

Zweistufige Trajektorienplanung für das automatische Einparken

Two-step trajectory generation for automatic parking

B. Müller, J. Deutscher, Erlangen

Kurzfassung

Dieser Beitrag beschreibt ein zweistufiges Verfahren zur Trajektoriengenerierung für das automatische Einparken eines zweiachsigen Fahrzeuges. Dabei wird zunächst eine kollisionsfreie Verbindung zwischen einem Start- und einem gewünschten Zielpunkt innerhalb der Parklücke bestimmt, ohne die kinematischen Beschränkungen in der Bewegung des Fahrzeuges zu beachten. Anschließend wird diese kollisionsfreie Verbindung in eine vom Fahrzeug abfahrbare, kollisionsfreie Trajektorie umgerechnet.

Abstract

In this contribution a two-step algorithm generating a trajectory for an autonomous parking maneuver is presented. First, a collision-free curve between a given start and a desired goal position within the parking space is planned ignoring the kinematic restrictions on the movement of the car. Second, the collision-free curve is converted into a feasible collision-free trajectory, which can be exactly followed by the car.

1. Einleitung

Die Entwicklung von Parkassistenzsystemen, die den Einparkvorgang vollständig oder teilweise übernehmen, wird von den Automobilherstellern mit Hochdruck vorangetrieben. Neben der Sensorik stellt die Planung einer geeigneten kollisionsfreien Bahn, die das Fahrzeug unter Berücksichtigung aller kinematischen Beschränkungen in die gewünschte Position innerhalb der Parklücke überführt, eine der größten Herausforderungen bei diesem Vorhaben dar. Bisher werden hierfür in der Praxis zumeist mehr oder weniger empirische Verfahren eingesetzt, die sich weitgehend am menschlichen Einparkvorgang orientieren (vgl. z.B. [1]).

In diesem Beitrag wird gezeigt, wie ein systematisches zweistufiges Verfahren aus der Robotik (vgl. [2,3]) zur Bahnplanung herangezogen werden kann. Die Methode ist modellbasiert und kann durch Anpassung weniger geometrischer bzw. dynamischer Fahrzeugparameter auf unterschiedliche Fahrzeugtypen übertragen werden. Zudem muss, im Gegensatz zu den

empirischen Lösungsansätzen, nicht zwischen Einparken in einem und in mehreren Zügen unterschieden werden. Eine Unterscheidung zwischen Einparkvorgängen parallel zur Fahrtrichtung und senkrecht dazu ist ebenfalls nicht erforderlich.

Im Folgenden wird das Verfahren in seinen Grundzügen vorgestellt; eine detaillierte Beschreibung findet sich in [4].

2. Problemstellung

Die in diesem Beitrag betrachtete Problemstellung lässt sich wie folgt spezifizieren. Gesucht ist eine Bahn (im Weiteren als Trajektorie bezeichnet), die ein Fahrzeug von einem Ausgangspunkt zu einem gewünschten Zielpunkt in der Parklücke überführt und dabei die folgenden Eigenschaften besitzt:

- Beim Abfahren der Trajektorie tritt keine Kollision mit einem Hindernis auf.
- Das Fahrzeug kann der Trajektorie mit einer beliebigen Geschwindigkeit v folgen, solange diese unterhalb einer festgesetzten Maximalgeschwindigkeit v_{\max} bleibt.
- Die Trajektorie enthält möglichst wenige Fahrtrichtungswechsel (d.h. nur wenige Fahrmanöver und damit Schaltvorgänge sind notwendig).
- Die Trajektorie ist kurz (d.h. die Dauer des resultierenden Einparkvorgangs ist klein).

Man beachte, dass die in b) aufgestellte Forderung nach Abfahrbarkeit mit variabler Geschwindigkeit die Einbindung in ein System erlaubt, bei dem der Fahrer die Geschwindigkeit v selbst vorgibt, während lediglich der zugehörige Lenkeinschlag automatisch bereitgestellt wird.

