

Trajektorienplanung mittels MPC und adaptiver Zielpunktgenerierung

Francois Mitterand Noumechi

Masterarbeit – 07. Februar 2018

Betreuer: Prof. Dr.-Ing. J. Adamy, M.Sc. F. Müller



TECHNISCHE
UNIVERSITÄT
DARMSTADT

REGELUNGSMETHODEN
UND ROBOTIK



Aufgabenstellung

Bei der Trajektorienplanung werden unterschiedliche Verfahren verwendet. Dazu gehört die modellprädiktive Regelung. Dieses Verfahren eignet sich sehr gut für die Behandlung beschränkter und nichtlinearer Systeme. Daher wird ihm zurzeit eine besondere Aufmerksamkeit in dem Forschungsgebiet Regelungstheorie gewidmet. In der Praxis wird jedoch die Anwendung dieses Verfahrens aufgrund der hohen Rechenzeit aus der Fassung gebracht. Aus diesem Grund wird in dieser Arbeit ein Planungskonzept entworfen, das basierend auf der modellprädiktiven Regelung die hohe Rechenzeit reduziert.

Beginn: 07. August 2017

Ende: 07. Februar 2018

Seminar: 07. Februar 2018

Prof. Dr.-Ing. Jürgen Adamy

Prof. Dr.-Ing. J. Adamy
M.Sc. F. Müller

Technische Universität Darmstadt
Institut für Automatisierungstechnik und Mechatronik
Fachgebiet Regelungsmethoden und Robotik
Prof. Dr.-Ing. Jürgen Adamy

Landgraf-Georg-Straße 4
64283 Darmstadt
Telefon 06151/16-25050
www.rmr.tu-darmstadt.de/



Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 07. Februar 2018

Francois Mitterand Noumechi

Kurzfassung

Heutzutage wird intensiv über die Entwicklung selbstfahrender Fahrzeuge geforscht. Eins der Themen, das dabei im Fokus steht, ist die Trajektorienplanung. Diese Aufgabenstellung stellt eine Herausforderung dar, denn das Finden einer hindernisfreien und fahrbaren Trajektorie wird in einer dynamisch veränderbaren Umgebung mit viel Rechenaufwand verbunden. Nach vielen Jahren der Forschung wurden unterschiedliche Konzepte und Verfahren zur Berechnung der Trajektorie eines Fahrzeugs entwickelt. Diese Arbeit richtet sich in die gleiche Richtung und schlägt daher ein Konzept zur Ermittlung der Trajektorie eines Fahrzeugs vor. Ausgangspunkt ist die Minimierung der Rechenzeit. Die Aufgabe der Trajektorienplanung wird daher in kleine Teilaufgaben zerlegt. Die Modellprädiktive Regelung(MPR) wird zur Lösung von Teilaufgaben verwendet. Dabei werden anhand eines überlagerten Verfahrens, straßenadaptierte Zwischenzielpunkte generiert, um ein lokales Ziel für die Planung zu ermitteln.

Schlüsselwörter: Trajektorienplanung, MPR, Optimierung, Zielpunktgenerierung, autonome fahrende Fahrzeuge, Hindernisvermeidung

Abstract

In the last few years many researches have been published on autonomous driving vehicles. One of the main topics treated is about trajectory planning. Trajectory planning in an dynamic changing environment is a challenging problem, due to the presence of multiple constraints. However significant efforts have been made to developing concepts and approaches for calculating the trajectory of a vehicle and reducing the computing efforts. This work follows the same goal and therefore proposes a concept for determining the trajectory of a vehicle. Reduction of the computing time will be one the challenges of this work. The task of trajectory planning is therefore to be divided in to small subtasks. Model predictive control (MPC) is used for solving subtasks. Using a superimposed procedure, street-adapted intermediate targets are generated in order to determine a local target for planning.

Keywords: trajectory planning, MPC, optimization, autonomous driving, goal point generation, obstacle avoidance

Inhaltsverzeichnis

Symbole und Abkürzungen	vii
1. Einleitung	1
1.1. Einführung in die Trajektorienplanung	1
1.1.1. Problemformulierung	1
1.1.2. Verfahren zur Trajektorienplanung	2
1.1.3. Motivation und Zielsetzung	3
1.2. Aufbau der Arbeit	4
2. Mathematische Grundlagen	5
2.1. Nichtlineare Systeme	5
2.1.1. Definition	5
2.1.2. Lösung nichtlinearer Differentialgleichungen	7
2.2. Optimierungsaufgaben	10
2.2.1. Statische Optimierungsprobleme	11
2.2.2. Optimalsteuerungsprobleme	12
2.3. Grundlagen der Modellprädiktiven Regelung	16
2.3.1. Funktionsweise der MPC	17
2.3.2. Mehrschießverfahren	21
2.4. Hindernisbeschreibung	24
2.4.1. Ellipsen- und Parabelförmige Hindernisse	24
2.4.2. Rechteckförmige Hindernisse	24
3. Modellbildung und Koordinatendefinition	27
3.1. Modellierung der Fahrzeugbewegung	27
3.2. Koordinatentransformation	30
3.2.1. Globale- und Lokale Betrachtung des Problems	30
3.2.2. Relativbewegung	31
3.3. Darstellung von Hindernissen im \mathcal{F}_s	33
4. Zielpunktadaptive Trajektorienplanung	37
4.1. Strategie zur Trajektorienplanung	37
4.2. Generierung von Zielpunkten	40
4.2.1. Beschreibung des Verfahrens	40
4.2.2. Lösungsansatz	41

4.3.	Formulierung der MPC	43
4.3.1.	Auswahl der Zielfunktion	45
4.3.2.	Kollisionsvermeidung	46
4.3.3.	Systembeschränkungen	47
4.4.	Regelungsentwurf	48
4.4.1.	Zielpunkt im Sichtbereich	48
4.4.2.	Zeitressource	48
4.5.	Kurzfassung	50
5.	Evaluation	51
5.1.	Verwendete Werkzeuge	51
5.2.	Evaluierung der Generierung von Zielpunkten	52
5.3.	Bewertung der MPC	53
5.4.	Ermittlung eines adaptiven Zielpunkts	54
5.5.	Evaluierung des Gesamtkonzeptes	55
5.5.1.	Geradeausfahrt	56
5.5.1.1.	Abschnitt S_1	56
5.5.1.2.	Abschnitt S_2	58
5.5.2.	Kurvenfahrt	59
5.5.2.1.	Abschnitt S_3	59
5.5.2.2.	Abschnitt S_5	61
5.6.	Gesamtstrecke	62
5.7.	Auftretende Probleme	63
5.8.	Vergleich des Planungskonzepts	64
6.	Zusammenfassung und Ausblick	69
A.	Anhang	71
A.1.	Polynomfunktionen	71
A.2.	Methode der kleinsten Quadrate	72
A.3.	Tangentengleichung	73
A.4.	Graphische Ermittlung der Tangente	73
B.	Transformation von Koordinatensystemen	75
C.	Frei wählbare Parameter und Planungskonzept	77
	Literaturverzeichnis	79

Symbole und Abkürzungen

Notation

Symbol	Beschreibung	Einheit
$(\cdot)_{max}$	maximal umsetzbare Komponente	
$(\cdot)_{min}$	minimal umsetzbare Komponente	
$(\cdot)_{soll}$	Sollkomponente	
$(\cdot)_{ref}$	Referenzkomponente	
$\tilde{(\cdot)}$	approximierte Komponente	
$(\cdot)^*$	optimale Komponente	
$(\cdot)_{ziel}$	Zielkomponente	
$(\cdot)_h$	Hinderniskomponente	
$(\cdot)_f$	Fahrzeugskomponente	
$\dot{(\cdot)}$	Zeitableitung	
Vektoren und Matrizen	fett geschriebenen Größen	
Skalare	nicht fett	

Lateinische Symbole und Formelzeichen

Symbol	Beschreibung	Einheit
x, y	straßenfeste Koordinaten	m
$X_{(\cdot)}, Y_{(\cdot)}$	weltfeste Koordinaten	m
v	lineare Geschwindigkeit	ms^{-1}
\mathbf{x}	Zustandsvektor	
\mathbf{u}	Stellgrößenvektor	
\mathbf{f}	Vektorfunktion	
$\min f(x)$	Minimum einer Funktion f	
$\max f(x)$	Maximum einer Funktion f	
$\frac{\partial y}{\partial x}$	partielle Ableitung von y nach x	

Griechische Symbole und Formelzeichen

Symbol	Beschreibung	Einheit
θ	Orientierung des Fahrzeugs bezüglich \mathcal{F}_s	rad
θ_s	Orientierung von \mathcal{F}_s bezüglich \mathcal{F}_w	rad
θ_h	Orientierung eines Hindernisses im \mathcal{F}_s	rad
ω	Drehwinkelgeschwindigkeit des Fahrzeugs im \mathcal{F}_s	rads ⁻¹

Abkürzungen

Kürzel	vollständige Bezeichnung
Dgl.	Differentialgleichung
KQV	Kleinste Quadrate Verfahren(engl. Least Squares, LS)
MIMO	Multiple Input Multiple Output
SISO	Single Input Single Output
AWP	Anfangswertproblem
RWP	Randwertproblem
RKV4	Runge-Kutta-Verfahren vierte Ordnung
MPC	Model predictive control
NMPC	Nonlinear model predictive control
NLP	Nichtlineares Programm(engl. Nonlinear programming problem)
KS	Koordinatensystem
OCP	Optimal control problem
u.B.	unter Beschränkungen
bzw.	beziehungsweise
d. h.	das heißt
z. B.	zum Beispiel
usw.	und so weiter

Sonstige Symbole

Symbol	Beschreibung
\mathbb{R}^n	n –dimensionaler reeller Zahlenraum
\mathcal{F}_w	Weltfestes Koordinatensystem
\mathcal{F}_s	Straßenkoordinatensystem
\mathcal{F}_f	Fahrzeugskordinatensystem
\mathcal{F}_h	Hindernisbezogenes Koordinatensystem
\mathcal{R}	Referenzweg
$J_{in}(\cdot)$	Integralrechnung mit bekannten Parametern
$S_{im}(\cdot)$	Integralrechnung mit unbekannten Parametern

1 Einleitung

1.1 Einführung in die Trajektorienplanung

Die Entwicklung von Fahrzeugen mit hoher Geschwindigkeit wird ohne Hilfe der maschinellen Wahrnehmung eindeutig mehr Probleme verursachen, als sie lösen werden. Die Anzahl der durch menschliches Fehlverhalten verursachten Unfälle ist ein Beweis dafür, dass der Mensch alleine in vielen Verkehrssituationen nicht mehr in der Lage ist, das Fahrzeug zu bedienen. Denn rechtzeitige Reaktionen und richtige Entscheidungen sind in kritischen Situationen zwingend notwendig. Daneben muss auch die Missachtung der Verkehrsordnung erwähnt werden. Denn sie hat dazu beigetragen, die Anzahl potenzieller Gefahrensituationen im Verkehr zu erhöhen. Diese Feststellung und viele andere sind ursächlich für den Fortschritt, der in den letzten Jahren bei automatischen Fahrsystemen gemacht wurde. Es hat sich dadurch gezeigt, dass automatisierte Fahrfunktionen in Automobilen sehr nützlich sind, weil sie den Mensch beim Fahren assistieren können. Ein wichtiges Teilproblem, das hierbei an zunehmender Bedeutung gewinnt, ist die Trajektorienplanung. Dabei wird anhand von Informationen aus einer Karte der Umgebung, die Solltrajektorie eines Fahrzeugs berechnet. Wenn diese Solltrajektorie genau geplant ist, wird die Fahrzeugbewegung dadurch eindeutiger. Somit lassen sich nicht nur Unfälle vermeiden, sondern wird auch Zeit gewonnen. Denn das Problem des stockenden Verkehrs kann beispielsweise gelöst werden, wenn alle im Verkehr befindlichen Fahrzeuge eine vordefinierte Trajektorie verfolgen würden. Da nicht alle möglichen Bewegungen eines Fahrzeugs geplant werden können, stellt die Trajektorienplanung ein umfangreiches Gebiet der Forschung dar.

1.1.1 Problemformulierung

Moderne Fahrzeuge werden mit automatisierten Fahrfunktionen gestaltet. Dies umfasst viele Komponenten, womit das Fahrzeug mit seinem Umfeld interagiert. Zu diesen Komponenten gehört die Trajektorienplanung, welche zur Ermittlung der zu fahrenden Solltrajektorie verwendet wird. Die Aufgabe der Trajektorienplanung ist nicht nur den fahrbaren Weg eines Fahrzeugs zu bestimmen, sondern beinhaltet auch die Berechnung der Geschwindigkeit und der Beschleunigung[1]. Das Ziel dabei ist das autonome Fahren zu ermöglichen. Allerdings kann es je nach Situation unterschiedliche Ansprüche geben, welche die Planung komplizierter machen. Es wird daher bei der Trajektorienplanung große Anstrengungen unternommen, um Strategien zu entwerfen, die sich in unterschiedlichen Situationen anpassen können. An dieser Stelle wird auf computergestützte Algorithmen zurückgegriffen. Die Herangehensweise besteht häufig darin Anforderungen, wie das Bestimmen des kürzesten Weges zwischen zwei vorgegebenen Stellen im

Raum, mathematisch zu formulieren. Die hiermit formulierte Aufgabe kann beispielsweise in eine Optimierungsaufgabe überführt und unter Einhaltung des zeitveränderlichen Verhaltens der Strecke (Modell des Fahrzeugs) und Hindernisse gelöst werden. Das Umfeld des Fahrzeugs spielt daher bei der Planung eine zentrale Rolle. Bei einer Langsamfahrt auf einer Parkfläche kann das Umfeld als statisch betrachtet werden. Das Planungsziel wird beispielsweise sein, ein Zielbereich beim Einparkmanöver (bzw. Ausparkmanöver) zu erreichen. Im Gegenteil dazu wird bei einer Autobahnfahrt (bzw. fließender Verkehr) die Umgebung als dynamisch veränderlich angenommen. In diesem Fall wird die Planung deutlich komplexer, denn die Trajektorie aller bewegten Objekten müssen zusätzlich explizit beschrieben werden. Die Zielsetzung der Planung kann darin bestehen, innerhalb einer Fahrspur zu bleiben, eine vorgegebene Geschwindigkeit (oder Abstand) bezüglich des vorderen Fahrzeugs zu halten oder die Fahrspur zu wechseln, wenn die Möglichkeit besteht [2]. Somit wird die Aufgabe der Trajektorienplanung eindeutig und kann zusammengefasst werden:

Die Aufgabe der Trajektorienplanung besteht darin, eine fahrbare und kollisionsfreie Bahn unter Einhaltung der Umfelderkennung zu ermitteln, um eine vordefinierte Bewegung eines selbstfahrenden Fahrzeugs zu ermöglichen. Hierbei wird die Angabe über die Zeitkomponente verlangt.

1.1.2 Verfahren zur Trajektorienplanung

Bei den in zukünftigen Fahrzeugen zu bauende Fahrerassistenzsystemen übernimmt die Trajektorienplanung eine der wichtigsten Rollen. In diesem Bereich wird sowohl für mobile Roboter als auch für selbstfahrende Fahrzeuge zurzeit intensiv geforscht. Dies hat dazu beigetragen, dass unterschiedliche Verfahren zur Lösung immer größer werdender Anzahl von Problemen entworfen wurden. Die Literatur bietet daher eine Übersicht über eine Vielzahl von Verfahren an [3, 4, 5], die durch ihre Ursprünge gruppiert werden können.

Schon früher wurden Zustandsgitter oder Gitter aus Grundgeometrien (engl. state lattices) bei der Bewegungsplanung verwendet. Dabei werden nach der Diskretisierung des Lösungsraums in kleinen Teilbereichen Graphsuchverfahren eingesetzt, um einen optimalen Pfad zu ermitteln. Sehr bekannte Suchalgorithmen wie Dijkstras oder A^* werden am häufigsten verwendet [4]. Die dadurch erhaltenen Trajektorien zeigen jedoch Ecken an manchen Stellen. Die mit dem Fahrzeug verbundenen Restriktionen lassen sich nur schwer behandeln. Das hat zur Folge, dass die Umsetzung problematisch wird. Bei solchen Methoden lassen sich gute Ergebnisse erzielen, wenn das Gitter fein genug (sehr kleine Auflösung) ist. Die Suche in dem Graphen wird dadurch aufwendiger [5].

In einigen Arbeiten werden probabilistische Bewegungsplanungsalgorithmen, wie z. B. der Rapidly-exploring Random Tree [6, 7, 3] oder Probabilistic Roadmap, verwendet. Es werden sehr viel Trajektorien erzeugt, wovon nur sehr wenige verwendet werden. Laut [5] existiert keine Garantie für das Finden optimaler Lösungen. Es wurden daher von diesen Verfahren abgeleitete Varianten entworfen [8].

Kurvenbasierte Verfahren stellen weitere Möglichkeiten dar, die Aufgabenstellung der Bewe-

gungsplanung zu lösen. Durch ihre Verläufe eignen sich Funktionen, wie z. B. Polynome, Splines, Klothoiden, Bezier-Kurven, Dubins-Pfad, oder Sigmoiden zur geometrischen Darstellung von Trajektorien[9, 2, 4]. Der Vorteil dabei ist der geringe Rechenaufwand, denn nur sehr wenige Parameter werden bei der Beschreibung einer solchen Kurve benötigt. Die Dynamik des Fahrzeugs wird jedoch nicht berücksichtigt. Zudem werden weitere Transformationen benötigt um die daraus entstandenen Trajektorien zu glätten.

Die bei der Trajektorienplanung formulierten Ansprüche können formelmäßig in eine Optimierungsaufgabe überführt werden. Um die dynamisch veränderbare Umgebung zu berücksichtigen, lässt sich diese Aufgabe zyklisch unter Rückführung des aktuellen Systemszustands lösen. Diese Vorgehensweise wird in der Literatur unter dem Begriff Modellprädiktive Regelung bezeichnet[2, 10, 11, 12, 8, 13, 14, 15]. Dieses Verfahren unterscheidet sich von den anderen dadurch, dass sich nicht nur die Restriktionen gut behandeln lassen, sondern die Dynamik des Fahrzeugs direkt mit in die Optimierung herangezogen wird. Dies führt dazu, dass die berechneten Trajektorien einfacher umsetzbar werden. Die Modellprädiktive Regelung stellt daher einen sehr guten Lösungsweg dar, um allgemeine nichtlineare Systeme zu behandeln. Bei der Trajektorienplanung gehört sie zu den beliebten Verfahren, wobei in den letzten Jahren viele Fortschritte gemacht wurden. Während der Fahrt begegnet man jedoch unterschiedliche Situationen, an denen bestimmte Fahrmanöver erforderlich sind. Die Modellprädiktive Regelung wird mit anderen Methoden kombiniert, wie z. B. die Potenzialfeldmethode, um die Durchführung bestimmter Aufgaben zu ermöglichen. Weitere Nachteile des Verfahrens liegen darin, dass in einer komplexen Umgebung sehr viel Restriktionen berücksichtigt werden müssen. Dies ist beispielsweise der Fall wenn, viele Hindernisse im Umfeld des Fahrzeugs liegen. Der Rechenaufwand steigt dadurch sehr enorm, was die Umsetzung des Verfahrens stark beeinflussen kann.

1.1.3 Motivation und Zielsetzung

In dieser Arbeit wird die Aufgabenstellung der Trajektorienplanung betrachtet. Besondere Aufmerksamkeit wird dabei auf die Modellprädiktive Regelung gerichtet. Dieses Verfahren eignet sich sehr gut für die Trajektorienplanung eines Fahrzeugs, insofern dass die Handhabung unterschiedlicher Restriktionen, die während der Fahrt auftreten können, garantiert ist. Allerdings wird der Rechenaufwand stark steigen, wenn zu viel Faktoren mit berücksichtigt werden müssen. Bei der Umsetzung des Verfahrens werden in einigen Arbeiten[14] eine Teststrecke mit vordefinierter Anzahl von Hindernissen global berücksichtigt. Um die Echtzeitfähigkeit des Verfahrens zu gewährleisten, werden sehr wenige Hindernisse angenommen. In der Realität würde eine globale Betrachtung der Teststrecke jedoch sehr viel Rechenaufwand verursachen, weil es sehr viele Hindernisse in jedem Schritt der Planung zu zählen gibt. Diese Arbeit lässt sich durch diese Feststellung motivieren. Sie hat sich daher das Ziel gesetzt, die Aufgabe der Trajektorienplanung mittels Modellprädiktive Regelung in kleinen Teilaufgaben zu zerlegen, um den Rechenaufwand Aufgrund vieler Hindernisse klein zu halten.

Eine lokale Betrachtung des Problems würde aber bedeuten, dass ein lokaler Zielpunkt vor-

gegeben wird. Es werden daher adaptive Zwischenzielpunkte auf der Fahrbahn definiert, um den Lösungsraum zu beschränken. Dadurch wird ein Hindernis nur dann in die Optimierung herangezogen, wenn es auch nötig wird.

1.2 Aufbau der Arbeit

Diese Arbeit wird folgenderweise gegliedert:

- In Kapitel 2 werden alle benötigten Grundlagen erläutert. Diese übermitteln eine Übersicht über die Verfahren, die in den nachfolgenden Kapiteln umgesetzt und implementiert werden. Dabei wird vor allem die allgemeine Formulierung und Lösung von Optimalsteuerungsproblemen betrachtet, sowie die Grundidee der Modellprädiktive Regelung.
- Kapitel 3 fasst zwei wichtige Themen dieser Arbeit zusammen. Zunächst wird aus der Literatur ein Fahrzeugmodell verwendet, um das Testfahrzeug vorzustellen. Danach wird ein Koordinatensystem zur lokalen Beschreibung des Fahrzeugs definiert. Die dabei benötigten Transformationen werden gegeben. Abschließen werden Hindernisse durch die vorgestellte Koordinatentransformation beschrieben.
- Kapitel 4 stellt das Gesamtkonzept der Trajektorienplanung vor. Die Umsetzung der Modellprädiktive Regelung wird beschrieben. Kombiniert mit anderen Verfahren zur adaptiven Zielpunktgenerierung wird erklärt, wie der Rechenaufwand klein gehalten wird. Am Ende des Kapitels wird ein Regelungssystem vorgestellt, womit das System geregelt wird.
- In Kapitel 5 werden zunächst die beschriebenen Verfahren einzeln in einem konkreten Beispiel getestet und bewertet. Anschließend wird das Gesamtkonzept getestet. Dabei wird an unterschiedlichen Szenarien untersucht, wie das entworfene Konzept funktioniert.
- Eine Ausführliche Zusammenfassung wichtiger Themen dieser Arbeit wird in Kapitel 6 dargestellt. Ein Ausblick schließt die Arbeit ab.

2 Mathematische Grundlagen

Dieses Kapitel bietet einen Überblick über einige mathematische Grundlagen über das Thema Trajektorienplanung. Mit der Entwicklung von mobilen Robotern und autonomen (bzw. selbstfahrenden) Fahrzeugen stellt die Trajektorienplanung heutzutage eine zentrale Aufgabe der Forschung dar. Zur Lösung dieser Aufgabe wurden zahlreiche Verfahren entwickelt und umgesetzt, woraus befriedigende Ergebnisse herausgezogen wurden. Alle diese Verfahren zeigen Vor- und Nachteile und werden abhängig von dem Verwendungszweck alleine oder kombiniert eingesetzt. Die Trajektorienplanung deckt daher ein großes Forschungsgebiet ab, an dem intensiv geforscht wird. Zum besseren Verständnis dieser Arbeit werden zunächst nichtlineare Differentialgleichungen eingeführt und die existierenden Lösungswege vorgestellt. Anschließend wird das Thema nichtlineare Optimierung detailliert besprochen und die darauf bauende Modell Prädiktive Regelung wird erläutert. Zum Abschluss des Kapitels wird das Thema Hindernis angesprochen. Dabei werden gängige Darstellungsmöglichkeiten gezeigt.

2.1 Nichtlineare Systeme

Fast alle in der Umgebung liegende Objekte ändern in der Zeit ihre Zustände. Die Zustände können beispielsweise Temperatur, Lage oder Menge eines Objekts sein, die verwendet werden können, um das Objekt zu beschreiben. Unter dem Einfluss externer oder interner Faktoren werden sich die Werte der Zustände ändern. Zur Beschreibung dieser Änderung lassen sich häufig eher nichtlineare Formeln herleiten. Dabei müssen sowohl die Zustände als auch alle externen und internen Faktoren mit berücksichtigt werden. Zu den externen Faktoren zählt beispielsweise die Kraft, die ein Körper verformen oder ihn in Bewegung setzen kann. Im Folgenden wird die Herleitung einer nichtlinearen Differentialgleichung beschrieben.

2.1.1 Definition

Zur Problemformulierung definieren wir zeitabhängige Variablen $x_i \in \mathbb{R}$ und frei wählbare Steuerefunktionen $u_i \in \mathbb{R}$, die durch die Gleichungen (2.1) und (2.2) beschrieben sind.

$$\begin{aligned} x_i : [t_0, t_f] &\rightarrow \mathbb{R} \\ t &\mapsto x_i(t) \end{aligned} \tag{2.1}$$

$$\begin{aligned} u_i : [t_0, t_f] &\rightarrow \mathbb{R} \\ t &\mapsto u_i(t) \end{aligned} \tag{2.2}$$

Diese Variablen können für eine Beobachtungszeit $t_f - t_0$ verwendet werden, um die Änderungen bei einem System darzustellen. Zur Beschreibung von Prozessen und Systemen, die mehr als einen Freiheitsgrad besitzen, kann für jeden Freiheitsgrad eine solche Variable definiert werden. Somit ergeben sich folgende Vektoren:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \quad \text{und} \quad \mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix}$$

Dabei beschreibt $\mathbf{x}(t)$ alle Änderungen des Systemszustands. Im Gegenteil dazu fasst $\mathbf{u}(t)$ alle Faktoren zusammen, die auf irgendwelcher Art das System beeinflussen können. Allgemein wird in dieser Arbeit ein Mehrgrößensystem(engl. MIMO) angenommen, das sich durch eine Differentialgleichung der Form (2.3) darstellen lässt.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (2.3)$$

Wobei die Systemgrößen \mathbf{x} , \mathbf{u} und \mathbf{f} als zeitabhängige Vektoren definiert und abhängig von den Parametern n und m dimensioniert sind.

\mathbf{x}	Zustandsvektor	$(n,1)$ – Vektor
\mathbf{u}	Stellgrößenvektor	$(m,1)$ – Vektor
\mathbf{f}	Vektorfunktion	$(n,1)$ – Vektor

Es wird ab hier darauf hingewiesen, dass der Parameter t teilweise weggelassen wird. Zudem werden die fett geschriebenen Größen immer Vektoren oder Matrizen darstellen. Die Struktur von $\mathbf{f}(\mathbf{x}, \mathbf{u})$ ermöglicht es lineare von nichtlinearen Systemen zu unterscheiden. Der große Teil von Prozessen wird aber in der Praxis als nichtlinear modelliert, somit stellt (2.3) eine nicht-lineare Dgl. dar. Im Vergleich zu linearen Systemen gibt es bei Nichtlinearen kein Verfahren zur Regelung, die für alle Systeme eingesetzt werden kann. Jedoch spielt die Untersuchung der Ruhelage in vielen Fällen eine wichtige Rolle, denn dadurch kann das System linearisiert werden und die bekannten Verfahren der linearen Regelung eingesetzt werden[16]. Die Ruhelage lässt sich durch Gleichung (2.4) ermitteln.

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0 \quad (2.4)$$

Zur Regelung eines nichtlinearen Systems wird der Stellgrößenvektor \mathbf{u} so ermittelt, dass das System eine gewünschte Bewegung durchführt. Somit kann der nächste Zustand des Systems entweder gemessen oder geschätzt werden. In der Theorie kann der Zustandsvektor \mathbf{x} leider nicht gemessen werden. Er wird eher dadurch ermittelt, dass die Dgl. (2.3) für eine Zeitspanne integriert wird.

2.1.2 Lösung nichtlinearer Differentialgleichungen

Zur Verallgemeinerung wird wieder ein System angenommen, das sich durch eine Dgl. der Form (2.3) beschreiben lässt. Wie schon erwähnt wurde, besitzt $\mathbf{f}(\mathbf{x}, \mathbf{u})$ eine nichtlineare Gestalt. Für ein System, das zu einem Anfangszeitpunkt t_0 den Zustandsvektor $\mathbf{x}(t_0)$ besitzt, kann eine Lösung des Problems in der Form (2.5) geschrieben werden.

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^t \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \quad (2.5)$$

Diese Form stellt die Lösung eines sogenannten Anfangswertproblem(AWP) zum Zeitpunkt t dar[16]. Der Nachteil dabei ist, dass dieses Gleichungssystem sich nicht generell analytisch lösen lässt. Aus der Literatur sind jedoch unterschiedliche Verfahren zu finden, die sich der Lösung anzunähern. Hierbei basieren sich diese Verfahren auf Integrationsmethoden. Der Ausgangspunkt ist die Diskretisierung der Zeitintervall in N Teilbereichen und das Problem in der Form (2.6) umzuwandeln, wobei h die Schrittweite darstellt.

$$\begin{aligned} \mathbf{x}(t_{i+1}) &= \mathbf{x}(t_i) + \int_{t_i}^{t_{i+1}} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \\ t_i &= t_0 + hi \quad \text{mit } i = 0, \dots, N \end{aligned} \quad (2.6)$$

Die zuletzt erhaltene Gleichung lässt sich iterativ lösen, indem eine Integralrechnung in jedem Schritt durchgeführt wird. Das dadurch erhaltene Ergebnis ist lediglich eine Näherung des richtigen Werts. Außerdem führen die einzeln aufsummierten Rechnungsfehler zu einem Gesamtfehler, der abhängig von der Schrittweite h aufsteigt. Zu den einfachen und wichtigen Lösungsverfahren zählen die Runge-Kutta-Verfahren(bzw. Einschrittverfahren), die eine Reihe von Verfahren enthält. Dazu gehören folgende Verfahren, die sich lediglich in der Integrationsmethode unterscheiden[16]:

- Polygonzugverfahren von Euler-Cauchy

In diesem Verfahren wird die Integralrechnung durch ein Rechteck approximiert. Die Methode ist zwar einfach aber ungenau.

$$\hat{\mathbf{x}}(t_{i+1}) = \hat{\mathbf{x}}(t_i) + h\mathbf{f}(\hat{\mathbf{x}}(t_i), \mathbf{u}(t_i))$$

- Verbessertes Euler-Cauchy-Verfahren

Wie der Name schon besagt, wird bei diesem Verfahren versucht, die große Ungenauigkeit des Euler-Cauchy-Verfahrens weiter zu verringern. Das Verfahren wird daher in folgenden zwei Schritten durchgeführt.

$$\begin{aligned} \hat{\mathbf{x}}(t_{i+\frac{1}{2}}) &= \hat{\mathbf{x}}(t_i) + \frac{h}{2}\mathbf{f}(\hat{\mathbf{x}}(t_i), \mathbf{u}(t_i)) \\ \hat{\mathbf{x}}(t_{i+1}) &= \hat{\mathbf{x}}(t_i) + h\mathbf{f}(\hat{\mathbf{x}}(t_{i+\frac{1}{2}}), \mathbf{u}(t_{i+\frac{1}{2}})) \end{aligned}$$

- Verfahren von Heun

Zur Verbesserung der Ungenauigkeit macht das Verfahren von Heun den Ansatz die Integralrechnung durch ein Trapez zu approximieren. Durch diese Annahme wird die Rekursionsgleichung (2.6) implizit definiert. Es wird daher in einem ersten Schritt das einfache Euler-Cauchy-Verfahren verwendet, um einen Zwischenvektor $\tilde{\mathbf{x}}(t_{i+1})$ zu ermitteln. Die damit verbundene Ungenauigkeit wird erst danach durch die Formel eines Trapezes verringert. Daraus ergeben sich die folgenden zwei Schritte:

$$\begin{aligned}\tilde{\mathbf{x}}(t_{i+1}) &= \mathbf{x}_i + h\mathbf{f}(\hat{\mathbf{x}}(t_i), \mathbf{u}(t_i)) \\ \hat{\mathbf{x}}(t_{i+1}) &= \hat{\mathbf{x}}(t_i) + \frac{h}{2}(\mathbf{f}(\hat{\mathbf{x}}(t_i), \mathbf{u}(t_i)) + \mathbf{f}(\tilde{\mathbf{x}}(t_{i+1}), \mathbf{u}(t_{i+1})))\end{aligned}$$

- Verfahren von Simpson

In diesem Verfahren wird ausgehend von einem polynomialen Ansatz die Integralrechnung mit besserer Genauigkeit approximiert. Folgende Schritte werden durchgeführt:

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{f}(\hat{\mathbf{x}}_i, \mathbf{u}_i) \\ \mathbf{k}_2 &= \mathbf{f}(\hat{\mathbf{x}}_i + \frac{h}{2}\mathbf{k}_1, \mathbf{u}_{i+1/2}) \\ \mathbf{k}_3 &= \mathbf{f}(\hat{\mathbf{x}}_i - h\mathbf{k}_1 + 2h\mathbf{k}_2, \mathbf{u}_{i+1}) \\ \hat{\mathbf{x}}_{i+1} &= \hat{\mathbf{x}}_i + \frac{h}{6}(\mathbf{k}_1 + 4\mathbf{k}_2 + \mathbf{k}_3)\end{aligned}$$

- Runge-Kutta-verfahren(RKV)

Genau wie bei allen anderen Einschrittverfahren geht es hier darum die Fläche F in Abbildung 2.1 genau zu schätzen. Dieses Verfahren versucht nicht nur eine bessere Schätzung von F zu erreichen, sondern stellt auch eine allgemeine Formulierung der Einschrittverfahren zur Lösung nichtlinearer Differentialgleichungen erster Ordnung dar. Die Durchführung erfolgt in $p + 1$ Schritte, die zusammengefasst in Gleichung (2.7) beschrieben sind.

$$\begin{aligned}\hat{\mathbf{x}}(t_{i+1}) &= \hat{\mathbf{x}}(t_i) + h \sum_{j=1}^p \gamma_j \mathbf{k}_j \\ \mathbf{k}_j &= \mathbf{f}(\hat{\mathbf{x}}(t_i) + h \sum_{l=1}^{j-1} \alpha_{j,l} \mathbf{k}_l, \mathbf{u}(t_i + \beta_j h))\end{aligned} \tag{2.7}$$

Für den Fall $\alpha_{j,l} = 0$ wird die Rekursionsgleichung explizit definiert. In den allgemeinen Fällen müssen die Parameter β_j und γ_j so ausgewählt werden, dass die Bedingung (2.8) gilt.

$$\sum_{j=1}^p \beta_j = 1 \quad \text{und} \quad \sum_{l=1}^{j-1} \alpha_{j,l} = \gamma_j \tag{2.8}$$

Am häufigsten wird in der Literatur das sogenannte Runge-Kutta-Verfahren vierte Ordnung eingesetzt, das eine klassische Form der RKV darstellt. Eine ausführliche Beschreibung dieses Verfahren bietet Algorithmus 1 an.

