

Vom reaktiven zum taktischen Trajektorienplaner

Janek Hudecek* und Lutz Eckstein†

Zusammenfassung: Der Beitrag stellt ein Verfahren zur Trajektorienplanung vor, das basierend auf einem beliebigen Fahrzeugmodell ohne Approximierung der Zustandsübergänge auskommt und so eine präzise Planung ermöglicht. Hierzu wird ein Optimalsteuerungsproblem formuliert, welches die kontinuierliche Überwachung von Nebenbedingungen gestattet. Um die Lösung des Problems in Echtzeit, es wird eine Zykluszeit von 20 ms angenommen, zu ermöglichen, wird eine Methodik vorgestellt, die die Dimension des Problems im Vergleich zu bisherigen Lösungen reduziert und die benötigte Zeit somit minimiert. Nach der Einführung dieser Methodik werden Details zu den konkreten Nebenbedingungen und der Kostenfunktion gegeben. Der Beitrag schließt mit der Vorstellung bisheriger Ergebnisse sowie einem Ausblick auf zukünftige Arbeiten.

Schlüsselwörter: Funktionsarchitektur, Heterogeneous Computing, Optimalsteuerung, Trajektorienplanung

1 Einleitung

Die Planung von Fahrtrajektorien ist ein zentraler Bestandteil des hochautomatisierten Fahrens. Die Aufgabe besteht darin, eine Steuerfunktion zu finden, mit der das Fahrzeug von seinem aktuellen Zustand auf sicherem und komfortablem Weg in einen Zielzustand überführt werden kann. Hierbei müssen sowohl Gegebenheiten aus dem Umfeld, definiert durch den Straßenverlauf, Hindernisse und Verkehrsteilnehmer, als auch kinematische und dynamische Einschränkungen des Fahrzeugs beachtet werden. In der Forschung und Entwicklung finden sich zahlreiche Ansätze zur Lösung dieses Problems, wobei jeweils spezifische Vor- und Nachteile bestehen. In den meisten Fällen wird zur Lösung ein Optimierungsproblem aufgestellt, welches entweder iterativ oder durch Diskretisierung und Suche nach dem Optimum gelöst wird.

Unabhängig von der Algorithmik basieren die meisten Planer auf dem gleichen Konzept: Statt die Steuerfunktion direkt zu ermitteln, werden Zwischenzustände gesucht, die das Fahrzeug der Reihe nach anfährt. Unterschiede finden sich vor allem in der Generierung dieser Zustände sowie in der Berechnung der Steuerfunktion, die benötigt wird, um den jeweiligen Zustandsübergang zu realisieren. Bei diskreten Verfahren werden die Zustände über Raster erstellt [8, 11], bei numerischen Verfahren werden sie durch die Optimierungsparameter beschrieben [3, 10]. Durch die fehlende Möglichkeit der analytischen Bestimmung der benötigten Steuerfunktion bei Verwendung nichtlinearer Fahrzeugmodelle muss diese approximiert werden, um die Nebenbedingungen zu überprüfen. Beispiele

*Janek Hudecek verantwortet am Forschungsinstitut Kraftfahrwesen mbH Aachen (fka) die Themen Optimalsteuerungen und heterogene Systeme (e-mail: hudecek@fka.de).

†Prof. Lutz Eckstein leitet das Institut für Kraftfahrzeuge der RWTH Aachen University (ika).

möglicher Approximationen sind dabei Linearisierungen [10, 3], die Trennung von lateralen und longitudinalen Bewegungen [9] oder die Vorabberechnung und Interpolation über Lookup-Tabellen zur Laufzeit [2].

Durch die Approximation der Fahrzeugmodelle sind die bisherigen Verfahren weitestgehend auf kinematisch geprägte Fahrmanöver beschränkt – für die Planung von Fahrten im fahrdynamischen Grenzbereich sind sie nur bedingt einsetzbar. Bei Verwendung von Lookup-Tabellen steigt der Bedarf an Speicherplatz mit der Anzahl an Zustandsparametern exponentiell an. Wird der Zustandsraum diskretisiert, kann die gefundene von der optimalen Lösung stark abweichen. Weiterhin werden bislang Randbedingungen nicht kontinuierlich, sondern lediglich an festgelegten Stellen überprüft.

Um sämtliche Dynamikbereiche abdecken zu können und den Lösungsraum nicht vorab einzuschränken, weicht der hier vorgestellte von existierenden Ansätzen ab. Statt den Zwischenzuständen wird direkt die Steuerfunktion gesucht, die Zwischenzustände über die hinterlegte Beschreibung der Systemdynamik berechnet. Der Ansatz formuliert hierfür ein Optimalsteuerungsproblem (OCP), wodurch sich ferner Zustandsbeschränkungen kontinuierlich überwachen lassen.

