

Studiengang: Elek. und Informationstechnik

Prüfer: Prof. Dr. rer. nat. habil. P. Levi

Betreuer: Dipl.-Inf. Andreas Tamke
Dr.-Ing. Thao Dang (Daimler AG)

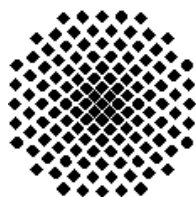
Diplomarbeit Nr. 3059

**Entwicklung und Vergleich von
Verfahren zur fahrzeugtauglichen
Bestimmung von Kritikalitätsmaßen
für komplexe Verkehrssituationen**

Adam Berthelot

begonnen am: 1. Juni 2010

beendet am: 22. Dezember 2010



Universität Stuttgart
Institut für Parallele und
Verteilte Systeme
Universitätsstraße 38
70569 Stuttgart, Germany



Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst
und nur die angegebenen Quellen benutzt zu haben.

(Adam Berthelot)

Sperrvermerk

Die vorliegende Arbeit beinhaltet interne vertrauliche Informationen der Firma Daimler AG. Die Weitergabe des Inhaltes der Arbeit und eventuell beiliegender Zeichnungen und Daten im Gesamten oder in Teilen ist grundsätzlich untersagt. Es dürfen keinerlei Kopien oder Abschriften - auch in digitaler Form - gefertigt werden. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma Daimler AG.

Die Diplomarbeit ist nur den Korrektoren sowie den Mitgliedern des Prüfungsausschusses zugänglich zu machen.

Zusammenfassung

Mit zunehmender Verkehrsdichte steigt die Gefahr von Verkehrsunfällen seit Jahren stetig an. Die Sicherheit von Fahrzeugen und deren Fähigkeit, den Fahrer zusätzlich durch Fahrerassistenzsysteme bei der Vermeidung von Verkehrsunfällen zu unterstützen, wird dadurch zu einem immer wichtigeren Kriterium bei der Auswahl eines Fahrzeugs.

Sicherheitsrelevante Fahrerassistenzsysteme können Kollisionen mit anderen Fahrzeugen oder Hindernissen abschwächen bzw. vermeiden. Sie verwenden häufig im Rahmen der Situationsanalyse sogenannte Zeitreserven, um die Kritikalität von potentiellen Kollisionssituationen zu bewerten und im Ernstfall einen Eingriff einer Assistenzfunktion auszulösen. Eine solche Zeitreserve kann beispielsweise den Kollisionszeitpunkt oder die Zeit bis zum letztmöglichen Zeitpunkt zur Durchführung einer kollisionsvermeidenden Vollbremsung definieren. Die Zeitreserven werden über verschiedene nichtlineare Methoden aus von Sensoren (Radar, Kamera, ...) eines Fahrzeugs gemessenen Umgebungsdaten errechnet.

Die Methoden zur Zeitreservenberechnung sind für komplexen Verkehrssituationen rechenaufwendig. Da die Rechenleistung der Steuergeräte der Fahrzeuge begrenzt ist, können nicht alle Verfahren fahrzeugtauglich ausgelegt werden. In dieser Arbeit wird eine auf neue innovativen Ansätze basierende effiziente Methode zur Zeitreservenberechnung vorgestellt. Diese Methode wurde im Rahmen der Arbeit konzipiert, umgesetzt und evaluiert.

Bei der Berechnung von Zeitreserven wird zumeist von der idealisierten Annahme ausgegangen, die Messwerte der Sensoren seien unverrauscht. In der Realität ist den Sensormesswerten allerdings stets ein Messrauschen überlagert. Aufgrund von Abhängigkeiten zwischen einzelnen Messgrößen, die auf Basis anderer Messgrößen berechnet werden, sind die Messgrößen zusätzlich untereinander korreliert.

Im zweiten Teil der Arbeit wird die zuvor vorgestellte Methode so erweitert, dass die Messunsicherheiten der verschiedenen Sensoren berücksichtigt werden. Hierfür wird unter der Annahme des multivariat normalverteilten Messrauschens, insbesondere die *Unscented Transformation* verwendet, um eine echtzeitfähige Approximation der Kollisionswahrscheinlichkeit zu berechnen. Die Verwendung von stochastischen Auslöseschwellen für die Fahrerassistenzsysteme erlaubt es, ihre Zuverlässigkeit und Robustheit zu erhöhen.

Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
1.2	Zielsetzung	4
1.3	Aufbau der Arbeit	5
2	Kritikalitätsbewertung mit Zeitreserven	6
2.1	Zeitreserven	6
2.2	Formulierung des Zeitreservenberechnungsproblems	7
2.3	Situationsprognose	10
2.4	Zeitreservenberechnung mit Vorwärtssimulation	10
2.4.1	Grundalgorithmus	11
2.4.2	Bewertung der Vorwärtssimulation	12
3	Effiziente Zeitreservenbestimmung	13
3.1	Mathematische Grundlagen	13
3.1.1	Aufbau der Distanzfunktion \mathcal{D}	13
3.1.2	Polynom Interporlation	17
3.1.3	Sturm-Algorithmus	20
3.2	Das kinematische Einspurmodell	22
3.2.1	Grundlagen des Einspurmodells	23
3.2.2	Euler-Cauchy Integration des kinematischen Einspurmosdells	24
3.2.3	Integration des Ackermann Modells	25
3.3	Zeitreservenbestimmung mit Interpolation der Distanzfunktion	29
3.3.1	Grundalgorithmus	29
3.3.2	Verwendung mehrere Polynome	30
3.3.3	Berechnung der Kollisionszeitreserve	31
3.3.4	Bestimmung von Reaktionszeitreserven	34
3.3.5	Direkte Berechnung von TTB	37
3.4	Versuchsaufbau : Auswertung der Zeitreservenbestimmung	39
3.4.1	Die Simulationsumgebung KSI	39
3.4.2	Implementierung der Zeitreservenberechnung	40
3.4.3	Integration in KSI	41
3.4.4	Szenario für die Auswertung : “Abbiegen”	43
3.4.5	Variationsbreite des Versuchsaufbaus	44
3.5	Ergebnisse	45

3.5.1	Relevante Auswertungsgrößen für TTC	45
3.5.2	Ergebnisse für die Berechnung der Zeitreserve TTC	46
3.5.3	Auswertung der Reaktionszeitreserven	49
3.5.4	Ergebnisse für die direkte Berechnung der Zeitreserve TTB	51
3.6	Fazit	51
4	Bestimmung der Kollisionswahrscheinlichkeit	53
4.1	Motivation	53
4.2	Zielsetzung	53
4.3	Stochastische Grundlagen	55
4.3.1	Normalverteilung	55
4.3.2	Multivariate Verallgemeinerung	56
4.3.3	Transformation von Wahrscheinlichkeitsverteilungen	57
4.3.4	Unscented Transformation (UT)	58
4.4	Bestimmung der Kollisionswahrscheinlichkeit	59
4.4.1	Prinzip der Methode	60
4.4.2	Untersuchung der Kollisionswahrscheinlichkeit mit Monte-Carlo Sampling	61
4.4.3	Approximationsmethode	62
4.4.4	Echtzeitfähige Approximationsmethode	64
4.5	Ergebnisse	66
4.5.1	Szenarien für die Auswertung	66
4.5.2	Bewertung des Ansatzes	67
4.5.3	Betrachtung einzelner Fällen	70
4.6	Fazit	75
5	Fazit und Ausblick	76
5.1	Fazit	76
5.2	Ausblick	76
	Literaturverzeichnis	78
A	Beweise	A-1
B	Bestimmung der Kollisionseigenschaften	B-1
C	Berechnung der Input Kovarianzmatrix	C-1
D	Linksseitige gesättigte Normalverteilung	D-1
E	Bestimmung der Aufenthaltswahrscheinlichkeit	E-1

Kapitel 1

Einleitung

1.1 Motivation

Mit zunehmender Verkehrsdichte auf den Straßen steigt die Gefahr von Verkehrsunfällen seit Jahren stetig an. Die Sicherheit von Fahrzeugen und deren Fähigkeit, den Fahrer zusätzlich durch Fahrerassistenzsysteme bei der Vermeidung von Verkehrsunfällen zu unterstützen, wird dadurch zu einem immer wichtigeren Kriterium beim Kauf eines Fahrzeugs. Seit vielen Jahren investiert die Daimler AG daher stark in die Entwicklung von elektronischen Sicherheitssystemen, die den Fahrer eines Fahrzeugs entweder durch Warnung, durch passiven Schutz oder durch aktives Eingreifen vor schweren, lebensgefährlichen Verletzungen bei einer möglichen Kollision schützen.

Zu den aktiv eingreifenden Systemen zählen sowohl solche, die sich nur auf den Fahrzustand des eigenen Fahrzeugs beschränken (z.B. ABS¹, ESP²) als auch Systeme, die durch die Erfassung und Berücksichtigung anderer Objekte in der Fahrzeugumgebung mittels Sensoren (z.B. Radar, Kamera) mögliche Kollisionspotenziale erkennen und rechtzeitig eingreifen, um eine Kollision abzuschwächen (Collision Mitigation) oder zu verhindern (Collision Avoidance). Aktuelle Collision Mitigation Systeme können hauptsächlich mit longitudinalen Verkehrssituationen umgehen. So kann z.B. mittels Blind Spot Monitoring (BSM) vor Fahrzeugen im „Toten Winkel“ gewarnt werden oder mittels Adaptive Cruise Control (ACC) der Abstand zu vorausfahrenden Fahrzeugen geregelt werden. Soll allerdings auch querender Verkehr beachtet werden, so werden die Anforderungen an ein Fahrerassistenzsystem deutlich höher, weil z.B. die Zeit von der ersten Erkennung eines potentiellen Kollisionspartners bis zu einer zur Abschwächung einer Kollision nötigen Reaktion deutlich geringer ist als bei longitudinalen Situationen.