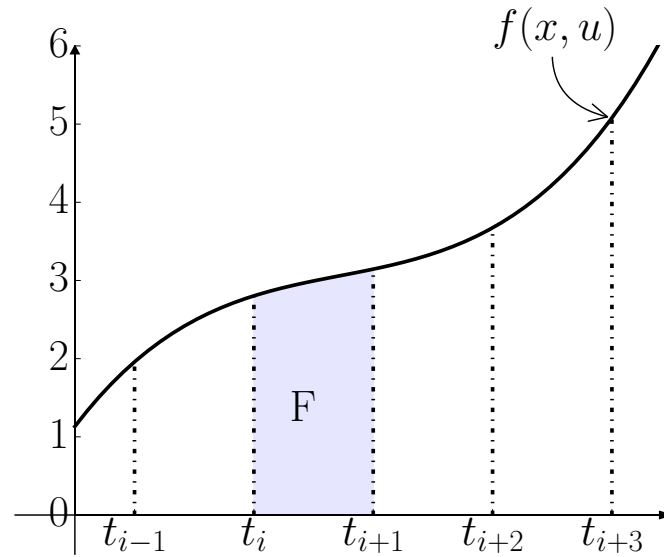


Abbildung 2.1.: Übersicht über die Einschrittverfahren, die markierte Fläche wird zum Zeitpunkt t_{i+1} approximiert

Aus Gründen der Übersichtlichkeit wurden teilweise die Zustandsgröße $\mathbf{x}(t_i)$ zum Zeitpunkt t_i durch ihre abgekürzte Form \mathbf{x}_i ersetzt. Neben den vorgestellten Einschrittverfahren existieren noch weitere Methoden zur Lösung nichtlinearer Differentialgleichungen. Für eine vollständige Beschreibung gängiger Methoden wird auf [16] verwiesen.

Das oben vorgestellte Thema wird heutzutage in der Forschung nicht mehr als alleinstehende Aufgabe gegeben, sondern in Kombination mit anderen Problemen behandelt. Das ist beispielsweise der Fall bei dieser Arbeit, in der die Dynamik eines Fahrzeugs in einer Optimierungsaufgabe als nichtlineare Dgl. berücksichtigt wird. In dem nächsten Abschnitt dieses Kapitels wird das Thema Optimierung besprochen, das ein zentrales Gebiet der angewandten Mathematik darstellt.

Algorithmus 1 RKV4

Eingabe: N (Anzahl Schritte), h (Schrittweite) und \mathbf{x}_0

Ausgabe: $\mathbf{x} = \mathbf{x}_N$

- 1: $i \leftarrow 0$
 - 2: wenn $i < N$ wiederholen
 - 3: $\mathbf{k}_1 \leftarrow \mathbf{f}(\hat{\mathbf{x}}_i, \mathbf{u}_i)$
 - 4: $\mathbf{k}_2 \leftarrow \mathbf{f}(\hat{\mathbf{x}}_i + \frac{h}{2}\mathbf{k}_1, \mathbf{u}_{i+1/2})$
 - 5: $\mathbf{k}_3 \leftarrow \mathbf{f}(\hat{\mathbf{x}}_i + \frac{h}{2}\mathbf{k}_2, \mathbf{u}_{i+1/2})$
 - 6: $\mathbf{k}_4 \leftarrow \mathbf{f}(\hat{\mathbf{x}}_i + h\mathbf{k}_3, \mathbf{u}_{i+1})$
 - 7: $\hat{\mathbf{x}}_{i+1} \leftarrow \hat{\mathbf{x}}_i + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$
 - 8: $i \leftarrow i + 1$
 - 9: ende wenn
-

2.2 Optimierungsaufgaben

Ein weiteres Thema das in dieser Arbeit behandelt wird, stellt das Optimierungsproblem dar. Es geht um ein sehr bekanntes Problem aus der Mathematik, das ein umfangreiches Gebiet der Forschung bildet. Unter dem Begriff Optimierung verstecken sich andere Begriffe wie Minimierung oder Maximierung, die durch ihre Bedeutungen schon das Ziel der Aufgabe definieren. Dabei werden Optimierungsvariablen ausgesucht, die eine Zielfunktion entweder minimieren oder maximieren. Es könnte gegebenenfalls Beschränkungen für die Optimierungsvariablen geben. Der Begriff Maximierung wird aber vernachlässigt, weil die Maximierung einer Gütemaße und die Minimierung ihrer negativen Form gleich bedeutend sind. Die Auffindung des Minimums(bzw. Maximums) einer Polynomfunktion der Ordnung n stellt das einfachste Beispiel einer Optimierungsaufgabe dar. Bei der Lösung dieser Aufgabe wird die erste Ableitung des Polynoms zu Null gesetzt. Daraus lassen sich alle Nullstellen bestimmen, die danach durch weitere Untersuchungen entweder als Minimum oder Maximum berücksichtigt werden. Das kann beispielsweise sehr hilfreich sein, um sich dem Verlauf einer Funktion anzunähern. Unterschiedliche Ausgangssituationen sind bei der Optimierung zu erwarten:

- Die Optimierungsaufgabe besitzt keine Lösung:

Das Minimum oder Maximum kann in vielen Fällen nicht gefunden werden. Dies kann beispielsweise daran liegen, dass keine gefundene Lösung im zulässigen Lösungsbereich liegt(wegen Beschränkungen) oder die Zielfunktion kein Optimum zeigt.

- Die Lösung ist lediglich in einem lokalen Bereich gültig:

Bei der Auffindung des Minimums einer Polynomfunktion kommt es häufig vor, dass mehrere Lösungen gefunden werden, die zunächst nur die lokalen Optima darstellen. Ausgehend davon, dass alle Lösungen gefunden wurden und dass der Lösungsraum begrenzt ist, müssen die einzelnen Lösungen weiter verglichen werden, um das globale Optimum zu finden. Diese Situation ist zu erwarten, wenn die Zielfunktion nicht quadratisch ist. Abbildung 2.2a veranschaulicht den Verlauf solcher Zielfunktionen, wobei zwei lokale Optima jeweils an der Stelle t_1^* und t_2^* liegen. Es ist offensichtlich, dass t_1^* lediglich in einem kleinen Bereich des globalen Lösungsraum als optimal gelten kann. An dieser Stelle wird es daher von lokalem Optimum gesprochen. Im Gegenteil dazu wird t_2^* überall optimal sein. Aus diesem Grund wird t_2^* als globales Optimum gezeichnet. Es muss beachtet werden, dass die Sattelpunkte der Zielfunktion sich als lokale Optima verhalten.

- Die Optimierungsaufgabe besitzt eine einzige Lösung:

Wie schon festgestellt wurde, hat die Zielfunktion eine entscheidende Bedeutung in der Optimierung. Es gibt beispielsweise die sogenannte quadratische Zielfunktionen, die durch ihre konvexe Verläufe ein einziges Optimum im Raum haben. Diese Funktionstypen werden häufig verwendet, denn die Bestimmung eines globalen Optimums wird dadurch garantiert. In Abbildung 2.2b ist der Verlauf einer quadratische Funktion zu sehen.

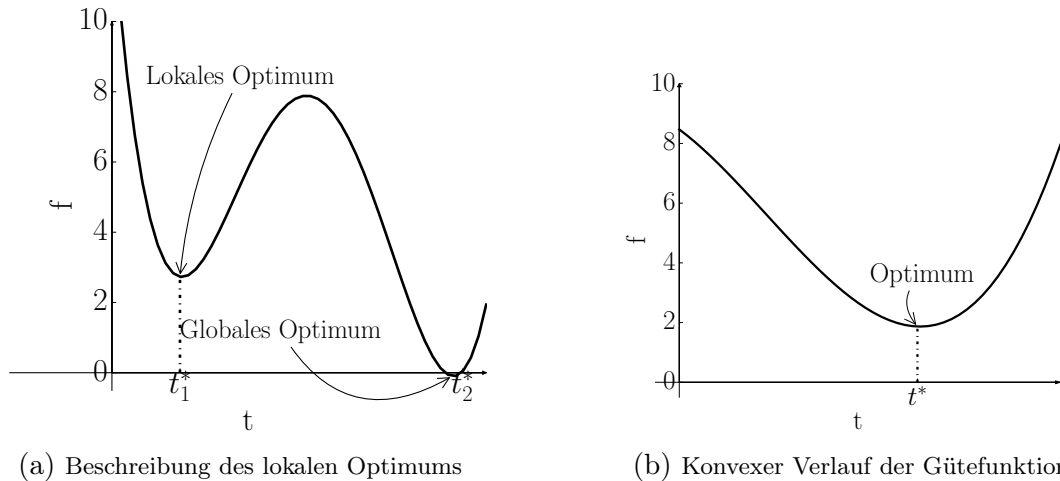


Abbildung 2.2.: Übersicht über mögliche Verläufe der Zielfunktion in eindimensionalem Fall

Aus der oben vorgestellten Ausgangssituationen lässt sich deutlich die lokale von der globalen Optimierung unterscheiden, sowie die konvexe von der nicht konvexen Optimierung. Sie sind außerdem mit unterschiedlichen Schwierigkeitsgraden(bzw. Konvergenzgrad) verbunden.

Bei komplexen Aufgaben kann die Optimierung nicht mehr analytisch gelöst werden. Es wurden daher abhängig von der Struktur der Optimierungsaufgabe numerische Verfahren entwickelt, die zur Lösung eingesetzt werden. Sowohl in der Wirtschaft als auch in der Industrie tauchen Optimierungsprobleme unterschiedlicher Art auf. Um einen groben Unterschied machen zu können, lassen sich die Optimierungsprobleme in folgenden zwei Hauptgruppen aufteilen[17].

- Statische Optimierungsprobleme:
Minimierung einer Funktion mit Optimierungsvariablen, die Elemente des euklidischen Raums sind.
- Dynamische Optimierungsprobleme:
Minimierung eines Funktional, bei dem die Optimierungsvariablen Elemente des Hilbert-Raumes sind(z. B. Zeitfunktionen)

Es wird wie in den vorangehenden Abschnitten anhand allgemeiner nichtlinearen Systeme dieses Thema weiter betrachtet. Jedoch wird lediglich der Teil der Optimierung näher besprochen, der für diese Arbeit relevant ist.

2.2.1 Statische Optimierungsprobleme

Die statische Optimierung hat die allgemeine mathematische Formulierung (2.9).

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad \text{Kostenfunktion} \quad (2.9a)$$

$$\text{u.B. } g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \quad \text{Gleichungsbeschränkungen} \quad (2.9b)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \quad \text{Ungleichungsbeschränkungen} \quad (2.9c)$$

Dabei wird die Zielfunktion (bzw. Kostenfunktion oder Gütemaße) $f(\mathbf{x})$ minimiert. Anders gesagt, werden Optimierungsvariablen $\mathbf{x}^* \in \mathbb{R}^n$ ausgesucht, die der kleinste Wert der Kostenfunktion unter Einhaltung von Nebenbedingungen (2.9b) und (2.9c) ergeben. Die Zielfunktion bestimmt daher, in welchem Bereich des Raums die optimalen Parameter eines Systems liegen. Diese Parameter können danach verwendet werden, um das System in einen gewünschten Zustand zu überführen. Die Gleichungs- und Ungleichungsbeschränkungen beschreiben dabei alle möglichen Restriktionen, die bei der Zustandsänderung eingehalten werden müssen. Man kann sich dabei einen Roboter vorstellen, der so schnell wie möglich einen Weg zurücklegen und gleichzeitig die im Weg stehenden Hindernisse umfahren muss. Die Zielfunktion könnte eine zeitoptimale Größe und die Restriktionen durch Hindernisse dargestellt sein. Die Größen \mathbf{x} , $f(\mathbf{x})$, $g_i(\mathbf{x})$ und $h_i(\mathbf{x})$ sind somit von besonderer Bedeutung. Zudem wird die statische Optimierung abhängig von der Struktur dieser Größen weiter unterteilen, wie folgt[17]:

- Lineare Programmierung:
Die Kostenfunktion und Nebenbedingungen sind linear.
- Quadratische Programmierung:
Die Kostenfunktion ist quadratisch, während die Nebenbedingungen linear sind.
- Nichtlineare Programmierung:
Die Kostenfunktion oder mindestens eine Beschränkung ist nichtlinear.
- Integer-Programmierung:
Alle Optimierungsvariablen sind diskret.
- Mixed-Integer-Programmierung:
Kontinuierliche und diskrete Optimierungsvariablen treten auf.

Die statische Optimierung wird im euklidischen Raum durchgeführt. Das bedeutet, dass der Lösungsraum zeitunabhängig definiert wird. Man kann beispielsweise zur Ermittlung eines optimalen Verstärkungsfaktors in der Regelungstechnik auf diese Art der Optimierung zurückgreifen[5]. Die statische Optimierung ist zwar in der Regelungstheorie einsetzbar, aber generell wird darauf bauend eine dynamische Formulierung gebildet, die für diese Arbeit am geeignetsten ist.

2.2.2 Optimalsteuerungsprobleme

Der Vollständigkeit halber wird hier noch die allgemeine Formulierung eines nichtlinearen Optimierungsproblems wiederholt und in der folgenden Form geschrieben.

$$\min_{\mathbf{x} \in \mathbb{R}^n} \phi(\mathbf{x}) \quad \text{Kostenfunktion} \quad (2.10a)$$

$$\text{u.B. } \mathbf{g}(\mathbf{x}) = 0 \quad \text{Gleichungsbeschränkungen} \quad (2.10b)$$

$$\mathbf{h}(\mathbf{x}) \leq 0 \quad \text{Ungleichungsbeschränkungen} \quad (2.10c)$$

Wobei mindestens eine von den Größen

$$\phi(\mathbf{x}), \quad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1 \\ \vdots \\ g_p \end{bmatrix} \quad \text{und} \quad \mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1 \\ \vdots \\ h_q \end{bmatrix} \quad (2.11)$$

nichtlinear ist und \mathbf{x} die Optimierungsvariablen darstellt. Die Kostenfunktion, Gleichungs- und Ungleichungsbeschränkungen haben, wie zuvor, die gleiche Bedeutung

Wir verfolgen nun das Ziel zeitabhängige Variablen, frei wählbare Steuerfunktionen und einen zusätzlichen Parameter a in die Optimierungsgrößen einzuführen. Dies lässt sich dadurch begründen, dass viele Prozesse (oder zu mindestens das System welches später betrachtet wird) die allgemeine Form (2.3) besitzen. Unter der Annahme, dass die Optimierung in einer Zeitspanne T stattfindet, in der das System von $\mathbf{x}(t_0)$ nach $\mathbf{x}(t_f)$ überführt wird und ausgehend von den in (2.1) und (2.2) definierten Vektoren $\mathbf{x}(t)$ und $\mathbf{u}(t)$ lassen sich die Nebenbedingungen (2.10b) und (2.10c) in Form von Randwerten (2.13) und (2.14) schreiben. Die Systemdynamik mit zusätzlichem Parameter a hat die Form (2.12).

$$\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), a) = 0 \quad (2.12)$$

$$\mathbf{g}(\mathbf{x}(t_0), a) = 0 \quad (2.13)$$

$$\mathbf{h}(\mathbf{x}(t_f), a) = 0 \quad (2.14)$$

Bei Prozesse oder Systeme, deren Verläufe in einem zulässigen Bereich liegen müssen, kann noch hinsichtlich der Trajektorienplanung eine Pfadbeschränkung der Form (2.15) in die Nebenbedingungen eingefügt werden.

$$\mathbf{r}(\mathbf{x}(t), \mathbf{u}(t), a) \leq 0 \quad (2.15)$$

Somit lässt sich die generelle Struktur eines dynamischen Optimierungsproblems für eine Zeitspanne $T = [t_0, t_f]$ in der Form (2.16) zusammenfassen.

$$\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, a \in \mathbb{R}^{n_a}} \phi(\mathbf{x}(t), \mathbf{u}(t), a) \quad (2.16a)$$

$$\text{u.B.} \quad \dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), a) = 0 \quad (2.16b)$$

$$\mathbf{g}(\mathbf{x}(t_0), a) = 0 \quad (2.16c)$$

$$\mathbf{h}(\mathbf{x}(t_f), a) = 0 \quad (2.16d)$$

$$\mathbf{r}(\mathbf{x}(t), \mathbf{u}(t), a) \leq 0 \quad (2.16e)$$

Die Anfangs- und Endbedingungen (2.16c) und (2.16d) sorgen dafür, dass der Anfangszustand und Endzustand des Systems vorgebar bleiben. Generell wird die Anfangsbedingung in der Form (2.17) beschrieben.

$$\underbrace{\mathbf{x}(t_0) - \mathbf{x}_0}_g = 0 \quad \implies \quad \mathbf{g}(\mathbf{x}(t_0), a) = 0 \quad (2.17)$$

Die Steuer- und Zustandsbeschränkungen werden durch Gleichung (2.16e) gewährleistet. Der Parameter a wird in der Optimierung entweder als Konstante auftreten oder freigegeben. Er wird daher gleich in die Steuerfunktion $\mathbf{u}(t)$ aufgenommen und nicht mehr explizit dargestellt [18]. Laut [17] lässt sich die Kostenfunktion $\phi(\cdot)$ in Meyer-Term (2.18a), der nur zum Endzeitpunkt bewertet wird und Lagrange-Term (2.18b) aufspalten. Sie können auch beide kombiniert sein und unter der Bezeichnung Bolza-Form genannt werden, wie Gleichung (2.18c) es darstellt.

$$\phi_M = E(t_f, \mathbf{x}(t_f)) \quad \text{Meyer-Term} \quad (2.18a)$$

$$\phi_L = \int_{t_0}^{t_f} L(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad \text{Lagrange-Term} \quad (2.18b)$$

$$\phi(\mathbf{x}(t), \mathbf{u}(t)) = \phi_M + \phi_L \quad \text{Bolza-Form} \quad (2.18c)$$

Abhängig von dem Ziel der Aufgabe sind einige Kriterien bei der Auswahl der Kostenfunktion zu berücksichtigen und somit ergeben sich unterschiedliche Darstellungen. Nachfolgend sind einige davon aufgelistet, die in der Regelungstheorie häufig verwendet werden [5]:

- Energie- bzw. verbrauchsoptimale Kostenfunktionen
- Kostenfunktionen mit Gewichtung von quadratischen Regelflächen
- Verlaufoptimale Kostenfunktionen
- Zeitoptimale Kostenfunktionen

Die verlaufoptimale Kostenfunktionen werden beispielsweise verwendet, um ein System auf einen gewünschten Weg zu halten. Die einzelnen Kostenfunktionen können auch kombiniert verwendet werden. Zusammengefasst geht es in der dynamischen Optimierung (2.16) darum, eine optimale Steuerfunktion $\mathbf{u}^*(t)$ so zu bestimmen, dass ein gewünschter Verlauf des Zustandsvektors $\mathbf{x}(t)$ erzeugt wird. Aus diesen Grund wird diese Aufgabe auch als Optimalsteuerungsproblem genannt (engl. optimal control problem, OCP). Abbildung 2.3 zeigt eine Übersicht über die Optimalsteuerung. Dabei sind die optimalen Steuerfunktionen $u_1(t)$, $u_2(t)$ und $u_3(t)$ als unstetige Funktionen dargestellt. Der Verlauf von $\mathbf{x}(t)$ und die Beschränkungen sind auch zu sehen.

Zur Lösung von Optimalsteuerungsproblemen werden die Lösungswege (a) dynamische Programmierung, (b) indirekte Methoden und (c) direkte Methoden verwendet [17, 10, 5], die nachfolgenden grob beschrieben werden.

- (a) Die dynamische Programmierung verwendet das sogenannte Optimalitätsprinzip von Bellman, das besagt, dass eine optimale Lösung aus optimalen Teillösungen besteht, um die Optimierungsaufgabe zu lösen. Die Zeitspanne wird in kleine Teilstücke aufgeteilt, sowie die vorliegenden kontinuierlichen Systemgrößen. Somit besteht das Problem nun aus gleichartigen Teilproblemen, deren optimale Steuergrößen rekursiv gesucht werden. Zur Erläuterung des zu Grunde liegenden Algorithmus nehmen wir ein diskretisiertes System an, das die Systemdynamik (2.19) hat.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \quad k = 0, \dots, N-1 \quad (2.19)$$

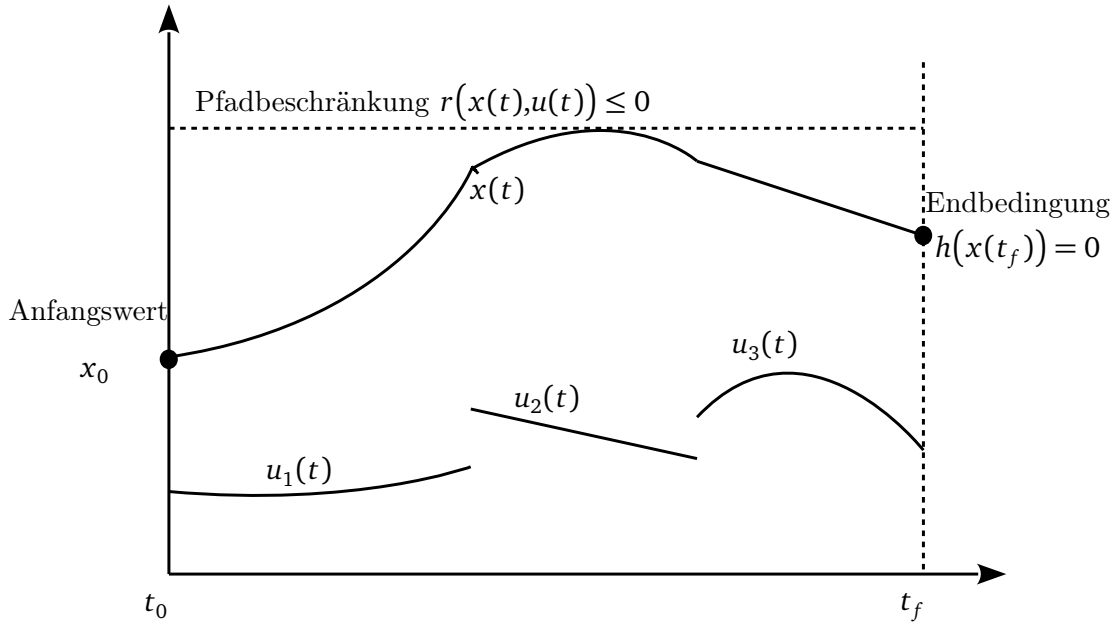


Abbildung 2.3.: Illustrierung der Optimalsteuerungsproblem, In Anlehnung an [10]

Zu jedem festen Zeitpunkt t_k steht eine endliche Menge von Steuerfunktionen zur Auswahl, wobei nur eine bestimmte Anzahl \mathbf{u}_k davon ausgewählt werden soll. Diese Auswahl führt dann zu dem zulässigen(bzw. gewünschten) Zustandsvektor \mathbf{x}_k . Das Ziel der Aufgabe besteht nun darin so fortzufahren, dass die N diskreten und optimalen Steuerfunktionen

$$\pi^* = \{\mathbf{u}_0^*, \dots, \mathbf{u}_{N-1}^*\} \quad (2.20)$$

ausgewählt werden und eine diskrete Kostenfunktion der Form (2.21) optimiert wird[18]. Damit sind die Kosten gemeint, die benötigt werden, um das System von einem festen(bzw. bekannten) Zustand \mathbf{x}_0 bis zu einem Endzustand \mathbf{x}_N zu überführen.

$$J_\pi(\mathbf{x}_0) = g_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} g_k(\mathbf{x}_k, \mathbf{u}_k) \quad (2.21)$$

Die daraus erhaltene Optimierungsaufgabe lässt sich in Gleichung (2.22) formulieren. Wo bei Π die Menge aller möglichen Steuerfunktionen darstellt.

$$J_{\pi^*}(\mathbf{x}_0) = \min_{\pi \in \Pi} J_\pi(\mathbf{x}_0) \quad (2.22)$$

Zur Lösung der Aufgabe wird ausgehend von $k = N - 1$ bis $k = 0$ die Fortsetzungskosten(engl. cost-to-go) berechnet und abgespeichert[18]. Diese Vorgehensweise vermeidet die Untersuchung aller Varianten der Lösung.

Der große Nachteil dieses Verfahren liegt daran, dass der Bedarf an Speicherplatz ziemlich hoch ist. Schon für Probleme mit kleiner Anzahl von Zuständen stellt sich diese Problematik dar. Aus diesen Grund findet dieses Verfahren weniger Anwendung in der Regelungstheorie [5].

- (b) Indirekte Methoden verwendet die Variationsrechnung um die Optimierungsaufgabe in ein Randwertproblem(RWP) zurückzuschließen, das mit bekannten Verfahren numerisch oder analytisch gelöst wird. Diese Verfahren liefern hochgenaue Lösungen, jedoch wird die Behandlung von Nebenbedingungen (bzw. Zustand- und Steuergrößenbeschränkung) eher mit großer Schwierigkeit verbunden[5]. Eine detaillierte Erklärung des Verfahrens ist in [17] erhältlich.
- (c) Direkte Methoden beruhen auf der Parametrisierung des Optimalsteuerungsproblems. Dadurch ergibt sich ein endlich dimensionales Problem. Dieses Problem lässt sich gut durch bekannte Verfahren der statischen Optimierung lösen. Bei der Parametrisierung des Optimalsteuerungsproblems werden drei Verfahren unterschieden, die für die Umsetzung von Modellprädiktiven Regelung verwendet werden[5].

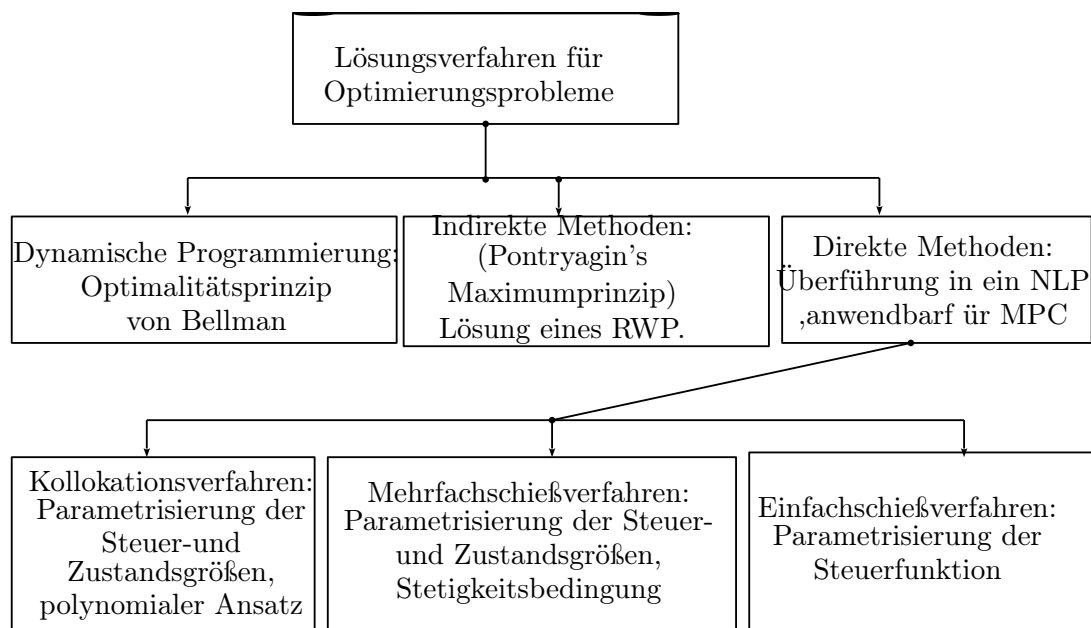


Abbildung 2.4.: Übersicht über die Lösungsverfahren der Optimalsteuerungsprobleme, in Anlehnung an [5, 10].

Die vorliegenden Beschreibung einzelner Verfahren zeigen Vorteile und Nachteile, die nach Verwendungszweck berücksichtigt werden müssen. Wie es in Abbildung 2.4 zusammenfassend zu sehen ist, werden die direkten Methoden weiter betrachtet. Im Abschnitt 2.3 wird das Thema Modellprädiktive Regelung erläutert, sowie die dabei anwendbaren direkten Einfachschießverfahren, Mehrzielverfahren und Kollokationsverfahren.

2.3 Grundlagen der Modellprädiktiven Regelung

Modellbasierte prädiktive Regelungen(MPR), auch kurz Modellprädiktive Regelungen, gehören zu den fortgeschrittenen Regelungsverfahren, die am häufigsten in der Industrie sowohl für lineare als auch für nichtlineare Prozesse eingesetzt werden[16]. Obwohl viele nichtlineare Prozesse an bestimmten Stellen des Zustandsraums linearisiert werden können und von linearen Reglern

geregelt werden, hat es sich gezeigt, dass kein sehr gutes Regelverhalten erreicht wird, wenn Beschränkungen von Stell- und Zustandsgrößen unterliegen. Durch die in dem Bereich Optimierung gemachten Fortschritte eignet sich MPR nicht nur für die Behandlung von Beschränkungen in Systemen, sondern auch für die Regelung allgemein nichtlinearer Prozesse, die vor allem kontinuierlich und ununterbrochen laufen müssen. Beispielsweise sind Heizungs- und Klimaanlage, die eine konstante Raumtemperatur tagsüber halten sollen, oder Herstellungsprozesse in der chemischen Verfahrenstechnik, mit MPR geregelt[18].

Mit der Bezeichnung Modellbasierte Prädiktive Regelung sind zwei Aufgaben aus der Regelungstheorie abgedeckt. Zum einen wird ein Prozess geregelt und zum anderen wird das Verhalten des Prozesses in einer nahen Zukunft beschrieben. Der Unterschied zu anderen prädiktiven Regelungen besteht darin, dass man in der MPR die Eigenschaften mathematischer Prozessmodelle verwendet, um sie während der Ausführung zu regeln und nicht einmalig das zukünftige Verhalten des Prozessmodells für einen Zeithorizont optimiert. Jedoch bleibt der Grundgedanke erhalten.

Es wird darauf hingewiesen, dass Aufgrund ihrer einfachen Struktur diskrete(bzw. diskretisierte) Systeme besser bei dem Verfahren behandelt werden als kontinuierliche Systeme, die eine komplexere Optimierungsaufgabe darstellen[16]. Außerdem unterscheiden sich lineare von nichtlinearen MPR. Wenn das zugrunde liegende Prozessmodell die lineare Gestalt (2.23) hat, spricht man von linearer MPR(engl. Model Predictive Control, MPC).

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \tag{2.23}$$

In dem Fall nichtlinearer Systeme der Form (2.3) spricht man von nichtlinearer MPR(engl. Nonlinear Model Predictive Control). Das Thema nichtlineare Modellprädiktive Regelung ist Gegenstand dieses Abschnitts und wird im Folgenden beschrieben. Da keine Verwechslungsgefahr mehr besteht, wird im weiteren Verlauf das Thema verallgemeinert und lediglich die Abkürzungen MPC für Modellprädiktive Regelung und OCP für Optimalsteuerungsproblem verwendet, um die Literatur möglich nah abzubilden.

2.3.1 Funktionsweise der MPC

Zur Beschreibung dieses Verfahren sei ein vereinfachtes Optimalsteuerungsproblem in der zeitkontinuierlichen Form (2.24) gegeben. Wobei Gleichungen (2.24b) und (2.24c) die Dynamik und der Anfangszustand eines nichtlinearen Systems darstellen. Die anderen Gleichungen be-

schreiben die Beschränkungen des Systems und die zu optimierende Zielfunktion (bestehend aus Meyer-Term und Lagrange-Term).

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_{t_0}^{t_0+T} L(\mathbf{x}(t), \mathbf{u}(t)) dt + E(\mathbf{x}(t_0 + T)) \quad (2.24a)$$

$$\text{u.B. } \dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = 0 \quad t \in [t_0, t_0 + T] \quad (2.24b)$$

$$\mathbf{x}(t_0) - \mathbf{x}_0 = 0 \quad (2.24c)$$

$$\mathbf{h}(\mathbf{x}(t_0 + T)) = 0 \quad (2.24d)$$

$$\mathbf{r}(\mathbf{x}(t), \mathbf{u}(t)) \leq 0 \quad t \in [t_0, t_0 + T] \quad (2.24e)$$

In der nichtlinearen Modellprädiktive Regelung wird das vorgegebene System (Systemdynamik (2.24b)) zyklisch geregelt, indem das oben dargestellte OCP für jede Zeitperiode T gelöst wird. In dieser Aufgabe wird eine optimale Steuergrößenfolge \mathbf{u}^* bestimmt. Das lässt sich noch in Abbildung 2.3 veranschaulichen. Vorausgesetzt, dass der Zeitaufwand zur Bestimmung einer optimalen Lösung (bzw. zur Konvergenz des Verfahrens) gering genug ist, wird das erste Element dieser Steuergrößenfolge auf die reale Strecke gegeben. Dadurch wird sich der Systemzustand ändern. Der neue Systemzustand wird ermittelt oder erfasst. Dieser zuletzt ermittelte Systemzustand (bzw. Ausgangsgröße) wird als Anfangszustand bei der Lösung eines erneut formulierten OCPs verwendet, wobei auch die Zeitspanne T um einen Zeitschritt ΔT nach vorne (bzw. in die Zukunft) verschoben wurde. T beschreibt die Länge des Horizonts, d. h. wie weit in die Zukunft geschaut werden muss (Dauer der Prädiktion) und wird aus diesem Grund als Zeithorizont genannt. Da zwischen zwei Schritten der Regelung T nach vorne verschoben wird, spricht man auch von einem gleitenden Horizont (engl. moving horizon) [16]. Abbildung 2.5 veranschaulicht die Vorgehensweise, wobei t_k der aktuelle Zeitpunkt darstellt. Die Fortsetzung des Verfahrens besteht aus endlicher Wiederholung der oben vorgestellten Schritte und dadurch entsteht ein geschlossener Regelkreis, wie es in der Regelungstechnik bekannt ist. Es stellt sich aber die Frage, warum nur das erste Element der Steuergrößenfolge verwendet wird und alle andere nicht berücksichtigt werden. Um diese Frage zu beantworten, soll man sich den Zustand der Umgebung eines Systems als dynamisch veränderbar vorstellen. Dies ist beispielsweise der Fall für mobile Roboter oder autonome Fahrzeuge, die durch aufgebaute Sensoren die Änderung ihres Umfelds wahrnehmen müssen. Zur Ermittlung der optimalen Steuergrößenfolge \mathbf{u}^* wurde ein bestimmter Zustand des Umfeldes berücksichtigt, der sich unmittelbar danach ändern kann. Außerdem gibt es keine Garantie, dass das System sich wie geplant verhält (Störungen können das Verhalten des Systems ändern). Die Lösung des OCPs gibt eine klare Vorstellung darüber, wie sich das System in der nahen Zukunft verhalten wird, wenn das Umfeld unveränderbar bleiben würde, oder keine Störung das System beeinflussen würde. Es wird daher nur das erste Element von \mathbf{u}^* auf das System gegeben und auf neue Informationen aus dem Umfeld gewartet. Diese zuletzt erhaltenen Kenntnisse über das Umfeld sowie der neu erfasste Zustand des Systems ermöglichen die Formulierung eines neuen OCPs. Somit wird das System ständig aktualisiert.