Um die Lösung des OCPs zu beschleunigen, wird in Abschnitt 2 eine allgemeine Methodik vorgestellt, die die Dimension des resultierenden Gleichungssystems reduziert. Danach werden in Abschnitt 3 Details der konkreten Umsetzung erläutert. In Abschnitt 4 werden Simulationsergebnisse des implementierten Planers präsentiert. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick auf ausstehende Arbeiten.

2 Lösung des Optimalsteuerungsproblems

Formal ist ein OCP durch die folgende Definition gegeben [1]: Gesucht ist eine Steuerfunktion $\mathbf{u}(t)$, welche das Kostenfunktional

$$J = \varphi(\mathbf{x}(t_0), \mathbf{x}(t_f)) + \int_{t_0}^{t_f} j(\mathbf{x}(t), \mathbf{u}(t), t) dt$$

unter den Differentialgleichungsnebenbedingungen

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad t_0 \leq t \leq t_f$$

sowie den Steuer- und Zustandsbeschränkungen

$$C(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0, \quad t_0 \leq t \leq t_f$$

und den Randbedingungen

$$\psi(\mathbf{x}(t_0), \mathbf{x}(t_f)) = 0$$

minimiert. Zur numerischen Lösung dieser Problemklasse existieren indirekte und direkte Verfahren, wobei im Folgenden nur auf die Letzteren eingegangen wird.

Direkte Verfahren reduzieren das OCP auf ein endlich-dimensionales Problem, indem sie zum einen den Suchraum für die Steuerfunktion \mathbf{u} durch die Vorgabe einer zu parameterisierenden Funktion beschränken, zum anderen das Kostenfunktional J durch eine

Funktion approximieren. Als weitere Vereinfachung wird die Überprüfung der Nebenbedingungen nur an einer begrenzten Anzahl an Stellen durchgeführt. Zur Lösung resultierenden Problems existieren zwei Ansätze: Schießverfahren¹ und direkte Kollokation². Beide Verfahren erstellen ein nichtlineares Programm unter Nebenbedingungen (NLP) und führen anschließend eine Parameteroptimierung der Steuerfunktion durch. Während bei den Schießverfahren nach einer initialen Schätzung der Parameter von \mathbf{u} die Zwischenzustände $\mathbf{x}(t_i)$ basierend auf dem Startzustand ermittelt werden müssen, stellen die Zwischenzustände $\mathbf{x}(t_i)$ bei der direkten Kollokation zusätzliche Parameter dar; der weitere Verlauf von \mathbf{x} wird hierbei durch eine Interpolation approximiert. Die Einhaltung der Differentialgleichungsnebenbedingungen muss somit als zusätzliche Nebenbedingung formuliert werden.

Die Dimension der resultierenden NLPs ist abhängig von der Schrittweite der Diskretisierung sowie der Anzahl an Bedingungen und Parameter. Im allgemeinen entsteht durch die Überführung ein dünn besetztes Gleichungssystem hoher Dimension, dessen Lösung in kurzer Zeit eine hohe Rechenleistung erfordert.

2.1 Reduzierung der Dimension des NLPs

Um die Berechnungszeit zu minimieren, wird statt der Aufnahme aller einzelnen Stützpunkte lediglich ein Maß A für die Verletzung der jeweiligen Beschränkung auf dem vollständigen Intervall als Nebenbedingung gefordert. Die kontinuierlichen Beschränkungen lassen sich somit als einzelne Randbedingung formulieren, die Dimension des NLPs wird um einen Faktor entsprechend der Anzahl an Stützpunkten im Ursprungssystem reduziert. Zur Verbesserung des Konvergenzverhaltens kann zusätzlich explizit die Einhaltung an dedizierten Stellen gefordert werden. Die Anzahl dieser Stellen kann durch die Filterfunktion gering gehalten werden.

Um eine stabile und schnelle Lösung des NLPs zu gewährleisten, müssen alle Funktionen die Bedingung $F \in C^2$ erfüllen. Für die benötigte Metrik wird daher eine kontinuierliche Filterfunktion A mit begrenzten Gradienten eingesetzt. Für jeden Zustand wird festgelegt, ab wann dieser als kritisch (x_{critical}) und ab wann als unzulässig (x_{invalid}) gilt; innerhalb dieser Grenzen steigt der Funktionswert von A von 0 auf 1 an. Über den kritischen Bereich hinaus verläuft die Funktion linear steigend, der Gradient ist dabei abhängig von der Intervallbreite $|x_{\text{invalid}} - x_{\text{critical}}|$.