Zur Zeit konzentriert sich die Forschung daher auf Ansätze, die sich nicht auf den Längsverkehr allein beschränken sondern auch eventuell auftretende laterale Kollisionen erkennen und abschwächen können sollen. Bei solchen Systemen ist es notwendig, mittels einer Situationsbewertung qualitative Aussagen über verschiedene Zeitmaße (wann muss beispielsweise spätestens gebremst werden, um einen Aufprall mit einem querenden Fahrzeug zu verhindern) machen zu können. Es wird hier von „Zeitreserven“ (siehe [1, 2]) gesprochen, die zur Quantifizierung der Kritikalität anhand eines Prädiktionsmodells angenommener potenzieller Kollisionssituationen eingesetzt werden. Solche Zeitreserven werden über verschiedene Ansätze aus von Sensoren (Radar, Kamera, ...) gemessenen Umgebungsdaten errechnet. Anhand der Zeitreserven kann in einer Situationsbewertung bestimmt werden, ob

¹Anti Blockiersystem

²Elektronisches Stabilitätsprogramm

ein Sicherheitssystem wie beispielsweise ein Notbremssystem eingreifen muss.

Wichtig dabei ist die Effizienz solcher Ansätze. Sie müssen dazu fähig sein, komplexere Verkehrssituationen, wie zum Beispiel eine Innenstadtkreuzung, mit wenig Rechenaufwand zu bearbeiten. Die Steuergeräte, mit welchen die Fahrzeuge ausgestattet sind, verfügen über beschränkte Rechenleistung.

Die idealisierte Annahme unverrauschter Messwerte ist für einen realen physikalischen Messprozess nicht haltbar, da es in der Realität keine Messung gibt, der nicht ein gewisses Messrauschen bzw. eine Messunsicherheit überlagert ist. Tatsächlich verhält es sich in der realen Welt eher wie in Abbildung 1.1. Ein Sensor misst die x - und y -Koordinaten eines querenden Fahrzeugs nur innerhalb einer gewissen Unschärfe, ihnen ist ein Messfehler überlagert. Es kann nicht mit Sicherheit gesagt werden, dass sich das Fahrzeug tatsächlich exakt an der Stelle befindet, an der der Sensor es vermutet.

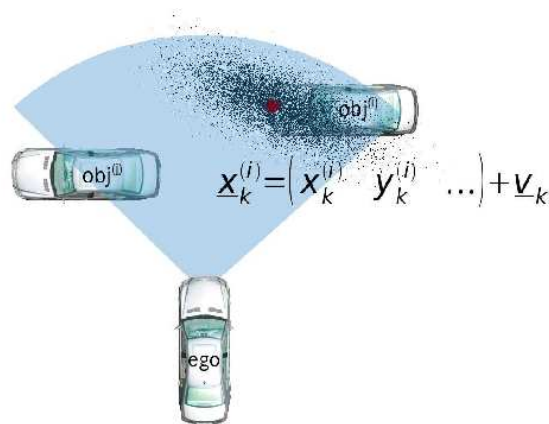


Abbildung 1.1: Realitätsgetreue Annahme verrauschter Messdaten.

Bei der Berechnung von Zeitreserven wird aktuell von der idealisierten Annahme unverrauschter Messwerte ausgegangen. Des Weiteren werden in der Realität nicht nur x - und y -Koordinaten gemessen, sondern auch Geschwindigkeiten, Beschleunigungen, Gierwinkel, Längen und Breiten von Fahrzeugen. Teilweise werden diese physikalischen Größen zusätzlich abhängig von anderen Messgrößen errechnet und sind deshalb untereinander korreliert.

1.2 Zielsetzung

Ziel und Aufgabenstellung dieser Arbeit ist es, ein Verfahren zur Berechnung von Zeitreserven zu entwickeln, das recheneffizienter als die schon existierenden Systeme wie die Vorwärtssimulation (siehe Abschnitt 2.4) ist. Ein solches Verfahren würde eine fahrzeugtaugliche Lösung zur Zeitreservenberechnung darstellen. Die Algorithmen sollen dabei in C implementiert werden, damit diese direkt auf dem Versuchsträger und mittels der Simulationsumgebung KSI getestet werden können. Die neue Methode soll im Vergleich zu der schon existierenden Methode zur Zeitreservenberechnung bewertet werden.

Des Weiteren soll auch ein Verfahren entwickelt werden, das die Kollisionswahrscheinlichkeit unter Berücksichtigung von Messrauschen auch in komplexen Verkehrssituationen bestimmen kann. Diese Kollisionswahrscheinlichkeit wird nämlich von Algorithmen benötigt, die die Verteilungen der Zeitreserven unter Berücksichtigung von Sensorunsicherheiten berechnen können.

1.3 Aufbau der Arbeit

Nach einer Hinführung zum Thema in Kapitel 1 befasst sich Kapitel 2 mit der Definition allgemeiner Begriffe der Situationsanalyse. Außerdem werden im Kapitel 2 die Prinzipien der Vorwärtssimulation erklärt.

Im Kapitel 3 wird der erste Teil der Arbeit, die Entwicklung einer neuen schnelleren Methode zur Zeitreservenberechnung, präsentiert. In diesem Kapitel werden zuerst die notwendigen theoretischen Grundlagen zu dieser Methode vorgestellt. Anschliessend werden eine ausführliche Vorstellung der Algorithmen zur Zeitreservenberechnung und ihre Ergebnisse ausführlich erklärt.