Nachteil dieses Verfahrens war immer der Rechenaufwand, der mit Zunahme der Anzahl von zu

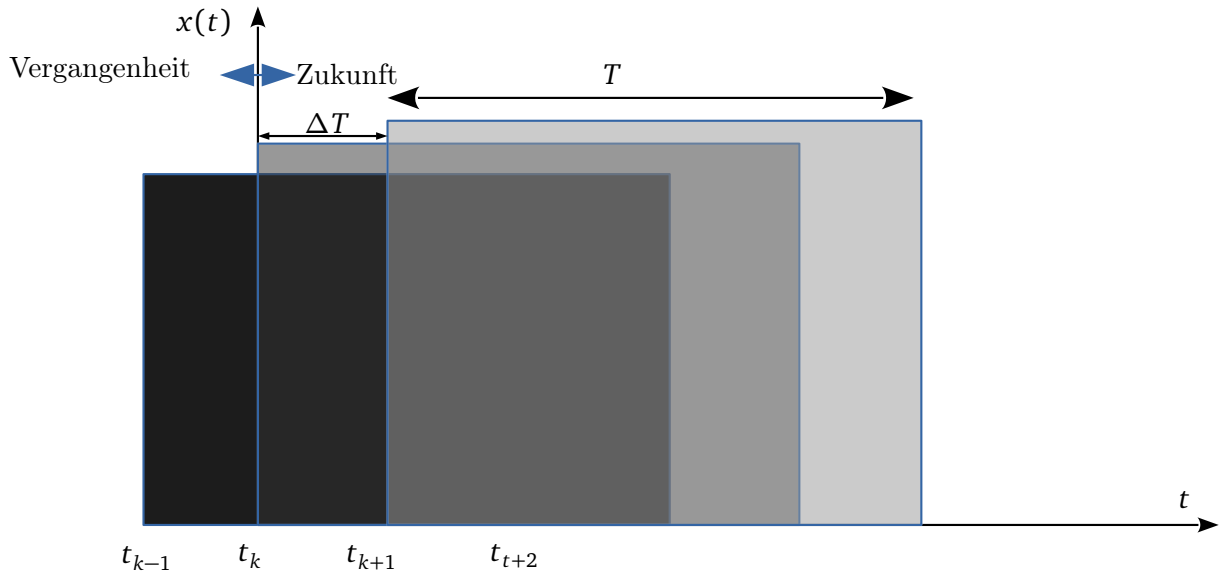


Abbildung 2.5.: Übersicht über den gleitenden Horizont. Es wird ein OCP zum Zeitpunkt t_k auf dem Horizont $[t_k, t_k + T]$ gelöst und der erste Teil von \mathbf{u}^* auf das reale System gegeben. Spätestens nach ΔT muss der neue Zustand des Systems erfasst, ein neues OCP auf dem Horizont $[t_{k+1}, t_{k+1} + T]$ formuliert und gelöst werden. Der erste Teil von \mathbf{u}^* wird verwendet, um das reale System zu steuern. Ein neues OCP wird wieder auf dem Horizont $[t_{k+2}, t_{k+2} + T]$ gelöst. Dieser Zyklus kann beliebig fortgesetzt werden, bis ein gewünschter Zustand erreicht wird.

behandelten Variablen schnell steigt. Dies hat zur Folge, dass die Lösung des OCPs viel Zeit kostet und die Echtzeitfähigkeit der MPC in Frage gestellt wird. Denn wenn die Rechenzeit eines OCPs zu lang ist, wird sich der erfasste Zustand des realen Systems in dieser Zeit deutlich verändert haben. Dies führt dazu, dass die zurückgelieferte Lösung \mathbf{u}^* nicht mehr unbedingt optimal wird.

In dem Forschungsgebiet nichtlineare Regelung wurde in den letzten Jahrzehnten intensiv über das Thema MPC geforscht. Dies hat dazu beigetragen, dass nichtlineare Systeme explizit behandelt werden können, sowie die damit verbundenen Beschränkungen. Die Entwicklung von Rechnern mit hohen Rechenleistungen bzw. Datenverarbeitungsgeschwindigkeit war ein entscheidender Punkt, der ermöglicht hat die Rechenzeit zu verringern. Durch alle diese Fortschritte wurde die Echtzeitfähigkeit der MPC sichergestellt.

Um ein in der kontinuierlichen Form (2.24) vorliegendes OCP zu lösen, werden unterschiedliche Verfahren der direkten Methoden verwendet. Dazu gehören insbesondere das Einfachschießverfahren (engl. direct simple shooting method), das Mehrfachschießverfahren (engl. direct multiple shooting method) und das Kollokationsverfahren (engl. direct collocation), die am häufigsten eingesetzt werden. Der Grundgedanke dieser Verfahren besteht darin, das Problem mittels Parametrisierung in ein nichtlineares Programm (engl. nonlinear programming problem, NLP) zurückzuführen, um existierende numerische Lösungsverfahren zur Lösung anwenden zu können [10]. Alle diese Verfahren haben daher gemeinsam die Tatsache, dass eine Diskretisierung

der Steuerfunktionen zunächst durchgeführt wird. Sie unterscheiden sich jedoch in der Art, wie die Zustände behandelt werden.

Bei direkten Einfachschießverfahren werden lediglich die diskreten Steuergrößen als Variablen der Optimierungsaufgabe berücksichtigt. Zur Ermittlung der Zustände wird abhängig von der optimalen Steuergröße in jedem Zeitintervall die Dynamik des Systems (nichtlineare Dgl.) gelöst, wie es im Abschnitt 2.1.2 erläutert wurde. Dieser zuletzt durchgeführte Schritt wird in der Praxis als Simulation bezeichnet. Somit werden die Optimierung und die Simulation fortlaufend oder hintereinander (engl. sequential) erfolgen. In Abbildung 2.6 wird die Vorgehensweise graphisch illustriert. Bei direkten Kollokationsverfahren werden sowohl die Steuer- als auch

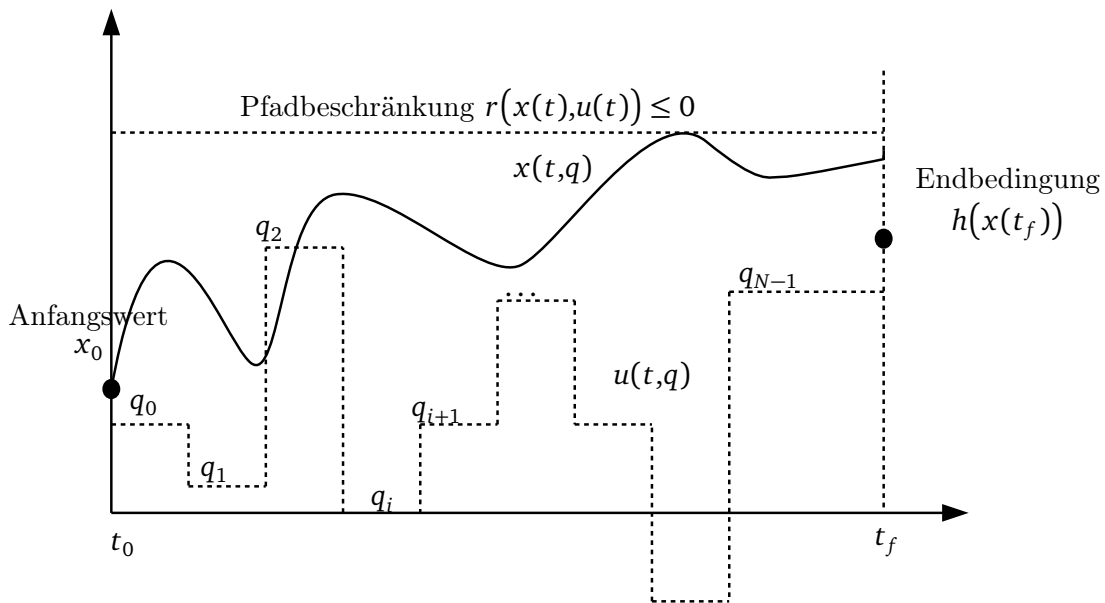


Abbildung 2.6.: Übersicht über das Einfachschießverfahren mit kontinuierlichem Verlauf der Zustände. Die Steuergrößen $\mathbf{u}(t, \mathbf{q})$ werden stückweise konstant angenommen und abhängig von \mathbf{q} parametrisiert. In Anlehnung an [10]

die Zustandsgrößen parametrisiert. Alle diese diskreten Größen werden als Variablen der Optimierung verwendet. Der Verlauf von Zuständen wird in jedem Zeitintervall durch Polynome geeigneter Ordnung angenähert, die an den sogenannten Kollokationspunkten die Dynamik des System einhalten. Es wird somit sehr viele Variablen geben. Die Optimierung und die Simulation erfolgen gleichzeitig.

Die ersten beiden Verfahren zeigen Vorteile und Nachteile, wie z. B. die Anzahl von Variablen oder die Behandlung des Anfangs- und Endzustands. Das direkte Mehrschießverfahren nutzt die Stärken beider Methoden um eine bessere Lösung des OCPs zu erhalten. Wir werden daher nur das letzte Verfahren in dieser Arbeit tiefer erklären.

2.3.2 Mehrschießverfahren

Das Mehrschießverfahren, auch Mehrzielmethode genannt, ist ein numerisches Lösungsverfahren aus der direkten Methoden, das zur Lösung von in der Form (2.16) vorliegenden Optimalsteuerungsproblemen verwendet wird. Der Ausgangspunkt ist wie bei allen direkten Methoden die Zerlegung des Lösungsraums in Teilbereiche. Dadurch werden endlich viele Knotenvariablen erzeugt und eine Lösung wird lediglich an diesen Stützstellen gesucht. Ausgehend von $N + 1$ Stützpunkten $0 = \tau_0 < \tau_1 < \dots < \tau_N = 1$ lässt sich der Zeithorizont T in N Teilintervalle der Form (2.25) unterteilen, wobei $t_i = T\tau_i$, $i = 0, \dots, N$ gesetzt wird.

$$0 = t_0 < t_1 < \dots, t_i, \dots < t_N = T \quad (2.25)$$

Das daraus erhaltene Zeitgitter kann sowohl zur Parametrisierung der Steuergrößen als auch der Zustände verwendet werden. Die Länge eines Intervalls soll vor allem keine besondere Vorschrift einhalten. Das bedeutet, dass es nicht vorgeschrieben wird, wie weit ein Stützpunkt von dem Nächsten liegen soll. Da es aber um Zeit geht, muss die Reihenfolge eingehalten werden.

Die Gleichungs- und Ungleichungsbeschränkungen müssen auch parametrisiert werden. Daraus ergeben sich endlich viele Nebenbedingungen, die nur noch an den $N + 1$ Stützpunkten eingehalten werden. Wenn das Zeitgitter jedoch fein genug ausgelegt wird, kann die Lösung des OCPs keine Bedingung verletzen. Und auch wenn das vorkommen würde, wird die Verletzung sehr gering bleiben, so dass die Lösung dadurch nicht verfälscht wird [10, 18].

Zur Diskretisierung der Steuergrößen wird in jedem Teilintervall $I_i = [t_i, t_{i+1}]$ eine Funktion \mathbf{q}_i definiert, die die gleiche Dimension wie $\mathbf{u}(t)$ besitzt und ihr lokaler Verlauf gut beschreiben kann. In der Praxis wird generell Aufgrund der Vereinfachung entweder eine lineare oder eine konstante Funktion \mathbf{q}_i verwendet. In dem Raum wird daher $\mathbf{u}(t)$ eine stückweise stetige Funktion darstellen, die mit der Gleichung (2.26) beschrieben wird.

$$\begin{aligned} \mathbf{u}(t) = \mathbf{q}_i \quad & \text{für } t \in [t_i, t_{i+1}] \\ i \in \{0, 1, \dots, N-1\} \end{aligned} \quad (2.26)$$

Es ist offensichtlich, dass die Steuergrößen nun von den Parametern

$$\mathbf{q} = \{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_{N-1}\} \text{ mit } \mathbf{q}_i \in \mathbb{R}^m$$

abhängt. Wir werden daher die Notation $\mathbf{u}(t, \mathbf{q})$ für die Steuergröße verwenden, um diese Abhängigkeit darzustellen. Es wird darauf hingewiesen, dass der Zeithorizont konstant angehalten wird, andernfalls muss T in die Parametrisierung eingeführt werden [10].

Die Diskretisierung der Zustände kann, wie schon erwähnt, mit dem gleichen Zeitgitter erfolgen. Ausgehend von der im Teilintervall $[t_i, t_{i+1}]$ definierten Steuergrößen $\mathbf{u}(t) = \mathbf{q}_i$, lässt sich das

Anfangswertproblem(AWP) (2.27) in diesem Teilintervall aufstellen. Dabei wird ein künstlicher Anfangswert \mathbf{s}_i angenommen.

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{f}(\mathbf{x}_i(t), \mathbf{q}_i) & t \in [t_i, t_{i+1}] \\ \mathbf{x}(t_i) &= \mathbf{s}_i & \text{mit } \mathbf{s}_i \in \mathbb{R}^n\end{aligned}\tag{2.27}$$

Somit wird jedes Teilintervall $[t_i, t_{i+1}]$ unabhängig von den anderen behandelt. Durch die Lösung des AWP's ergeben sich N Funktionen $\mathbf{x}_i(t; \mathbf{s}_i, \mathbf{q}_i)$, die offensichtlich von den Parametern \mathbf{s}_i und \mathbf{q}_i abhängig sind. Eine Betrachtung des ganzen Lösungsraums wird zeigen, dass es möglicherweise Sprünge oder Ecken an den inneren Stützpunkten gibt. Um dieses Problem zu vermeiden wird die Stetigkeit durch Einführung von sogenannten Schießbedingungen versichert. Diese Bedingungen lassen sich in der Form (2.28) für alle Knoten schreiben. In Abbildung 2.7 sind die diskreten Größen dargestellt und dadurch ist auch die Vorgehensweise gut erklärt.

$$\mathbf{s}_{i+1} = \mathbf{x}_i(t_{i+1}; \mathbf{s}_i, \mathbf{q}_i), \quad \forall i \in \{0, 1, \dots, N-1\}\tag{2.28}$$

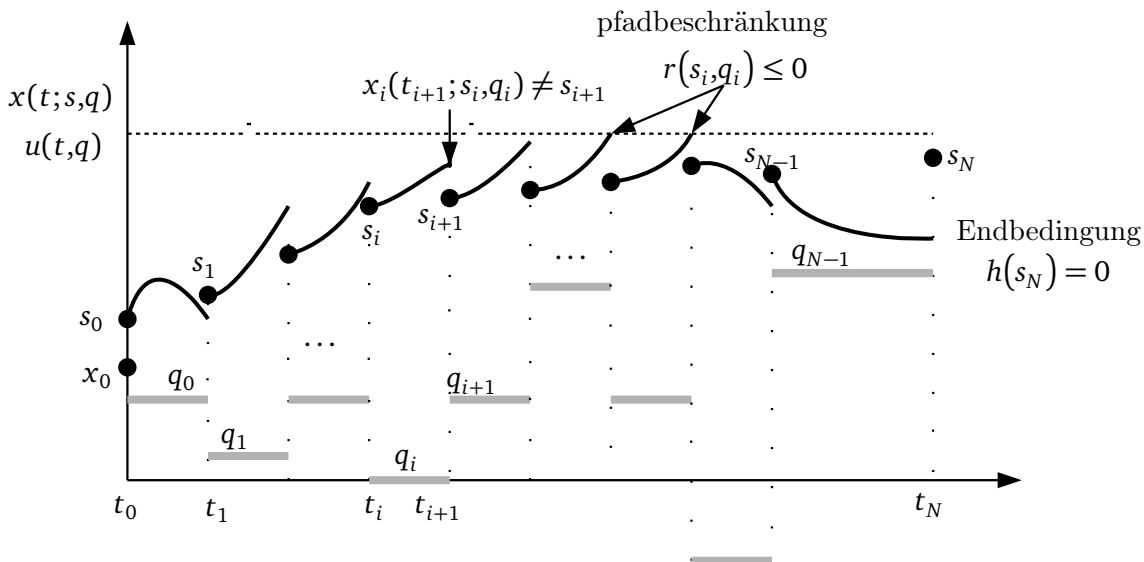


Abbildung 2.7.: Übersicht über das Mehrschießverfahren(eindimensionaler Fall). Die Steuergrößen $\mathbf{u}(t, \mathbf{q})$ werden stückweise konstant angenommen und abhängig von \mathbf{q} parametrisiert, die Zustände haben in jedem Zeitintervall unterschiedliche Verläufe. Die Stetigkeit wird durch die Schießbedingungen gewährleistet. In Anlehnung an [10]

Wie alle anderen Systemgrößen wird auch eine Zielfunktion in jedem Teilbereich definiert. Der Lagrange- und Meyer-Term werden zerlegt, für die N Teilbereiche summiert und in Gleichung (2.29) geschrieben.

$$\int_{t_0}^{t_0+T} L(\mathbf{x}(t), \mathbf{u}(t)) dt + E(\mathbf{x}(t_0 + T)) = \sum_{i=0}^{N-1} l_i(\mathbf{s}_i, \mathbf{q}_i) + E(\mathbf{s}_N)\tag{2.29}$$

Wobei $l_i(\mathbf{s}_i, \mathbf{q}_i)$ der im Teilbereich $[t_i, t_{i+1}]$ definierte Lagrange-Term darstellt und die folgende Form hat:

$$l_i(\mathbf{s}_i, \mathbf{q}_i) = \int_{t_i}^{t_{i+1}} L(\mathbf{x}_i(t; \mathbf{s}_i, \mathbf{q}_i), \mathbf{q}_i) dt$$

Die Ungleichungsbeschränkungen werden dementsprechend parametrisiert. Die Gleichung (2.30) beschreibt die daraus erhaltenen Bedingungen an den Stützpunkten.

$$\mathbf{r}_i(\mathbf{s}_i, \mathbf{q}_i) \leq 0 \quad \forall i \in \{0, \dots, N\} \quad (2.30)$$

Dazu kommen die Anfangs- und Endbedingungen (2.31) und (2.32).

$$\mathbf{s}_0 - \mathbf{x}_0 = 0 \quad (2.31)$$

$$\mathbf{h}(\mathbf{s}_N) = 0 \quad (2.32)$$

Die Gleichungen (2.28) bis (2.32) werden nun zusammengefügt, daraus entsteht das nichtlineare Programm(NPL) (2.33), mit $\mathbf{s} = \{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_N\}$

$$\min_{\mathbf{s}, \mathbf{q}} \sum_{i=0}^{N-1} l_i(\mathbf{s}_i, \mathbf{q}_i) + E(\mathbf{s}_N) \quad (2.33a)$$

$$\text{u.B.} \quad \mathbf{s}_{i+1} = \mathbf{x}_i(t_{i+1}; \mathbf{s}_i, \mathbf{q}_i) \quad \forall i \in \{0, 1, \dots, N-1\} \quad (2.33b)$$

$$\mathbf{s}_0 - \mathbf{x}_0 = 0 \quad (2.33c)$$

$$\mathbf{h}(\mathbf{s}_N) = 0 \quad (2.33d)$$

$$\mathbf{r}_i(\mathbf{s}_i, \mathbf{q}_i) \leq 0 \quad \forall i \in \{0, \dots, N\} \quad (2.33e)$$

Das NLP (2.33) stellt die diskrete Form des allgemeinen OCPs (2.24) dar. Es hat den Vorteil, dass der Lösungsraum endlich ist. Denn die Anzahl von Optimierungsvariablen und Nebenbedingungen ist bekannt. Für ein sehr feines Zeitgitter($N \rightarrow \infty$) konvergiert die Lösung des diskreten Problems (2.33) gegen die gleiche Lösung von (2.24). Der Parameter N bestimmt, wie viele Schritte man in die Zukunft planen will und wird daher auch als Horizont genannt.

In der diskreten Form wird die numerische Handhabung von Variablen möglich. Das OCP (2.33) ist generell nicht konvex und wird durch die aus dem Newton-Verfahren abgeleiteten Verfahren gelöst. Dazu gehört das SQP-Verfahren(engl. sequential quadratic programming), wobei das Problem fortlaufend als konvex quadratisches Programm in Teilproblemen angenähert wird[19]. Das Interior-Point-Verfahren wird auch zur Lösung des NLPs eingesetzt. Wir können hier nicht das gesamte Verfahren erläutern und weisen daher interessierte Leser auf [20] hin.

2.4 Hindernisbeschreibung

Die Trajektorie eines Fahrzeugs ändert sich bezüglich des Fahrzeugumfelds. Die Beschreibung aller im Umfeld des Fahrzeugs befindlichen Objekte muss explizit sein, um Kollision zu vermeiden. Diese umliegenden Objekte werden von Sensoren erfasst. Sie bestehen zum großen Teil aus anderen Verkehrsteilnehmern (mit unterschiedlichen Größen und Volumen) und Gegenständen, die auf der Fahrbahn liegen. Da die genaue Form eines Hindernisses nur schwer erkannt wird, lassen sich Hindernisse nur annähernd beschreiben. In zweidimensionaler Ebene eignen sich Ellipsen, Parabeln, Hyperbeln, Vielecke, etc. zur Abgrenzung eines Bereichs der Fahrbahn, wo das Kollisionsrisiko besteht. Es wird zwischen statischen und dynamischen Hindernissen unterschieden. Zu den Statischen zählen Objekte deren Positionen zeitunabhängig sind, wie geparkte Fahrzeuge. Im Gegenteil dazu sind dynamische Hindernisse meistens in Bewegung und haben zeitabhängige Positionen.

2.4.1 Ellipsen- und Parabelförmige Hindernisse

Ellipsen sind geometrische Figuren, deren Verläufe die Konturen eines geschlossenen Bereichs des zweidimensionalen Raums darstellen. Sie beschreiben eine Menge von Punkten, welche die Ellipsengleichung erfüllen. Ellipsen haben einen glatten Verlauf und lassen sich lediglich aus vier Parametern beschreiben. Der Bereich der Fahrbahn, der durch ein Hindernis besetzt ist, lässt sich gut durch die Hüllkurve einer Ellipse beschreiben. Gleichung (2.34) beschreibt alle Punkte $(x, y)^T$, die außerhalb einer Ellipse liegen und kann somit verwendet werden, um ein Hindernis abzusperren. Der Mittelpunkt des Hindernisses liegt im Punkt $(x_h, y_h)^T$ und die Brennpunkte jeweils im $(r_a, 0)^T$ und $(0, r_b)^T$. Abbildung 2.8a illustriert eine Verkehrssituation, wo ein Hindernis durch eine Ellipse abgebildet wird.

$$\frac{(x - x_h)^2}{r_a^2} + \frac{(y - y_h)^2}{r_b^2} > 1 \quad (2.34)$$

In der Literatur werden auch anstelle Ellipsen Kreise verwendet, um Hindernisse abzubilden. Dieser Ausnahmefall lässt sich erreichen, indem $r_a = r_b$ als Radius des Kreises ausgewählt wird. Parabeln (bzw. Hyperbeln) sind weitere Hüllkurven, die zur Beschreibung von Hindernissen gebraucht werden. In Abbildung 2.8b ist ein Hindernis durch die Kurve einer Parabel gehüllt.

2.4.2 Rechteckförmige Hindernisse

Neben den oben vorgestellten Hüllkurven lassen sich Hindernisse auch durch allgemeine Vielecke beschreiben. Wenn die Anzahl von Ecken unendlich wird, konvergiert ein Vieleck zu einer Ellipse. Vielecke zeigen gegenüber Ellipsen den Vorteil, dass sie sich besser den Fahrzeugrahmen anpassen. In diesem Abschnitt wird gezeigt, wie Hindernisse durch Rechtecke beschrieben

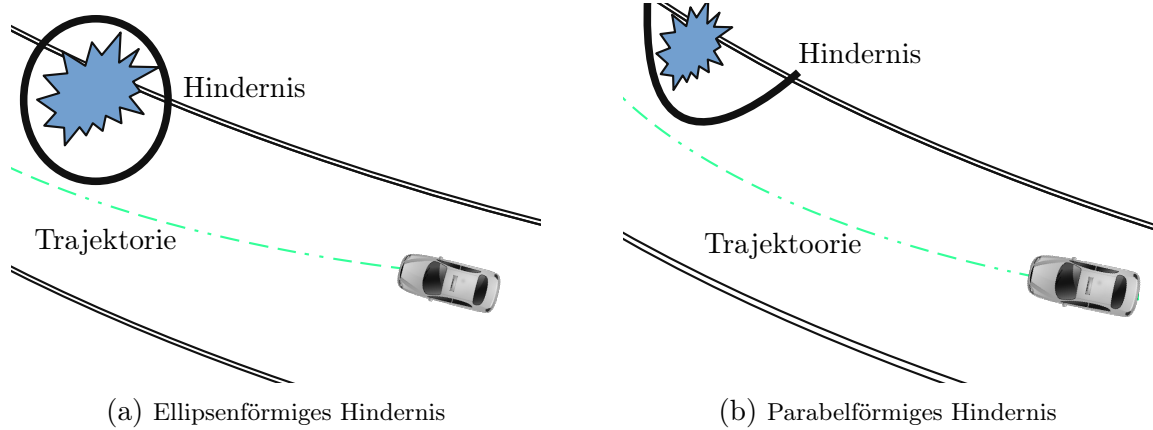


Abbildung 2.8.: Abgebildete Hindernisse in einer Verkehrssituation. Die genauen Größen des betrachteten Hindernisses sind nicht bekannt.

werden.

Gegeben sei ein im Mittelpunkt eines Rechtecks liegendes Koordinatensystem, wie es in Abbildung 2.9 zu sehen ist. Jeder Punkt der Ebene $\mathbf{x} = (x, y)^T$, der im Rechteck liegt, erfüllt Gleichung (2.35). Anhand dieser Gleichung kann eine Bedingung zur Kollisionsvermeidung gestellt werden.

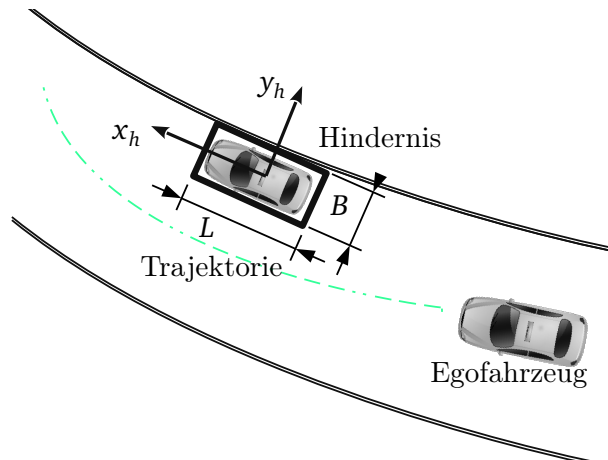


Abbildung 2.9.: Lokale Abbildung eines Hindernisses(bzw. Fahrzeug) als Rechteck

$$\mathbf{A}_0 \mathbf{x} \leq \mathbf{b}_0 \quad (2.35)$$

$$\text{wobei } \mathbf{A}_0 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \text{und} \quad \mathbf{b}_0 = \frac{1}{2} \begin{bmatrix} L \\ L \\ B \\ B \end{bmatrix}$$

L und B stellen jeweils die Länge und Breite eines Hindernisses dar. Es ist ersichtlich, dass Rechtecke für eine gute Beschreibung von Hindernissen geeignet sind.

3 Modellbildung und Koordinatendefinition

Im Kapitel 2 wurden anhand allgemeiner nichtlinearen Systeme einige mathematische Grundlagen beschrieben. Da in dieser Arbeit an einem konkreten Beispiel gearbeitet wird, beschäftigt sich dieses Kapitel damit geometrische und physikalische Parameter der Umgebung zu definieren. Zu den physikalischen Parametern gehören vor allem alle Verkehrsteilnehmer, wie Fahrzeuge oder Passanten. Die geometrischen Parameter ermöglichen eine Raumanordnung einzelner Teilsysteme.

Im Folgenden wird zunächst ein Fahrzeugmodell aus der Literatur ausgesucht, das ausreichende Eigenschaften eines realen Fahrzeugs besitzt und zugleich keine große Komplexität darstellt. Anschließend werden unterschiedliche Koordinatensysteme zur Beschreibung der Fahrzeugbewegung eingeführt und abschließend wird eine Angabe darüber gemacht, wie eine Kollision zwischen Teilsystem entstehen kann.

3.1 Modellierung der Fahrzeugbewegung

In der Literatur[3] gibt es eine Vielzahl an Fahrzeugmodellen mit unterschiedlicher Dynamik. Um die Realität besser abzubilden, muss die Gleichung der Fahrzeugbewegung mindestens von zweiter Ordnung sein, damit die Beschleunigung berücksichtigt wird.

Der Einfachheit halber wird in dieser Arbeit ein einrädriiges Fahrzeugmodell verwendet, das in Abbildung 3.1 graphisch dargestellt ist. Aus dieser Abbildung ist ersichtlich, dass keine

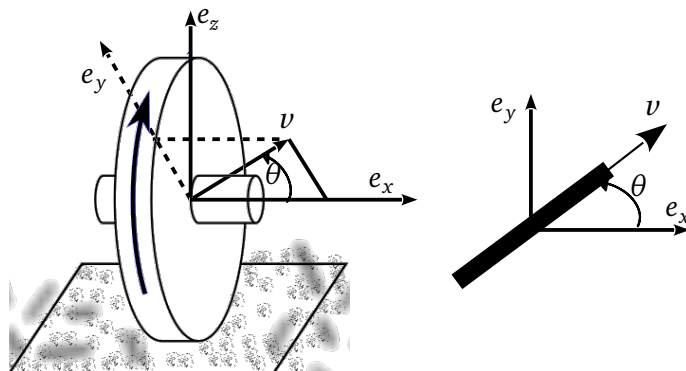


Abbildung 3.1.: Darstellung eines einrädriigen Fahrzeugmodells(links) und der Draufsicht(rechts)

Bewegung in e_z -Richtung vorhanden ist. Vorausgesetzt, dass die Ebene flach genug ist, um einen möglichen Sprung des Rades zu vermeiden. Das Fahrzeug lässt sich daher in xy -Ebene modellieren. Während der Fahrt sind folgende Änderungen in der Bewegung zu erwarten:

- Die Orientierung des Rads bewirkt eine Drehung um die e_z -Achse. Daraus entsteht der Drehwinkel θ
- Durch die Vorwärtsdrehung des Rads wird das Fahrzeug in Bewegung mit linearer Geschwindigkeit v gesetzt
- Die Position des Fahrzeugs kann zu jedem Zeitpunkt als Koordinaten x und y des Mittelpunkts des Rades angenommen werden.

Die vorgeschriebenen Aktionen lassen sich in die Bewegungsgleichung (3.1) zusammenfassen, wobei ω die Drehwinkelgeschwindigkeit darstellt.

$$\begin{aligned}\dot{x}(t) &= v(t)\cos(\theta(t)) \\ \dot{y}(t) &= v(t)\sin(\theta(t)) \\ \dot{\theta}(t) &= \omega(t)\end{aligned}\tag{3.1}$$

Diese Differentialgleichung(Dgl.) hat die gleiche Struktur wie bei einem punktförmigen Fahrzeug, denn nur die Bewegung des Referenzpunkts(bzw. Zentrum oder Mittelpunkt) des Rads wird berücksichtigt. Dieser Punkt bewegt sich in der momentanen Richtung $\theta(t)$ mit der momentanen Geschwindigkeiten $v(t)$ und $\omega(t)$ fort. Für eine gegebene Anfangssituation, wo der Zustand des Fahrzeugs die Gleichung (3.2) erfüllt, lassen sich die Position und Orientierung des Fahrzeugs in jedem Zeitpunkt t durch Gleichung (3.3) bestimmen.

$$\begin{bmatrix} x(t_0) \\ y(t_0) \\ \theta(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix}\tag{3.2}$$

$$\begin{aligned}x(t) &= x_0 + \int_{t_0}^t v(\tau)\cos(\theta(\tau))d\tau \\ y(t) &= y_0 + \int_{t_0}^t v(\tau)\sin(\theta(\tau))d\tau \\ \theta(t) &= \theta_0 + \int_{t_0}^t \omega(\tau)d\tau\end{aligned}\tag{3.3}$$

Daraus erkennt man die Struktur eines Anfangswertproblems, das beispielsweise bei konstanten Geschwindigkeiten v und ω analytisch integriert werden kann, um die zeitabhängigen Größen $x(t)$, $y(t)$ und $\theta(t)$ zu ermitteln.