In das NLP wird schließlich die Ungleichungsbedingung $C = \int_{t_0}^{t_f} A(x(t)) \leq T$ aufgenommen, wobei T ein Toleranzparameter ist, über den sichergestellt wird, dass sobald ein unzulässiger Funktionswert erreicht wurde, die Bedingung nicht mehr erfüllt ist, kritische Bereiche jedoch in Maßen toleriert werden. Abbildung 1 zeigt den Verlauf einer beispielhaften Filterfunktion und ihres Gradienten.

2.2 Steuerfunktion

Ziel des reduzierten Problems ist die Optimierung der Parameter einer gegebenen Steuerfunktion \mathbf{u} . Der Lösungsraum für \mathbf{u} hängt dabei von der Funktion $P : \mathbb{R}^n \rightarrow \mathcal{U}$, also

¹Shooting-Methods

²Direct Collocation/Direct Transcription

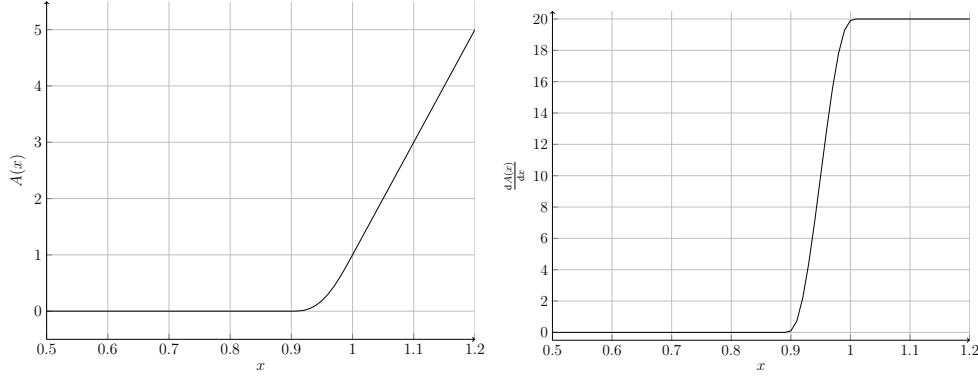


Abbildung 1: Verlauf einer Filterfunktion mit zugehörigem Gradienten für $x_{\text{critical}} = 0.9$ und $x_{\text{invalid}} = 1$.

der Abbildung der Parameter auf die konkrete Steuerfunktion ab. Im vorgestellten Ansatz dient ein Spline als Basis für \mathbf{u} , der über Polynome fünften Grades zusammengesetzt wird.

Die Parameter werden als Funktionswerte des Splines an äquidistanten Stützstellen interpretiert, die Form des Splines kann durch die, aus dem hohen Grad der verwendeten Polynome resultierende, hohe Anzahl an Freiheitsgraden angepasst werden. Das Optimierungskriterium der Interpolation kann dabei situationsabhängig gewählt werden, um so z. B. einen besonders glatten Verlauf oder die Minimierung von Überschwingern zu erreichen (vgl. [4]).

3 Formulierung des Optimalsteuerungsproblems

Wie in Abschnitt 2 beschrieben, definiert sich das OCP durch eine Kostenfunktion, den Systemzustand, die Zustandsübergänge sowie die Randbedingungen. Im Folgenden wird deren konkrete Formulierung beschrieben.

3.1 Systemzustand und -übergänge

Die Implementierung als Optimalsteuerungsproblem erlaubt die Verwendung beliebiger Differentialgleichungsbedingungen zur Modellierung der Systemübergänge. Auf diese Weise können Steuertrajektorien sowohl für kinematisch geprägte Szenarien, wie z. B. Parkvorgänge oder Fahren in der Stadt, als auch für hochdynamische Szenarien, z. B. zur Planung von Ausweichtrajektorien, berechnet werden. Die Bestandteile des Systemzustands $\mathbf{x}(t)$ sind abhängig vom verwendeten Fahrdynamikmodell.