Kapitel 4 befasst sich mit dem zweiten großen Teil der Arbeit, der approximativen Bestimmung der Verteilungsfunktion der Kollisionswahrscheinlichkeit unter Berücksichtigung von Messunsicherheit. Hierbei werden Ideen und Ansätze diskutiert und analysiert, die bei der Approximation aufgegriffen werden. Schließlich werden die erzielten Ergebnisse der Kapitel 3 und 4 im Kapitel 5 zusammengefasst. Abschliessend wird ein Ausblick auf mögliche zukünftige Weiterentwicklungen gegeben.

Kapitel 2

Kritikalitätsbewertung mit Zeitreserven

In einem modernen Fahrerassistenzsystem erfüllt die Situationsbewertung die Aufgabe, die Kritikalität der vorliegenden Verkehrssituation für das eigene Fahrzeug zu bestimmen. Dabei benötigen Assistenzsysteme, die insbesondere in kritischen Verkehrersituationen aktiv werden, also Kollisionsminderung oder -vermeidung zum Ziel haben, die Ergebnisse der Kritikalitätsbewertung als Entscheidungsgrundlagen für die Aktivierung und Steuerung ihres Verhaltens, z.B. Einleitung einer Fahrerwarnung oder eines Vollbremsmanövers. Aber auch die Assistenzsysteme, die eher den Fahrkomfort adressieren, sind auf die Kritikalitätsbewertung der Situation angewiesen, vor allem um die Konfliktfreiheit der von ihnen geplanten autonom auszuführenden Manövern, wie z.B. Spurwechsel, verifizieren zu können.

Die Bewertung und Repräsentation der Kritikalität einer Verkehrssituation kann nur anhand geeigneter Kritikalitätsmaße erfolgen. Die Zeitreserven, die häufig auch als Zeitmaße bezeichnet werden stellen solche Maße dar. Die Zeitreserven bewerten die Kritikalität einer Situation anhand ihrer physikalisch-kinematischen Beschaffenheit, indem sie die relativen Zeitspannen bis zum Eintreten von bestimmten signifikanten Ereignissen quantisieren. Eine bevorstehende potentielle Kollision ist ein einfaches Beispiel für ein solches Ereignis, das durch die Kollisionszeitreserve Time-To-Collision (TTC) repräsentiert werden kann. Komplexere Zeitreserven können mögliche Aktionen bzw. Kollisionsvermeidungsmanöver des Fahrers oder des Assistenzsystems berücksichtigen.

Als Maße für die Kritikalität von Verkehrssituationen im Sinne eines Fahrerassistenzsystems weisen Zeitreserven zahlreiche positive Eigenschaften auf. So stellen sie quantitative und einfach zu interpretierbare Entscheidungsgrundlagen für das Systemverhalten dar.

2.1 Zeitreserven

Im Kontext von Fahrerassistenzsystemen charakterisieren Zeitreserven die Kritikalität einer vorliegenden Verkehrssituation aus der Sicht eines der Verkehrsteilnehmer. Dabei quantisiert eine Zeitreserve stets eine Zeitspanne bis zum Eintreten eines prognostizierten Ereignisses, welches für die Kritikalität der Situation aus der Perspektive des jeweiligen Verkehrsteilnehmers signifikant ist. Diese Zeitspanne wird in der Regel implizit relativ zum aktuellen Zeitpunkt t_0 angegeben, so dass sie durch einen skalaren Zeitwert repräsentiert werden kann. Üblicherweise gilt dabei $t_0 = 0$, und das durch die

Zeitreserve charakterisierte prognostizierte Ereignis muss innerhalb eines gewissen zeitlichen Prognosehorizonts t_H liegen.

In Abhängigkeit von der Art des charakterisierten Ereignisses können zwei Typen von Zeitreserven unterschieden werden:

1. Reaktionsfreie Zeitreserven:

Der reaktionsfreie Zeitreserven wird prognostizierten kritischen Ereignissen zugeordnet, die keine Reaktion des betrachteten Verkehrsteilnehmers, also des Fahrers oder eines Assistenzsystems des Egofahrzeugs, auf das Ereignis berücksichtigen.

Die für Assistenzsysteme vermutlich wichtigste Zeitreserve dieses Typs stellt die Kollisionszeitreserve Time-To-Collision (TTC) dar. Sie charakterisiert den Zeitpunkt an dem das Egofahrzeug mit einem Objekt kollidiert, falls der Fahrer oder ein Assistenzsystem keine Reaktion auf die bevorstehende Kollision zeigen, d.h. keine kollisionsvermeidende Manöver einleiten.

2. Reaktionszeitreserven:

Der zweite Typ von Zeitreserven wird Ereignissen zugeordnet, die eine potentielle Reaktion des Fahrers oder eines Assistenzsystems beinhalten.

So kann jedem Manöver eine Zeitreserve zugeordnet werden, die den spätesten Zeitpunkt charakterisiert, an dem durch das Einleiten dieses Manövers eine potentielle Kollision noch vermieden werden kann. Zum Beispiel können den Manövern Vollbremsen, Vollbeschleunigen oder Ausweichen durch maximalen Lenkeinschlag die Zeitreserven Time-To-Brake (TTB), Time-To-Kickdown (TTK) und Time-To-Steer (TTS) zugeordnet werden.

