with an electronic steering system (EPAS – Electronic Power Assist Steering), with dSPACE AutoBox and with ultrasonic sensors. These systems are connected to the CAN (Controller Area Network) bus of the vehicle. The four components in Fig. 1 (state estimator, map construction, path planning and controller) and the additional supervisory logic run on the processor board of an AutoBox.

The graphical user interface is provided by an ultra mobile personal computer (UMPC) of Asus.

The software components are developed using the Matlab/Simulink and the Real-Time Interface and ControlDesk products of dSPACE.

The functioning of the PAS was tested in several situations. In the presented example, the car moves backwards to park in a lane (see Fig. 9). The driver generates the velocity (see Fig. 8), the steering input is influenced by the controller.

Our controller is able to eliminate the huge initial error using time-scaling. The velocity of the reference motion is slowed down due to the error (see Fig. 10). Thanks to this slower reference motion, the vehicle is able to track the desired path.

7 Conclusions

Our goal was to design and implement a parking assist system for a passenger car. We had to solve an open theoretical problem, namely the design of a tracking controller for the single input kinematic car. Our solution is based on time-scaling: the time distribution of the reference is modified according to

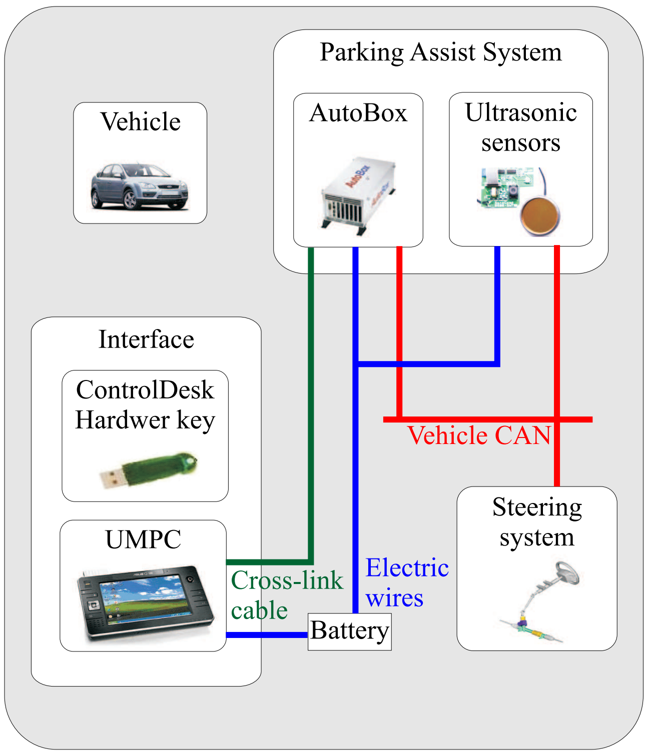


Fig. 7. Hardware components of PAS

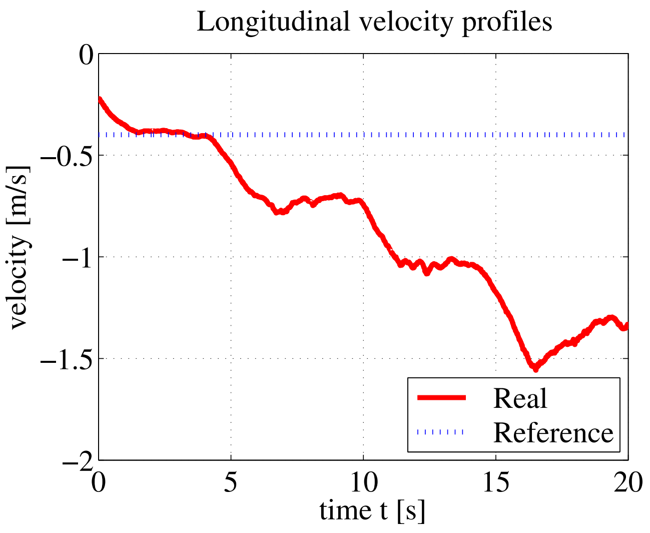


Fig. 8. Velocity profiles in the test on a real car (backward maneuver)

the closed loop behavior of the system. Moreover, using time-scaling, a new so-called scaling input was introduced.

The time-scaling based controller was implemented using Matlab/Simulink and dSPACE products, such that it can be used in real-time. Tests were performed both in simulations and on the real vehicle. Exponentially tracking was achieved.

We have still some work. Several additional tests have to be performed to verify the whole parking assist system. Our future work is to adopt the time-scaling based control method for other class of systems.