3. Fahrzeugmodell und Steuerungsentwurf

3.1. Nichtlineares Fahrzeugmodell

Die ebene Bewegung eines schlupffrei rollend angenommenen zweiachsigen Fahrzeugs mit dem Achsabstand L kann in Zustandsraumdarstellung gemäß

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{1}{L} \tan \phi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (1)$$

beschrieben werden. Wie in Bild 1 gezeigt, sind mit x und y die Position des Hinterachsmittelpunktes, mit θ die Ausrichtung der Fahrzeuglängsachse gegenüber der raumfesten

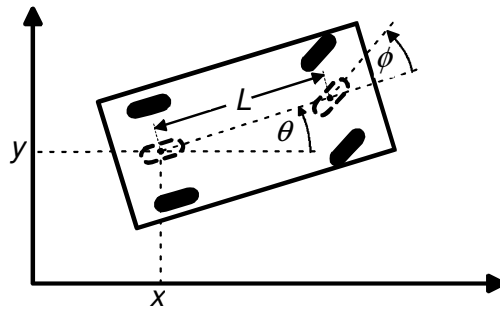


Bild 1: Nichtlineares Fahrzeugmodell

x -Achse und mit ϕ der Lenkwinkel der Vorderachse bezeichnet. Eingangsgrößen sind die Lenkwinkelgeschwindigkeit ω und die Fahrzeuggeschwindigkeit v (gemessen am Hinterachsmittelpunkt). Zu beachten sind zudem die Beschränkung des Lenkwinkels und die durch den Lenkaktor begrenzte Lenkwinkelgeschwindigkeit gemäß

$$|\phi| \leq \phi_{\max} < \frac{1}{2}\pi \quad \text{bzw.} \quad |\omega| \leq \omega_{\max} . \quad (2)$$

3.2. Flachheitsbasierter Steuerungsentwurf

Das Fahrzeugmodell (1) ist ein sog. flaches System, wobei die Fahrzeugkoordinaten x und y einen flachen Ausgang bilden. Für die Definition der Flachheit und einen Nachweis für das Vorliegen dieser Systemeigenschaft sei auf [5] verwiesen. Eine direkte Folgerung ist, dass bei Vorgabe von mindestens dreifach differenzierbaren Sollverläufen für die Fahrzeugposition $x(t)$ und $y(t)$ die Sollverläufe der restlichen Zustandsgrößen ohne Integration von Differentialgleichungen bestimmt werden können. Darüber hinaus lässt sich die zugehörige Steuerung direkt anhand einer Darstellung der Systemeingänge durch den flachen Ausgang und endlich vielen von dessen Zeitableitungen ermitteln. Angewendet auf das in Abschnitt 2 spezifizierte Problem bedeutet dies, dass zur Planung einer Trajektorie für (1) nicht der gesamte Zustandsraum betrachtet werden muss, sondern die Bestimmung von geeigneten genügend oft differenzierbaren Sollzeitverläufen für die Positionskoordinaten $x(t)$ und $y(t)$ ausreichend ist, sofern die daraus resultierenden Verläufe für den Lenkwinkel $\phi(t)$ und die Eingangsgröße $\omega(t)$ die Beschränkungen (2) einhalten.

Die Äquivalenz von (1) zu einem bogenlängenbezogenen Modell gleicher Struktur ermöglicht eine weitgehend geschwindigkeitsunabhängige Trajektorienplanung. Mit der Bogenlänge

$$s(t) = \int_0^t |v(\tau)| d\tau \quad (3)$$

bzw. $|v(t)| = ds(t)/dt$ können die Zustandsgleichungen (1) nach einer Zeit-Bogenlängen-Transformation auch als Differentialgleichungen in s geschrieben werden. Auf dieser Basis kann man eine Trajektorie alternativ als eine entsprechend glatte, in s parametrisierte Kurve $x(s)$ und $y(s)$ auffassen und eine zugehörige bogenlängenbezogene Steuerung herleiten, welche nur das Vorzeichen der Geschwindigkeit vorgibt. Diese Steuerung kann bei Messung der tatsächlichen Geschwindigkeit v mit Hilfe der Beziehung (3) online in die zugehörigen Zeitverläufe umgerechnet werden. Bei Einhaltung einer für die Trajektorienplanung angenommenen Maximalgeschwindigkeit v_{\max} sowie Beachtung der Punkte, an denen ein Fahrtrichtungswechsel erfolgen muss (d.h. an denen v das Vorzeichen wechselt), kann eine geplante zulässige Trajektorie so im Prinzip mit einer beliebigen Geschwindigkeit v durchfahren werden.