Eine für Assistenzsysteme, die Kollisionsminderung oder -vermeidung zum Ziel haben, sehr gut geeignete und häufig verwendete Zeitreserve ist die Reaktionszeitreserve Time-To-React (TTR). Diese quantisiert die Zeitspanne bis zum letztmöglichen Zeitpunkt, an dem die Kollision durch ein letztmögliches vom Fahrer oder vom Assistenzsystems eingeleitetes Vermeidungsmanöver gerade noch verhindert werden kann.

Die Gesamtbewertung der Kritikalität einer Verkehrssituation kann sich aus dem Zusammenwirken von mehreren unterschiedlichen Zeitreserven ergeben. In dieser Arbeit liegt das Augenmerk, in Anlehnung an die Arbeiten von Hillenbrand [1, 2], auf der Kritikalitätsbewertung anhand der Kollisionszeitreserve TTC und der Reaktionszeitreserve TTR sowie den Zeitreserven, die für die Bestimmung bzw. Approximation der beiden notwendig sind.

2.2 Formulierung des Zeitreservenberechnungsproblems

Die Basis für die Berechnung von Zeitreserven bilden die Beschreibung der aktuell vorliegenden Verkehrssituation sowie Annahmen über ihre zeitliche Weiterentwicklung (siehe *Situationsprognose* im Abschnitt 2.3).

Eine Situationsbeschreibung zum aktuellen Zeitpunkt t_0 basiert dabei auf den Ergebnissen der Umgebungserfassung und der Objektbildung. Sie kann im Sinne der Zeitreservenberechnung als eine endliche Menge von N kollidierbaren Objekten betrachtet werden, die sich im Raum befinden:

Ein Objekt der Situation wird dabei durch seinen Zustandsvektor $\mathbf{x}_i^c(t)$ repräsentiert, welcher sowohl veränderliche (z.B. Position) als auch konstante Größen (z.B. Form) beinhalten kann. Die Menge der Zustandsvektoren wird im Folgenden mit \mathbf{Z} bezeichnet.

Jedem Objekt kann eine der Klassen *statisch* (*stat*) oder *dynamisch* (*dyn*) zugeordnet werden. Die Objekte der Klasse *statisch* repräsentieren unbewegliche Bauobjekte. Ihre Zustandvektoren \mathbf{x}_i^{stat} bestehen nur aus zeitinvarianten Größen. Die Klasse *dynamisch* repräsentiert Verkehrsteilnehmer, die sich im Ablauf der Situation im Raum bewegen können. Die Weiterentwicklung ihrer Zustände und damit der Situation wird im Zeitintervall $\mathbf{T} = [t_0, t_0 + t_H]$ betrachtet, wobei t_H dem Zeithorizont der Zeitreservenberechnung entspricht. In der Regel gilt dabei $t_0 = 0$. Ein kompletter Zustandsverlauf eines Objektes i in einem Zeitintervall wird als Trajektorie γ_i bezeichnet. Die Menge aller möglichen Trajektorien eines Objektes i im Zeitintervall \mathbf{T} wird mit Γ_i angegeben.

Die beweglichen Verkehrsteilnehmer können in weitere Unterklassen (*Ego*-)Fahrzeug, Zweirad, Fußgänger eingeteilt werden. Die Zugehörigkeit zu einer der Unterklasse bestimmt dabei vor allem über die physikalisch-kinematischen Bewegungsmöglichkeiten sowie über mögliche Annahmen ihres Verhaltens. Die Zeitreserven werden zwar in der Regel nur für Fahrzeuge bestimmt, allerdings muss bei der Bestimmung von Zeitreserven das Verhalten aller Verkehrsteilnehmerklassen berücksichtigt werden.

Die Zustandsveränderung der Verkehrsteilnehmer, also insbesondere ihre Bewegung, wird durch die Abbildung

$$m_i : (\mathbf{x}_i(t), \mathbf{u}_i(t)) \mapsto m_i(\mathbf{x}_i(t), \mathbf{u}_i(t)) = \dot{\mathbf{x}}_i(t) \quad (2.2.1)$$

charakterisiert. Die Funktion m_i wird als Bewegungsmodell bezeichnet. Für Fahrzeuge wird das Bewegungsmodell auch Fahrzeugmodell genannt. Der Vektor $\mathbf{u}_i(t)$ wird als Steuereingangsvektor bezeichnet. Für Fahrzeuge beschreibt er die Steuergrößen, mit welchen der Fahrer oder ein Assistenzsystem, das Fahrzeugverhalten steuern können.

Die Menge \mathbf{I} bezeichnet die Menge aller Steuereingangsvektoren. Die Abbildung

$$\mathbf{u}_i : t \mapsto \mathbf{u}_i(t) \quad (2.2.2)$$

wird als Steuerfunktion bezeichnet. Sie ordnet jedem Zeitpunkt des betrachteten Zeitintervalls \mathbf{T} einen Steuereingangsvektor zu, entspricht also für Fahrzeuge dem (Steuer-)Verhalten des Fahrers oder eines Assistenzsystems. Die Menge \mathbf{U} ist die Menge aller möglichen Steuerfunktionen für das Zeitintervall \mathbf{T} .