Zunächst kommt ein einfaches Einspurmodell zum Einsatz. Der Systemzustand ist hier durch $\mathbf{x}(t) = [x, y, \psi, \kappa, \omega, \alpha]$, also der Position und Orientierung des Fahrzeugs sowie der Bahnkrümmung inklusive zeitlicher Ableitungen, definiert; die Geschwindigkeit wird als konstant angenommen. Die Systemübergänge sind durch die Gleichungen (1) gegeben.

$$\begin{aligned} \dot{x} &= v \cdot \cos \psi \\ \dot{y} &= v \cdot \sin \psi \\ \dot{\psi} &= \kappa, \quad \dot{\kappa} = \omega, \quad \dot{\omega} = \alpha \end{aligned} \tag{1}$$

3.2 Kostenfunktion

Um eine vom Fahrer akzeptierte Steuer- und damit Fahrtrajektorie zu berechnen, wird in der Kostenfunktion wie allgemein üblich die Minimierung von Querbeschleunigung und Ruck gefordert. Weiterhin soll das Fahrzeug nach Möglichkeit in der Mitte der Spur, bzw. entlang einer Referenzkurve Γ fahren. Gleichung 2 formalisiert die genannten Bedingungen.

$$\begin{aligned}
 J = \int_{t_0}^{t_f} & w_\kappa \cdot \mathbf{x}^\kappa(t)^2 + \\
 & w_\omega \cdot \mathbf{x}^\omega(t)^2 + \\
 & w_d \cdot p(\|\Gamma^{xy}(\tilde{s}(t)) - \mathbf{x}^{xy}(t)\|) + \\
 & w_\psi \cdot (\Gamma^\psi(\tilde{s}(t)) - \mathbf{x}^\psi(t))^2 dt
 \end{aligned} \tag{2}$$

$\mathbf{x}^x(t)$ bezeichnet eine Komponente des Fahrzeug-Zustandsvektors zum Zeitpunkt t , $\Gamma^x(\tilde{s}(t))$ eine Komponente des der Fahrzeugposition nächstgelegenen Punktes auf der über die Länge parametrisierten Referenzkurve. Da dieser Punkt im Allgemeinen nicht eindeutig bestimmt ist, erfolgt eine Ermittlung nach der in Abschnitt 3.4.3 erläuterten Methode. Die Funktion p bestraft ein Abweichen von der Referenz abhängig von der Entfernung, die Abweichung in der Orientierung geht quadratisch in den Strafterm ein. Durch die Faktoren w_x können die einzelnen Terme je nach Situation gewichtet werden. Eine zeitliche Abhängigkeit ist ebenfalls möglich, derzeit jedoch nicht implementiert.

3.3 Steuerfunktion

Als Steuerfunktion dient der Lenkwinkel $\delta(t)$. Die Einhaltung der kinematischen und dynamischen Grenzen kann über die in Abschnitt 2.1 vorgestellten Filterfunktionen sichergestellt werden. Da in (1) die Bahnkrümmung benötigt wird, muss der Lenkwinkel zunächst über $\kappa = l/\tan \delta$ umgerechnet werden, wobei l den Radstand des Fahrzeugs bezeichnet. Der zusätzliche Aufwand gegenüber der direkten Verwendung von κ lohnt sich jedoch aufgrund verbesserter numerischer Stabilität.

Damit die Optimierung möglichst schnell konvergiert, wird ein guter Startwert benötigt. Im Kaltstart, also beim ersten Durchlauf der Optimierung, wird der Lenkwinkel zu 0 angenommen, die zurückzulegende Strecke wird anhand der euklidischen Distanz zwischen Start- und Endpunkt bestimmt. In den darauf folgenden Schritten kann die Optimierung warm, also bereits mit guten Schätzwerten, gestartet werden. Hierzu werden die Parameter aus der jeweils vorherigen Iteration um die Zykluszeit in die Zukunft prädictiert. Bei plötzlichem Auftauchen von Hindernissen liegt der Startwert weiter vom Optimum entfernt, so dass in diesen Fällen mehr Iterationen benötigt werden.

3.4 Zustandsbewertung

In der Zustandsbewertung wird die Einhaltung der Nebenbedingungen C_k des OCPs überprüft. Folgende Aspekte müssen berücksichtigt werden:

- Einhaltung kinematischer und dynamischer Grenzwerte: Über die in 2.1 beschriebene Filterfunktion kann die Einhaltung direkt überprüft werden.

- Kollisionsfreiheit: Zur Überprüfung der Kollisionsfreiheit müssen alle Zwischenpositionen mit zugehörigen Zeitpunkten bekannt sein. Für jeden Zeitpunkt wird überprüft, ob sich das Fahrzeug zu einem auf der Straße befindet, zum anderen mit keinem Objekt kollidiert.
- Abstand zur Referenzkurve Γ : Der für das Gütemaß benötigte Abstand zur Referenzkurve muss bestimmt werden.

Ein zentraler Aspekt bei der Zustandsbewertung sind die zugrunde liegenden Abstandsmetriken; die Berechnung ist dabei abhängig davon, ob der Abstand zu einem Objekt, einem Polygonzug oder einer Referenzkurve bestimmt werden soll.