4. Zweistufige Trajektorienplanung

4.1. Motivation und Prinzip

Die in Abschnitt 2 formulierten Forderungen lassen sich grob in zwei Gruppen gliedern: die mathematisch exakt formulierbaren Eigenschaften a) und b) sowie die qualitativen Anforderungen c) und d). Eine zweistufige Vorgehensweise zur Trajektorienplanung kann direkt anhand der fundamentalen Forderungen a) und b) motiviert werden. Die Grundidee besteht darin, diese nicht gleichzeitig zu erfüllen, sondern sie nacheinander zu befriedigen, d.h. das komplexe Gesamtproblem wird in zwei deutlich einfacher zu lösende Teilprobleme aufgespalten. In einem ersten Schritt wird zur Erfüllung von a) eine kollisionsfreie Verbindung, ein sog. Pfad, zwischen dem Ausgangs- und dem Zielpunkt in der Parklücke berechnet. Dabei wird nur auf Kollisionsfreiheit geachtet, die Abfahrbarkeit durch das Fahrzeug jedoch nicht berücksichtigt, d.h. beispielsweise wäre eine von einem PKW nicht umzusetzende seitliche Bewegung zulässig. Erst im zweiten Schritt wird der kollisionsfreie Pfad in eine abfahrbare Trajektorie gemäß Forderung b) umgerechnet. Die zweistufige Konstruktion wird aus Bild 2 ersichtlich. In Bild 2 ist zudem angedeutet, auf welche Weise die Verknüpfung zwischen den beiden Teilschritten realisiert ist. Auf dem kollisionsfreien Pfad (durchgezogene Linie) werden geeignete Stützstellen (als Kreise angedeutet) ausgewählt, die paarweise mit einem abfahrbaren Trajektorienteilstück (gestrichelte und gepunktete Linien) verbunden werden. Tritt auf einem Trajektorienteilstück eine Kollision mit einem Hindernis auf, so werden neue, näher aneinander liegende Stützstellen auf dem kollisionsfreien Pfad ausgewählt. Dies geschieht solange, bis die resultierende Gesamttrajektorie kollisionsfrei ist. Es lässt sich zeigen,

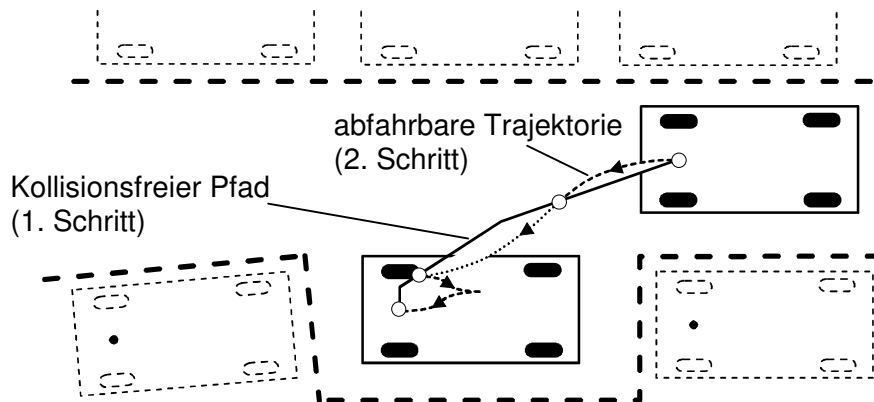


Bild 2: Prinzip der zweistufigen Trajektorienplanung

dass so jeder kollisionsfreie Pfad in eine abfahrbare Trajektorie überführt werden kann. Somit ist mit Hilfe dieses Vorgehens immer eine kollisionsfreie Gesamtrajektorie und damit eine Lösung des Bahnplanungsproblems ermittelbar.