Durch die zeitliche Integration des Bewegungsmodells m_i entsteht die Abbildung

$$\mathcal{M}_i : (\mathbf{x}_i(t_0), \mathbf{u}_i, t) \mapsto \mathcal{M}_i(\mathbf{x}_i(t_0), \mathbf{u}_i, t) = \mathbf{x}_i(t) \quad , \quad (2.2.3)$$

welche ausgehend von einem Startzustand $\mathbf{x}_i(t_0)$ und unter Vorgabe einer Steuerfunktion \mathbf{u}_i zu jedem Zeitpunkt t in \mathbf{T} den Objektzustand $\mathbf{x}_i(t)$ bestimmt und somit die Bewegungstrajektorie des Objektes beschreibt. Dies bedeutet insbesondere, dass für einen gegebenen Startzustand $\mathbf{x}_i(t_0)$ die Trajektorie γ_i eines Objektes ausschließlich von der Steuerfunktion \mathbf{u}_i bestimmt wird.

Bei der Modellierung der Weiterentwicklung der aktuellen Situationsbeschreibung werden die Startzustände $\mathbf{x}_i(t_0)$ der Objekte durch die Ergebnisse der Umgebungserfassung festgelegt. Das impliziert, dass die Bewegung eines Objekts im Zeitintervall \mathbf{T} mit Hilfe einer Prognosesteuerfunktion $\mathbf{u}_{P,i}$ geeignet modelliert werden muss. Diese spiegelt dabei das priori-Wissen bzw die Annahmen über das Verhalten des Objektes wider. Für Fahrzeuge kann $\mathbf{u}_{P,i}$ als Verhaltensmodell des Fahrers oder eines Assistenzsystems betrachtet werden.

Ausgehend von einem gegebenen Startzustand, kann jede Zeitreserve genau einmal pro angenommene

Funktion $\mathbf{u}_{P,i}$ bestimmt werden. Eine Trajektorie, die auf der Prognosesteuerfunktion $\mathbf{u}_{P,i}$ basiert, wird abgekürzt als $\gamma_{P,i}$ dargestellt.

Es wird angenommen, dass die Verkehrsteilnehmer in sich star sind, also eine konstante geometrische Form besitzen. Die Parameter der Objektform sind im Zustandsvektor $\mathbf{x}_i(t)$ inkludiert. Die Abbildung

$$\mathbf{s} : \mathbf{x}_i(t) \mapsto \mathbf{s}(\mathbf{x}_i(t)) = \mathbf{s}_i(t) \quad (2.2.4)$$

ordnet einem Objekt i die Raummenge (die Silhouette) zu, welche dieses zum Zeitpunkt t belegt. Die Funktion \mathbf{s} ist sowohl für statische als auch für bewegliche Objekte definiert. Das Ergebnis von \mathbf{s} kann anhand geometrischer Formen, z.B. anhand von Rechtecken oder Polygonen, oder auch direkt als Raummenge z.B. in einem Belegungsgitter repräsentiert werden.

Die Distanzfunktion

$$\mathcal{D} : (\mathbf{s}_i(t), \mathbf{s}_j(t)) \mapsto \mathcal{D}(\mathbf{s}_i(t), \mathbf{s}_j(t)) = \mathcal{D}_{i,j}(t) \quad \text{mit } i \neq j \quad (2.2.5)$$

beschreibt den minimalen Abstand zwischen den Silhouetten von zwei unterschiedlichen Objekten zu einem Zeitpunkt t . Das nicht zwangsläufig euklidische Abstandmaß ist dabei folgendermaßen definiert:

- $\mathcal{D}_{i,j}(t) = 0 \iff \mathbf{s}_i(t)$ und $\mathbf{s}_j(t)$ berühren sich
- $\mathcal{D}_{i,j}(t) < 0 \iff \mathbf{s}_i(t)$ und $\mathbf{s}_j(t)$ überlappen sich
- $\mathcal{D}_{i,j}(t) > 0$ sonst ($\mathbf{s}_i(t) \cap \mathbf{s}_j(t) = \emptyset$)

Die Funktion \mathcal{D} bildet einen Abstandmaß, und muß nicht einer Distanz im mathematischen Sinne entsprechen.

Wenn e dem Egofahrzeug entspricht wird $\mathcal{D}_{e,i}$ einfach \mathcal{D}_i bezeichnet.

Mit den erfolgten Definitionen lässt sich die Kollisionszeitreserve TTC für einen Verkehrsteilnehmer als Ergebnis folgender Gleichung formulieren:

$$TTC_i = \min\{t \in \mathbf{T} \mid d_i(t) = 0 \wedge \exists n \in \mathbb{N}, \forall k < n, d_i^{(k)}(t) = 0, d_i^{(n)}(t) < 0, \} \quad (2.2.6)$$

Dabei gilt die Annahme $d_i(t_0) \geq 0$.

Im Fall, dass kein Zeitpunkt in \mathbf{T} die Bedingung der Gleichung (2.2.6) erfüllt, also das Objekt i im Zeithorizont t_H unter den getroffenen Steuervorgabeannahmen $\mathbf{u}_{P,i}$ nicht kollidiert, ergibt sich $TTC_i = \min\{\emptyset\} = \infty$.