3.4.1 Abstand zu Objekten

Erkannte Objekte werden als (zusammengesetzte) konvexe Polygone beschrieben. Als Abstandsmaß kommt eine pseudosemantische Metrik [5] zum Einsatz, welche außerhalb des Objekts durch den euklidischen Abstand, innerhalb durch den relativen Abstand zum Schwerpunkt des Polygons definiert ist. Damit das Abstandsmaß auch an den Eckpunkten zweifach stetig differenzierbar ist, werden Ecken mittels Polynom fünften Grades abgerundet, der Abstand kann somit nicht analytisch berechnet werden. Stattdessen wird dieser über die Länge $\|\mathbf{x} - \mathbf{s}\|$, bestimmt durch den Schnittpunkt S des Vektors vom Abrundungszentrum C der jeweiligen Ecke zur Referenzposition X , approximiert (siehe Abbildung 2). Für die folgende Beschreibung stellt x einen Wert im Ursprungskordinatensystem, \tilde{x} im transformierten System dar.

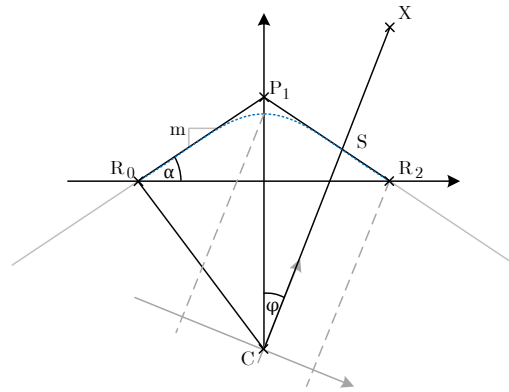


Abbildung 2: Skizzierung des Prinzips zur Abrundung der Polygonecken.

Anstelle der direkten Berechnung des Schnittpunktes im Ursprungskordinatensystem wird das Interpolationspolynom um φ , gegeben durch den Winkel zwischen $\mathbf{p}_1 - \mathbf{c}$ und $\mathbf{x} - \mathbf{c}$, verdreht und an der Stelle $\tilde{x} = 0$ ausgewertet. Es muss somit lediglich der Koeffizient \tilde{a} des verdrehten Polynoms $\tilde{\mathcal{P}}$ berechnet werden. $\tilde{\mathcal{P}}$ ist eindeutig durch die Funktionswerte sowie die erste und zweite Ableitung an den Stellen $\tilde{\mathcal{P}}(\tilde{0})$ und $\tilde{\mathcal{P}}(\tilde{x}_{R_2})$ definiert. Um diese Werte zu erlangen, wird das Ursprungspolynom \mathcal{P} an den Stellen 0 und x_{R_2} ausgewertet, die Werte entsprechend in das neue Koordinatensystem transformiert. Für die Funktionswerte sowie die erste Ableitung ist dieser Schritt trivial. Die zweite Ableitung ist

lediglich an der Stelle $x = 0$ von 0 verschieden und muss daher nur hier verdreht werden. Dies geschieht durch Differenzierung der ersten Ableitung in Richtung der Abszisse des verdrehten Koordinatensystems, siehe (3).

$$\frac{d^2\mathcal{P}}{d\tilde{x}^2}(\tilde{x}_{R_2}) = -\frac{3m \cdot \sec \alpha}{2 \cdot x_{R_2}} \quad (3)$$

Mithilfe der Stützpunkte kann jetzt \tilde{a} und somit der gesuchte Abstand berechnet werden. Um der C^2 -Forderung zu genügen, muss abschließend noch eine Normierung gemäß

$$\text{dist} = \frac{\|\mathbf{x} - \mathbf{c}\| - \tilde{a}}{\tilde{a} \cdot \|\mathbf{r}_2 - \mathbf{c}\|} \quad (4)$$

durchgeführt werden. Auf das ermittelte Maß wird ein Sicherheitsabstand addiert, so dass die zuvor beschriebene Filtermethodik angewandt werden kann.

3.4.2 Abstand zu einem Polygonzug

Sowohl die Referenzkurve als auch die Randbegrenzung können als Polygonzug gegeben werden. Im Gegensatz zum äußeren Bereich bei konvexen Polygonen ist der euklidische Abstand bei Polygonzügen auch nach Abrundung nicht zwingend differenzierbar. Zur Vermeidung nicht differenzierbarer Stellen wird um diesen ein Toleranzband gelegt, das eine eindeutige Zuordnung des nächstgelegenen Punktes ermöglicht. Die Bestimmung des Abstandes erfolgt dann gemäß Abschnitt 3.4.1.