Bei der Regelung des Systems in (3.1) würde man die Geschwindigkeiten $v(t)$ und $\omega(t)$ als Stellgrößen annehmen. Was realistisch keinen Sinn macht[3], denn auch beim Fahren lässt sich die Geschwindigkeit eher durch Treten des Gaspedals ändern. Angenommen, dass die Koordinaten x und y des Mittelpunkts des Rades zweimal differenzierbar sind, bietet sich die Möglichkeit, ein I-Glied vor $v(t)$ und vor $\omega(t)$ zu plazieren[3]. Somit wird keine Änderung von $v(t)$ oder $\omega(t)$

direkt erfolgen, sondern durch Vorgabe ihrer Ableitungen. Diese neu eingeführten Systemgrößen sind die Drehbeschleunigung und lineare Beschleunigung, die nun als Stellgröße verwendet werden. Wie alle Stellgrößen, müssen sie begrenzt sein (Stellgrößenbegrenzung). Die Geschwindigkeiten sind zeitabhängige kontinuierliche Funktionen. Die letztendliche Bewegungsgleichung besitzt die Form (3.4)

$$\begin{aligned}
 \dot{x}(t) &= v(t)\cos(\theta(t)) \\
 \dot{y}(t) &= v(t)\sin(\theta(t)) \\
 \dot{\theta}(t) &= \omega(t) \\
 \dot{v}(t) &= u_1(t) \\
 \dot{\omega}(t) &= u_2(t)
 \end{aligned} \tag{3.4}$$

Diese nichtlineare Dgl. kann noch kompakter geschrieben werden, indem alle Zustandsgrößen zu einem Vektor zusammengefasst werden. Es ergibt sich Gleichung (3.5).

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.5}$$

$$\text{wobei } \mathbf{x}(t) = \begin{bmatrix} x \\ y \\ \theta \\ v \\ \omega \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \text{und} \quad \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \\ \omega(t) \\ u_1 \\ u_2 \end{bmatrix}$$

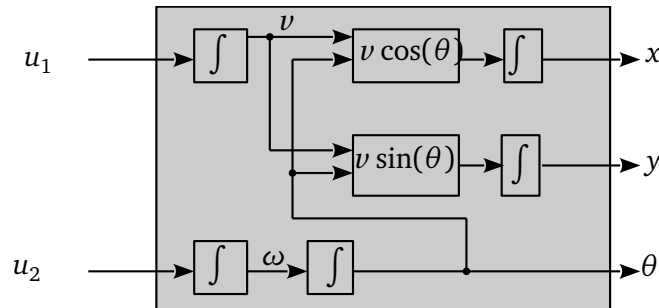


Abbildung 3.2.: Nichtlineares Modell des Fahrzeugs

Aus dem vorgestellten Fahrzeugmodell erhält man das Blockschaltbild in Abbildung 3.2. In dieser Arbeit wird eine Ausgangsgröße \mathbf{y} angenommen, welche die Form (3.6) besitzt.

$$\mathbf{y} = \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} = \mathbf{C}^T \mathbf{x}, \quad \text{mit } \mathbf{C}^T = [1, 1, 1, 0] \tag{3.6}$$

x	Zustandsvektor	(5,1) – Vektor
u	Stellgrößenvektor	(2,1) – Vektor
f	Vektorfunktion	(5,1) – Vektor
y	Ausgangsvektor	(4,1) – Vektor

Das Gleichungssystem (3.5) beschreibt die Bewegung des Fahrzeugschwerpunkts relativ zu einem Koordinatensystem (liegt irgendwo im Raum), das die gleiche Orientierung wie $\{e_x e_y e_z\}$ hat und bezüglich dessen das Fahrzeug die Koordinaten (3.7) besitzt.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (3.7)$$

Für eine genaue Beschreibung der Fahrzeugbewegung werden im nächsten Abschnitt mögliche KS vorgestellt.

3.2 Koordinatentransformation

Die Räumliche Anordnung des Fahrzeugs umfasst die Position und die Orientierung, die in einem festen oder beweglichen KS beschrieben werden können. Bewegliche KSe haben den Vorteil, dass sie durch ihre Flexibilität eine einfache Darstellung des Fahrzeugs ermöglichen. Es wird aber häufig eine Rücktransformation in ein festes KS verlangt, die aufwändig sein könnte. Es ist daher notwendig sich darüber Gedanken zu machen, welches KS zur Beschreibung des Fahrzeugs geeigneter ist. Bevor diese Frage beantwortet wird, muss zunächst der Arbeitsraum grob definiert werden.

3.2.1 Globale- und Lokale Betrachtung des Problems

Die Ermittlung einer befahrbaren Trajektorie ist mit großem Rechenaufwand verbunden. Es muss daher auf unnötige Rechnungen verzichtet werden. Die Verwendung eines weltfesten kartesischen KS, hat den Nachteil, dass die Fahrbahn nur schwer beschrieben werden kann. Es ist beispielsweise ersichtlich, dass wenn starke Krümmungen in dem Straßenverlauf gibt, kann die Trajektorie nicht mehr durch das inertielle KS (engl. base frame oder inertial frame) beschrieben werden. Viele Arbeiten wie [21, 5] führen deswegen das Frenet-Koordinatensystem ein, um den Straßenverlauf zu entkrümmen und somit die Berechnung der Trajektorien zu erleichtern. Das hat natürlich den Nachteil, dass die Beschreibung des Systems komplizierter oder eventuell, dass das System um neue Variablen erweitert wird. Eine alternative zu dieser Vorgehensweise wäre, dass das Problem auf einer lokalen Ebene betrachtet wird. Durch Verwendung klassischer Transformationsbeziehungen von Punktkoordinaten (bzw. Koordinaten eines Punkts) beim Wechseln von globalem nach lokalem KS werden die gleichen Ergebnisse erreicht. Außerdem müssen keine zusätzlichen Variablen in die Aufgabe eingeführt werden. Aus diesem Grund wird die Beschreibung des Systems in einem lokalen KS des Raums stattfindet.

3.2.2 Relativbewegung

Die globale Lokalisierung eines Fahrzeugs ist in vielen Situationen nicht relevant, denn bei der Durchführung bestimmter Aufgaben, wie Einparken, zählt eher die lokale Position des Fahrzeugs. Aus diesem Grund ist auch eine lokale Lokalisierung von besonderer Bedeutung. In dieser Arbeit werden vier Koordinatensysteme zur Beschreibung der Fahrzeugbewegung verwendet:

1. $\mathcal{F}_f = \{\mathbf{O}_f x_f y_f\}$

Das Fahrzeugkoordinatensystem wird mit \mathcal{F}_f bezeichnet und liegt im Schwerpunkt des Fahrzeugs. Es ermöglicht die Position und Orientierung des Fahrzeugs zu bestimmen. Daher hat die x_f -Achse die gleiche Richtung wie die momentane lineare Geschwindigkeit des Fahrzeugs.

2. $\mathcal{F}_h = \{\mathbf{O}_h x_h y_h\}$

Genauso wie das Fahrzeug besitzt ein Hindernis eine Lage und eine Orientierung. Aus diesem Grund dient das hindernisbezogene KS zur Darstellung von Hindernissen. Es liegt im Mittelpunkt(bzw. Schwerpunkt) des Hindernisses. Die x_h -Achse wird in Längsrichtung des Hindernisses ausgerichtet.

3. $\mathcal{F}_s = \{\mathbf{O}_s x y\}$

Das Straßenkoordinatensystem liegt auf der Mittellinie der Straße(Referenzkurve \mathcal{R}). Der Ursprung wird in jedem Zeitpunkt durch \mathbf{O}_s beschrieben, der sich entlang der Referenzkurve \mathcal{R} bewegt. Die x -Achse zeigt in Richtung der longitudinalen Bewegung und bleibt dabei immer tangential zur \mathcal{R} ausgerichtet. Die y -Achse steht senkrecht und wird nach links gezeigt. Die einzelnen Teilsysteme werden in diesem KS beschrieben, wo die Positionen des Fahrzeugs und Hindernisses jeweils durch $\mathbf{O}_{f/s}$ und $\mathbf{O}_{h/s}$ definiert werden.

$$\mathbf{O}_{f/s} = \begin{bmatrix} x \\ y \end{bmatrix} \text{ Relative Position des Fahrzeugs bezüglich } \mathcal{F}_s$$
$$\mathbf{O}_{h/s} = \begin{bmatrix} x_h \\ y_h \end{bmatrix} \text{ Relative Position eines Hindernisses bezüglich } \mathcal{F}_s$$

4. $\mathcal{F}_w = \{\mathbf{O}_w XYZ\}$

Das weltfeste KS, auch kurz Weltkoordinatensystem oder Inertialsystem, ermöglicht eine

globale Beschreibung des Fahrzeugs. In diesem KS werden die Lagen(bzw. Ursprünge) von \mathcal{F}_s , \mathcal{F}_f und \mathcal{F}_h durch die Vektoren $\mathbf{O}_{s/w}$, $\mathbf{O}_{f/w}$ und $\mathbf{O}_{h/w}$ definiert, wie folgt:

$$\begin{aligned}\mathbf{O}_{s/w} &= \begin{bmatrix} X_s \\ Y_s \end{bmatrix} \text{ Ursprung des Straßenkoordinatensystems bezüglich } \mathcal{F}_w, \\ \mathbf{O}_{f/w} &= \begin{bmatrix} X_f \\ Y_f \end{bmatrix} \text{ Position des Fahrzeugs bezüglich } \mathcal{F}_w \text{ und} \\ \mathbf{O}_{h/w} &= \begin{bmatrix} X_h \\ Y_h \end{bmatrix} \text{ Position eines Hindernisses bezüglich } \mathcal{F}_w\end{aligned}$$

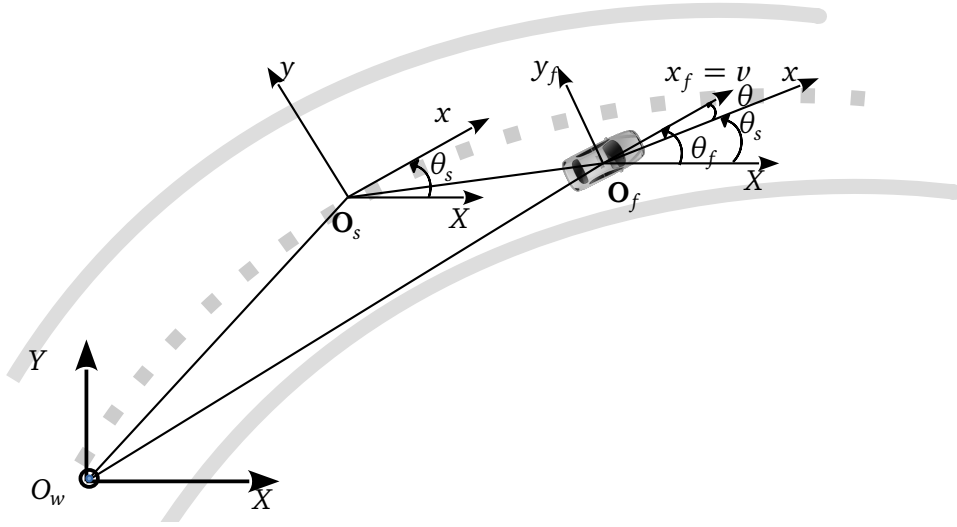


Abbildung 3.3.: Position und Orientierung des Fahrzeugs jeweils im Welt- und Straßenkoordinatensystem dargestellt

\mathcal{F}_w , \mathcal{F}_s , \mathcal{F}_f und \mathcal{F}_h sind alle rechtwinklige, kartesische Koordinatensysteme. Die Einheitsvektoren sind zu einander orthogonal, haben die Länge eins und bilden ein Rechtssystem. Abbildung 3.3 veranschaulicht die geometrische Darstellung aller verwendeten KSe.

Da die Beschreibung der Fahrzeugbewegung in \mathcal{F}_s erfolgt, ist es notwendig, die erhaltenen Größen wieder in \mathcal{F}_w umzuschreiben, um eine globale Lokalisierung zu ermöglichen. Zu jedem Zeitpunkt entsteht die Anordnung von \mathcal{F}_s aus \mathcal{F}_w durch eine Rotation(Drehung) und eine Translation(Verschiebung). Der Translationsvektor $\mathbf{O}_{s/w}$ stellt die Koordinaten des Ursprungs von \mathcal{F}_s bezüglich \mathcal{F}_w dar. Daraus gilt Gleichung (3.8), die die Koordinaten des Fahrzeugs in \mathcal{F}_s und \mathcal{F}_w verbindet.

$$\mathbf{O}_{f/w} = \mathbf{O}_{s/w} + \mathbf{R}_{s/w} \mathbf{O}_{f/s} \quad (3.8)$$

$$\begin{bmatrix} X_f \\ Y_f \end{bmatrix} = \begin{bmatrix} X_s \\ Y_s \end{bmatrix} + \begin{bmatrix} \cos(\theta_s) & -\sin(\theta_s) \\ \sin(\theta_s) & \cos(\theta_s) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$\mathbf{R}_{s/w}$ ist die Rotationsmatrix von \mathcal{F}_s bezüglich \mathcal{F}_w um die Z-Achse. Der Drehwinkel dieser Rotation hängt von der Krümmung der Straße ab und beträgt θ_s . Der tatsächliche Drehwinkel des Fahrzeugs im \mathcal{F}_w lässt sich durch (3.9) bestimmen.

$$\theta_f = \theta_s + \theta \quad (3.9)$$

θ_s : Drehwinkel von \mathcal{F}_s bezüglich \mathcal{F}_w

θ_f : Drehwinkel von \mathcal{F}_f (bzw. Fahrzeug) bezüglich \mathcal{F}_w und

θ : Drehwinkel von \mathcal{F}_f (bzw. Fahrzeug) bezüglich \mathcal{F}_s

Mit (3.8) und (3.9) wird die Rücktransformation durchgeführt. Alle anderen Größen bleiben unverändert.

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{C}^T \mathbf{x} \end{aligned} \quad (3.10)$$

Eine Ausführliche Beschreibung dieser Transformation ist im Abschnitt B erhältlich.

Die dargestellten Transformationen ermöglichen, das Problem von globalem nach lokalem KS zu überführen und umgekehrt. Da eine lokale Betrachtung Vorrang vor der Globale hat, lässt sich das betrachtete System durch die Dgl. (3.10) im \mathcal{F}_s beschreiben. Die Systemgrößen \mathbf{f} , \mathbf{x} , \mathbf{u} und \mathbf{C} behalten dabei die gleiche Struktur aus (3.5) und (3.6).

3.3 Darstellung von Hindernissen im \mathcal{F}_s

Im Abschnitt 2.4 wurden einigen Möglichkeiten zur Darstellung von Hindernissen vorgestellt. Es kommt zwar häufig vor, dass Ellipsen[9, 15] und Parabeln[21] zur Beschreibung von Hindernissen eingesetzt werden, aber in Verkehrssituationen wie auf der Autobahn, soll eher mit anderen Fahrzeugen gerechnet werden. Da Ellipsen nicht die Konturen eines Fahrzeugs richtig abbilden, werden Hindernisse in dieser Arbeit durch Rechtecke dargestellt. Laut [21] können ellipsenförmige Hindernisse dazu führen, dass lokale Minima entstehen. Dies kann die Lösung der Optimierungsaufgabe so schlecht beeinflussen, dass die Regelung mittels MPC problematisch wird. Dieser Abschnitt stellt vor, wie Rechtecke zur Beschreibung von Hindernissen in dem nachfolgenden Kapitel verwendet werden

Im Folgenden wird ein Hindernis angenommen, das sich durch das Koordinatensystem \mathcal{F}_h lokalisieren lässt. Der Vektor \mathbf{O}_h ist im \mathcal{F}_s durch (3.11) definiert.

$$\mathbf{O}_{h/s} = \begin{bmatrix} x_h \\ y_h \end{bmatrix} \quad (3.11)$$

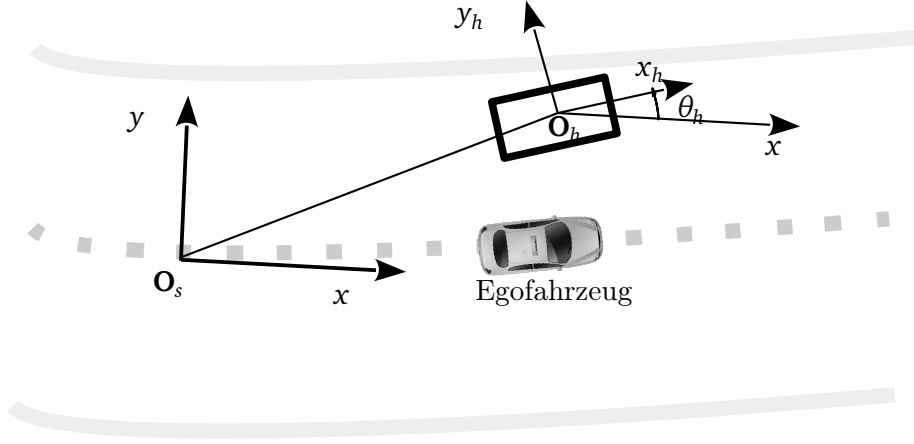


Abbildung 3.4.: Darstellung eines Hindernisses im \mathcal{F}_s

Der Drehwinkel von \mathcal{F}_h bezüglich \mathcal{F}_s wird durch θ_h dargestellt, wie es in Abbildung 3.4 zu sehen ist.

Wie in Abschnitt 2.4 gezeigt wurde, gilt Gleichung (3.12) im \mathcal{F}_h , wobei die Parameter L und B jeweils die Länge und Breite des Hindernisses darstellen. \mathbf{x}_h ist ein im \mathcal{F}_h dargestellter Punkt des Hindernisses.

$$\mathbf{A}_0 \mathbf{x}_h \leq \mathbf{b}_0 \quad (3.12)$$

$$\text{mit } \mathbf{A}_0 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \text{und} \quad \mathbf{b}_0 = \frac{1}{2} \begin{bmatrix} L_h \\ L_h \\ B_h \\ B_h \end{bmatrix}$$

Die Fahrzeugbewegung wird aber im \mathcal{F}_s beschrieben, daher muss die Bedingung (3.12) umgeschrieben werden. Die Transformationsbeziehung (3.8) lässt sich in diesem Fall zwischen \mathcal{F}_h und \mathcal{F}_s in der Form (3.13) schreiben.

$$\mathbf{x}_{h/s} = \mathbf{O}_{h/s} + \mathbf{R}_{h/s} \mathbf{x}_h \quad (3.13)$$

$$\text{wobei } \mathbf{R}_{h/s} = \begin{bmatrix} \cos(\theta_h) & -\sin(\theta_h) \\ \sin(\theta_h) & \cos(\theta_h) \end{bmatrix} \quad \Rightarrow \quad \mathbf{R}_{h/s}^{-1} = \begin{bmatrix} \cos(\theta_h) & \sin(\theta_h) \\ -\sin(\theta_h) & \cos(\theta_h) \end{bmatrix}$$

$\mathbf{x}_{h/s}$ ist der Punkt \mathbf{x}_h (ein Punkt des Hindernisses), deren Koordinaten im \mathcal{F}_s dargestellt werden. Aus (3.13) ergibt sich (3.14)

$$\mathbf{x}_h = \mathbf{R}_{h/s}^{-1} (\mathbf{x}_{h/s} - \mathbf{O}_{h/s}) \quad (3.14)$$

In der Ungleichung (3.12) kann nun \mathbf{x}_h durch (3.14) ersetzt werden. Nach einigen Umformulierungen erhält man die Ungleichung (3.15), die die gleiche Struktur wie (3.12) besitzt. Mit dieser Ungleichung wird die Bedingung formuliert, die erfüllt sein muss, damit ein im \mathcal{F}_s dargestellter Punkt $\mathbf{x}_{h/s}$ in einem Hindernis liegt, das im \mathcal{F}_s durch den Vektor \mathbf{b} beschrieben wird.

$$\mathbf{A}\mathbf{x}_{h/s} \leq \mathbf{b} \quad (3.15)$$

$$\text{wobei } \mathbf{A} = \mathbf{A}_0 \mathbf{R}_{h/s}^{-1} = \begin{bmatrix} \cos(\theta_h) & \sin(\theta_h) \\ -\cos(\theta_h) & -\sin(\theta_h) \\ -\sin(\theta_h) & \cos(\theta_h) \\ \sin(\theta_h) & -\cos(\theta_h) \end{bmatrix} \in \mathbb{R}^{4 \times 2}$$

$$\text{und } \mathbf{b} = \mathbf{b}_0 + \mathbf{A}_0 \mathbf{R}_{h/s}^{-1} \mathbf{O}_{h/s} = \begin{bmatrix} \frac{1}{2}L_h + \cos(\theta_h)x_h + \sin(\theta_h)y_h \\ \frac{1}{2}L_h - \cos(\theta_h)x_h - \sin(\theta_h)y_h \\ \frac{1}{2}B_h - \sin(\theta_h)x_h + \cos(\theta_h)y_h \\ \frac{1}{2}B_h + \sin(\theta_h)x_h - \cos(\theta_h)y_h \end{bmatrix} \in \mathbb{R}^{4 \times 1}$$

Diese Ungleichungsbeschränkung kann verwendet werden, um zu Prüfen, ob es eine Kollision gibt. Wenn diese Ungleichung für zwei oder mehrere unterschiedliche Figuren(bzw. Rechtecke) irgendwann erfüllt wird, heißt es, dass sie miteinander kollidieren.

Für ein global(im \mathcal{F}_w) lokalisiertes Hindernis müssen die Orientierung θ_h und die Position $\mathbf{O}_{h/s}$ von \mathcal{F}_w nach \mathcal{F}_s umgewandelt werden. Dies lässt sich durch Gleichungen (3.16) und (3.17) erreichen, wobei $\theta_{h/w}$ die globale Orientierung des Hindernisses ist.

$$\theta_{h/w} = \theta_s + \theta_h \quad (3.16)$$

$$\begin{bmatrix} X_h \\ Y_h \end{bmatrix} = \begin{bmatrix} X_s \\ Y_s \end{bmatrix} + \begin{bmatrix} \cos(\theta_s) & -\sin(\theta_s) \\ \sin(\theta_s) & \cos(\theta_s) \end{bmatrix} \begin{bmatrix} x_h \\ y_h \end{bmatrix} \quad (3.17)$$

In dem nächsten Kapitel werden die hergeleitete Gleichung der Fahrzeugbewegung und die beschriebenen Transformationen verwendet, um die Trajektorienplanung zu realisieren.



4 Zielpunktadaptive Trajektorienplanung

In den oben liegenden Kapiteln wurden zahlreiche Verfahren sowie deren Anwendungsbereiche beschrieben, die für diese Arbeit notwendig sind. Im Folgenden werden diese theoretisch beschriebenen Verfahren kombiniert und bei der Trajektorienplanung eingesetzt.

Eine Trajektorie ist nichts anderes als ein Pfad mit Zeitkomponenten, der durch ein Fahrzeug fahrbar wird. Verfahren zur Positionsbestimmung wie GPS liefern schon Fahrwege für Fahrzeuge. Jedoch kann man nicht vorhersagen, wann das Fahrzeug an welcher Position und mit welcher Geschwindigkeit sein wird oder ob der Weg frei oder fahrbar ist. Daher werden diese Informationen ohne Hilfe eines Fahrers, der das Fahrzeug bedient, unvollständig. Für selbstfahrende Fahrzeuge werden global gelieferte Informationen über die Bewegungsplanung eher als Referenzwege verwendet. Darauf basierend können Trajektorien ermittelt werden, welche die Dynamik des Fahrzeugs berücksichtigen. Die Trajektorienplanung unterscheidet sich daher von der Pfadplanung dadurch, dass zusätzlich zu den über die Zeit gemachten Angaben auch die Möglichkeit besteht, die umgebenden Objekte, wie Passanten oder andere Verkehrsteilnehmer, zu berücksichtigen. Eine exakte Planung beinhaltet außerdem zahlreiche Restriktionen über alle Zustandsgrößen des Systems, sowie Einhaltung der Stellgrößenbegrenzung. Dabei werden die Komfortaspekte, Sicherheitsaspekte und Energieverbrauch mit einbezogen.

Dieses Kapitel verfolgt daher das Ziel einen Algorithmus zur Trajektorienplanung herzuleiten, der auf die Lösung eines Optimalsteuerungsproblem(OCP) basiert. Zur formelmäßigen Beschreibung der Fahrzeugbewegung wurde das Koordinatensystem(KS) \mathcal{F}_s definiert, dessen Ursprung sich entlang der Straßenmittellinie bewegt. In diesem KS werden Zielpunkte erzeugt, die sich dem Straßenverlauf anpassen und zur Formulierung eines OCPs dienen.

Im ersten Schritt wird die unmittelbare Umgebung des Fahrzeugs diskretisiert, um einen Zielpunkt zu ermitteln. Dabei wird basierend auf den bekannten Positionen von Hindernissen ein hindernisfreier Weg auf der Fahrbahn gesucht, wodurch sich das Fahrzeug bewegen kann. Dies wird durch Optimierung heuristischer Ansätze erfolgen.

Anschließend wird ein adaptiver Punkt auf dem hindernisfreien Weg bestimmt und als Zielpunkt angenommen. Um die tatsächliche fahrbare Trajektorie des Fahrzeugs zu bestimmen wird auf das Verfahren der MPC zurückgegriffen. Dabei wird eine Zielfunktion formuliert, die für die Durchführung bestimmter Aufgaben geeignet ist.

4.1 Strategie zur Trajektorienplanung

Die Tatsache, dass die Trajektorienplanung mittels MPC mit hohem Aufwand verbunden wird, ist uns längst bekannt. Es ist uns auch bewusst, dass der Aufwand abhängig von der Anzahl

von zu behandelten Variablen variiert. Diese Feststellung hat das in dieser Arbeit vorgestellte Konzept motiviert. Die Grundidee besteht darin, das Problem der Ressourcenknappheit bei der Trajektorienplanung zu lösen. Aus diesem Grund wurde die Planung in kleine Teilprobleme zerlegt. Bevor das Planungskonzept erläutert wird, sind einige Voraussetzungen zu klären, um den Lösungsraums zu definieren.

- Vorgegebener Anfangspunkt und Zielpunkt als globale Koordinaten im Raum:
Es wird in dieser Arbeit lediglich das Fahren eines Fahrzeugs von einem vorgegebenen globalen Anfangspunkt zum Zielpunkt des Raums betrachtet. Alle anderen Fahrmanöver, wie Einparken, Rückwärtsfahren, Überholen oder Spurwechsel werden nicht berücksichtigt.
- Fahren in statischer Umgebung:
Die Straße ist einspurig und ihr Verlauf wird strukturiert dargestellt. Somit werden beispielsweise die Breite der Straße oder eine vorhandene Kreuzung auf der Straße eindeutig beschrieben. Es muss nicht auf Gegenverkehr geachtet werden, denn Hindernisse in der Umgebung bewegen sich nicht. Es werden nur Langsamfahrten mit vorgeschriebener Höchstgeschwindigkeit erlaubt.
- Es existiert eine Lösung:
Es muss mindestens eine Lösung zu der Planungsaufgabe geben. Die Straße muss beispielsweise nicht gesperrt sein.

Daraus wird offensichtlich, dass das Planungsziel darin besteht, eine fahrbare Trajektorie bis zum vorgegebenen Zielpunkt zu berechnen. Um die Anzahl von behandelten Parametern zu reduzieren, wird die Gesamtplanungsaufgabe in kleine Planungsaufgaben unterteilt. Die Gesamtlösung wird dann aus Summe einzelner Teillösungen bestehen. Hierbei wird ein Zwischenzielpunkt adaptiv für jeden Schritt der Planung erzeugt. Die lokale durchgeführte Planung wird ab dieser Stelle als Teilaufgabe genannt. Das folgende Planungskonzept beschreibt eine Teilaufgabe, die durch mehrere Wiederholungsschritte die Gesamtlösung ergibt.

- Koordinatentransformation
Für die Formulierung einer Teilaufgabe werden zunächst der Ursprung von \mathcal{F}_s und die Orientierung θ_s festgelegt. Danach werden alle benötigten Informationen über das Umfeld des Fahrzeugs anhand der vorgestellten Koordinatentransformation (3.8), (3.9), (3.16) und (3.17) in das Straßenkoordinaten \mathcal{F}_s überführt. Dazu gehören beispielsweise der Zielpunkt, die aktuellen Zustände des Fahrzeugs, die Positionen von relevanten Hindernissen und Parameter der Strecke. Das vereinfacht deutlich die Beschreibung des Straßenverlaufs. Daraus ergibt sich eine Karte der lokalen Umgebung und entspricht dem ersten Schritt der Planung. Nach der Planung müssen die nächsten Zustände des Fahrzeugs zurück nach \mathcal{F}_w umgerechnet werden.
- Gitter aus Grundgeometrien
Anhand der Informationen aus der erstellten Karte wird ein straßenadaptiertes Gitter erzeugt. Durch die Optimierung einer Kostenfunktion werden aus den Gitterpunkten mög-

- Regelsystem

Die erhaltene Punktfolge wird zu einem Regelsystem weiter geleitet, das abhängig vom aktuellen Zustand des Fahrzeugs, von der Position der Hindernisse und von verfügbaren Ressourcen entschieden wird, wo der Zwischenzielpunkt sein wird. Es wird nur die Zeitressource in dieser Arbeit berücksichtigt. Das Regelsystem wird dann den Zwischenzielpunkt weit vom Fahrzeug auswählen, wenn die Berechnungen weniger Zeit kosten. Wenn dagegen viel Zeit in Anspruch genommen wird, reagiert das Regelsystem entsprechend und wählt den Zwischenzielpunkt nah am Fahrzeug aus. Dadurch kann die zur Berechnung verantwortliche Komponente der Planung entlastet werden und somit wird die Ressource effizient ausgenutzt.

- MPC

Die MPC stellt die Komponente dieses Planungskonzepts dar, welche die Lösung der Optimierungsaufgabe übernimmt. Aus der erstellten Karte und dem vordefinierten Zwischenzielpunkt wird ein OCP formuliert und gelöst. Dadurch entsteht der genaue Verlauf der Trajektorie in der lokalen Umgebung. Dieses Ergebnis wird als Anfangszustand(bzw. Anfangsposition) des Fahrzeugs für die nächste Teilaufgabe verwendet. Eine Planung wird erneut durchgeführt. Dies wiederholt sich bis der globale Zielpunkt erreicht wird.

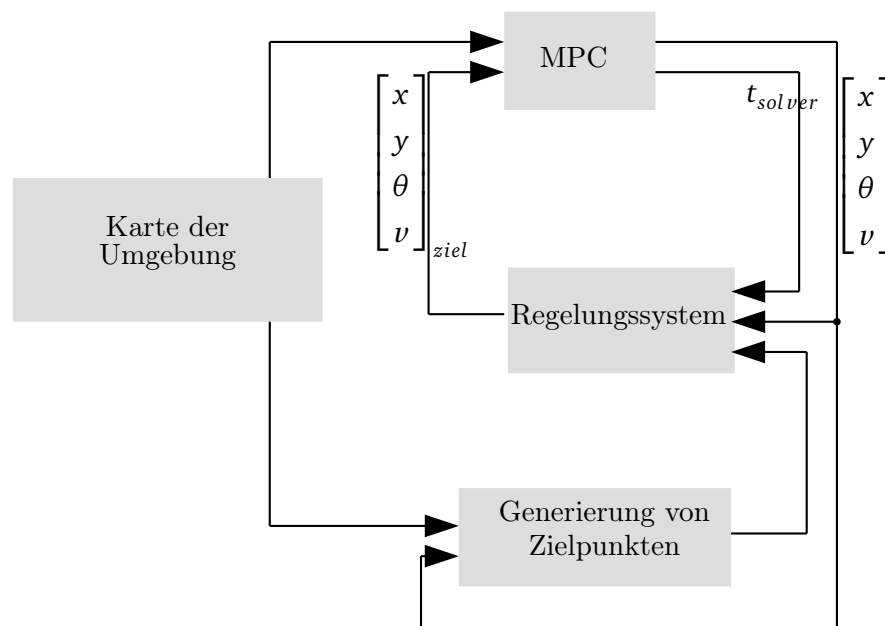


Abbildung 4.1.: Darstellung des gesamten Planungsalgorithmus. t_{Solver} ist die vom Solver benötigte Zeit zur Lösung eines OCPs.

Eine solche vorgestellte Teilaufgaben kann formuliert und unabhängig von dem globalen Zielpunkt gelöst werden. Abbildung 4.1 zeigt eine Übersicht über die einzelnen Komponenten. Die genaue Durchführung einzelner Teilaufgaben wird in dem nächsten Abschnitt erläutert. Da die Koordinatentransformation schon vorgestellt wurde, wird sie hier nicht mehr besprochen.

4.2 Generierung von Zielpunkten

In jeder Teilaufgabe muss ein Zwischenzielpunkt vorgegeben werden, wohin die Planung führen muss. In dem vorgestellten Planungskonzept wird in dem zweiten Schritt der Planung ein straßenadaptiertes Gitter erzeugt. Die Herangehensweise basiert sich auf der Verwendung von Gittern aus Grundgeometrien(engl. state lattices). Dieses Verfahren ist bekannt und wird schon in Arbeiten wie [4, 9] eingesetzt. Ein davon abgeleitetes Verfahren zur Generierung straßenadaptierter Punkte wird im Folgenden beschrieben.

4.2.1 Beschreibung des Verfahrens

Die in dieser Komponente zu beantwortende Frage lautet, wo sich die möglichen Stellen auf der Fahrbahn befinden und wohin sich das Fahrzeug bewegen kann. Ganz grob lautet die Antwort, überall wo kein Hindernis liegt. Durch diese Antwort ist zu verstehen, dass es unendlich viele Lösungen gibt. Alle hindernisfreien Stellen der Fahrbahn müssen daher untersucht werden.

Bei der geometrischen Bahnplanung[1, 22] werden typische Verfahren, wie z. B. das Roadmap-Verfahren, die exakte oder approximative Zellzerlegung und Voronoi Diagramme, zur Untersuchung eines Konfigurationsraums verwendet. Durch diese Verfahren lässt sich den freien Raum sehr gut finden. Das Roadmap-Verfahren liefert jedoch einen Weg, der sehr nah am Hindernis liegt. Außerdem hängen diese Verfahren sehr viel von der Geometrie des Hindernisses ab und untersuchen daher nur eine sehr begrenzte Anzahl von Punkten des Konfigurationsraums.