Die qualitativen Anforderungen c) und d) können durch Verwendung geeigneter Algorithmen in den beiden Teilschritten des zweistufigen Verfahrens sichergestellt werden. Zunächst werde die Forderung c) nach einer minimalen Anzahl von Fahrtrichtungswechseln betrachtet. Hierfür muss die Anzahl der Stützstellen, die auf dem kollisionsfreien Pfad zur Umrechnung in eine Trajektorie nötig sind, möglichst klein gehalten werden. Schließlich führt dies auf eine Gesamtbahn, die sich aus wenigen Teiltrajektorien zusammensetzt und somit wenige Fahrmanöver beinhaltet. Viele Stützstellen werden jedoch erzeugt, wenn der kollisionsfreie Pfad nahe an den Hindernissen verläuft, da dann bei der Verbindung zweier Punkte mit einer Teiltrajektorie schon bei einer kleinen Abweichung vom Pfad eine Kollision auftreten und die Wahl von weiteren Stützstellen erforderlich sein kann. Um dies zu vermeiden, wird der kollisionsfreie Pfad abstandsoptimiert geplant, d.h. mit möglichst großem Abstand zu allen Hindernissen (vgl. [6]). Hierauf wird im nachfolgenden Abschnitt näher eingegangen.

Die letzte Anforderung d) nach einer möglichst kurzen Einparktrajektorie wird sichergestellt, indem für den zweiten Schritt ein geeigneter Trajektorienplaner zum Einsatz kommt, der per Konstruktion kurze Teiltrajektorienstücke liefert (vgl. [7]). In Abschnitt 4.3. ist das Prinzip dieses Trajektorienplaners kurz beschrieben.

4.2. Schritt 1: Abstandsoptimierte kollisionsfreie Pfadplanung

Bevor eine abstandsoptimierte Planung erfolgen kann, muss zunächst ein geeignetes Abstandsmaß definiert werden. Auf der linken Seite in Bild 3 ist angedeutet, warum die herkömmliche euklidische Metrik für das vorliegende Problem nur bedingt geeignet ist. Obwohl

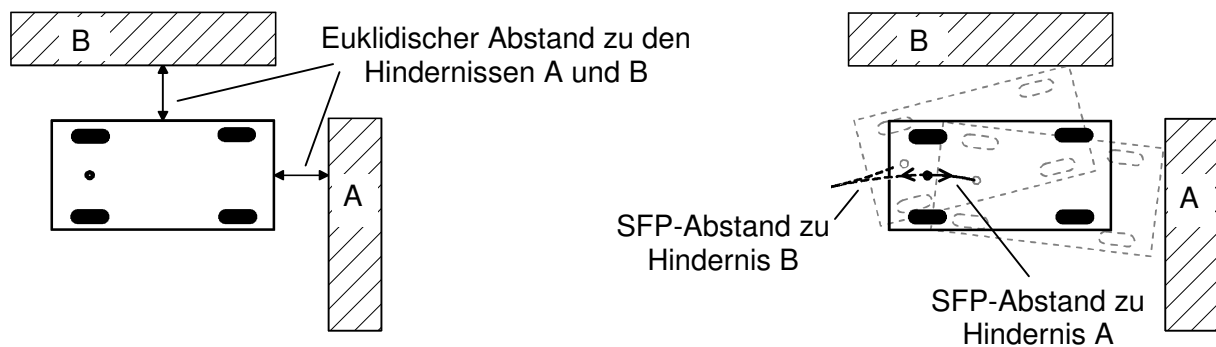


Bild 3: Motivation und Definition des SFP-Abstandsmaßes

die Hindernisse A und B im euklidischen Sinn etwa gleich weit von dem dargestellten Fahrzeug entfernt sind, ist intuitiv offensichtlich, dass die Gefahr einer Kollision mit dem Hindernis A größer ist als mit dem Hindernis B. Der Grund hierfür ist die kinematische Beschränkung des Fahrzeuges, die eine seitliche Bewegung nicht zulässt. Dieser Aspekt wird von der sog. SFP-Metrik (engl.: shortest feasible path) berücksichtigt. Der Hindernisabstand im Sinne der SFP-Metrik (im folgenden als SFP-Hindernisabstand bezeichnet) ist definiert als die Bogenlänge der kürzesten abfahrbaren Trajektorie bis zur Kollision. Auf der rechten Seite in Bild 3 sind die jeweils kürzesten Verbindungstrajektorien zum Hindernis A und zum Hindernis B gezeigt. Es ist deutlich zu erkennen, dass sich ein wesentlich größerer SFP-Abstand zu B als zu A ergibt und die SFP-Metrik somit im Sinne der vorherigen Betrachtungen ein geeignetes Abstandsmaß darstellt. Für eine exakte mathematische Definition der SFP-Metrik sei auf [6] verwiesen.