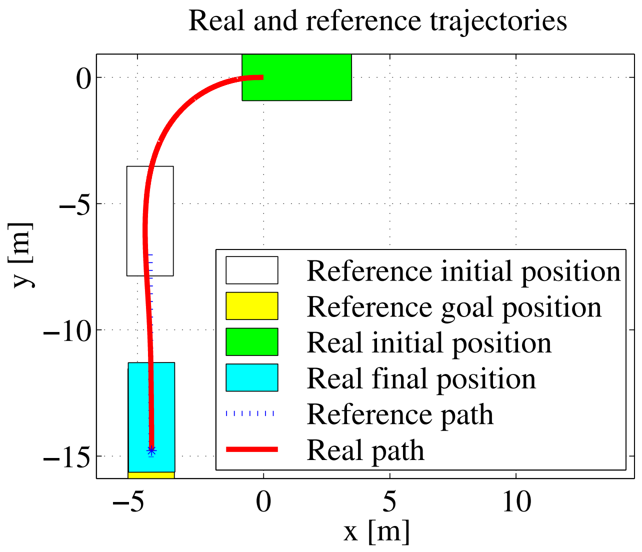


Fig. 9. The paths in the x–y plane

References

1 **AISIN SEIKI Co. Ltd. Intelligent Parking Assist**, available at `http://www.aisin.com`.

2 **Boissonnat J D, Cérézo A, Leblond J**, *Shortest paths of bounded curvature in the plane*, J. of Intelligent and Robotics Systems **11** (1994), 5–20, DOI 10.1007/BF01258291.

3 **Cuesta F., Ollero A.**, *Intelligent mobile robot navigation*, Springer tracts in advanced robotics, 2005.

4 **Dahl O., Nielsen L.**, *Torque-limited path following by on-line trajectory time scaling*, IEEE Trans. on Robotics and Automation **6** (1990), no. 5, 554–561, DOI 10.1109/70.62044.

5 **Dubins L. E.**, *On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents*, American J. of Mathematics **79** (1957), 497–517, DOI 0.2307/2372560.

6 **Evolve**, available at `http://www.ikp.liu.se/evolve`.

7 **Fliess M., Lévine J., Martin P., Rouchon P.**, *Flatness and defect of non-linear systems: Introductory theory and examples*, International J. of Control **61** (1995), no. 6, 1327–1361, DOI 10.1080/00207179508921959.

8 **Hollerbach J. M.**, *Dynamic scaling of manipulator trajectories*, Trans. of the ASME, J. of Dynamic Systems, Measurement, and Control **106** (1984), no. 1, 102–106, DOI 10.1115/1.3149652.

9 **Kiss B., Szádeczky-Kardoss E.**, *Tracking control of the orbitally flat kinematic car with a new time-scaling input*, Proc. of IEEE conf. on decision and control, 2007, pp. 1969–1974, DOI 10.1109/CDC.2007.4434463, (to appear in print).

10 **Nissan Around View Monitor**, available at `http://www.nissan-global.com`.

11 **Oetiker M. B., Baker G. P., Guzzella L.**, *A navigation-field based semi-autonomous non-holonomic vehicle-parking assistant*, IEEE Trans. on Vehicular Technology **accepted for future publication** (2008).

12 **Oroszi S., Szádeczky-Kardoss E.**, *Map construction and parking lot identification algorithm for autonomous parking system*, Proc. of the international carpathian control conf., 2007, pp. 512–515.

13 **Reeds J. A., Shepp L. A.**, *Optimal paths for a car that goes both forwards and backwards*, Pacific J. of Mathematics **145** (1990), no. 2, 367–393.

14 **Sampei Mitsuji, Furuta Katsuhisa**, *On time scaling for nonlinear systems: Application to linearization*, IEEE Trans. on Automatic Control **AC-31** (1986), 459–462, DOI 10.1109/TAC.1986.1104290.

15 **Szádeczky-Kardoss E., Kiss B.**, *Continuous-curvature turns in motion*

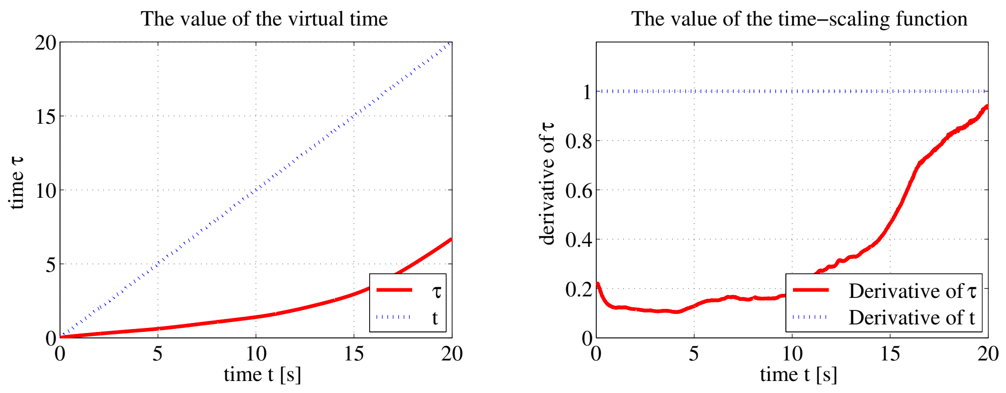


Fig. 10. Time-scaling in the test on a real car(a) The value of τ (b) The value of derivative of τ

planning for car like mobile robots, Proc. of the international carpathian control conf., 2007, pp. 684–687.

16 ______, *Path planning and tracking control for an automatic parking assist system*, Springer tracts in advanced robotics: European robotic symposium, 2008, pp. 175–184.

17 **Volkswagen Park Assist**, available at `http://www.volkswagen.com`.