Im Gegenteil zu den genannten Verfahren wird durch die Verwendung von Gittern aus Grundgeometrien viel mehr Stellen der Fahrbahn(Konfigurationsraum) untersucht und die Geometrie des Hindernisses spielt dabei keine große Rolle. Aus diesem Grund wird in dieser Arbeit basierend auf straßenadaptierten Punkten[9] ein Verfahren hergeleitet.

Ausgangspunkt des Verfahrens ist die aktuelle Position des Fahrzeugs \mathbf{P}_0 in (4.1) Bezüglich des Straßenkoordinatensystems \mathcal{F}_s , das irgendwo auf der Straße und nah genug am Fahrzeug liegt

$$\mathbf{p}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (4.1)$$

Es werden hintereinander in Längsrichtung M Schichte(engl. layer) von Stichpunkten(engl. samples) erzeugt, die in einem Abstand von Δs zueinander liegen. Für jede Schicht sind insgesamt K Stichpunkte vorgesehen. Hiermit wird eine vorgegebene Entfernung s_{max} in Fahrtrichtung abgedeckt.

Um eine reguläre Abdeckung der Fahrbahn zu ermöglichen müssen die Schichten immer senkrecht zu dem Verlauf der Straße stehen. Eine mögliche Darstellung solcher Datenpunkte ist in Abbildung 4.2 zu sehen. Wobei ein stückweise stetiger Weg, der durch einige ausgewählte Datenpunkte verläuft, zu sehen ist. Es wird bei jeder Schicht $\{L_i\}_{i<1,\dots,M}$ ein Punkt p_i ausgewählt, der als optimal gilt. Die nächste Herausforderung bei dieser Vorgehensweise besteht darin, die M optimalen Punkte durch eine mathematische Formel zu beschreiben. Denn sie können nicht beliebig ausgewählt werden. Die Lösung dieser Aufgabe wird im nächsten Abschnitt besprochen. Es wird darauf hingewiesen, dass die Größen Δs , Δl , M , s_{max} , und K Parameter der Entwicklung darstellen und beispielsweise zur Verfeinerung des Gitters verwendet werden können.

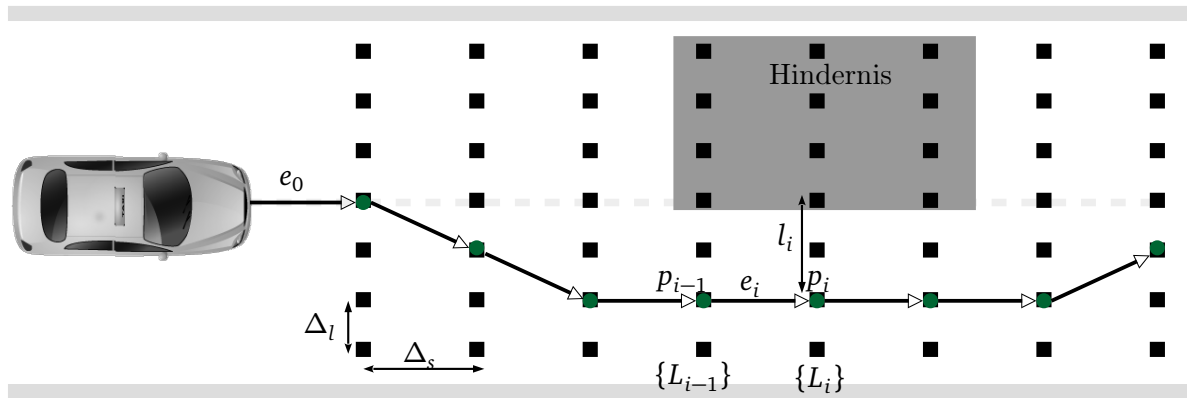


Abbildung 4.2.: Straßenadaptiertes Gitter

4.2.2 Lösungsansatz

In diesem Abschnitt wird die Annahme gemacht, dass das Fahrzeug vorzugsweise immer auf der Mittellinie der Straße bleibt (Spurhalten), wenn sie frei ist. Daher wird sie als Referenzweg \mathcal{R} angesehen. Die Verteilung von Stichpunkten muss in der Art erfolgen, dass jede Schicht immer ein Stichpunkt auf \mathcal{R} besitzt. Dadurch ist es offensichtlich, dass der Parameter K ungerade sein muss. Die Aufgabenstellung weist die gleiche Struktur auf, wie die Suche nach dem kürzesten hindernisfreien Weg zwischen zwei Punkten. Wobei in diesem Fall nur der Anfangspunkt vorgegeben wird. Viele Suchalgorithmen, wie z. B. A^* oder Dijkstras, eignen sich gut für solche Aufgabenstellungen. Dabei muss zunächst eine Kostenfunktion definiert werden, womit für alle Bezugspunkte die entsprechenden Kosten berechnet werden. Erst danach wird das Verfahren eingesetzt. Es wird daher sehr viele Rechnungen geben, denn für jede Planung werden neue Bezugspunkte definiert. Als alternativer Weg wird die Auswahl optimaler Stichpunkte durch Optimierung einer heuristischen Ansatzfunktion beschrieben. Zur Herleitung der zu optimierenden Gütemaße sind folgende Überlegungen zusammenzuführen:

- Die Stichpunkte werden bevorzugt, wenn diese auf \mathcal{R} liegen.

- kein optimaler Stichpunkt darf im Hindernis liegen
- Die Kante zwischen zwei hintereinander liegenden optimalen Stichpunkten darf kein Hindernis berühren.

Die vorgestellten Voraussetzungen führen dazu die Hilfsvariablen e_i und l_i einzuführen, mit Hilfe deren die Optimierungsaufgabe (4.2) definiert wird[9].

$$\{\mathbf{p}^*\} = \min_{\mathbf{p}_i \in \{L_i\}} \sum_{i=1}^M \omega_1 l_i^2 + \omega_2 e_i^2 + \mathcal{C}_h(e_i) \quad (4.2)$$

Wobei ω_1 und ω_2 beliebig auswählbare Parameter sind. Die Entfernung zwischen \mathbf{p}_i und \mathcal{R} wird mit l_i bezeichnet. Die Kante von \mathbf{p}_{i-1} nach \mathbf{p}_i wird mit e_i bezeichnet. $\mathcal{C}(e_i)$ ist ein Maß, die dafür sorgt, das optimale Stichpunkte nicht im Hindernis liegen. Sie beträgt Null, wenn e_i kein Hindernis berührt. Im Gegenteil dazu wird sie auf unendlich gesetzt. Da für jede Schicht K Stichpunkte untersucht werden müssen, müssen zu viele Kombinationen verglichen werden. Um unnötige Rechnungen zu vermeiden, wird zur Lösung der Optimierungsaufgabe (4.2) das Verfahren der dynamischen Programmierung eingesetzt werden[9]. Denn der Lösungsraum wurde schon parametrisiert(bzw. diskretisiert) und es wird lediglich ein optimaler Stichpunkt pro Schicht gesucht. Dieses Verfahren wurde schon grob im Abschnitt 2.2.2 zur Lösung eines OCPs eingeführt und wird für die zu lösende Aufgabe angepasst. Es wird beruhend auf dem Optimalitätsprinzip von Bellman die Optimale Lösung in optimalen Teillösungen zerlegt. Ausgehend von der Schicht $M - 1$ wird die Fortsetzungskosten (4.3) definiert. Dadurch wird der Stichpunkt in Schicht $\{L_M\}$ gesucht, womit die kleinsten Kosten erreicht werden. Dabei werden alle Stichpunkte von $\{L_M\}$ untersucht, wie Abbildung 4.3a es zeigt.

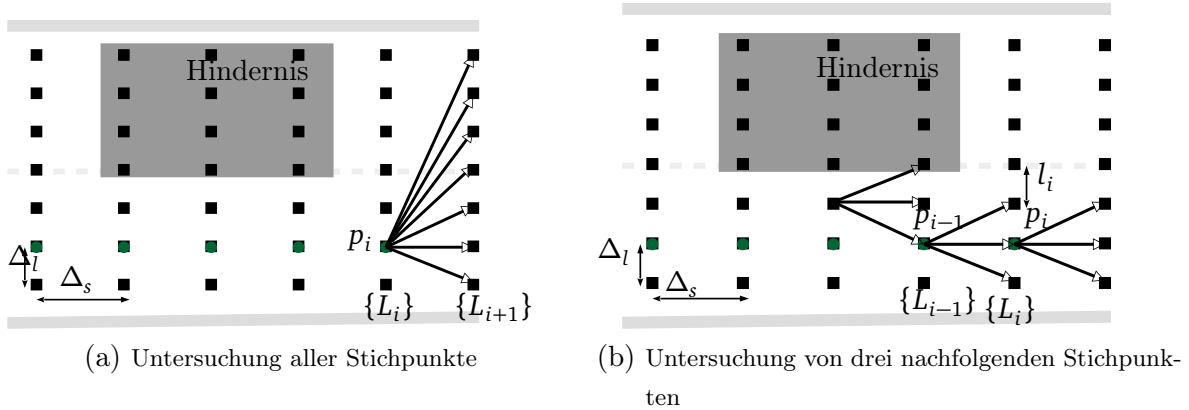


Abbildung 4.3.: Verwendung der dynamischen Programmierung zur Auffindung optimaler Stichpunkte

$$J_{M-1,j} = \min_{\nu=0}^{K-1} \omega_1 l_\nu^2 + \omega_2 e_\nu^2 + \mathcal{C}_h(e_\nu) \quad \forall j \in \{0, \dots, K-1\} \quad (4.3)$$

Vorausgesetzt, dass der Stichpunkt j von $\{L_{M-1}\}$ optimal ist, stellt $J_{M-1,j}$ die Kosten dar, die benötigt werden, um den Stichpunkt v von $\{L_M\}$ als Nachfolger auszuwählen. Die entsprechenden Indizes (j, v) müssen dann tabelliert und gespeichert werden, sowie die entsprechenden Kosten. Für die Schicht $\{L_{M-2}\}$ lassen sich auf gleicherweise die Fortsetzungskosten in der Form (4.4) definieren.

$$J_{M-2,j} = \min \sum_{v=0}^{K-1} \{\omega_1 l_v^2 + \omega_2 e_v^2 + \mathcal{C}_h(e_v) + J_{M-1,v}\} \quad \forall j \in \{0, \dots, K-1\} \quad (4.4)$$

Es wird auf die gleiche Weise bis zur Schicht $\{L_1\}$ weiter fortgesetzt, an der die Kostenfunktion die Form (4.5) hat.

$$J_{1,j} = \min \sum_{v=0}^{K-1} \{\omega_1 l_v^2 + \omega_2 e_v^2 + \mathcal{C}_h(e_v) + J_{2,v}\} \quad \forall j \in \{0, \dots, K-1\} \quad (4.5)$$

Die Zwischenlösungen sind die optimalen Teillösungen. Als letzter Schritt wird abhängig von der aktuellen Position des Fahrzeugs \mathbf{p}_0 (die schon als optimal gilt) die optimalen Gesamtkosten berechnet. Dies wird in Gleichung (4.6) dargestellt.

$$J_0 = \min \sum_{v=0}^{K-1} \{\omega_1 l_v^2 + \omega_2 e_v^2 + \mathcal{C}_h(e_v) + J_{1,v}\} \quad (4.6)$$

Die tabellierten Indizes können verwendet werden, um die optimalen Stichpunkte zu finden. Der Nachteil dieses Verfahrens liegt darin, dass zu viel Indizes und Fortsetzungskosten gespeichert werden müssen. Es ist offensichtlich, dass für jede Schicht K^2 Kostenfunktionen berechnet werden müssen. Daraus ergibt sich einen Aufwand in Höhe von $\mathcal{O}(MK^2)$. Eine Verfeinerung des Gitters (Erhöhung von M oder K) ist daher mit Kosten verbunden. Eine Maßnahme kann aber ergriffen werden, um den Aufwand zu verringern. Es kann festgestellt werden, dass in der Kostenfunktion lange Kanten mehr bestraft werden. Daher könnte man sich dafür entscheiden, nur eine bestimmte Anzahl $a < K$ von Nachfolgern zu untersuchen. Daraus sinkt der Aufwand in Höhe von $\mathcal{O}(MKa)$ ab. Abbildung 4.3b veranschaulicht diese Maßnahme, wobei nur die drei nachfolgenden Knoten untersucht werden. Es könnten auch unmittelbar nach der Erzeugung des Gitters die Knoten entfernt werden, die im Hindernis liegen.

4.3 Formulierung der MPC

Bei der Trajektorienplanung müssen viele Restriktionen eingehalten werden, denn die bestimmte Trajektorie muss befahrbar und hindernisfrei sein. Wie in Kapitel 1 schon erwähnt wurde, kommen unterschiedliche Verfahren bei dieser Aufgabenstellung zum Einsatz. Dazu zählen Graphsuchverfahren (A^* und Variante), kurvenbasierte Verfahren, Straßenkarte, Modell prädiktive Regelung, usw.. Das zuletzt genannte Verfahren hat einen großen Vorteil gegenüber den

anderen, da es die Restriktionen besser handhabt. Zudem wird die Dynamik des Systems berücksichtigt, was die Umsetzung erleichtert. Die MPC stellt aus diesem Grund den Hauptteil der Planung dar. Sie wird für die Berechnung der tatsächlichen Trajektorie des Fahrzeugs eingesetzt. Die Funktionsweise der MPC wurde im Abschnitt 2.3 ausführlich erklärt. An dieser Stelle wird es daher lediglich erklärt, wie das Verfahren bei der Trajektorienplanung eingesetzt wird. Die Grundidee besteht darin, das mit der Gleichung (3.5) beschriebene Fahrzeugmodell so zu regeln, dass es auf einer vorgegebenen Fahrbahn bleibt und sich bewegt. Durch Änderung ihres Zustandes wird sich das Fahrzeug in Richtung eines Zielpunkts bewegen. Diese Bewegung wird in mehreren Schritten solange durchgeführt, bis das Fahrzeug den Zielpunkt erreicht. Das Fahrzeugmodell stellt die kontinuierliche Dynamik¹ des Systems dar. Ein kontinuierliches OCP der Form (2.24) kann zur Regelung des Systems formuliert werden. In dieser Form wird die numerische Handhabung von Variablen sehr komplex sein, denn dieses Problem besitzt eine unendliche Dimension. Es wird daher durch das vorgestellte Mehrschießverfahren eine Diskretisierung des OCPs durchgeführt. Daraus ergibt sich ein NLP, welches in Gleichung (4.7) für einen Prädiktionshorizont N dargestellt wird.

$$\min_{\mathbf{X}, \mathbf{U}} \sum_{i=0}^{N-1} l_i(\mathbf{x}_i, \mathbf{u}_i) + l(\mathbf{x}_N) \quad (4.7a)$$

$$\text{u.B. } \mathbf{x}_{i+1} - \phi_i(\mathbf{x}_i, \mathbf{u}_i) = 0 \quad \forall i \in \{0, \dots, N-1\} \quad (4.7b)$$

$$\mathbf{x}_0 - \bar{\mathbf{x}}_0 = 0 \quad (4.7c)$$

$$\mathbf{h}(\mathbf{x}_N) = 0 \quad (4.7d)$$

$$\mathbf{r}_i(\mathbf{x}_i, \mathbf{u}_i) \leq 0 \quad \forall i \in \{0, \dots, N\} \quad (4.7e)$$

$$\text{Wobei } \mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} \quad \text{und} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N-1} \end{bmatrix} \quad (4.8)$$

\mathbf{X} und \mathbf{U} stellen die Menge aller Variablen dar, die bei dieser Aufgabe gesucht werden. Mit diesen Variablen lässt sich das nächste Verhalten des Systems beschreiben. Die Größen $\mathbf{x}_{0 \leq i \leq N}$ und $\mathbf{u}_{0 \leq i \leq N-1}$ sind jeweils die Knotenvariablen (ähnliche Bedeutung wie den künstlichen Anfangswert des AWP in (2.27)) und konstanten Stellgrößen. $\bar{\mathbf{x}}_0$ und \mathbf{x}_N sind jeweils der Anfangs- und Endzustand des Systems auf den Prädiktionshorizont N . Die Schießbedingung (4.7b) entsteht durch die Lösung einer nichtlinearen Dgl. der Form (2.27). Dieser Schritt wird in der Durchführung als Simulation bezeichnet. Dafür wird Algorithmus 1 verwendet, wobei \mathbf{x}_i als Variable und \mathbf{u}_i als konstante Stellgröße betrachtet werden. Die Gleichungen (4.7c) und (4.7d) sind die Anfangs- und Endbedingungen. Die Ungleichungsbeschränkungen und Pfadbeschränkungen lassen sich durch (4.7e) darstellen. Die Zielfunktion in Gleichung (4.7a) kann unterschiedlich modelliert werden.

¹ Wirkende Kräfte und Momente werden nicht berücksichtigt

Die Schrittweite des RKVs wird mit $h = \frac{T}{N\sigma}$ berechnet, wobei σ die Anzahl der Schritte pro Simulation darstellt. Da eine Änderung des Zeithorizonts oder Prädiktionshorizonts die Schrittweite des RKVs beeinflusst, werden diese Parameter in dieser Arbeit als konstant angenommen. Die Aufgabe wird zu jedem Zeitpunkt k formuliert, gelöst und der Horizont wird um einen Schritt $\Delta T = \frac{T}{N}$ nach vorne geschoben. Die zu verwendete Zielfunktion wird im nächsten Abschnitt eingeführt.

4.3.1 Auswahl der Zielfunktion

Die Zielfunktion spielt eine wichtige Rolle und muss daher sorgfältig formuliert werden. In der Trajektorienplanung sorgt man dafür, dass das Fahrzeug auf der Straße bleibt. Wie schon erwähnt wurde, wird die Mittellinie der Straße als Referenzweg(\mathcal{R}) angenommen. Das Fahrzeug soll daher auf \mathcal{R} bleiben, solange nichts im Weg steht. Der Komfortaspekt muss auch berücksichtigt werden. Durch sprunghafte Stellgrößen kann das Fahrzeug unerwünschte Stöße spüren. Die in dieser Arbeit verwendete Zielfunktion hat die Form (4.9).

$$l_i(\mathbf{x}_i, \mathbf{u}_i) = (\mathbf{x}_i - \mathbf{C}^T \mathbf{x}_{ziel})^T \mathbf{Q} (\mathbf{x}_i - \mathbf{C}^T \mathbf{x}_{ziel}) \quad (4.9a)$$

$$+ q_d D(x_i, y_i, \mathcal{R})^2 \quad (4.9b)$$

$$+ q_\phi \phi(x_i, y_i, \theta_i, \mathcal{R})^2 \quad (4.9c)$$

$$+ \mathbf{u}_i^T \mathbf{R}_1 \mathbf{u}_i + (\mathbf{u}_{i+1} - \mathbf{u}_i)^T \mathbf{R}_2 (\mathbf{u}_{i+1} - \mathbf{u}_i) \quad (4.9d)$$

$$l(\mathbf{x}_N) = q_\theta (\theta_N - \theta_d)^2 \quad (4.9e)$$

Diese allgemeine Zielfunktion besteht aus unterschiedlichen Teilen, die je nach geplanter Bewegung verwendet werden können. Sie lassen sich wie folgt beschreiben:

- In der Optimierungsaufgabe sorgt (4.9a) dafür, dass das System das Ziel erreicht. Des Weiteren wird dadurch eine Wunschgeschwindigkeit eingehalten.
- (4.9b) ist der Abstand zwischen dem aktuellen Position des Fahrzeugs $[x_i, y_i]^T$ und \mathcal{R} . Dieser Term kann verwendet werden, um das Fahrzeug auf \mathcal{R} zu halten.
- Der Term in (4.9c) beschreibt die lokale Abweichung zwischen \mathcal{R} und der aktuellen Fahrtrichtung. Die Minimierung dieses Terms ermöglicht, dass sich das Fahrzeug an die Kontur von \mathcal{R} anpasst. Dabei kann die Tangentengleichung verwendet werden, wie es im Abschnitt A.3 erklärt wird.
- Gleichung (4.9d) sorgt dafür, dass der Energieverbrauch klein bleibt. Durch die Minimierung dieses Terms werden die Stellgrößen klein gehalten. Weiterhin werden sprunghafte Änderungen der Stellgrößen vermieden.
- Der Meyer-Term (4.9d) wird verwendet, damit das Fahrzeug bei der Knotenvariable θ_N einen gewünschten Drehwinkel einhält. Es kann in vielen Situationen sehr hilfreich sein.

q_θ, q_ϕ, q_d sind frei wählbare Parameter. Die Matrix $\mathbf{Q} \in \mathbb{R}^{5 \times 5}$ wird positiv semidefinit. $\mathbf{R}_1 \in \mathbb{R}^{2 \times 2}$ und $\mathbf{R}_2 \in \mathbb{R}^{2 \times 2}$ werden positiv definit ausgewählt.

4.3.2 Kollisionsvermeidung

Zur Sicherstellung der Hindernisvermeidung wird die Ungleichung (3.15) verwendet. Da ein Fahrzeug in dieser Arbeit rechteckförmig dargestellt wird, kann eine solche Ungleichung verwendet werden, um alle Punkte des Raums zu beschreiben, die im Fahrzeug liegen. Daraus folgt, dass ein Punkt des Raums \mathbf{z} zu jedem Knoten i des Prädiktionshorizonts gleichzeitig in einem Hindernis und im Fahrzeug liegen kann, nur dann wenn Ungleichung (4.10) erfüllt wird.

$$\begin{bmatrix} \mathbf{A}_{f,i} \\ \mathbf{A}_h \end{bmatrix} \mathbf{z} \leq \begin{bmatrix} \mathbf{b}_{f,i} \\ \mathbf{b}_h \end{bmatrix} \quad (4.10)$$

Zum Zweck der Hindernisvermeidung müssen alle Bereiche des Raums als unzulässig angenommen werden, wo die Ungleichung (4.10) erfüllt wird, damit sich ein Fahrzeug und ein Hindernis nicht kollidieren. Eine Bedingung, die ein unzulässiger Bereich darstellt, ist leider für die Optimierungsaufgabe(bzw. MPC) nicht geeignet[2]. Laut [23] gibt es glücklicherweise in der Dualitätstheorie für Optimierungsaufgaben das Farkas Lemma 1, das die Bedingung (4.10) umformuliert.

Lemma 1 Gegeben seien die Matrizen und Vektoren $\mathbf{A}_1 \in \mathbb{R}^{4 \times 2}$, $\mathbf{A}_2 \in \mathbb{R}^{4 \times 2}$, $\mathbf{b}_1 \in \mathbb{R}^{4 \times 1}$ und $\mathbf{b}_2 \in \mathbb{R}^{4 \times 1}$

Für alle Punkte des Raums $\mathbf{z} \in \mathbb{R}$ besitzt Gleichung

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{z} \leq \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

keine Lösung, nur wenn es ein positiver Vektor $\beta \in \mathbb{R}^{8 \times 1}$ gibt, sodass die folgenden Bedingungen erfüllt sind:

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}^T \beta = 0 \quad \text{und} \quad \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}^T \beta < 0$$

Für eine Verkehrssituation, in der q Hindernisse vermieden werden müssen, lässt sich durch Einhaltung der Bedingungen (4.11) die Kollisionsvermeidung erreichen.

$$\begin{bmatrix} \mathbf{A}_{f,i}^T \\ \mathbf{A}_{h,j}^T \end{bmatrix} \beta_{j,i} = 0 \quad \forall j \in \{1, \dots, q\} \quad (4.11a)$$

$$\begin{bmatrix} \mathbf{b}_{f,i}^T \\ \mathbf{b}_{h,j}^T \end{bmatrix} \beta_{j,i} < 0 \quad \forall j \in \{1, \dots, q\} \quad (4.11b)$$

$$\beta_{j,i} \geq 0 \quad \forall j \in \{1, \dots, q\} \quad (4.11c)$$

Es muss beachtet werden, dass Hindernisse statisch sind. Jedes Hindernis j lässt sich einzeln durch die konstanten Größen $\mathbf{A}_{h,j}$ und $\mathbf{b}_{h,j}$ beschreiben. Die genaue Struktur dieser Größen ist aus (3.15) zu entnehmen. Im Gegenteil dazu bewegt sich das Fahrzeug. Die Darstellung des Fahrzeugs wird bei jedem Knoten i des Horizonts durch veränderbare Größen $\mathbf{A}_{f,i}$ und $\mathbf{b}_{f,i}$ erfolgen, welche die Form (4.12) und (4.13) besitzen.

$$\mathbf{A}_{f,i} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\cos(\theta_i) & -\sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \\ \sin(\theta_i) & -\cos(\theta_i) \end{bmatrix} \quad \mathbf{b}_0 = \frac{1}{2} \begin{bmatrix} L_f \\ L_f \\ B_f \\ B_f \end{bmatrix} \quad (4.12)$$

$$\mathbf{b}_{f,i} = \mathbf{b}_0 + \mathbf{A}_{f,i} \mathbf{x}_i \quad (4.13)$$

Der Vektor

$$\boldsymbol{\beta}_i = \begin{bmatrix} \beta_{1,i} \\ \vdots \\ \beta_{q,i} \end{bmatrix} \in \mathbb{R}^{8q \times 1}$$

stellt neue Variablen dar und muss daher als weitere Optimierungsvariablen im OCP berücksichtigt werden. Aus diesem Grund entstehen die erweiterten Knotenvariablen \mathbf{x}_e in (4.14)

$$\mathbf{x}_{e,i} = \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\beta}_i \end{bmatrix} \in \mathbb{R}^{(5+8q)} \quad \Rightarrow \quad \mathbf{X}_e = \begin{bmatrix} \mathbf{x}_{e,0} \\ \vdots \\ \mathbf{x}_{e,N} \end{bmatrix} \quad (4.14)$$

4.3.3 Systembeschränkungen

Um der Realität nah zu bleiben werden einige Beschränkungen für das System festgelegt. Wie jedes andere System muss die Stellgrößenbeschränkung gefordert werden. Ungleichungen der Form (4.15) müssen eingehalten werden, um die Beschleunigungen zu beschränken.

$$\mathbf{u}_i \geq -\mathbf{u}_{max} \quad \text{und} \quad \mathbf{u}_i \leq \mathbf{u}_{max} \quad (4.15)$$

Die Wunschgeschwindigkeit muss unter einer vorgeschriebenen Höchstgeschwindigkeit bleiben. Dafür müssen Ungleichungen der Form (4.16) eingeführt werden.

$$v_i \geq 0 \quad \text{und} \quad v_i \leq v_{max} \quad (4.16)$$

Es gibt aber keine Garantie, dass das Fahrzeug die Fahrbahn nicht verlässt. Für eine Geradeausfahrt werden Ungleichung der Form (4.17) berücksichtigt.

$$y_i \geq -y_{max} \quad \text{und} \quad y_i \leq y_{max} \quad (4.17)$$

Für eine Kurvenfahrt wird der Parameter q_d groß ausgewählt. Dies vermeidet, dass sich das Fahrzeug weit von \mathcal{R} entfernt. Abhängig von der Planung sollten alle Variablen \mathbf{X} und \mathbf{U} beschränkt sein.

4.4 Regelungsentwurf

Die Zielfunktion (4.9a) wird abhängig von einem Zielvektor \mathbf{x}_{ziel} dargestellt. In diesem Abschnitt wird daher erklärt, wie dieser Vektor ermittelt wird. Dabei werden vor allem zwei Faktoren berücksichtigt:

- Ein Zielpunkt muss im Sichtbereich liegen
- Die Rechenzeit muss gesteuert werden

4.4.1 Zielpunkt im Sichtbereich

In der Planung werden zunächst viele mögliche Zielpunkte berechnet. Um den letztendlichen Zielpunkt zu ermitteln, wird der Sichtbereich des Fahrzeugs gesucht. Hiermit werden alle möglichen Zielpunkte, die außerhalb dieses Bereichs liegen ausgeschlossen. In Abbildung 4.4 wird eine Verkehrssituation dargestellt, wo die Sichtweite des Fahrzeugs zum großen Teil durch ein Hindernis verdeckt wird. Eine gute Planung wäre in dieser Situation lediglich in Richtung des dargestellten Sichtbereichs möglich.

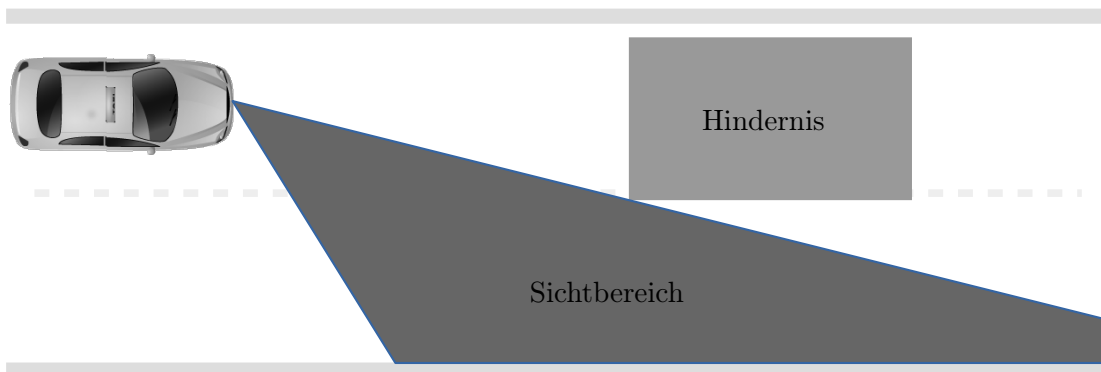


Abbildung 4.4.: Darstellung des Sichtbereichs in einer Verkehrssituation

Ein in dem Sichtbereich ausgewählter Zielpunkt trägt bei, dass die Planung dahin führt, wo man auch sieht.

4.4.2 Zeitressource

Ein weiterer Faktor, der bei der Ermittlung des Zielpunkts berücksichtigt wird, ist die Zeitressource. Damit ist die Zeit gemeint, die bei der Lösung eines OCPs gebraucht wird. Die Rechenzeit hat bei der Trajektorienplanung eine entscheidende Bedeutung, denn die Echtzeitfähigkeit des verwendeten Algorithmus hängt davon ab. Die Nutzung der Zeitressource muss daher geregelt werden. Grundsätzlich wird dabei eine Sollrechenzeit t_{soll} festgelegt. Wenn bei der Lösung eines OCPs diese Sollzeit überschritten wird(bzw. wenn die Planung viel Zeit kostet), muss der

nächste Zielpunkt in Richtung des Fahrzeugs verschoben werden. Das bedeutet, dass der nächste gewählte Zielpunkt näher am Fahrzeug liegen wird. Im Gegenteil dazu wird der Zielpunkt weiter weg ausgewählt, wenn die Rechenzeit weniger als t_{soll} beträgt. Jedes weitere Hindernis, das durch die Verschiebung des Zielpunkts berücksichtigt wird, kostet zusätzliche Rechenzeit, denn $8N$ weitere Variablen werden der Optimierung hinzugefügt. Dies lässt sich durch Gleichung (4.14) verdeutlichen. Wenn ein Hindernis nicht mehr berücksichtigt wird, sinkt ebenfalls die Anzahl von Variablen um $8N$.

Die Regelung besteht lediglich darin, abhängig davon wie lange die Lösung des OCPs dauert, ein Zielpunkt aus den Möglichen auszuwählen, der entweder weit oder nah am Fahrzeug liegt. Anders gesagt, wird es eine Ganzzahl(bzw. Indexzahl) n gesucht, die bestimmt, welcher von den M möglichen Zielpunkten verwendet wird. Ungleichung (4.18) muss daher erfüllt werden.

$$1 \leq n \leq M \quad (4.18)$$

Zur Ermittlung dieser Indexzahl kann die Zeitabweichung als Skalierungsfaktor oder eine Fermi-Funktion der Form (4.19) verwendet werden. Die letzte Variante bietet eine bessere Möglichkeit, denn ihre Steigung lässt sich durch einen Parameter a ändern. Dadurch kann der Bereich definiert werden, in dem die Zeitabweichung berücksichtigt wird.

$$f = \frac{1}{1 + \exp(-\frac{\Delta t}{a})} \implies 0 \leq f \leq 1 \quad (4.19)$$

Der Parameter a ist frei zu wählen und muss die gleiche Einheit wie Δt haben. Zu jedem Zeitpunkt wird die Regeldifferenz mit $\Delta t_k = t_{soll} - t_{solver}$ bezeichnet. Eine Stellgeschwindigkeit lässt sich in der Form (4.20) darstellen. Diese Form hat den Vorteil, dass die vergangenen Zeitabweichungen auch berücksichtigt werden können und das Verhalten eines PID-Reglers dadurch erreicht werden kann.

$$\Delta t = b_0 \Delta t_k + b_1 \Delta t_{k-1} + b_2 \Delta t_{k-2} \quad (4.20)$$

Die Parameter b_0 , b_1 und b_2 lassen sich wiederum frei wählen. Die Indexzahl n lässt sich letztendlich in (4.21) berechnen.

$$n = \begin{cases} \lceil Mf \rceil \\ 1 & \text{wenn } f = 0 \end{cases} \quad (4.21)$$

Die Wunschgeschwindigkeit in (4.22) wird als Bruchteil von der Höchstgeschwindigkeit v_{max} berechnet. Dies führt dazu, dass das Fahrzeug langsamer wird, wenn der Zielpunkt nah liegt.

$$v_{ziel} = \frac{n}{M} v_{max} \quad (4.22)$$

$$\theta_{ziel} = \text{atan2}(y_{ziel}, x_{ziel}) \quad (4.23)$$

Um eine Drehung des Fahrzeugs in Richtung des Zielpunktes zu erhalten, wird der Drehwinkel durch (4.23) ermittelt.

4.5 Kurzfassung

Die einzelnen Komponenten des Planungskonzepts wurden in obigen Abschnitten ausführlich erklärt. Die Entwurfparameter und ein Algorithmus Zur Durchführung der Planung werden in Kapitel C zusammengefasst. In diesem Abschnitt wird daher nur eine Zusammenfassung des Planungskonzepts dargestellt. Auf einem Horizont der Länge N werden für jeden Knoten i die Variablen(bzw. Knotenvariablen) $\mathbf{x}_{e,i}$ ((4.14)) erzeugt. Zwischen je zwei Knoten werden die Variablen \mathbf{u}_i erzeugt. Insgesamt wird es $(5+8q)N+2(N-1)$ Variablen für die Optimierung geben, wobei q die Anzahl von Hindernissen darstellt. Alle anderen Gleichungen und Ungleichungen werden abhängig von diesen Variablen definiert. In dieser Arbeit wird für jede Variable und jede Restriktion eine obere und eine untere Grenze geben. Die Anwendung von Algorithmus 1 wird als $\mathcal{S}_{im}(\cdot)$ bezeichnet. Zur Steuerung des Systems werden die berechneten Stellgrößen an das System übertragen, um den nächsten Zustand des Systems zu ermitteln. Dieser Schritt kann durch eine normale Integralrechnung erfolgen. Er wird als $\mathcal{J}_{in}(\cdot)$ bezeichnet. $\mathcal{J}_{in}(\cdot)$ und $\mathcal{S}_{im}(\cdot)$ sind beide Integrationsverfahren und unterscheiden sich lediglich von der Tatsache, dass $\mathcal{J}_{in}(\cdot)$ bekannte Parameter verwendet, während das zweite Verfahren die Optimierungsvariablen verwendet.