Zur eigentlichen Berechnung des SFP-Abstandes kann auf die Arbeiten von Reeds/Shepp [8] und Sussmann/Tang [9] zurückgegriffen werden, die durch Lösen eines Optimierungsproblems die Verbindungstrajektorien mit der kürzesten Bogenlänge zwischen zwei beliebigen Punkten für das Fahrzeugmodell (1) explizit hergeleitet haben. Um den SFP-Abstand zu einem Hindernis zu berechnen, geht man nun wie folgt vor. Die Oberfläche des Hindernisses wird diskretisiert und der SFP-Abstand zu allen diskreten Hindernispunkten bestimmt, wobei der kleinste dabei auftretende Wert dem SFP-Hindernisabstand entspricht. Da dieses Verfahren sehr rechen- und damit zeitintensiv ist, muss dieser Schritt offline ausgeführt werden. Hierzu werden geeignete Hinderniselemente definiert, der SFP-Abstand zu diesen offline bestimmt und das Ergebnis in Form von Look-up-Tabellen abgelegt. Wie auf der linken Seite in Bild 4 dargestellt kann dann durch Verschieben, Rotieren und Aneinanderfügen der einzelnen Hinderniselemente online eine vorliegende Parkumgebung approximiert und ein zugehöriger SFP-Hindernisabstand durch einfaches Nachschlagen schnell ermittelt werden.

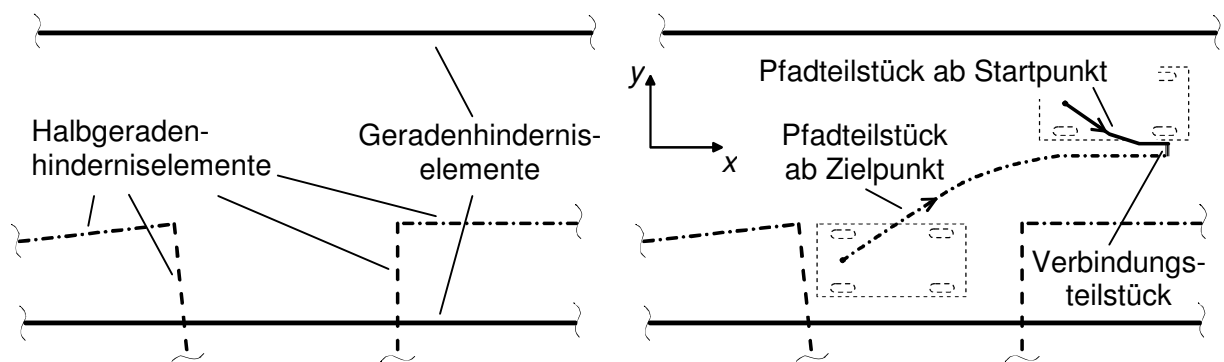


Bild 4: Approximation einer Parkumgebung mit elementaren Hinderniselementen (links); Prinzip der Planung des abstandsoptimierten Pfades (rechts)

Auf der rechten Seite von Bild 4 ist am Beispiel des parallelen Einparkens gezeigt, auf welche Weise die eigentliche Planung des kollisionsfreien Pfades erfolgt. Ausgehend vom Startpunkt außerhalb der Parklücke wird zunächst ein Teilpfad in Richtung steigender x -Werte und größer werdender oder gleich bleibender SFP-Hindernisabstände geplant. Auf ähnliche Weise wird ein zweites Teilstück ausgehend vom Zielpunkt in der Parklücke bestimmt. Schließlich werden beide Teilstücke mit einem geeigneten Verbindungsstück zu einem Gesamtpfad vom Start- zum Zielpunkt verbunden. Man beachte, dass ein Punkt des Pfades nicht nur durch die Position x und y , sondern auch durch die Ausrichtung θ des Fahrzeuges festgelegt ist. Dies ist in der zweidimensionalen Darstellung in Bild 4 nur schlecht zu erkennen.