3.4.3 Abstand zu einer geschlossenen Referenzkurve

Liegt eine Referenzkurve oder die Begrenzung der Straße als Spline vor, muss eine Projektion der Fahrzeugposition auf diesen erfolgen. Hierfür wird ein Verfahren ähnlich dem in [7] vorgestellten angewandt. Bei Krümmungen unterhalb eines Grenzwertes wird hierbei direkt eine Projektion mittels Lot auf die Fahrbahn gefällt.

Die Verwendung geschlossener Kurven hat sowohl Vor- als auch Nachteile gegenüber den Polygonzügen. Auf der einen Seite entstehen im Straßenverlauf keine kantigen Züge, was in langgezogenen Kurven bei entsprechender Gewichtung von w_ψ und w_d zu ruckartigen Lenkbewegungen führen kann, auf der anderen Seite ist die Projektion nur sequentiell berechenbar und wirkt sich somit negativ auf die Laufzeit des Trajektorienplaners aus.

3.5 Implementierungsdetails

Durch die hohe Anzahl an Stützpunkten, die zur Bestimmung der Zwischenzustände benötigt werden, sowie der teilweise aufwändig zu berechnenden Metriken wird eine entsprechend leistungsstarke Hardware zur Lösung des OCPs benötigt. Die konkrete Implementierung ist dabei auf ein heterogenes System bestehend aus einer CPU und einer GPU hin optimiert [6], wobei der Solver auf der CPU, die restlichen Berechnungen parallelisiert auf der GPU ausgeführt werden. Um die maximale Stabilität bei der Optimierung zu erreichen, werden neben den Funktionswerten auch deren erste und zweite Ableitungen in Richtung der Parameter berechnet. Hierdurch steigt der Berechnungsaufwand mit zunehmender Anzahl n an Parametern proportional zu $n \cdot (n+1)/2$ an. Aktuell wird die Steuerungsfunktion über 14 Stützpunkte variiert.

4 Ergebnisse

Der beschriebene Ansatz zur Trajektorienplanung befindet sich noch in der Entwicklung und wird derzeit ausschließlich in der Simulation getestet. Die in diesem Abschnitt vorgestellten Ergebnisse stützen sich daher nicht auf Realversuche. Die Streckendaten werden durch den Fahr Simulator Silab des WIVW erzeugt; die Modellierung der Straße basiert bei diesem nicht auf Geraden und Klothoiden, vor Kurveneingängen kommt es weiterhin zu gegengerichteten Überschwingern. Da der Trajektorienplaner dem Verlauf der Strecke folgt, sind somit auch diese Überschwinger in der resultierenden Fahrtrajektorie sichtbar. Im Folgenden werden die Ergebnisse aus zwei Anwendungsfällen gezeigt: die Fahrt durch ein Stadtszenario sowie die Durchführung eines Ausweichmanövers im fahrdynamischen Grenzbereich.

4.1 Stadtszenario

Zum Test des Fahrverhaltens in der Stadt wurde ein Rundkurs mit Hindernissen und Engstellen erstellt, siehe Abbildung 3. Um zu testen, wie gut das Fahrzeug dem Straßenverlauf folgt, ist die Platzierung der Hindernisse auf den ersten Teil der Strecke beschränkt. Sie werden erst ab einer Entfernung von 20 m erkannt, um reale Sensorik nachzubilden.

Um die Stabilität beim Kaltstart zu gewährleisten, erfolgt die Planung hier in zwei Schritten: Zunächst werden lediglich die Streckendaten berücksichtigt und eine Route auf der Straße geplant. Anschließend findet eine Berücksichtigung der Hindernisse statt und die Fahrtrajektorie wird darauf angepasst. Der Lenkwinkelverlauf gestaltet sich sehr glatt,