Die Ungleichungen (4.11b) werden in (4.24d) umgewandelt, wobei $\epsilon > 0$ einen sehr kleinen Wert nimmt und als obere Grenze von (4.24d) gesetzt wird. Alle benötigten Gleichungen- und Ungleichungsbeschränkungen werden in (4.24) zusammengefasst.

$$\mathbf{x}_{i+1} - \mathcal{S}_{im}(\mathbf{x}_i, \mathbf{u}_i) = 0 \quad (4.24a)$$

$$\mathbf{x}_0 - \bar{\mathbf{x}}_0 = 0 \quad (4.24b)$$

$$\begin{bmatrix} \mathbf{A}_{f,i}^T \\ \mathbf{A}_{h,j}^T \end{bmatrix} \beta_{j,i} = 0 \quad \forall j \in \{1, \dots, q\} \quad (4.24c)$$

$$\begin{bmatrix} \mathbf{b}_{f,i}^T \\ \mathbf{b}_{h,j}^T \end{bmatrix} \beta_{j,i} \leq -\epsilon \quad \forall j \in \{1, \dots, q\} \quad (4.24d)$$

$$\beta_{j,i} \geq 0 \quad \forall j \in \{1, \dots, q\} \quad (4.24e)$$

$$y_i \geq -y_{max} \quad \text{und} \quad y_i \leq y_{max} \quad (4.24f)$$

$$v_i \geq 0 \quad \text{und} \quad v_i \leq v_{max} \quad (4.24g)$$

$$\mathbf{u}_i \geq -\mathbf{u}_{max} \quad \text{und} \quad \mathbf{u}_i \leq \mathbf{u}_{max} \quad (4.24h)$$

Mit den ausgewählten Entwurfparametern und der aktuellen Position des Fahrzeugs wird der Zielvektor berechnet. Zusammen mit der Zielfunktion in (4.9) wird ein OCP formuliert. Nach der Lösung des OCPs wird die Integralrechnung $\mathcal{J}_{in}(\bar{\mathbf{x}}_0, \mathbf{u}_0^*)$ durchgeführt. Daraus ergibt sich der neue Zustand des Fahrzeugs $\bar{\mathbf{x}}_0$. Die gleichen Schritte werden solange wiederholt, bis das Fahrzeug am Zielpunkt eintrifft. Der Referenzweg \mathcal{R} wird dabei als Polynom geeigneter Ordnung angenommen, denn Polynome beschreiben die Konturen einer Fahrbahn recht gut. Eine Beschreibung von Polynomfunktionen ist im Abschnitt A.1 erhältlich.

5 Evaluation

In der Umsetzung wurde für das vorgestellte Fahrzeugmodell eine Strategie zur Trajektorienplanung theoretisch beschrieben. An dieser Stelle ist soweit klar, wie die Planung durchgeführt wird. Zur Evaluierung des Konzepts wird in diesem Kapitel an unterschiedlichen Beispielen getestet, wie gut die Umsetzung funktioniert.

5.1 Verwendete Werkzeuge

Prinzipiell wurden zwei Verfahren zum Aufbau des Planungskonzepts kombiniert, die sich unterschiedlich implementieren lassen. Die Zielpunktgenerierung lässt sich direkt als ein allein stehendes Programm schreiben. Im Gegenteil dazu wird bei der MPC eine Optimierungsaufgabe behandelt. Bei dieser Aufgabe wird basierend auf Drittanbieterprogramme ein kontinuierliches OCP zuerst parametrisiert (Überführung in ein NLP) und danach gelöst. Die Literatur bietet Werkzeuge, wie z. B. CVX[24] und IPOPT[20] zur Lösung von Optimierungsaufgaben an. Der Open Source Solver IPOPT eignet sich besonders für die Lösung nichtlinearer und nicht konvexer Optimierungsaufgaben. Er wird daher in dieser Arbeit verwendet, weil das zu lösende OCP aufgrund des nichtlinearen Fahrzeugmodells und der Restriktionen eine nicht konvexe und nichtlineare Optimierung darstellt. Zur Diskretisierung des kontinuierlichen OCPs kommt das Werkzeug CasADi[25] zum Einsatz. Damit werden alle zur Lösung erforderlichen Größen (wie z. B. die Ableitung von Nebenbedingungen, Implementierung von $\mathcal{S}_{im}(\cdot)$ und $\mathcal{J}_{in}(\cdot)$) berechnet. Alle Verfahren werden auf einem Rechner mit Core i5 – 4200M CPU 8GB RAM und in der Programmiersprache Python implementiert.

Um die Graphiken übersichtlicher und unterscheidbar zu zeichnen, werden folgende Farben und

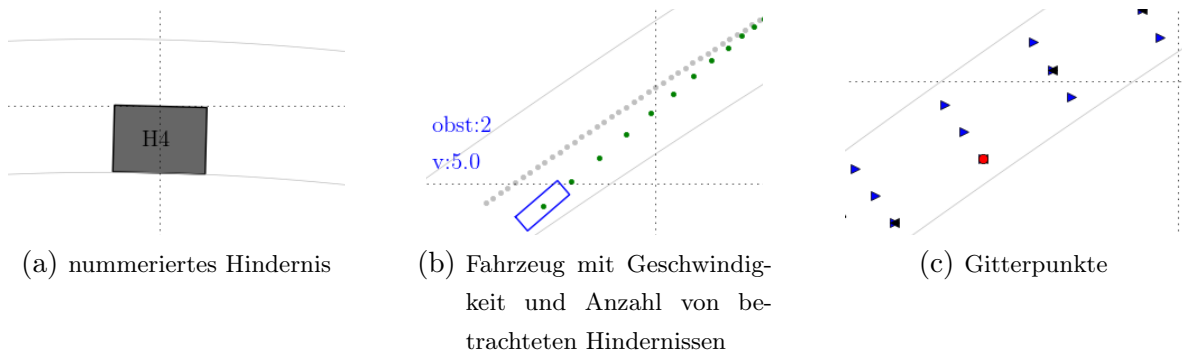


Abbildung 5.1.: Darstellung von Graphikelementen.

Formen verwendet, wie es in Abbildung 5.1 zu sehen ist:

- Hindernisse werden als graue Vierecke dargestellt (Abbildung 5.1a).
- Das Fahrzeug wird rechteckig und blau gezeichnet, denn damit wird recht gut ein Fahrzeug abgebildet. Die aktuelle Geschwindigkeit und die Anzahl von Hindernissen werden links oben angezeigt (Abbildung 5.1b). Es wird nur die Anzahl von Hindernissen gezeigt, die gerade in der MPC berücksichtigt werden. Die aktuellen berechneten Knotenvariablen des Systems werden durch grüne Punkte auf der Fahrbahn markiert. \mathcal{R} wird durch eine graue punktierte Kurve gezeichnet.
- Die Stichpunkte werden blau auf der Fahrbahn dargestellt. Die optimalen Punkte werden schwarz markiert. Der letztendliche Zielpunkt wird rot dargestellt. Abbildung 5.1c zeigt, wie das Gitter aussieht.

In unterschiedlichen Szenarien wird das Planungskonzept evaluiert. Zunächst wird eine Planung auf einer relativ kurzen Strecke durchgeführt. Dabei wird überprüft, wie gut die einzelnen Komponenten ihre Aufgaben erledigen können. Anschließend wird das Gesamtkonzept für eine lange Strecke getestet und bewertet. Schließlich wird die bei der Planung benötigte Rechenzeit analysiert.

5.2 Evaluierung der Generierung von Zielpunkten

Bei der Trajektorienplanung wird die Fahrbahn aufgrund der Suche nach befahrbaren Punkten diskretisiert. Dadurch wird ein Gitter erzeugt, womit optimale Punkte bestimmt und zur Regelung weitergegeben werden. Dabei wurde ein Verfahren beschrieben, das nun getestet wird. Die Fahrbahn wird mit einem Gitter belegt, das aus 10 Schichten von jeweils 3 Untersuchungs-

Tabelle 5.1.: Verwendete Parameter zur Zielpunktgenerierung

M	K	ω_1	ω_2	Δs	Δl	s_{max}
10	3	0.5	1	3	0.9	30

punkten besteht. Da das verwendete Fahrzeug ungefähr 0.5 m breit ist, reicht dieses Gitter zur Abdeckung der Fahrbahn aus. Alle anderen verwendeten Parameter sind aus Tabelle 5.1 zu entnehmen. Abbildung 5.2 zeigt, wie die Fahrbahn während der Fahrt untersucht wird. In Abbildung 5.2a wird die Straße durch ein Gitter belegt. Abhängig von der Position des Fahrzeugs werden optimale Stichpunkte ausgesucht. Abbildung 5.2b zeigt eine Verkehrssituation, in der die linke Seite der Fahrbahn durch ein Hindernis gesperrt ist. Da sich das Fahrzeug auf \mathcal{R} befindet, liegen die optimalen Gitterpunkte auf \mathcal{R} bis auf der Höhe des Hindernisses, wo sie auf der rechten Straßenseite liegen. In Abbildung 5.2c wird gleich festgestellt, dass die rechte Straßenseite auch gesperrt ist. Die optimalen Gitterpunkte liegen um die Hindernisse, auf beiden Seiten und auf dem Mittelstreifen (\mathcal{R}) der Straße. Etwas später werden die optimalen Punkte in dem freien Bereich so ausgewählt, wie es in Abbildung 5.2d zu sehen ist. Es ist nun offensichtlich, dass alle optimalen Bezugspunkte in hindernisfreien Bereichen liegen. Die Generierung von

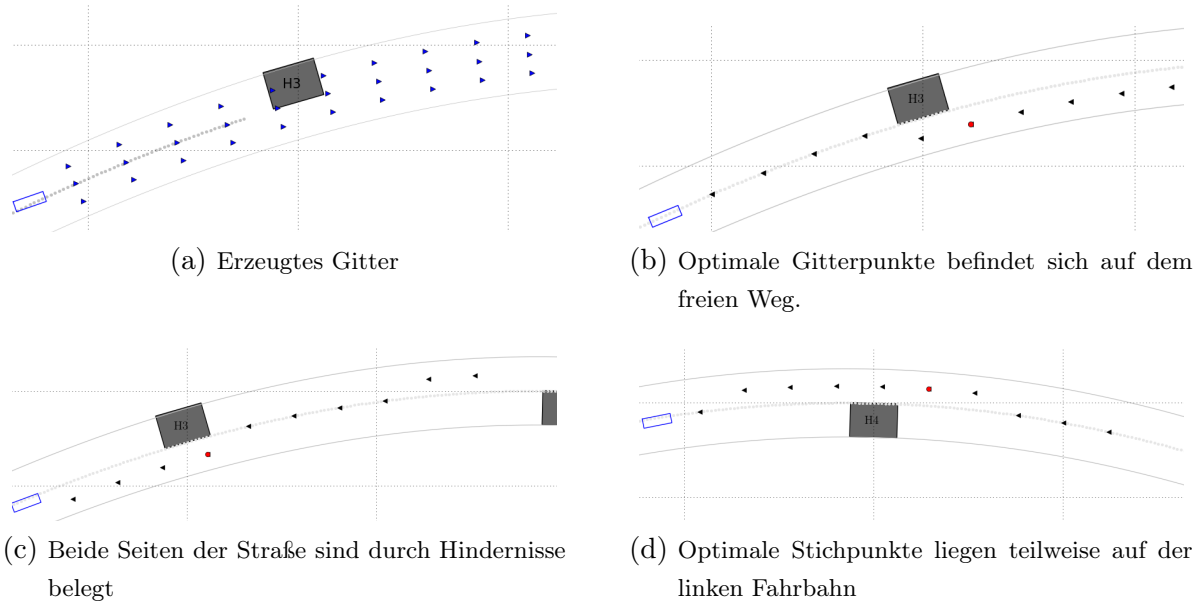


Abbildung 5.2.: Generierung von Zielpunkten. Die optimalen Punkte werden lediglich in dem freien Bereich ausgewählt.

Zielpunkten funktioniert daher wie erwartet. Einer der optimalen Stichpunkte wird ausgewählt, um die tatsächliche Planung durchzuführen.

5.3 Bewertung der MPC

In dem vorgestellten Planungskonzept wird die kollisionsfreie und fahrbare Trajektorie mittels MPC berechnet. In diesem Abschnitt wird auf einer kurzen Strecke getestet, ob die MPC in der Lage ist, diese Aufgabe zu lösen. Zur Einstellung der MPC wurden die Parameter aus Tabelle 5.2 entnommen. Der Referenzweg \mathcal{R} wird als Polynom vierter Ordnung verwendet. Der Zielpunkt wird dabei willkürlich vorgegeben. In Abbildung 5.3 wird die Trajektorie des Fahrzeugs in

Tabelle 5.2.: Verwendete Parameter zur Einstellung der MPC

\mathbf{Q}	$\mathbf{R}_1 = \mathbf{R}_2$	v_{max}	q_θ	q_ϕ	q_d	\mathbf{u}_{max}	y_{max}	N	T	σ
$\begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	5	5	50	1	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	2	15	3	1

Anwesenheit eines Hindernisses berechnet. Das Fahrzeug nähert sich dem Hindernis an, verlässt den Mittelstreifen der Straße (Abbildung 5.3a und 5.3b), um auf dem freien Weg zu fahren. Es fährt am Hindernis vorbei (Abbildung 5.3c) und fährt wieder auf den Mittelstreifen (Abbildung 5.3d). Die in jedem Schritt geplante Trajektorie (grüne Punkte) zeigt einen hindernisfreien Weg.

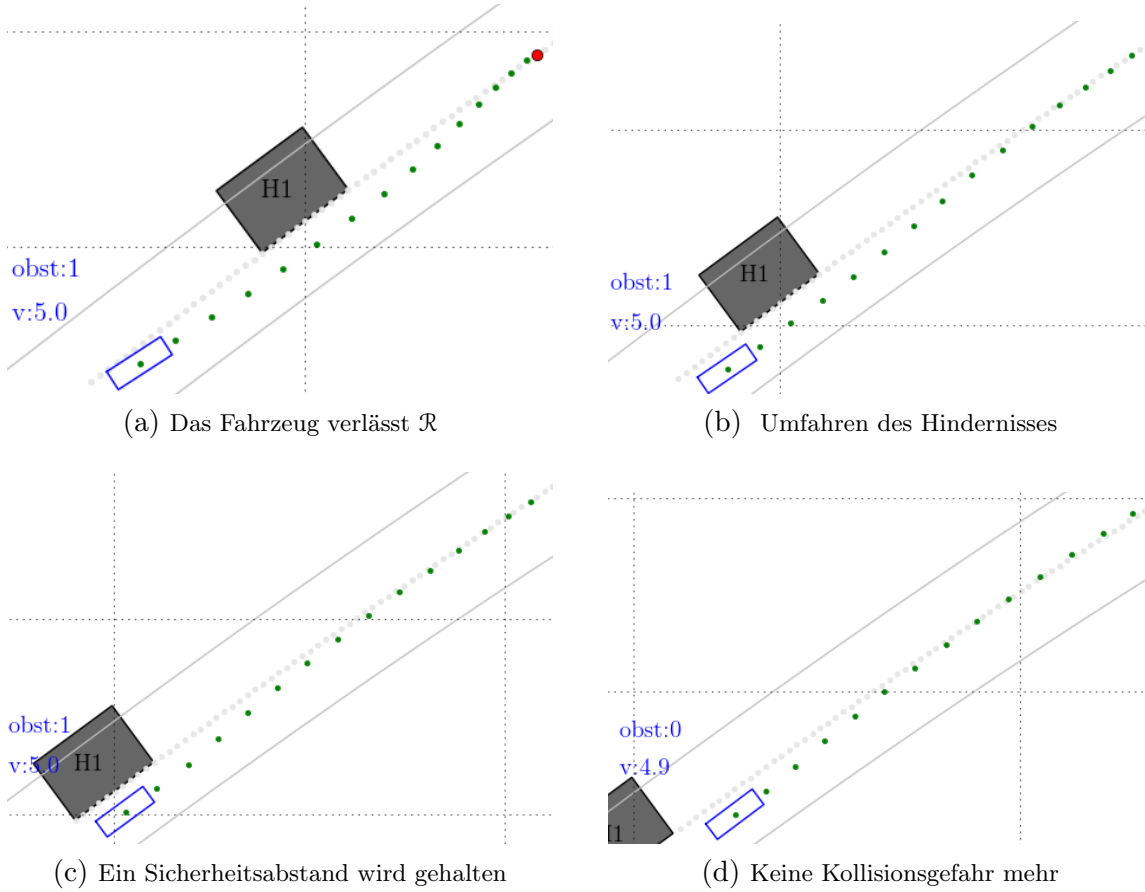
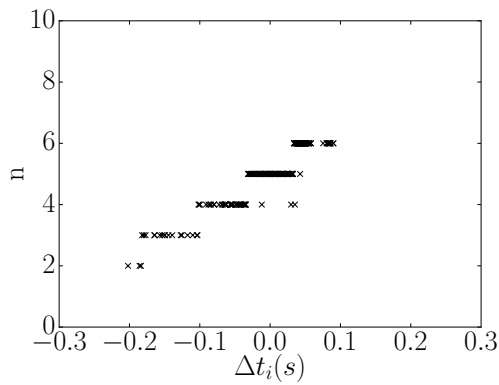


Abbildung 5.3.: Umfahren eines Hindernisses, das Fahrzeug verlässt \mathcal{R} , fährt am Hindernis vorbei und fährt wieder auf \mathcal{R} .

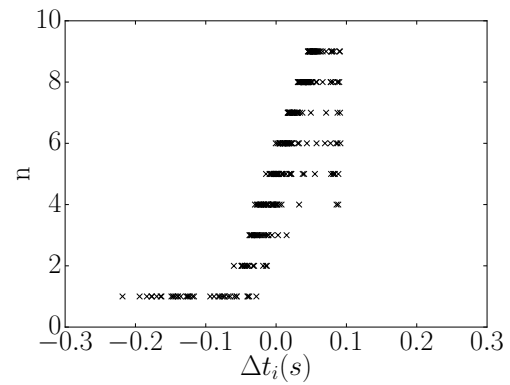
Durch die angezeigte aktuelle Geschwindigkeit des Fahrzeugs wird es erkannt, dass die vorgegebene Höchstgeschwindigkeit (v_{max}) nicht überschritten wird. Das Fahrzeug bleibt dabei auf der Fahrbahn. Durch das durchgeführte Szenario wird gezeigt, dass die MPC die Trajektorie des Fahrzeugs berechnen kann.

5.4 Ermittlung eines adaptiven Zielpunkts

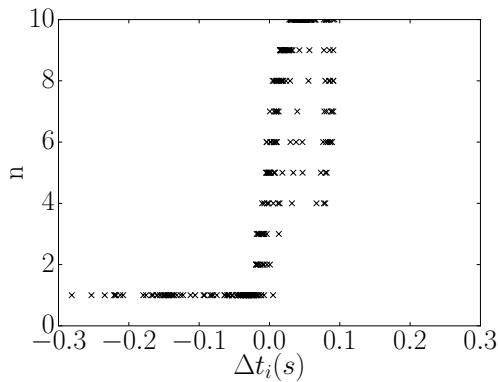
Zum Testen des Regelungssystems wird eine Sollrechenzeit von $t_{soll} = 0.1$ festgelegt. Wie in Abbildung 5.2b, 5.2c und 5.2d eindeutig zu sehen ist, verschiebt sich der Zielpunkt während der Fahrt. Diese Verschiebung ist abhängig von den Parametern b_0 , b_1 , b_2 und a , womit die Regeldifferenz definiert wird. Sie können zwar frei gewählt werden, aber es wurde festgestellt, dass eine sprunghafte Verschiebung entsteht, wenn die Parameter nicht gut ausgewählt wurden. Genau aus diesem Grund müssen für jede Teststrecke die Parameter angepasst werden. Um diese Feststellung zu erläutern, wird die gleiche Strecke mit unterschiedlichen Parametern vier mal gefahren. Daraus lassen sich die ermittelten Indizes (n) abhängig von der Regeldifferenz Δt_i graphisch darstellen. Abbildung 5.4 zeigt die Verteilung von Indizes. Daraus wird ein treppenförmiger Verlauf erkannt, was auch zu erwarten war. In Abbildung 5.4d überschneiden sich die



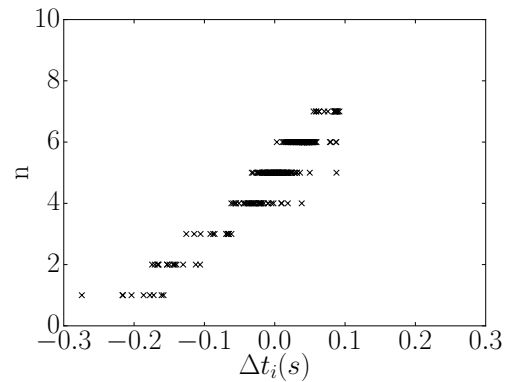
(a) $b_0 = 6, b_1 = 0, b_2 = 0$ und $a = 1$



(b) $b_0 = 6, b_1 = 0.5, b_2 = 1$ und $a = 0.2$



(c) $b_0 = 100, b_1 = 10, b_2 = 1$ und $a = 1$



(d) $b_0 = 10, b_1 = 1, b_2 = -1$ und $a = 1$

Abbildung 5.4.: Verteilung von Indizes des Zielpunkts bezüglich Regeldifferenz Δt_i

einzelnen Stufen. Die Überschneidung ist in Abbildung 5.4c und 5.4b noch stärker zu sehen. Im Gegenteil dazu ist in Abbildung 5.4a nur eine leichte Überschneidung gesehen. Zudem sollte der genaue Zielpunkt nicht zu weit und nicht zu nah am Fahrzeug liegen. Wenn der Zielpunkt zu weit liegt, besteht die Möglichkeit, dass zusätzliche Hindernisse berücksichtigt werden müssen. Wenn er zu nah am Fahrzeug liegt, wird sich das Fahrzeug kaum bewegen. Die frei wählbaren Parameter b_0, b_1, b_2 und a werden daher so gewählt, dass die Indizes ungefähr wie in Abbildung 5.4a aussehen.

5.5 Evaluierung des Gesamtkonzeptes

Zur Evaluierung des Gesamtkonzeptes wird das Fahrzeug auf einer langen Teststrecke gefahren. Die dabei zu realisierenden Bewegungen bestehen aus:

- **Spurhalten:**
Es wird überprüft, wie gut das Fahrzeug auf einer vordefinierten Fahrbahn bleiben kann.
- **Hindernisvermeidung:**
Es wird erwartet, dass das Planungskonzept eine hindernisfreie Trajektorie bestimmt.

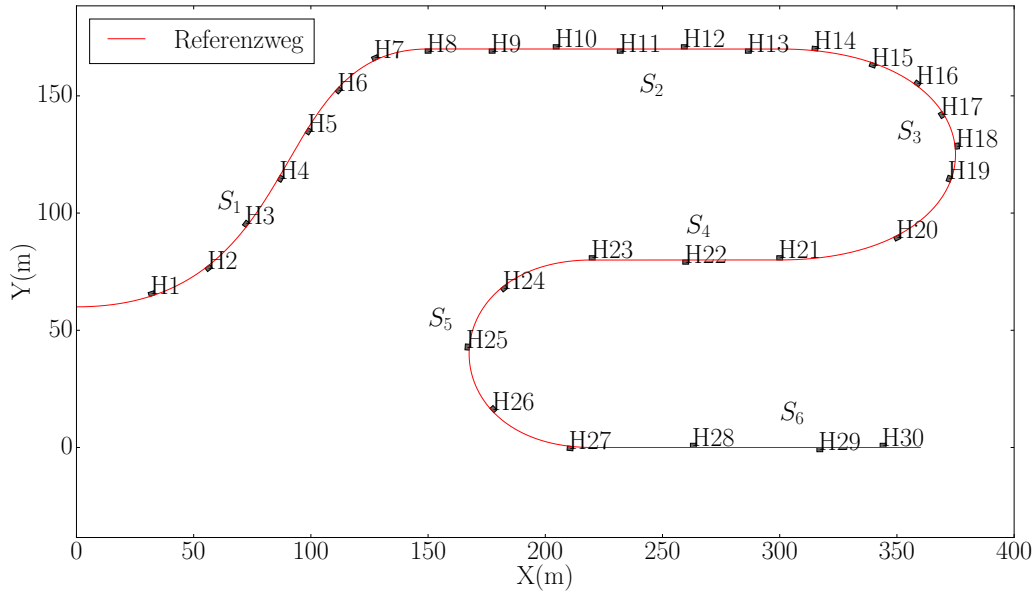


Abbildung 5.5.: Übersicht über die Teststrecke und die Hindernisse

Die Teststrecke ist ca. 905 m lang und 2 m breit. Es werden abwechselnd Geradeausfahrt und Kurvenfahrt durchgeführt. Die Teststrecke besteht deshalb aus den Streckenabschnitten S_1 , S_2 , S_3 , S_4 , S_5 und S_6 , die nacheinander gefahren werden. In Abbildung 5.5 werden die Gesamtstrecke und die Hindernisse dargestellt. Die Kurvenabschnitte S_1 , S_3 und S_5 sind kubische Bezier-Kurven. Insgesamt werden 30 Hindernisse auf der Fahrbahn platziert. Um den Lösungsraum weiter zu beschränken, werden abwechselnd die linke und die rechte Fahrbahnseite durch Hindernisse gesperrt. Das Fahrzeug wird daher lediglich eine Möglichkeit haben, ein Hindernis zu überwinden. Die Streckenabschnitte mit unterschiedlichen Verläufen werden zuerst einzeln gefahren und anschließend wird die komplette Strecke gefahren. Es werden die gleichen Parameter verwendet, die beim Testen einzelner Komponenten gebraucht wurden ($b_0 = 6$, $b_1 = 0$, $b_2 = 0$, $a = 1$, Tabelle 5.2 und 5.1).

5.5.1 Geradeausfahrt

Bei einer Strecke mit weniger Krümmung lässt sich eine Geradeausfahrt testen. Die Streckenabschnitte S_1 , S_2 , S_4 und S_6 eignen sich gut für eine Geradeausfahrt. Da S_2 , S_4 und S_6 ähnliche Verläufe haben, wird für eine Geradeausfahrt lediglich S_1 und S_2 getestet.

5.5.1.1 Abschnitt S_1

Die Teststrecke S_1 weist eine leichte Krümmung am Anfang und am Ende auf. Sie wird von unten nach oben gefahren. Dabei sind insgesamt acht Hindernisse zu vermeiden. Abbildung 5.6 zeigt die globalen Koordinaten (X und Y) des Fahrzeugs während der Fahrt. Wie erwartet,

bleibt das Fahrzeug auf der Fahrbahn und vermeidet alle Hindernisse. Es wird aber festgestellt, dass das Fahrzeug manchmal zu weit am Straßenrand entlang fährt. Dies geschieht in der Höhe von H2 und H7, wo die Strecke leichte Krümmung aufweist. Die Geschwindigkeitsprofile werden in Abbildung 5.7 veranschaulicht. Das Fahrzeug fährt mit Null-Geschwindigkeit (lineare Geschwindigkeit) ab, beschleunigt bis zur Höchstgeschwindigkeit und bremst kurz ab, um das Hindernis H1 zu überwinden. Die meiste Zeit wird mit der Höchstgeschwindigkeit gefahren, bis das Fahrzeug aufgrund der Hindernisvermeidung die Fahrbahnseite wechseln muss. Es wird keine Restriktion verletzt und die Stellgrößenbegrenzung wird eingehalten, wie Abbildung 5.8 verdeutlicht.

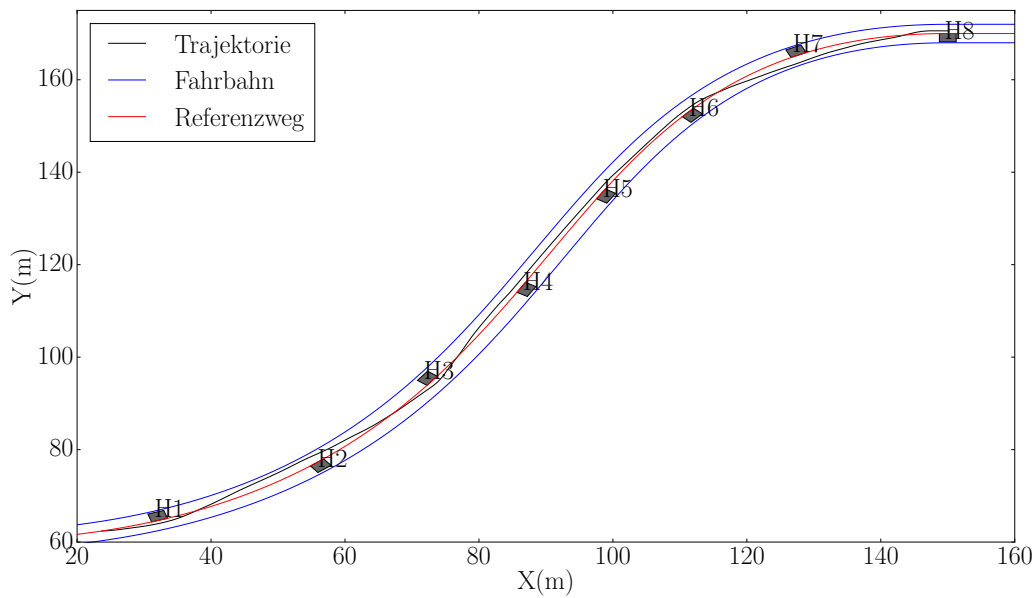


Abbildung 5.6.: Position des Fahrzeugs auf S_1

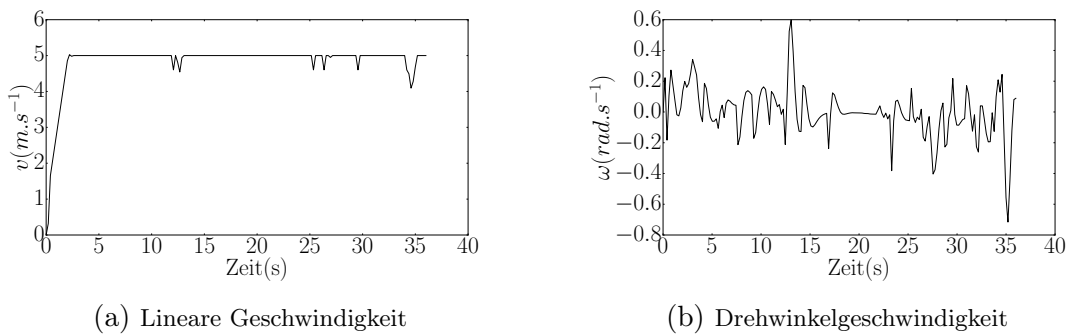
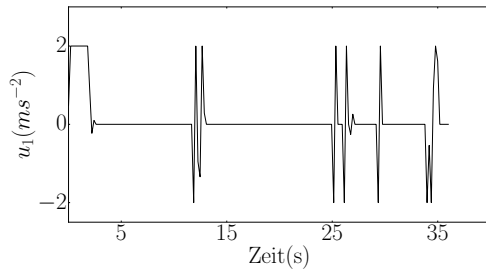
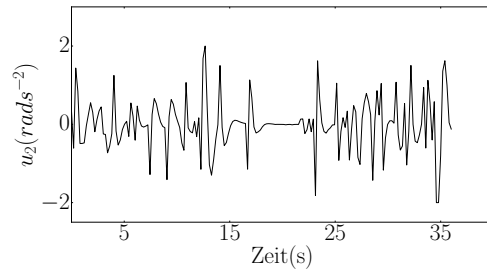


Abbildung 5.7.: Geschwindigkeitsprofile auf S_1

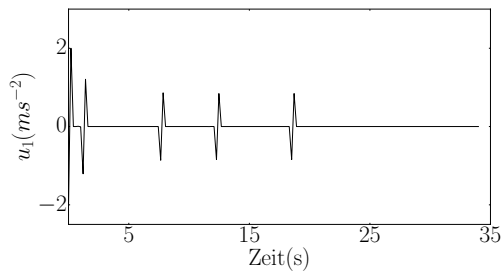


(a) Lineare Beschleunigung

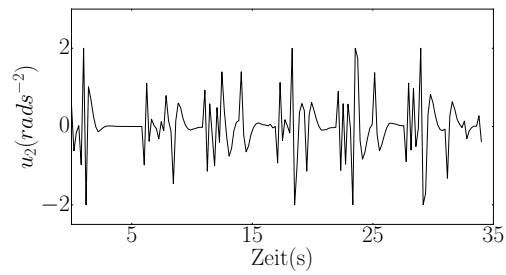


(b) Drehwinkelbeschleunigung

Abbildung 5.8.: Stellgrößen auf S_1



(a) Lineare Beschleunigung

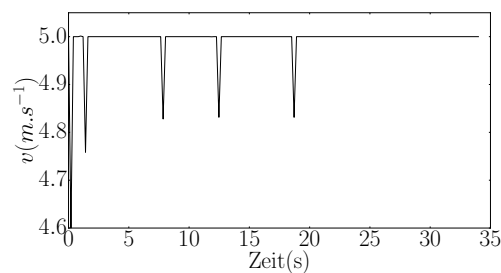


(b) Drehwinkelbeschleunigung

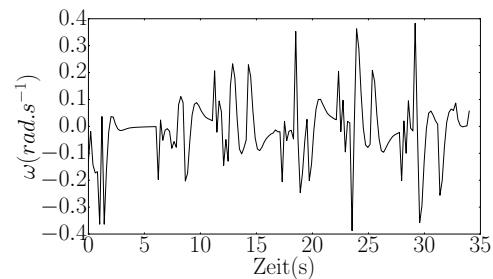
Abbildung 5.9.: Stellgrößen auf S_2

5.5.1.2 Abschnitt S_2

Diese Strecke zeigt einen geradlinigen Verlauf und wird von links nach recht gefahren. In Abbildung 5.11 wird die globale Position des Fahrzeugs gezeigt. Eine hindernisfreie Fahrt ist dabei erkennbar. Hierbei ist zudem sehr gut zu sehen, dass ein Sicherheitsabstand immer eingehalten wird. Wie Abbildung 5.10a es eindeutig zeigt, fährt das Fahrzeug mit $4,6 \text{ ms}^{-1}$ in die Strecke ein und hält die Höchstgeschwindigkeit fast überall, obwohl Hindernisse überwinden werden müssen. Dies liegt daran, dass die Strecke geradlinig ist und die Lage des Zielpunkts im Vorfeld



(a) Lineare Geschwindigkeit



(b) Drehwinkelgeschwindigkeit

Abbildung 5.10.: Geschwindigkeitsprofile auf S_2

verhindert, dass das Fahrzeug in Richtung eines Hindernisses fährt. In Abbildung 5.9 sind die Stellgrößen veranschaulicht. Dabei wird erkannt, dass die lineare Beschleunigung null bleibt, wenn das Fahrzeug die Höchstgeschwindigkeit(lineare Geschwindigkeit) erreicht hat. Im Gegenteil dazu wird eine Drehwinkelbeschleunigung benötigt, um Hindernisse auszuweichen. Die Stellgrößenbegrenzung wird dabei eingehalten.