4.3. Schritt 2: Trajektoriengenerierung

Wie in Abschnitt 4.1. beschrieben, wird der kollisionsfreie Pfad abschnittsweise in eine abfahrbare Trajektorie umgerechnet, indem jeweils zwei Stützstellen auf dem Pfad miteinander verbunden werden. Hierfür wird ein Trajektorienplaner benötigt, d.h. ein Algorithmus, der zu zwei beliebigen Punkten eine abfahrbare Verbindungstrajektorie liefert. Um die qualitative Anforderung d) nach einem möglichst kurzen Gesamtweg zu erfüllen, sollten die vom Trajektorienplaner erzeugten Teilstücke zudem eine möglichst kleine Bogenlänge besitzen. Wegen dieser letzten Anforderung liegt es zunächst nahe, einen Planer zu verwenden, der die im letzten Abschnitt bereits erwähnten bogenlängenoptimalen Trajektorien nach Reeds/Shepp erzeugt. Diese optimalen Trajektorien haben jedoch den Nachteil, dass ein Fahrzeug an mehreren Punkten anhalten und im Stand lenken muss, um ihnen zu folgen. Dies kann bei sehr engen Parklücken vorteilhaft sein, ist aber bei größeren Parklücken in der Regel nicht

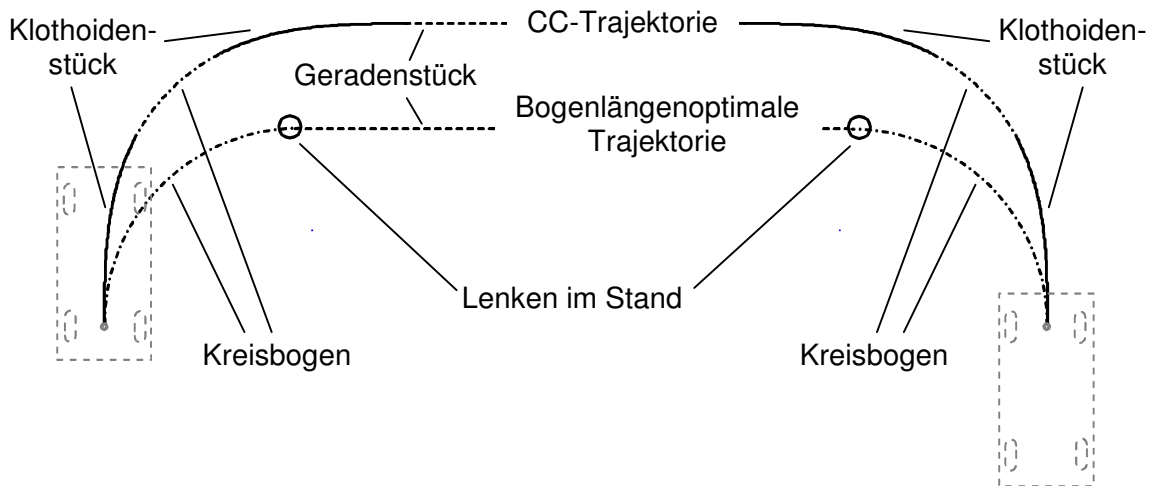


Bild 5: Beispiel für bogenlängenoptimale Trajektorie und CC-Trajektorie

akzeptabel. Daher wird für größere Parklücken eine Verallgemeinerung dieser optimalen Trajektorien verwendet, die sog. CC-Trajektorien (vgl. [7]). Diese haben im Prinzip die gleiche Form wie die optimalen Trajektorien, glätten diese jedoch in geeigneter Weise, so dass ein Fahrzeug sie abfahren kann, ohne Lenkbewegungen im Stand auszuführen. In Bild 5 ist dies an einem Beispiel in der x - y -Ebene gezeigt. Die optimale Trajektorie besteht in diesem Fall aus einem Kreisbogen, einem Geradenstück und einem abschließenden Kreisbogen, wobei an den jeweiligen Schnittpunkten der Teilstücke der Lenkwinkel im Stand verändert werden muss. Bei der entsprechenden CC-Trajektorie sind diese Schnittpunkte durch Einfügen von Klothoidenstücken geglättet, so dass die Trajektorie ohne anzuhalten durchfahren werden kann, solange die Geschwindigkeit die Schranke $|v(t)| \leq v_{\max}$ für alle Zeiten t einhält. Aufgrund ihrer Ähnlichkeit zu den längenoptimalen Trajektorien sind auch die CC-Trajektorien in der Regel relativ kurz und erfüllen somit zumindest qualitativ die oben gestellte Anforderung.