Ist die prognostizierte Trajektorie eines Objektes nicht kollisionsfrei, so können für das Objekt weitere Reaktionszeitreserven bestimmt werden. Bei der Formulierung solcher Zeitreserven geht man von einem hypothetischen Manöver aus, das das Objekt ausführen könnte, um drohende Kollision zu vermeiden. Ein Manöver \mathbf{m} wird im Sinne der Zeitreservenbestimmung als eine Funktion aufgefasst, die einer Vorgabe zur Veränderung der Steuerfunktion \mathbf{u}_i entspricht. Dabei kann ein Manöver zum beliebigen Zeitpunkt τ des Zeitintervalls \mathbf{T} ausgeführt werden. Der Manöverausführungszeitpunkt τ bestimmt die durch die Manöverausführung entstehende Trajektorie.

Anschauliches Beispiel: Die *Vollbremsung* entspricht einem einfachen Manöver, welches für sich allein gesehen nur die Vorgabe enthält, ab dem Manöverausführungszeitpunkt die maximale Bremsung ($a = a_{min}$) durchzuführen. Die *Vollbremsung* kann zu jedem Zeitpunkt τ einer gegenen Objekttrajektorie eingeleitet werden. Sie erzeugt dabei für jedes τ eine andere (Manöver-)Trajektorie.

Die Menge aller Manöver wird mit \mathbf{M} bezeichnet. Eine Manöverausrührung \mathcal{U} stellt formal folgende Abbildung dar:

$$\mathcal{U} : (\mathbf{u}_{P,i}, \mathbf{m}, \tau) \mapsto \mathcal{U}(\mathbf{u}_{P,i}, \mathbf{m}, \tau) = \mathbf{u}_{m,\tau,i} \quad (2.2.7)$$

Sie erzeugt also aus der vorgegebenen Prognosesteuerfunktion $\mathbf{u}_{P,i}$ eine neue Steuerfunktion, die die Einleitung eines Manövers \mathbf{m} zum Zeitpunkt τ beinhaltet. Die Trajektorie die aus $\mathbf{F}_c(\mathbf{x}_i(t_0), \mathbf{u}_{m,\tau,i}, t) = \mathbf{x}_{m,\tau,i}(t)$ resultiert wird im Folgenden mit $\gamma_{m,\tau,i}$ bezeichnet. Die Distanzfunktionen für Objektsilhouetten, die durch Manöverausrührung entstehen, werden mit $d_{m,\tau,i,j}(t)$ bzw $d_{m,\tau,i}(t)$ abgekürzt.

Jedem Manöver \mathbf{m} kann eine Zeitreserve Time-To-Maneuver (TTM oder häufig TTx) zugeordnet werden, welche den letztmöglichen Zeitpunkt zur Einleitung von \mathbf{m} charakterisiert, so dass die prognostizierte Kollision vermieden wird. Dies entspricht folgender Optimierung:

$$TTM_i = \max\{\tau \in]-\infty, TTC_i[\mid \min_{t \in \mathbf{T}} \{d_{m,\tau,i}(t)\} \geq 0\} \quad (2.2.8)$$

Dabei wird also gefordert, dass die durch das Manöver erzeugte Trajektorie bis zum Zeithorizont kollisionsfrei bleibt. Der Zeitpunkt der Manövereinleitung τ kann auch kleiner t_0 sein, was bedeutet, dass das Manöver vor t_0 hätte eingeleitet werden müssen.

2.3 Situationsprognose

Die Situationsprognose besteht darin, eine quantitative und/oder qualitative Weiterentwicklung der Situation anhand der Situationsbeschreibung so gut wie möglich zu präzisieren. Eine quantitative Situationsprognose könnte sein : "Objekt o_1 wird nach τ Sekunden auf der Position (x, y) sein". Ebenso könnte eine qualitative Prognose sein : "Fahrzeug f_1 wird abbiegen".

Eine robuste und effiziente Situationsprognose kann aus verschiedenen Komponenten bestehen, wie zum Beispiel :

- Fahrerabsichterkennung :
 - Überwachung der Blinker
 - Route wurde im Navi programmiert
- Benutzung von GPS Daten :
 - Fahrzeug bremst während er sich einer Kreuzung oder einer Kurve nähert
- ein System erkennt die Fahrmuster eines Fahrers

Im Rahmen dieser Arbeit wird es angenommen, dass die Situationprognose daraus besteht, dass die Steuergröße konstant bleiben:

$$\mathbf{u}(t) = \mathbf{u}_0, \forall t \in \mathbf{T} \quad (2.3.1)$$

2.4 Zeitreservenberechnung mit Vorwärtssimulation

Die Vorwärtssimulation [3] stellt ein sehr flexibles iteratives numerisches Verfahren zur Zeitreservenbestimmung dar. Sie verfügt folgende positive Eigenschaften :

- Sie kommt mit beliebigen Bewegungstrajektorien zurecht
- Sie kann die Bewegung der Verkehrsteilnehmer anhand komplexer Fahrzeugmodelle genau abbilden
- Sie bestimmt Zeitreserven für beliebig komplexe Kollisionsvermeidungsmanöver
- Sie berücksichtigt implizit alle Objekte, also auch Folgekollisionen, die auf Grund eines Kollisionsvermeidungsmanövers entstehen können

In diesem Abschnitt werden die grundsätzliche Funktionsweise der Vorwärtssimulation beschrieben.