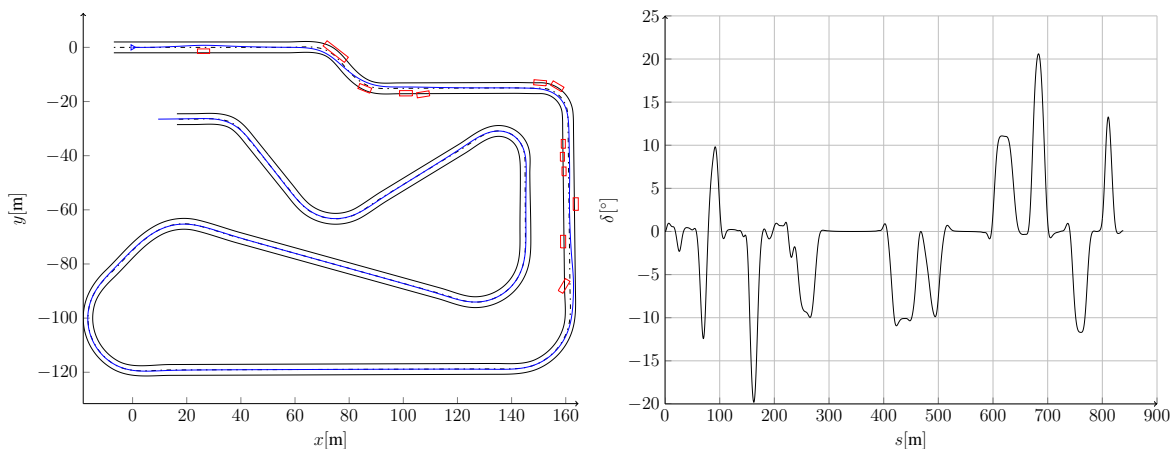


Abbildung 3: Links: Strecke des Stadtszenarios mit Referenzkurve (gestrichelt), Hindernissen (rot) und Fahrtrajektorie (blau). Rechts: Resultierender Krümmungsverlauf.

die Überschwinger vor sowie die Oszillationen in den Kurven sind durch die beschriebene Übergabe der Streckendaten begründet. Beim Übergang in die Gerade nähert sich das Fahrzeug schnell der Referenzkurve an ohne Überschwinger zu erzeugen. Den platzierten Hindernissen wird unter Einhaltung des festgelegten Sicherheitsabstands ausgewichen, ohne unnötig weit von der Spurmitte abzuweichen. Die Anzahl an Iterationen bleibt in diesem Szenario bei einer Vorausschau von 70 m unter 10, wodurch sich Gesamtlaufzeiten von 15 ms im Worst Case ergeben. Abbildung 3 zeigt den Verlauf der Trajektorie über die gesamte Strecke an.

4.2 Ausweichmanöver

Im zweiten Szenario wird ein Ausweichmanöver bei konstanter Geschwindigkeit durchgeführt, siehe Abbildung 4. Das Hindernis ist in rot dargestellt, die resultierende Fahrtrajektorie in blau. Damit die maximal zulässige Querbewegung $a_{\text{lat}}^{\text{max}}$ nicht überschrit-

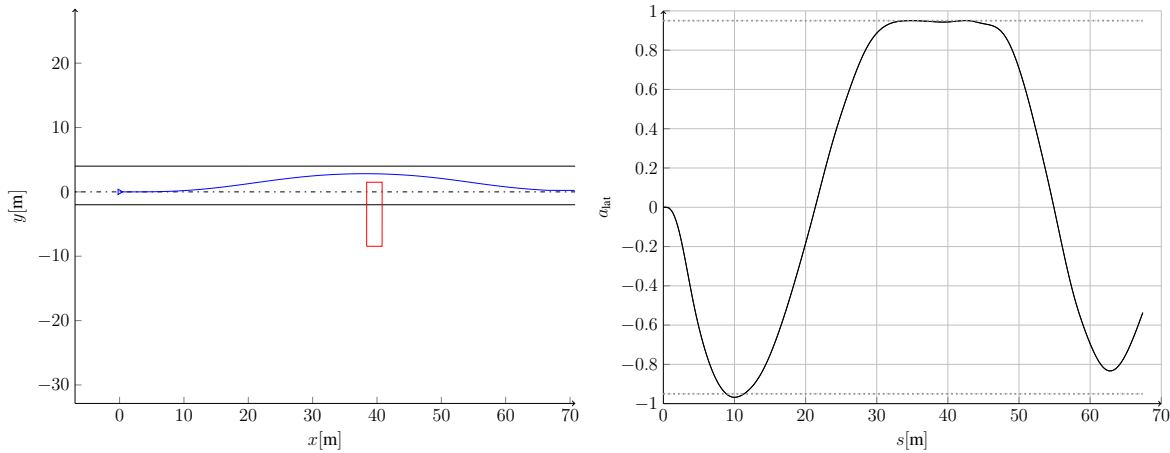


Abbildung 4: Links: Berechnete Ausweichtrajektorie (blau) um Hindernis (rot). Rechts: Querbewegung des geplanten Manövers.

ten wird, muss der Lenkwinkel in diesem Szenario beschränkt werden. $a_{\text{lat}}^{\text{max}}$ wird über die Filterfunktion ab 0.95 g als kritisch, ab 1 g als unzulässig eingestuft. Um die Konvergenz zu stabilisieren und zu beschleunigen, wird die Einhaltung zusätzlich an dedizierten Stellen explizit gefordert. Die resultierende Anzahl an Iterationen beträgt im Kaltstart 10, die gesamte Berechnungszeit liegt unter 15 ms. Abbildung 4 stellt die berechnete Trajektorie sowie die resultierende Querbewegung im ersten Planungsschritt dar. Im Grenzbereich schmiegt sich a_{lat} an $a_{\text{lat}}^{\text{max}}$ an und überschreitet den maximal zulässigen Wert von 1 g nicht.