Es ist einsichtig, dass bei Vorgabe einer großen Maximalgeschwindigkeit v_{\max} lange Glättungsteilstücke eingefügt werden müssen und die Länge der CC-Trajektorien somit mit v_{\max} zunimmt. Bei der Verbindung von nahe aneinanderliegenden Punkten bzw. Stützstellen - ein Problem, das meist bei Einparksituationen mit kleinen Parklücken auftritt - kann es daher bei Vorgabe einer zu großen Maximalgeschwindigkeit v_{\max} zu unerwünscht langen Trajektorienstücken mit vielen Fahrtrichtungswechseln kommen. Um diesen Effekt zu vermeiden, wird die Maximalgeschwindigkeit für jedes Trajektorienteilstück adaptiv bestimmt, abhängig davon wie weit entfernt bzw. wie nahe die zu verbindenden Stützstellen liegen. Auf diese Weise werden automatisch Trajektorien erzeugt, die eine hohe Geschwindigkeit erlauben, sofern genügend Platz zum Rangieren vorhanden ist, an Engstellen jedoch nur eine kleine

Geschwindigkeit zulassen. Bei sehr eng aneinanderliegenden Stützstellen kann alternativ auf einen Trajektorienplaner umgeschaltet werden, der die längenoptimalen Trajektorien liefert und somit Lenkbewegungen im Stand erlaubt.

5. Simulationsergebnisse

Das oben beschriebene Verfahren wurde in MATLAB implementiert und in der Simulation getestet. Dabei wurden Fahrzeugparameter verwendet, die ungefähr der Konfiguration eines VW Golf V entsprechen. Für den Lenkaktor wurde angenommen, dass eine Änderung des Lenkwinkels von Volleinschlag links nach Volleinschlag rechts innerhalb von 3s erfolgen kann. In Bild 6 sind exemplarisch die sich ergebenden Trajektorien für zwei parallele Einparksituationen mit engen Parklücken dargestellt. Diese zeigen, dass das vorgestellte Verfahren auch für kleine Parklücken geeignet ist. Anhand der in den Einzelbildern angegebenen maximalen Geschwindigkeit v_{max} , die für das jeweils aktuelle Trajektorienteilstück gültig ist, lässt sich erkennen, wie die automatische Adaption die zulässige Geschwindigkeit v_{max} reduziert, wenn wenig Raum zum Rangieren vorhanden ist. Da der Algorithmus eine Trajektorie von einer beliebigen Startposition aus berechnen kann, liefert er auch für die zunächst ungünstige Ausgangsposition des auf der rechten Seite von Bild 6 gezeigten Beispiels eine ansprechende Lösung. Die Berechnungsdauer betrug bei einem Speicherbedarf von ca.