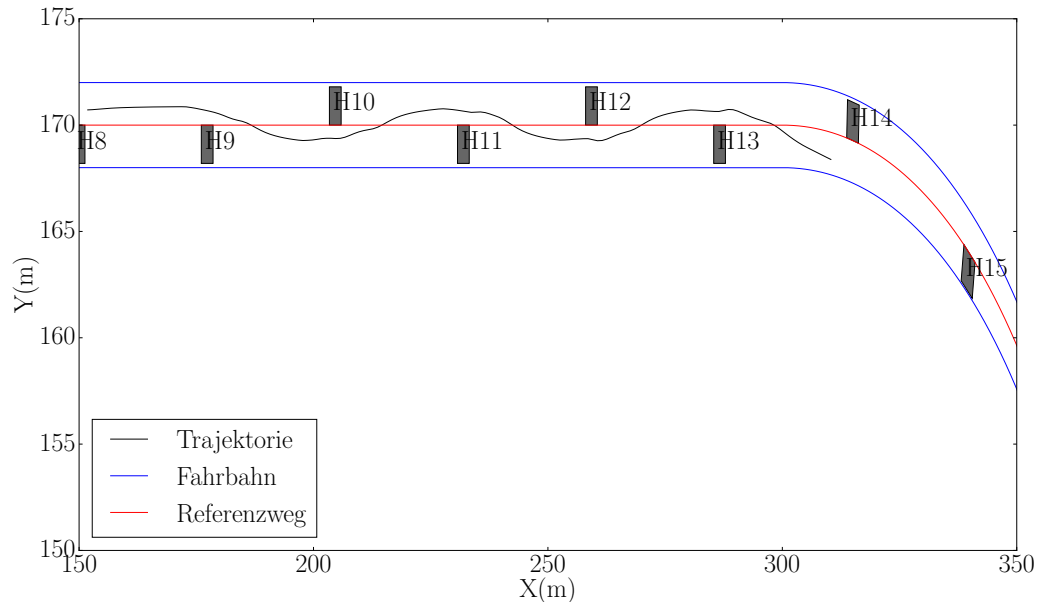


Abbildung 5.11.: Position des Fahrzeugs auf S_2

5.5.2 Kurvenfahrt

Es wird in diesem Abschnitt das Verhalten des Fahrzeugs auf Strecken, deren Verläufe eine starke Krümmung aufweist, untersucht. Streckenabschnitte S_3 und S_5 werden dabei verwendet.

5.5.2.1 Abschnitt S_3

Auf dem Streckenabschnitt S_3 wird die Fahrt von oben nach unten durchgeführt. Die Position des Fahrzeugs während der Fahrt wird in Abbildung 5.12 dargestellt. Wie erwartet, wird eine hindernisfreie Trajektorie gezeigt. Wie es bei S_1 schon der Fall war, wird jedoch eine dauerhafte Querabweichung in der Höhe von H20 gesehen, wo das Fahrzeug eine Zeit lang am Straßenrand fährt. Es verlässt aber die Fahrbahn nicht. In Abbildung 5.13 sind die Geschwindigkeitsprofile zu sehen. Dabei ist ersichtlich, dass das Fahrzeug auf der Höhe von H18 stark bremst. Dort weist die Strecke die größte Krümmung auf. Außerdem muss ein Hindernis vermieden werden. In Abbildung 5.14 wird gezeigt, dass die beiden Stellgrößen in dem erlaubten Bereich gehalten werden.

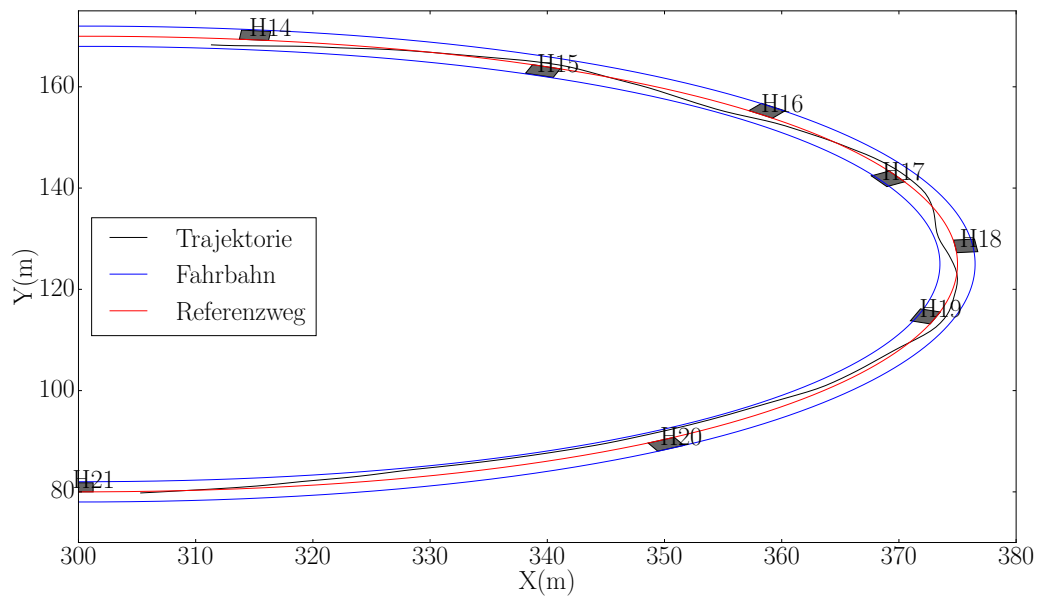
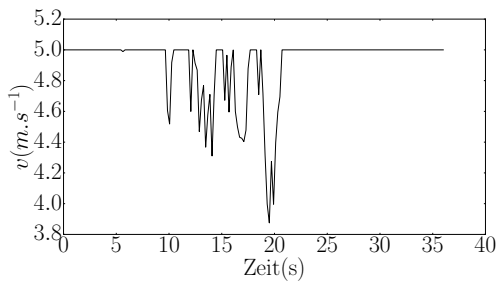
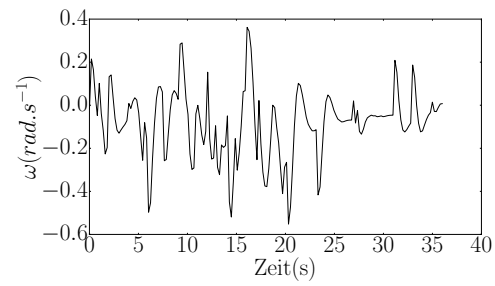


Abbildung 5.12.: Position des Fahrzeugs auf S_3

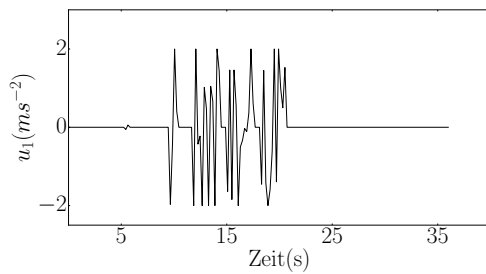


(a) Lineare Geschwindigkeit

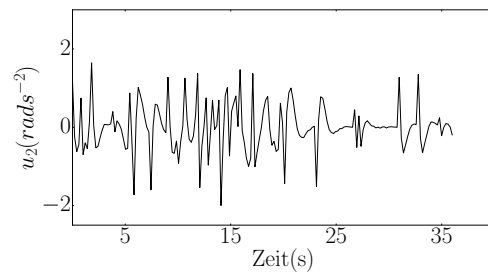


(b) Drehwinkelgeschwindigkeit

Abbildung 5.13.: Geschwindigkeitsprofile auf S_3



(a) Lineare Beschleunigung



(b) Drehwinkelbeschleunigung

Abbildung 5.14.: Stellgrößen auf S_3

5.5.2.2 Abschnitt S_5

Dieser Streckenabschnitt wird ebenfalls von oben nach unten gefahren. Er unterscheidet sich aber von S_3 dadurch, dass er nach rechts geöffnet ist. Abbildung 5.15a veranschaulicht die Position des Fahrzeugs. Es fährt teilweise am Straßenrand, ohne dabei die Fahrbahn zu verlassen. Genau wie bei S_3 wird die Trajektorie nur in Richtung des inneren Rands der Straße ausweichen. Dies liegt zum großen Teil daran, dass an dieser Stelle der Zielpunkt nicht mehr auf \mathcal{R} liegt, sondern auf der linken Straßenseite. Abbildung 5.15b veranschaulicht diese Fahrsituation. In Abbildung 5.16 werden die Geschwindigkeiten dargestellt. Es ist ersichtlich, dass es während der Fahrt viel gebremst wird. Es wird teilweise bis zu 4.6 ms^{-1} gefahren. Die Stellgrößen sind in Abbildung 5.20 dargestellt. Alle Restriktionen wurden eingehalten und das Fahrzeug hat die Hindernisvermeidung überstanden.

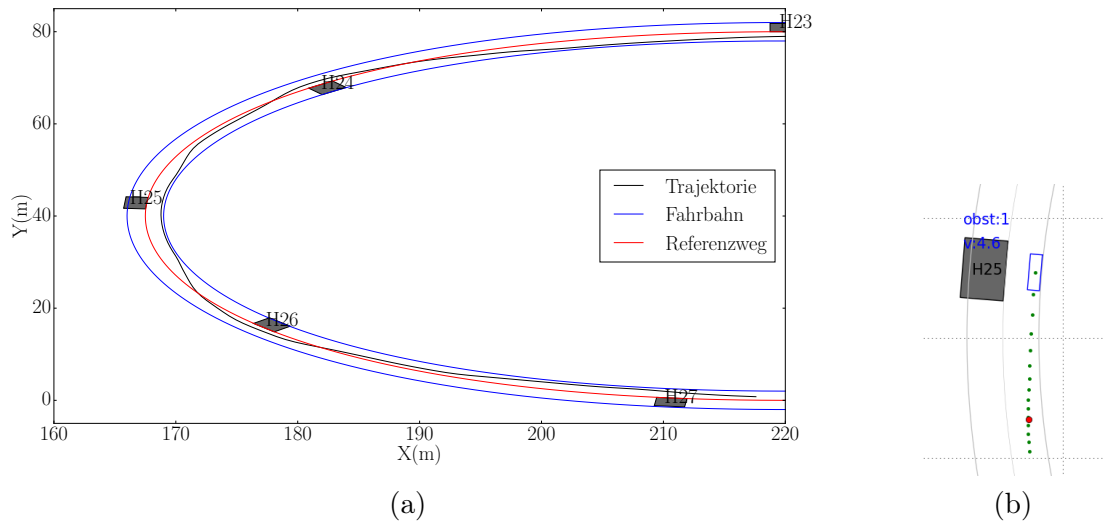


Abbildung 5.15.: Position des Fahrzeugs auf S_5

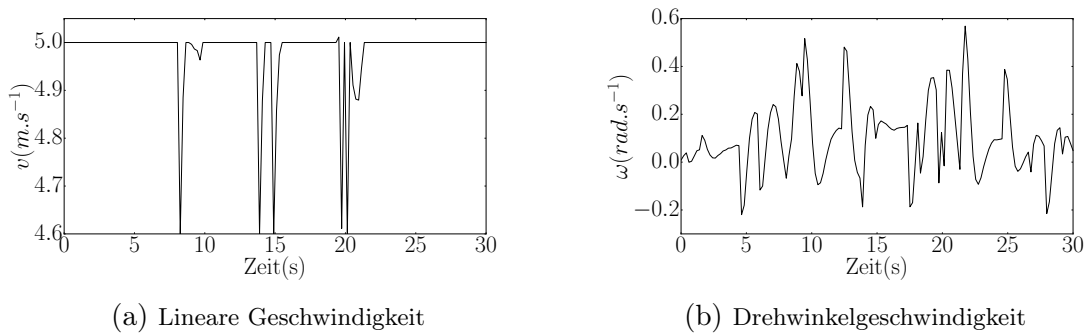


Abbildung 5.16.: Geschwindigkeitsprofile auf S_5

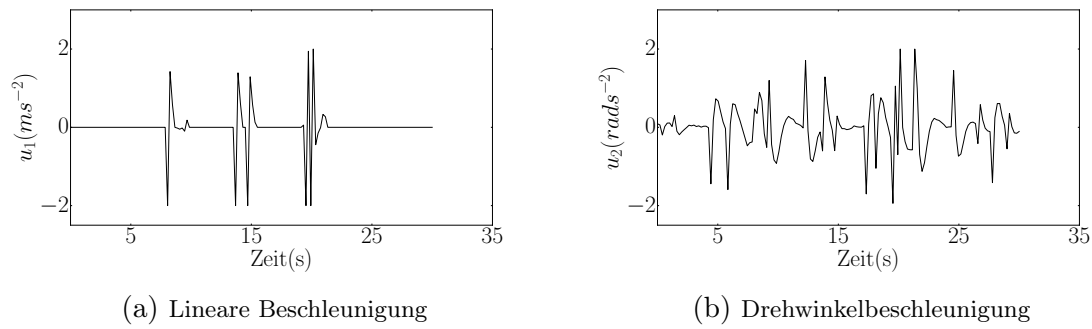


Abbildung 5.17.: Stellgrößen auf S_5

5.6 Gesamtstrecke

Zusammengefasst zeigt Abbildung 5.18 die berechnete Trajektorie auf der Gesamtstrecke. Es wurde gezeigt, dass das Fahrzeug während der Fahrt auf der abgegrenzten Fahrbahn bleibt. Es wird eine hindernisfreie Trajektorie berechnet. Das Fahrzeug bleibt häufig auf dem Mittelstreifen und lenkt nur dann ab, wenn ein Hindernis zu vermeiden gibt. Während der Kurvenfahrt weist jedoch die Trajektorie eine Querabweichung auf. Diese Abweichung gefährdet aber die Hindernisvermeidung nicht, sondern könnte lediglich dazu führen, dass das Fahrzeug die Fahrbahn verlässt, wenn es starke Krümmungen in dem Straßenverlauf gibt.

Die vorgeschriebene Höchstgeschwindigkeit von 5 ms^{-1} wurde nicht überschritten. Offensicht-

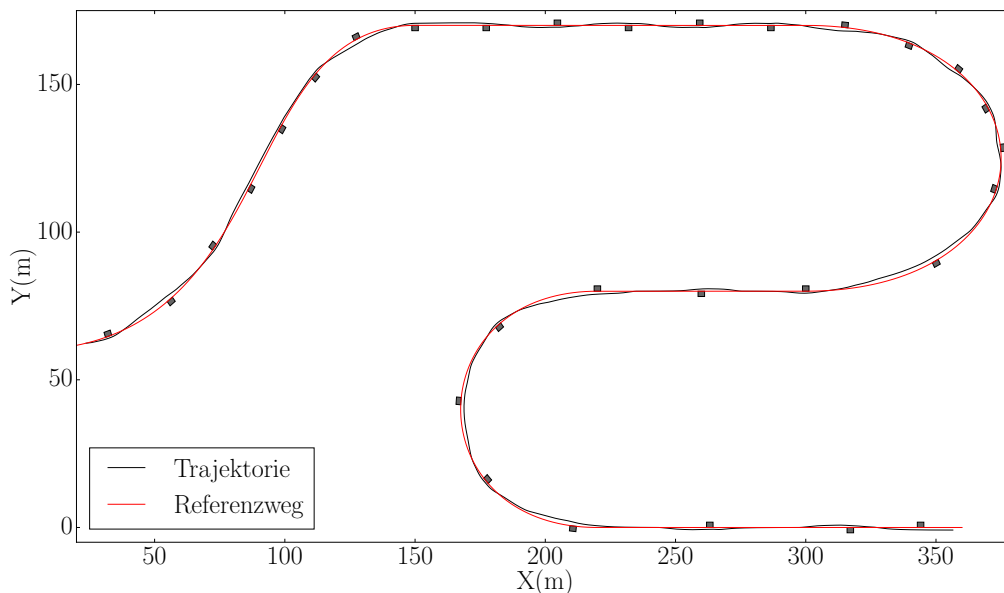


Abbildung 5.18.: Übersicht über die berechnete Trajektorie

lich bremst das Fahrzeug kurz vor Hindernissen ab. Vor allem wenn sie in der Kurve stehen. Während einer Kurvenfahrt wird daher das Fahrzeug langsamer. Die Wahl der Höchstgeschwin-

digkeit lässt sich dadurch begründen, dass wenn das Fahrzeug eine höherer Geschwindigkeit besitzt, es häufig zur Kollision kommt. Dies liegt daran, dass die Hindernisse relative nah zueinander liegen und werden in der Optimierung nur dann berücksichtigt, wenn ein Kollisionsrisiko besteht. Somit kann es vorkommen, dass der Bremsweg zu kurz ist. In der Realität kommt es auch vor, dass trotz einer Vollbremsung eine Kollision nicht vermieden werden kann. Es muss daher beachtet werden, dass abhängig von der aktuellen Geschwindigkeit ein ausreichender Bremsweg vorgenommen wird, wobei das nächste in Fahrtrichtung liegende Hindernis unbedingt berücksichtigt werden muss. Alle Restriktionen und die maximalen Stellgrößen von jeweils 2 ms^{-2} und 2 rads^{-2} werden eingehalten.

5.7 Auftretende Probleme

Es wurde verdeutlicht, dass eine Kollisionsgefahr besteht, wenn das Fahrzeug schnell fährt und kein langer Bremsweg eingehalten wird. Das Problem der Querabweichung bei einer Kurvenfahrt wurde auch schon besprochen. Neben diesen Problemen wurden weitere unerwünschte Situationen festgestellt:

- (a) Keine optimale oder gar keine Lösung der Optimierungsaufgabe wird gefunden:
Dies liegt daran, dass die zu lösende Aufgabe eine nicht konvexe und nichtlineare Optimierungsaufgabe darstellt. Denn die Dynamik des Systems und die Ungleichungsbeschränkungen zur Hindernisvermeidung sind nichtlinear. Die Ausgabe des Solvers sieht in diesem Fall ungefähr wie in Abbildung 5.20a aus. Dieses Problem kann häufig vermieden werden, indem eine gute(bzw. zulässige) Startlösung verwendet wird.
- (b) Zielpunktadaption führt zum Fehlverhalten:
Wenn das Fahrzeug schnell fährt und der Zielpunkt in der Nähe liegt, sieht die berechnete Trajektorie wie in Abbildung 5.19a aus. Es könnte noch schlimmer werden und die Situation in Abbildung 5.19c entsteht. Dabei ist zu sehen, dass das Hindernis nicht berücksichtigt wird, weil es weiter weg vom Zielpunkt liegt. Die berechnete Trajektorie verläuft jedoch im Hindernis. Die Trajektorie in Abbildung 5.20c könnte die Fahrbahn verlassen.
- (c) Diskretisierung:
Die Restriktionen werden lediglich an der Stellen eingehalten, an der die Knotenvariablen liegen. Dies kann dazu führen, dass die zwischen Knotenvariablen liegenden Hindernisse nicht berücksichtigt werden. Abbildung 5.20b verdeutlicht diese unerwünschte Situation. Dabei ist zu sehen, dass die obere Ecke des Hindernisses die Trajektorie fast berührt.

Zur Vermeidung des Problems (a) wurde folgende Vorsichtsmaßnahme in dieser Arbeit getroffen:

- Der letzte Systemzustand $\bar{\mathbf{x}}_0$ wird immer als Startlösung vorgegeben. Die Knotenvariable \mathbf{x}_0 muss dann so beschränkt werden, dass sie nicht von $\bar{\mathbf{x}}_0$ abweicht. Das bedeutet, dass die obere und die untere Grenze von \mathbf{x}_0 auf $\bar{\mathbf{x}}_0$ gesetzt werden müssen. Dies führt dazu, dass die Startlösung einen zulässigen Wert darstellt und kann bei dem Finden anderer Lösungen helfen. Anders gesagt, werden nicht nur die obere und die untere Grenze der

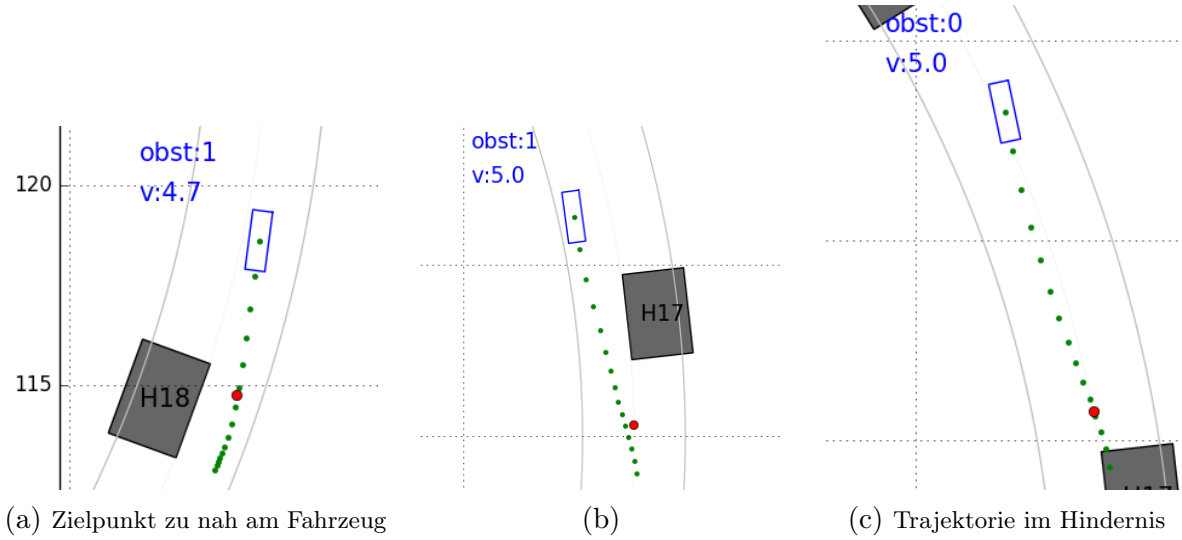


Abbildung 5.19.: Durch Zielpunktadaption auftretende Probleme

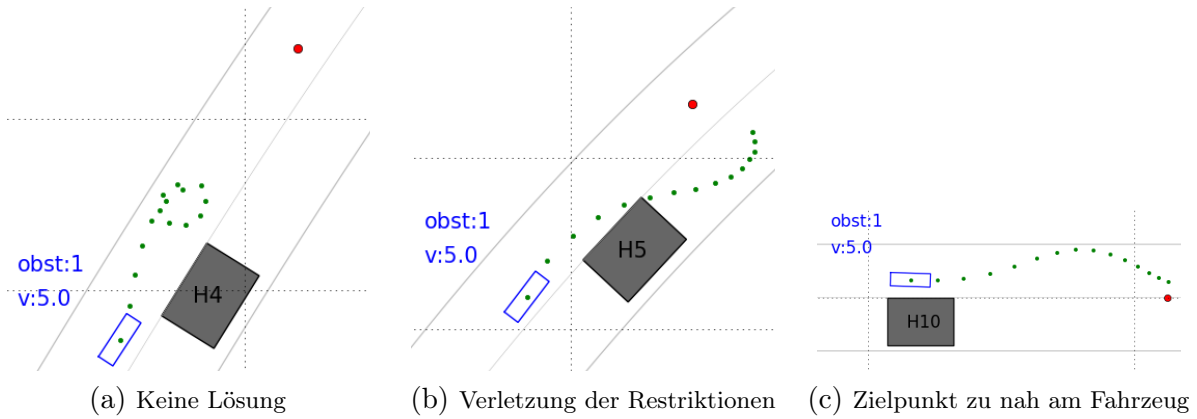


Abbildung 5.20.: Fehlverhalten

Gleichungsbeschränkungen $\mathbf{x} - \bar{\mathbf{x}}_0 = 0$ (Gleichung (4.24b)) auf Null gesetzt, sondern wird zusätzlich die Startlösung (wo der Solver bei der Suche nach einer Lösung beginnen muss) auf $\bar{\mathbf{x}}_0$ gesetzt.

Allerdings muss aufgepasst werden, dass wenn der Solver aus irgendeinem Grund keine zulässige Lösung findet, die nächste Startlösung wahrscheinlicher unzulässig sein wird, sofern der Fehler durch $J_{in}(\cdot)$ nicht behoben wird. Trotz allem gibt es keine Garantie, dass eine Lösung gefunden wird.

5.8 Vergleich des Planungskonzepts

Ein entscheidender Faktor beim Einsatz der MPC ist die Rechenzeit, die erheblich hoch sein kann. Diese Arbeit wurde zum Teil in der Tatsache motiviert, dass die Rechenzeit durch das vorgestellte Planungskonzept verringert werden kann. Um diese Vermutung zu begründen, wird

untersucht, wie lange die Planung dauert. Das implementierte Planungskonzept besteht aus wiederholenden Teilaufgaben. Bei jeder Neuplanung wird ein Zielpunkt generiert, ein OCP formuliert und gelöst. Das Lösen von Teilaufgaben wird abhängig von der Anzahl der Variablen unterschiedliche Rechenzeit kosten. Es wird eine aus S_2 und S_3 bestehende Strecke gefahren. Zunächst wird es zweimal mit adaptiv generierten Zielpunkten gefahren, wobei eine unterschiedliche Anzahl von Hindernissen verwendet wird. Zum Vergleich werden die gleichen Fahrten wiederholt, mit dem einzigen Unterschied, dass die Zielpunkte immer in der gleichen Entfernung in Fahrtrichtung liegen. Die festen Zielpunkte werden dabei immer 30 m (Länge des Gitters) weit vom Fahrzeug liegen. In Abbildung 5.22 werden die daraus entstandenen

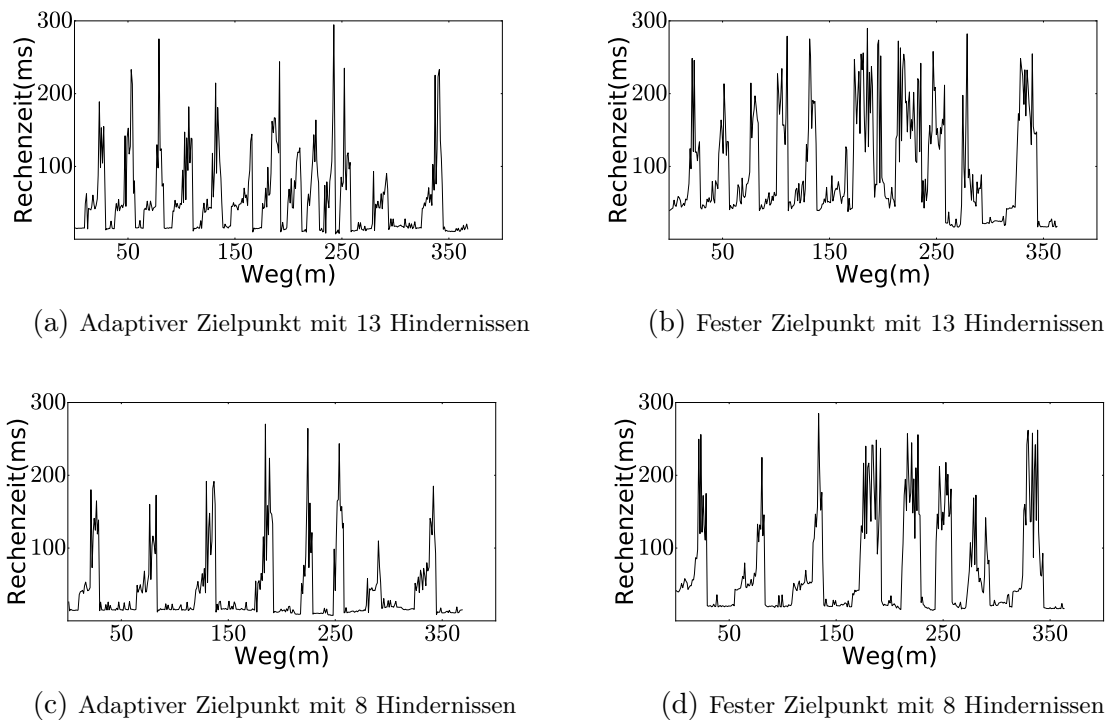


Abbildung 5.21.: Rechenzeit in Anwesenheit von Hindernissen

Rechenzeiten bezüglich des zurückgelegten Wegs dargestellt. Bei dem ersten Durchgang werden dreizehn Hindernisse auf der Fahrbahn platziert. Abbildung 5.21a und 5.21b zeigen sehr unterschiedliche Verläufe. Bei der Planung mit adaptiver Zielpunktgenerierung liegen die Rechenzeiten zum großen Teil unter 100 ms und überschreiten 200 ms sehr selten. Im Gegenteil dazu liegen die Rechenzeiten bei festem Zielpunkt meistens über 100 ms. Sie überschreiten außerdem sehr oft 200 ms. Abbildung 5.21c und 5.21d zeigen den zweiten Versuch, wobei weniger Hindernisse zu vermeiden gibt. Wie vermutet, wird bei der Planung mit festen Zielpunkten mehr Rechenzeit benötigt. Dies liegt daran, dass Hindernisse immer berücksichtigt werden, obwohl keine Kollisionsgefahr besteht. Je weiter der Zielpunkt liegt, desto höher wird die Rechenzeit. Bei den beiden Planungen wird festgestellt, dass die Rechenzeit erheblich steigt, wenn sich das Fahrzeug dem Hindernis annähert. Bei der Planung mit adaptivem Zielpunkt steigt jedoch die Rechenzeit nur ganz kurz.

Die MPC funktioniert mit einer Abtastperiode von $h = 200 \text{ ms}$. Wenn in der Praxis eine Neuplanung alle 200 ms stattfinden würde, würde das heißen, dass eine Planung mit einem festen Zielpunkt nicht echtzeitfähig wäre, wenn der Zielpunkt in 30 m oder mehr liegt.

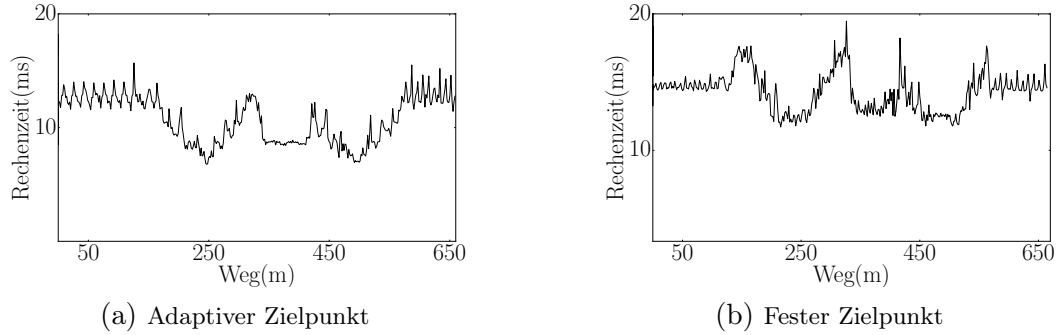


Abbildung 5.22.: Rechenzeit ohne Hindernis

Tabelle 5.3.: Zurückgelegter Weg bei gleicher Zeit

	Zeit(s)	Weg(m) mit Hindernissen	Weg(m) ohne Hindernisse
Adaptiver Zielpunkt	104	517,86	856,33
Fester Zielpunkt	104	514,17	862

Zum weiteren Vergleich wurde 104 s lang die Teststrecke gefahren. Dabei wurden die zurückgelegten Wege berechnet und in Tabelle 5.3 aufgetragen. Daraus ist es offensichtlich, dass die Fahrt in Anwesenheit von Hindernissen schneller wird, wenn eine Zielpunktadaption durchgeführt wird. Der Grund dafür ist die lange Rechenzeit, die bei der Planung mit einem festen Zielpunkt entsteht, wenn Hindernisse vorhanden sind. Zudem wird die Kollisionsgefahr bei der Planung mit festem Zielpunkt sehr groß. Um das Fahrzeug auf der Fahrbahn zu halten wurde der Parameter $q_d=50$ (Gleichung (4.9b)) ausgewählt. Dies hat zur Folge, dass das Fahrzeug häufig auf \mathcal{R} bleibt und auf Hindernisse zufährt. Im Gegenzug wird die Zielpunktadaption bei einer Planung ohne Hindernisse die Fahrt verzögern. Dies liegt daran, dass durch die Zielpunktadaption das Fahrzeug bremst, auch wenn kein Hindernis im Weg liegt. Dieses Problem lässt sich lösen, indem t_{soll} verkleinert wird.

Abbildung 5.22a und 5.22b zeigen die bei den Planungen ohne Hindernisse entstandenen Rechenzeiten. Die Rechenzeit bei der Planung mit Zielpunktadaption liegt durchschnittlich bei 10 ms . Bei der Planung mit festem Zielpunkt wird mehr Rechenzeit gebraucht. Diese Rechenzeiten liegen bei den beiden Planungen in einem sehr guten Bereich. Durch die Zielpunktadaption wird das Fahrzeug eventuell langsamer fahren und somit wird für die gleiche Strecke mehr Zeit benötigt. Jedoch wird die Rechenzeit gering gehalten. Bei der Zielpunktgenerierung werden die durchschnittlichen Rechenzeiten von jeweils 10 ms bei einer Planung mit Hindernissen und 1 ms bei einer Planung ohne Hindernisse gebraucht. Die Zielpunktgenerierung verursacht daher keine große Rechenzeit.