5 Zusammenfassung und Ausblick

Der Beitrag beschreibt einen Trajektorienplaner, der im Gegensatz zu bisher bestehenden Algorithmen nicht die Zwischenpositionen auf der Straße, sondern die Steuerfunktion direkt bestimmt, wodurch auf eine Approximation der Zustandsübergänge verzichtet werden kann. Das Verfahren basiert dabei auf der echtzeitfähigen Lösung eines Optimalsteuerungsproblems. Die bisherigen Tests in einer Simulationsumgebung zeigen, dass die Optimierung im Warmstart auch bei plötzlich erkannten Hindernissen unter 20 ms durchgeführt werden kann und dass das resultierende Fahrverhalten nachvollziehbar und frei von unkomfortablen Störeinflüssen ist, womit alle Anforderungen erfüllt werden.

Obwohl relativ hohe Anforderungen an die zugrundeliegende Hardware³ gestellt werden, bietet der Planer großes Potential. Neben der Möglichkeit, beliebige Fahrzeugmodelle zu hinterlegen, können Randbedingungen und Kostenterme frei formuliert werden. Die kontinuierliche Restriktionsüberwachung garantiert darüber hinaus, dass die geplante

³Bei entsprechender Spezialisierung auf derartige Probleme könnten dabei die Hardwarekosten zukünftig stark reduziert werden.

Trajektorie auch tatsächlich fahrbar ist; konservative Abschätzungen, welche bei ausschließlich stützpunktbasierter Überprüfung notwendig sind, können entfallen.

Um den Trajektorienplaner weiter zu verbessern, wird im nächsten Schritt zum einen die Optimierung des Geschwindigkeitsprofils mit einbezogen, zum anderen wird die Berücksichtigung dynamischer Hindernisse implementiert. Um hierbei weitestgehend unabhängig vom Startwert zu sein, muss eine entsprechende Modellierung dieser Hindernisse entwickelt werden, damit lokale Minima vermieden werden.

Literatur

- [1] Betts, J.: Practical Methods for Optimal Control Using Nonlinear Programming. Advances in design and control. Society for Industrial and Applied Mathematics, Philadelphia, PA (2010)
- [2] Ferguson, D., Howard, T., Likhachev, M.: Motion Planning in Urban Environments: Part I. In: Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems. IEEE (2008)
- [3] Gu, T., Snider, J., Dolan, J., Lee, J.: Focused Trajectory Planning for Autonomous On-Road Driving. In: Intelligent Vehicles Symposium. IEEE (2013)
- [4] Hudecek, J., Eckstein, L.: Ein hybrider Ansatz zur Realisierung echtzeitfähiger Optimalsteuerungen im Kraftfahrzeug. In: AAET Braunschweig (2015)
- [5] Hudecek, J., Eckstein, L.: GPU Based Tactical Motion Planning - Das Fundament für hochintegrierte Fahrerassistenz. In: Elektronik im Fahrzeug - Die neue Welt der Automobilindustrie (2015)
- [6] Hudecek, J., Eckstein, L.: Real-Time Capable Solution for Optimal Control problems. In: WKM Symposium (2015)
- [7] Irle, P., Gröll, L., Werling, M.: Zwei Zugänge zur Projektion auf 2d-Kurven für die Bahnregelung autonomer Fahrzeuge. In: AT - Automatisierungstechnik (2009)
- [8] Schwesinger, U., Ruffli, M., Furgale, P., Siegwart, R.: A Sampling-Based Partial Motion Planning Framework for System-Compliant Navigation Along a Reference Path. In: Intelligent Vehicles Symposium, pp. 391–396. IEEE (2013)
- [9] Werling, M., Kammel, S., Ziegler, J., Gröll, L.: Optimal Trajectories for Time-Critical Street Scenarios Using Discretized Terminal Manifolds. I. J. Robot. Res. **31**(3), 346–359 (2012)
- [10] Ziegler, J., Bender, P., Dang, T., Stiller, C.: Trajectory planning for Bertha - A local, continuous method. In: Intelligent Vehicles Symposium, pp. 450–457. IEEE (2014)
- [11] Ziegler, J., Stiller, C.: Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In: Intelligent Robots and Systems. IEEE (2009)