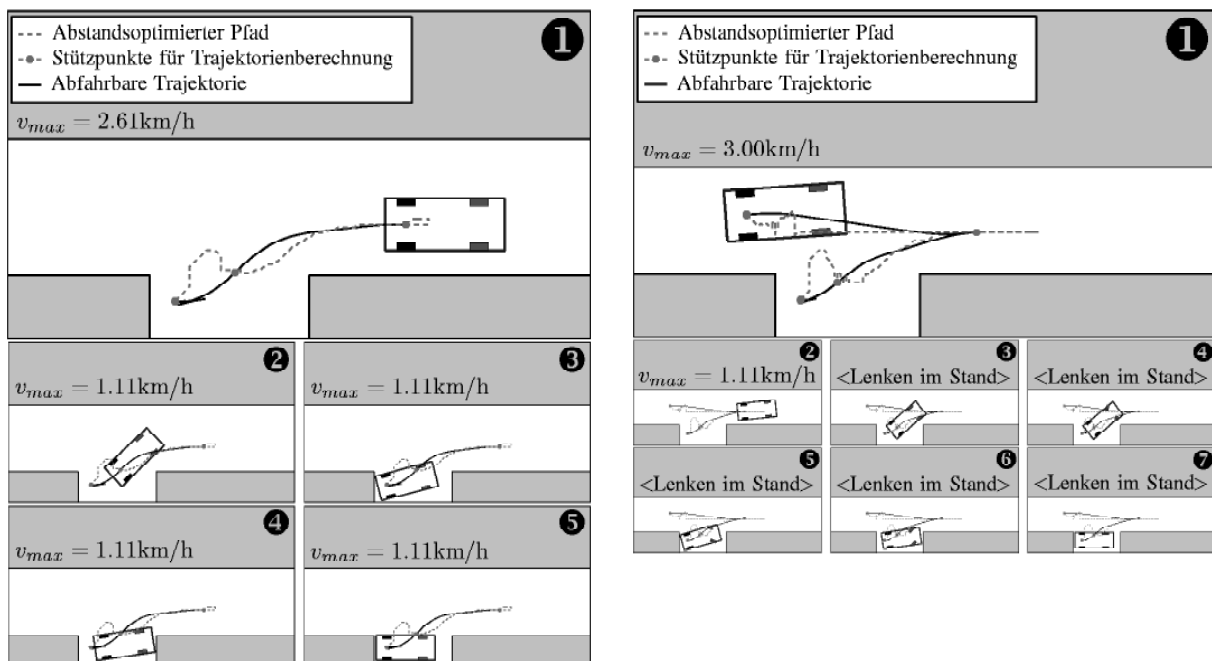


Bild 6: Resultierende Trajektorien für zwei parallele Einparksituationen; die Parklücke ist jeweils 0,4m breiter und 1,4m (links) bzw. 0,9m (rechts) länger als das Fahrzeug

3Mbyte für beide Beispiele weniger als 2,5s auf einem AMD Athlon XP 1800+. Hierbei ist jedoch zu berücksichtigen, dass die vorliegende MATLAB-Implementierung noch ein großes Optimierungspotential im Hinblick auf die Rechenzeit besitzt. Mit einer entsprechend optimierten und beispielsweise in C realisierten Software sollte daher auch auf weniger leistungsfähigen Prozessoren eine Berechnungsdauer erreichbar sein, die für eine reale Anwendung im Fahrzeug akzeptabel ist.

6. Zusammenfassung

Es wurde ein zweistufiger Algorithmus zur Berechnung von Trajektorien für das automatische Einparken vorgestellt und dessen Leistungsfähigkeit sowie dessen Einsetzbarkeit bei kleinen Parklücken in der Simulation nachgewiesen. Neben der Eignung für enge Parksituationen sind weitere Vorteile des Verfahrens, dass eine Trajektorie von jeder beliebigen Ausgangsposition aus geplant und das Einparken senkrecht bzw. schräg zur Fahrtrichtung in gleicher Weise wie das parallele Einparken behandelt werden kann.

7. Literatur

- [1] Stäbler T, Grimm W., Schmid D., Nixdorf B., Uhler W.: Manövrierunterstützung des Fahrers beim Einparken, AUTOREG 2002, VDI-Berichte Nr. 1672, S. 597-606, 2002.
- [2] Laumond J.: Feasible trajectories for mobile robots with kinematic and environment constraints, Proc. Conf. on Intelligent Autonomous Systems, S. 346-354, 1986.
- [3] Latombe J.-C.: Robot Motion Planning, Kluwer Academic Publishers, 1991.
- [4] Müller B., Deutscher J., Grodde S.: Continuous curvature trajectory design and feed-forward control for parking a car, eingereicht bei IEEE Trans. Control Systems Technology.
- [5] Rouchon P., Fliess M., Lévine J., Martin Ph.: Flatness and motion planning: the car with n-trailers, Proc. European Control Conference, S. 1518-1522, 1993.
- [6] Mirtich B., Canny J.: Using skeletons for nonholonomic path planning among obstacles, Proc. IEEE Int. Conf. on Robotics and Automation, S. 2533-2540, 1992.
- [7] Fraichard T., Scheuer A.: From Reeds and Shepp's to continuous-curvature paths, IEEE Trans. on Robotics and Automation, Vol. 20, Nr. 6, 2004.
- [8] Reeds J. A., Shepp L. A.: Optimal paths for a car that goes both forward and backward, Pacific Journal of Mathematics, 1990.
- [9] Sussmann H. J., Tang G.: Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control, Rutgers University, SYCON-Rutgers Center for Systems and Control, 1991.