In dieser Arbeit wurden die Parameter $N=15$ und $T=3$ s lediglich abhängig von dem Systemverhalten ausgewählt. Dies führt dazu, dass unerwartete Situationen beobachtet werden. Es wurde beispielsweise eine Strecke von 856,33 m mit einer Höchstgeschwindigkeit von 5 ms^{-1} in 104 s gefahren, wenn keine Hindernisse im Weg liegen. In der Realität ist diese Feststellung unvorstellbar. Dies liegt daran, dass die gewählten Parameter (T und N) für eine Fahrt ohne Hindernisse schlecht konditioniert sind. Diese Strecke (ohne Hindernisse) wurde tatsächlich länger als 104 s gefahren, denn der Solver war dabei ungefähr zehn mal schneller als bei einer Planung mit Hindernissen.



6 Zusammenfassung und Ausblick

Die vorliegende Arbeit vermittelt einen Einblick in den Bereich der Trajektorienplanung. Es wurde basierend auf der MPC ein Konzept zur Trajektorienplanung entworfen. Hierbei wurden zunächst ein verfahren zur Ermittlung eines Zwischenzielpunkts vorgestellt, das sich im Bezug auf die Geometrie der Fahrbahn durchführen lässt. Dieses überlagerte verfahren, die MPC, die Koordinatentransformation und ein Regelungssystem wurden als Hauptteile des Planungskonzepts vorgestellt. Zudem wurde die Aufgabe der Trajektorienplanung in kleine Teilaufgaben zerlegt, um eine lokale Betrachtung des Problems zu ermöglichen.

Um das Thema Trajektorienplanung abzugrenzen und somit der Lösungsraum zu definieren, wurde das Umfeld als statisch und strukturiert betrachtet, in dem alle Verkehrsteilnehmer sich bezüglich eines weltfesten Koordinatensystems lokalisieren lassen. Zur lokalen Beschreibung des Systems wurde eine Koordinatentransformation hergeleitet, womit alle Parameter von einem beweglichen Straßenkoordinatensystem nach dem Weltkoordinatensystem dargestellt werden können und umgekehrt. Somit wurde die Aufgabe der ersten Komponente der Planung beschrieben.

Der lokale Zielpunkt ließ sich in zwei Schritten ermitteln. Durch ein straßenadaptiertes Gitter wurden Bezugspunkte der Fahrbahn untersucht. Dabei wurden optimale Bezugspunkte anhand der Optimierung einer heuristischen Ansatzfunktion bestimmt. Diese optimalen Bezugspunkte sowie die verfügbare Zeitressource wurden zu einem Regelungssystem weiter geleitet, wo der letztendliche Zwischenzielpunkt bestimmt wurde.

Abhängig von dem Zielpunkt, dem aktuellen Zustand des Fahrzeugs und dem Straßenverlauf wurde ein zeitkontinuierliches OCP formuliert. Dabei wurden hergeleitete Gleichungen und Ungleichungen zur Hindernisvermeidung sowie vorgeschriebene Höchstgeschwindigkeit als Restriktionen des Systems gesetzt. Die Stellgrößenbeschränkung wurde auch berücksichtigt. Durch die vorgestellte Mehrzielmethode wurde das OCP parametrisiert, um die Handhabung des Problems zu erleichtern. Das daraus entstandene NLP wurde durch die Kombination von Drittanbieterprogrammen(CasADi und IPOPT) gelöst. Somit wurde die zu fahrende Trajektorie bestimmt. Diese zusammengefassten Schritte mussten für den gleichen Horizont so weiter fortgesetzt werden, bis ein vorgegebenes Ziel erreicht wird

Des Weiteren wurde basierend auf einem beschriebenen Fahrzeugmodell die Umsetzung durch Implementierung getestet. Die daraus entstandenen Ergebnisse wurden in Kapitel 5 ausführlich präsentiert. Es wurde gezeigt, dass jede Komponente des Planungskonzepts das erwartete Verhalten aufweist. Eine Teststrecke mit unterschiedlichen Abschnitten wurde gefahren. Dabei wurden hauptsächlich die Geradeausfahrt und die Kurvenfahrt untersucht. Bei der Geradeausfahrt konnten sehr gute Ergebnisse erzielt werden. Es wurde eine hindernisfreie Trajektorie

berechnet, wobei alle vorgenommenen Restriktionen eingehalten wurden. Die Kurvenfahrt zeigte auch eine hindernisfreie Trajektorie mit eingehaltenen Beschränkungen. Jedoch wurde an bestimmten Stellen der Fahrbahn eine starke Querabweichung bezüglich des Mittelstreifens(\mathcal{R}) festgestellt. Dieses unerwünschte Verhalten des Systems könnte dazu führen, dass das Fahrzeug die Fahrbahn verlässt.

Es wurden Vergleiche angestellt, um zu dem Schluss zu kommen, dass das vorgestellte Planungskonzept die Rechenzeit signifikant reduziert. Hierbei wurden zweimal die gleiche Strecke mit unterschiedlicher Anzahl von Hindernissen gefahren. Durch die Zielpunktadaption ließ sich während der Fahrt eine kürzere Rechenzeit im Vergleich zu einem festen Zielpunkt erzielen. Dabei tritt jedoch das Problem auf, dass der Zielpunkt häufig nah am Fahrzeug liegt. Dadurch könnten Restriktionen verletzt werden. Zudem könnte die Zielpunktadaption die Fahrt verzögern, wenn keine Hindernisse im Weg liegen.

In dem vorgestellten Planungskonzept wurden viele Beschränkungen bei der Definition des Lösungsraums gemacht. Es wurde lediglich das Spurhalten bei der Umsetzung berücksichtigt. Zur Verbesserung des Planungskonzepts könnten andere Fahrmanöver, wie z. B. Kreuzung überqueren, Spurwechsel, Ein- und Ausparken dem Konzept hinzugefügt werden. Die Bestimmung des lokalen Zielpunkts erfolgt durch Diskretisierung der Fahrbahn. Dabei könnten nicht zu viele Bezugspunkte untersucht werden, da die Rechenzeit dadurch erheblich steigen würde. Es könnte daher in der Zukunft an dieser Stelle zwischen nachfolgenden Bezugspunkten interpoliert werden. Somit werden viel mehr Stellen der Fahrbahn untersucht. Diese Vorgehensweise könnte für ein Fahrmanöver, wie Einparken, helfen. Denn in dieser Fahrsituation wird ein sehr feines Gitter benötigt. Um die Realität näher nachzubilden, muss die Umgebung als dynamisch veränderbar angenommen werden.

Es wurde im Abschnitt 5.7 erwähnt, dass eine unzulässige Startlösung dazu führen könnte, dass der Solver die Optimierungsaufgabe nicht lösen kann. Die vorgeschlagene Vorsichtsmaßnahme bietet jedoch keine Garantie dafür, dass immer eine Lösung gefunden wird. Eine weitere Möglichkeit eine zulässige Startlösung zu finden wäre, dass eine oder mehrere vereinfachte Optimierungsaufgaben vorher gelöst werden. Hiermit werden Optimierungsaufgaben der gleichen Struktur wie die Hauptaufgabe gemeint, wobei entweder die Zielfunktion einen konstanten Wert annimmt oder kein Hindernis mehr berücksichtigt wird[2]. Diese Vorgehensweise ist aber mit zusätzlichen Rechnungen verbunden. Des Weiteren wurde gezeigt, dass die Diskretisierung des zeitkontinuierlichen OCPs dazu führt, dass Zwischen Knotenvariablen die Restriktionen nicht eingehalten werden. Dieses Problem kann verhindert werden, indem das Zeitgitter verfeinert wird. Dafür müssen die Parameter T und N als Optimierungsvariablen gesetzt werden. Hierbei muss beachtet werden, dass die in RKV verwendete Schrittweite dadurch veränderbar wird. Es könnte daher in zukünftigen Arbeiten die Optimierung von Parametern T und N behandelt werden. Das durch die Zielpunktadaption verursachte Problem kann vermindert werden, indem ein geeigneter minimaler Abstand zum Fahrzeug festgelegt wird, unterhalb dessen sich der Zielpunkt nicht befinden darf.

A Anhang

A.1 Polynomfunktionen

Polynome sind Funktionen mit einfacher Gestalt. Sie lassen sich allgemein als Summe der Vielfachen von Potenzen einer Variable darstellen. Aus einer Variable x ergibt sich das Polynom

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0.$$

$$a_{i=0,\dots,n} \text{ mit } n \in \mathbb{R}$$

Die Koeffizienten a_i bestimmen die Eigenschaften des Polynoms und n wird als Ordnung(bzw. Grad) des Polynoms genannt. Somit lässt sich ein Polynom durch die Koeffizienten und der Grad vollständig definieren. Polynome sind differenzierbar und die Ableitung ist wiederum ein Polynom der Gestalt:

$$p'(x) = a_n n x^{n-1} + a_{n-1} (n-1) x^{n-2} + \cdots + a_1.$$

$p'(x)$ ist die erste Ableitung von $p(x)$ und dient zur Untersuchung sowohl der Krümmung als auch der Monotonie. Polynome können die Geometrie einer Fahrbahn sehr gut beschreiben, wie es in Abbildung A.1 zu sehen ist. Zur Darstellung eines Straßenverlaufs kann daher ein Polynom

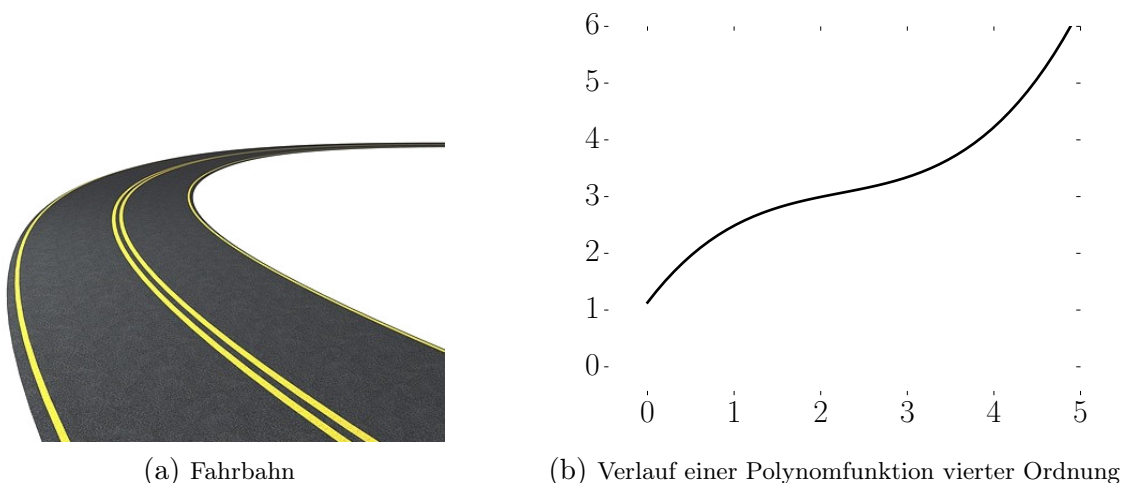


Abbildung A.1.: Der Verlauf eines Polynoms sieht dem Straßenverlauf ähnlich

geeigneter Ordnung verwendet und die Koeffizienten können lokal angepasst werden, wie es in Abschnitt A.2 erläutert wird.

A.2 Methode der kleinsten Quadrate

In der Literatur wird die Methode der kleinsten Quadrate mit dem Begriff Least Squares (LS) bezeichnet. Das LS-Verfahren ist ein mathematisches Optimierungsverfahren, das zur Bestimmung von Parametern einer Funktion verwendet wird. Das Ziel besteht darin, eine möglichst glatte Kurve auf eine Reihe von Messdaten so zu legen, dass sie die Messdaten gut abbildet. Dies lässt sich in Abbildung A.2 veranschaulichen.

Die gesuchte Funktion $F(x)$ kann unterschiedlich parametrisiert werden. Eine einfache und für

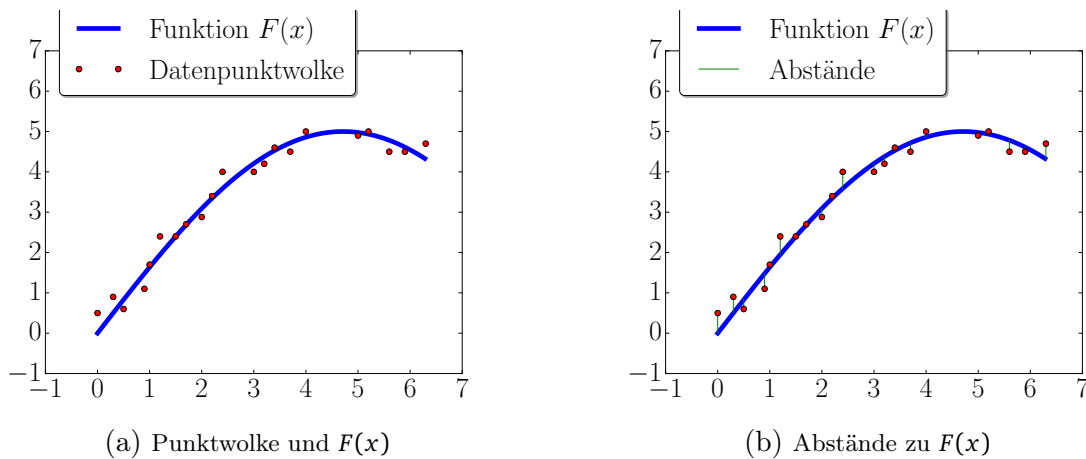


Abbildung A.2.: Verlauf von $F(x)$ durch eine Reihe von Datenpunkten

diese Arbeit geeignete Überlegung wäre, dass $F(x)$ ein Polynom der Ordnung n darstellt.

$$F: \mathbb{R} \rightarrow \mathbb{R}$$

$$x \mapsto y = F(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

Ausgehend von einer Reihe bekannter Punkte $(x_i, y_i)_{i=0, \dots, n}$ kann das LS-Verfahren eingesetzt werden, um die Parameter $(a_i)_{i=0, \dots, n}$ zu ermitteln. Zu diesem Zweck wird eine Ausgleichsrechnung durchgeführt, wie folgt:

- Der Abstand aller Messdaten zu der Kurve von $F(x)$ wird bestimmt und deren Quadrate summiert.
- Die erhaltene Summe wird minimiert

Diese Optimierungsaufgabe (bzw. Minimierungsaufgabe) kann für kleine Mengen von Datenpunktwolken analytisch gelöst werden. Dabei soll die erste Ableitung nach allen Parametern auf Null gesetzt werden. Somit ergeben sich folgende $n + 1$ Gleichungen:

$$\left(\frac{\partial F(x)}{\partial a_i} \right)_{i=0, \dots, n} = 0$$

Es werden daher mindestens $n + 1$ Messdaten benötigt, um dieses Gleichungssystem lösen zu können. Jedoch hat es sich gezeigt, dass die Berechnungsfehler mit steigender Anzahl von Messdaten geringer werden. Aus diesem Grund werden sehr viel Messdaten verwendet und daraus entsteht ein überbestimmtes Gleichungssystem, das nur numerisch gelöst werden kann.

A.3 Tangentengleichung

Tangenten sind Geraden, welche die Steigung des Verlaufs einer Funktion in einem bestimmten Punkt beschreiben. Abhängig von der Krümmung der Kurve wird immer mindestens ein Berührungspunkt erwarten. Im allgemeinen Fall gehören viel andere Punkte der Kurve zu der Tangente, vor allem diejenigen die in der unmittelbaren Nähe der Stelle liegen, wo die Tangente berechnet wird. Tangenten lassen sich beispielsweise graphisch oder durch eine Taylorreihe bestimmen.

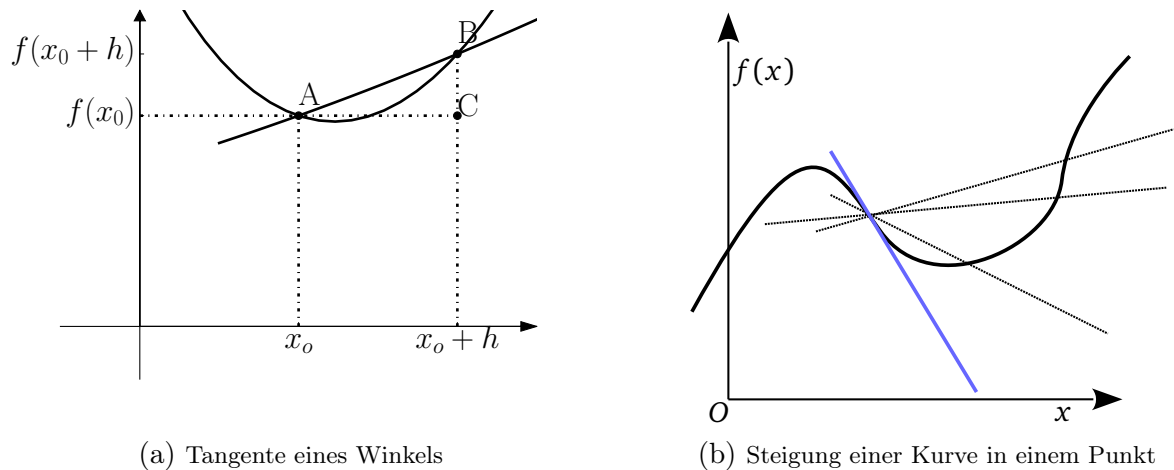


Abbildung A.3.: Graphische Ableitung der Tangente

A.4 Graphische Ermittlung der Tangente

In Abbildung A.3a ist der Verlauf einer Funktion $f(x)$ dargestellt. Zur Beschreibung der Tangente wurden zwei Punkte $(x_0, f(x_0))$ und $(x_0+h, f(x_0+h))$ dieser Funktion ausgewählt, wodurch eine Gerade verläuft. Die Punkte A, B und C bilden ein rechtwinkliges Dreieck, das verwendet werden kann, um die Tangente des Winkels \widehat{BAC} zu bestimmen. Wenn a die Tangente dieses Winkel ist, dann gilt die folgende Formel:

$$a = \frac{f(x_0+h) - f(x_0)}{x_0+h - x_0} = \frac{f(x_0+h) - f(x_0)}{h}$$

Nun wird der Abstand h verkleinert, mit der Absicht eine Gerade zu bekommen, welche die Kurve nur noch leicht berührt. Abbildung A.3b beschreibt dieses Verhalten. Es ist ersichtlich,

dass wenn h unendlich klein wird, die blaue Gerade sich ergibt. Diese Gerade stellt die Tangente der Kurve von $f(x)$ im Punkt A dar und lässt sich schreiben, wie folgt:

Die Tangente der Funktion $f(x)$ in einem Punkt $(x_o, f(x_o))$ ist die Gerade, die den Verlauf der Funktion in diesem Punkt berührt und folgende Steigung besitzt.

$$\lim_{h \rightarrow 0} \frac{f(x_o + h) - f(x_o)}{h}$$

Diese Formel kann graphisch interpretiert werden, wie folgt:

Gegeben sei die Funktion

$$\begin{aligned} F : \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto f(x) \end{aligned}$$

Für alle relativ nah zueinander liegenden zwei Punkte $(x_1, f(x_1))$ und $(x_2, f(x_2))$ von $f(x)$, hat die Tangente im Punkt $(x_1, f(x_1))$ die Steigung

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

Die erste Ableitung stellt eine Alternative zu dieser Formel dar. Denn die Steigung der Tangente einer Funktion $f(x)$ in einem Punkt $(x_o, f(x_o))$ kann auch als $f'(x_o)$ berechnet werden.

B Transformation von Koordinatensystemen

Die Fahrzeugbewegung findet in einer Ebene statt, in der das Fahrzeug als Starrkörper zu betrachten ist. Die Verformung eines Starrkörpers wird in jedem Zeitpunkt während der Bewegung vernachlässigt, weil der Abstand zwischen zwei beliebig angenommenen Punkten des Fahrzeugs sich nicht ändern wird. Das führt dazu, dass die Bewegung des ganzen Körpers durch ein Koordinatensystem(KS) beschrieben werden kann, das an einen beliebigen Punkt des Körpers angebracht ist. Dabei werden die Position und die Orientierung des körperbezogenen KSs zur formelmäßigen Beschreibung der Bewegung benötigt. Die umgebenden Objekte werden auch durch ein ähnliches KS beschrieben.

Im Folgenden werden rechtwinklige kartesische KS verwendet, um die Rotationen und die Translationen darzustellen. Die Basisvektoren rechtwinkliger KS sind Einheitsvektoren, stehen orthogonal zu einander und bilden ein Rechtssystem.

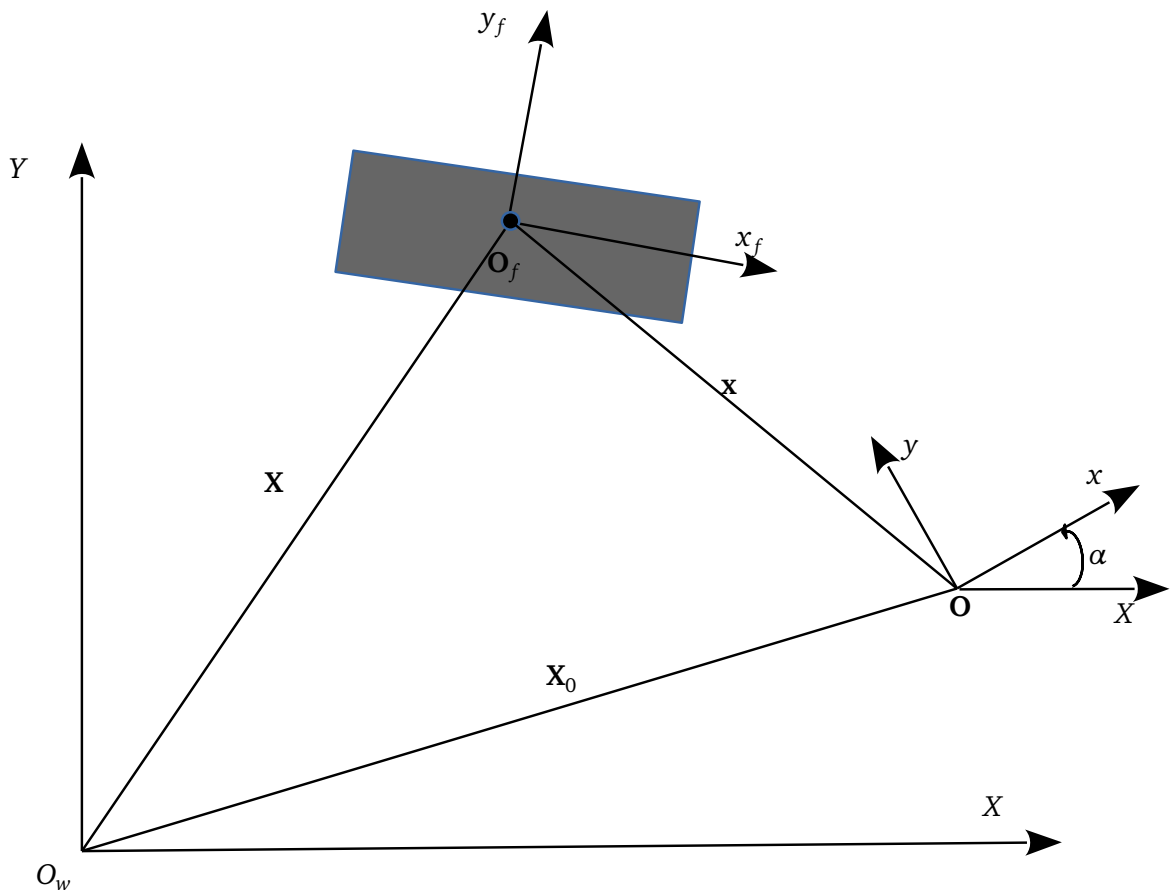


Abbildung B.1.: Räumliche Anordnung eines Objekts beim Wechsel eines KSs von $\{O_fx_fy_f\}$ nach $\{O_wXY\}$

Zur Vereinfachung wird der Körper durch ein körperbezogenes KS $\{\mathbf{O}_f x_f y_f\}$ dargestellt, deren Orientierung und Position in einem beweglichen KS $\{\mathbf{O}xy\}$ bekannt ist. Die globale Betrachtung erfolgt durch das Weltkoordinatensystem $\{\mathbf{O}_w XY\}$. Die Bewegung des Körpers(bzw. $\{\mathbf{O}_f x_f y_f\}$) ist eine Reihe von Abbildungen. Wobei jede Abbildung sich aus Translation und Rotation des körperbezogenen KSs zusammenfasst. Diese Art der Abbildung wird in der Literatur unter dem Begriff affine Abbildung bezeichnet. Abbildung B.1 zeigt einen starren Körper, der in zwei unterschiedlichen KS beschrieben wird. Das Objektkoordinatensystem $\{\mathbf{O}xy\}$, in dem eine lokale Beschreibung des Körpers stattfindet, wird durch einen Drehwinkel α und einen Translationsvektor \mathbf{X}_0 bezüglich $\{\mathbf{O}_w XY\}$ lokalisiert. Die in $\{\mathbf{O}xy\}$ beschriebene Bewegung lässt sich anhand einer Transformationsbeziehung der Form (B.1) in $\{\mathbf{O}_w XY\}$ darstellen.

$$\mathbf{X} = \mathbf{R}\mathbf{x} + \mathbf{X}_0 \quad (\text{B.1})$$

Wobei \mathbf{x} der Zustandsvektor des Körpers(enthält die Position und Orientierung des Körpers bezüglich $\{\mathbf{O}xy\}$) und \mathbf{R} die Rotationsmatrix sind. In dem Rechtssystem ergibt sich bei einer positiven Rotation um den Winkel α folgende Beziehungen:

$$\begin{aligned} \mathbf{R}_x(\alpha) &= \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \end{pmatrix}^T \\ \mathbf{R}_y(\alpha) &= \begin{pmatrix} -\sin(\alpha) & \cos(\alpha) & 0 \end{pmatrix}^T \\ \mathbf{R}_z(\alpha) &= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T \end{aligned}$$

Hierbei ist es ersichtlich, dass die Drehung um die Z-Achse stattfindet. Zusammengefasst hat \mathbf{R} für eine positive Drehung folgende Gestalt:

$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{B.2})$$

und

$$\mathbf{R}^{-1}(\alpha) = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{B.3})$$

C Frei wählbare Parameter und Planungskonzept

Entwurfsparameter

Symbol	Bedeutung	Einheit
MPC		
T	Zeithorizont	s
N	Prädiktionshorizont	
σ	Anzahl der Schritte von RKV4	
$\mathbf{Q} \in \mathbb{R}^{5 \times 5}$	positiv semidefinite Matrix	
$\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{2 \times 2}$	positiv definite Matrizen	
q_d, q_ϕ, q_θ	Parameter der Zielfunktion	
\mathbf{u}_{max}	maximale Stellgrößen	ms^{-2} , rads^{-2}
y_{max}	halbe Breite der Fahrbahn	m
v_{max}	Höchstgeschwindigkeit	ms^{-1}
Generierung von Zielpunkten		
Δs	Längsabstand zwischen Stichpunkten	m
Δl	Seitenabstand zwischen Stichpunkten	m
M	Anzahl von Schichten eines Gitters	
K	Anzahl von Stichpunkten einer Schicht	
s_{max}	maximale Länge des Gitters	m
Regelung		
b_0, b_1, b_2	Reglerparameter	
a	Parameter zur Änderung der Steigung der Fermi-Funktion	s
t_{soll}	Sollrechenzeit	s

Algorithmus 2 Planungskonzept

Eingabe: $\bar{\mathbf{x}}_0$, Entwurfparameter, Hindernisse, \mathcal{R} , t_{solver}

Ausgabe: \mathbf{X}_e und \mathbf{U}

- 1: Ursprung und Drehwinkel(Abschnitt A.4) von \mathcal{F}_s bezüglich \mathcal{F}_w festlegen
 - 2: Koordinatentransformation(Hindernisse, \mathcal{R} und $\bar{\mathbf{x}}_0$ in \mathcal{F}_s darstellen)
 - 3: Berechnung des Zielpunkts
 - 4: Gitter erzeugen
 - 5: Optimale Gitterpunkte durch Dynamische Programmierung ermitteln
 - 6: Zielpunkt in Regelungssystem bestimmen
 - 7: Formulierung des OCPs
 - 8: Zielfunktion in Gleichung (4.9)
 - 9: Beschränkungen in Gleichung (4.24)
 - 10: Lösung des NLPs in Gleichung (4.7)
 - 11: $\bar{\mathbf{x}}_0 \leftarrow \mathcal{J}_{in}(\bar{\mathbf{x}}_0, \mathbf{u}_0^*)$
 - 12: Koordinaten von $\bar{\mathbf{x}}_0$ zurück in \mathcal{F}_w transformieren und speichern
 - 13: Horizont verschieben und wiederholen
-

Literaturverzeichnis

- [1] Stryk, O. V.: Grundlagen der Robotik. Technische Universität Darmstadt.
- [2] C. Liu, S. Lee, S Varnhagen H. E. Tseng: Path Planning for Automous Vehicles using Model Predictive Control.
- [3] LaValle, S. M.: Planning Algorithms. Cambridge University Press, 2006.
- [4] Ziegler, J.: Optimale Bahn- und Trajektorienplanung für Automobile. Doktorarbeit.
- [5] Rathgeber, Christian: Trajektorienplanung und -folgeregulung für assistiertes bis hochautomatisiertes Fahren. Doktorarbeit, Technische Universität Berlin, 07.2016.
- [6] Y. Kuwata, J.Teo, S. Karaman G. Fiore E. Frazzoli J. How: Motion Planning in Complex Environments using Closed-loop Prediction, 2007.
- [7] Y. Kuwata, J.Teo, S. Karaman G. Fiore E. Frazzoli J. How: Real-Time Motion Planning with Applications to Autonomous Urban Driving, 2007.
- [8] D. D. Dunlap, C. V. Caldwell und E. G. Collins Jr: Nonlinear Model Predictive Control Using Sampling and Goal-Directed Optimization.
- [9] Qian, X., I. Navarro, A. De la Fortelle und F. Moutarde: Motion Planning for Urban Autonomous Driving using Bézier Curves and MPC, 2016.
- [10] M. Diehl, H. Bock, H.Diedam und P. B. Wieber: Fast Direct Multiple Shooting Algorithms for Optimal Robot Control, 2009.
- [11] Meyer, H. F.: Echtzeioptimierung für Ausweichtrajektorien mittels der Sensitivitätsanalyse eines parametergestörten nichtlinearen Optimierungsproblems. Doktorarbeit.
- [12] A. Katriniok, P.Kleinbaum und M. Josevski: Distributed Model Predictive Control for Intersection Automation Using a Parallelized Optimization Approach, 2017.
- [13] Le, Hoang Long: Vergleich und Implementierung existierender numerischer NMPC-Lösungsverfahren, 14. September 2015.
- [14] Frasch, J.V., A. Gray, M. Zanon, H.J. Ferreau, S. Sager, F. Borrelli und M. Diehl: An Auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles.
- [15] Geisert, M. und N. Mansard: Trajectory generation for quadrotor based Systems using Numerical Optimal control.
- [16] Adamy, J.: Systemdynamik und Regelungstechnik III. Shaker, 2007.
- [17] Graichen, Knut: METHODEN DER OPTIMIERUNG UND OPTIMALEN STEUERUNG. Universität Ulm, Wintersemester 2012/2013.

-
- [18] Buchner, Alexander: Auf Dynamischer Programmierung basierende Nichtlineare Modell-prädiktive Regelung für LKW. 2010.
- [19] Quirynen, R., M. Vukobratovic, M. Zanon und M. Diehl: Autogenerating microsecond solvers for nonlinear MPC: a tutorial using ACADO integrators.
- [20] Wächter, A. und L. T. Biegler: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. http://www.optimization-online.org/DB_HTML/2004/03/836.html, 2006.
- [21] F. Althé, P. Polack und A. de la Fortelle: High-Speed Trajectory Planning for Autonomous Vehicles Using a Simple Dynamic Model.
- [22] Adamy, J.: Robotik und Künstliche Intelligenz. Technische Universität Darmstadt.
- [23] M Gerds, R. Henrion, D Hömberg und C. Landry: Path Planning And Collision Avoidance For Robots, September 2012.
- [24] Grant, M. und S. Boyd: Graph implementations for nonsmooth convex programs. in Recent Advances in Learning and Control, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd and H. Kimura, Eds. Springer-Verlag Limites, pp. 95-110, 2008.
- [25] J. Anderson, J. Gillis und M. Diehl: User Documentation for CasADi. http://casadi.sourceforge.net/users_guide/casadi-users_guide.pdf, v3.2.0+435.131e5db00 Auflage, 2017.
- [26] Albaker, B. M. und N. A. Rahim: A Survey of Collision Avoidance Approaches for Unmanned Aerial Vehicles.
- [27] Konigorski, Ulrich: Mehrgrößenreglerentwurf im Zustandsraum. Skript, Darmstadt, 2007.
- [28] Lunze, Jan: Regelungstechnik 1. Springer, Berlin etc., 6. Auflage, 2007.
- [29] Lunze, Jan: Regelungstechnik 2. Springer, Berlin etc., 4. Auflage, 2006.
- [30] Sezer, Volkan und Metin Gokasan: A novel obstacle avoidance algorithm: Follow the Gap Methode.
- [31] Ohtsuka, Toshiyuki: A continuation/GMRES method for fast computation of nonlinear receding horizon control.
- [32] Gao, Y., A. Gray, J.V. Frasch, T. Lin, E. Tseng und J.K. Hedrick: Spatial predictive control for agile semi-autonomous ground vehicles.
- [33] Oubbati, Mohamed: Einführung in die Robotik, WS 2009/2010.
- [34] Kretschmann, Ralf: Zeitoptimale Bahnplanung für Industrieroboter, 18. Dezember 2007.
- [35] <http://mazzoni.blogspot.de/2013/07/installing-aruco-augmented-reality.html>, August 2016.
- [36] Mitchell, Harvey B: Multi-sensor data fusion: an introduction, 2007.