2.4.1 Grundalgorithmus

Die Vorwärtssimulation ist ein flexibles modulares Verfahren, das es erlaubt, eine vorliegende Situation unter Verwendung von Verhaltensannahmen in zeitdiskreten Schritten iterativ in die Zukunft zu extrapolieren (simulieren). Das Ziel liegt darin, getroffene Hypothesen auf Eintreten bestimmter Ereignisse zu untersuchen. Mit der Vorwärtssimulation können ganz unterschiedliche Fragestellungen der Situationsanalyse untersucht werden. Die Bestimmung von Zeitreserven stellt also nur eine mögliche Anwendung der Vorwärtssimulation dar.

Diskretisierung

Das bei der Zeitreservenbestimmung betrachtete Zeitintervall $\mathbf{T} = [t_0, t_0 + t_H]$ wird mit einer gewählten Schrittweite Δt diskretisiert: $(t_0, \dots, t_k, \dots, t_n)$

Für den Index k gilt dabei $k \in \{0, \dots, n\}$ mit $n = \frac{t_H}{\Delta t}$. Die Zeitpunkte t_k werden im Folgenden auch als Zeitstützstellen bezeichnet.

In der Regel wird $t_0 = 0$ gewählt. Der für die Zeitreservenberechnung interessanter Zeithorizont liegt bei $3 \text{ s} \leq t_H \leq 5 \text{ s}$. Für die Schrittweite Δt gilt, je kleiner desto genauer die Rechenergebnisse. Aber der Rechenaufwand und der Speicherbedarf wachsen mit sinkendem Δt .

Iteration über die Zeitstützstellen

In der Vorwärtssimulation wird die Entwicklung des Zustandes $\vec{x}(t)$ eines bewegten Objektes i nur für die Zeitstützstellen t_k explizit berechnet. Es gilt $\mathbf{x}_i(t_k) = \mathbf{x}_{k,i}$.

Das Bewegungsmodell $m(\mathbf{x}_i(t), \mathbf{u}_i(t)) = \dot{\mathbf{x}}_i(t)$ wird in der Vorwärtssimulation als ein zeitdiskretes numerisches Modell $m_N(\mathbf{x}_{k,i}, \mathbf{u}_{k,i}, \Delta t) = \Delta \mathbf{x}_{k+1,i}$ umgesetzt. Die Abbildung $\mathcal{M}_{N,i}(\mathbf{x}_i(t_0), \mathbf{u}_i, t) = \mathbf{x}_i(t)$ wird durch die iterative k -fache Hintereinanderausführung von m_N umgesetzt. Sie kann als $\mathcal{M}_{N,i}(\mathbf{x}_{0,i}, \mathbf{u}_i, \Delta t, k) = \mathbf{x}_{k,i} = \mathbf{x}_{k-1,i} + \Delta \mathbf{x}_{k,i}$ dargestellt werden.

Die Steuervorgabefunktionen \mathbf{u}_i werden nur an den Zeitstützstellen t_k abgetastet. Es gilt $\mathbf{u}_i(t_k) = \mathbf{u}_{k,i}$.

Eine Zustandstrajektorie γ_i kann als n -Tupel der entwickelten Objektzustände dargestellt werden. $\gamma_i = (\mathbf{x}_{0,i}, \dots, \mathbf{x}_{k,i}, \dots, \mathbf{x}_{n,i})$.

Zu jedem diskreten Zustand $\mathbf{x}_{k,i}$ wird die Silhouette $\mathbf{s}(\mathbf{x}_{k,i}) = \mathbf{s}_{k,i}$ bestimmt.

Die Distanzfunktion $\mathcal{D}_{k,i,j} = \mathcal{D}(\mathbf{s}_{k,i}, \mathbf{s}_{k,j})$, die den Kollisionsabstand zwischen zwei Silhouetten zum Zeitpunkt t bestimmt, wird auf die diskreten Silhouetten $\mathbf{s}_{k,i}, \mathbf{s}_{k,j}$ angewendet. Im Fall der

Vorwärtssimulation ist der Abbildungsraum der Distanzfunktion die Menge $\{0, 1\}$. Dabei wird die Distanzfunktion folgendermaßen definiert:

$$\begin{aligned} \mathcal{D}_{k,i,j} = 1 &\iff \mathbf{s}_{k,i} \cap \mathbf{s}_{k,j} = \emptyset \\ \mathcal{D}_{k,i,j} = 0 &\iff \mathbf{s}_{k,i} \cap \mathbf{s}_{k,j} \neq \emptyset \end{aligned} \tag{2.4.1}$$

Durch die Berechnung von $d_{k,i,j}$ für alle Zeitpunkte k entsteht ein n -Tupel von Distanzwerten $(d_{0,i,j}, \dots, d_{k,i,j}, \dots, d_{n,i,j})$. Der erste Nullwert dieses n -Tupels wird im Sinne der Vorwärtssimulation als Kollision gewertet.

Die Suche nach TTM für ein bestimmtes Manöver \mathbf{m} erfolgt durch wiederholtes testen von Manöverausführungen mit unterschiedlichen Startzeitpunkten τ entstandenen Trajektorien $\gamma_{\mathbf{m},\tau,k,i}$ auf kollisionsfreiheit. Entscheidend dabei ist die Suchstrategie. Zwei Strategien sind möglich :

- eine lineare Suche
- Binärsuche

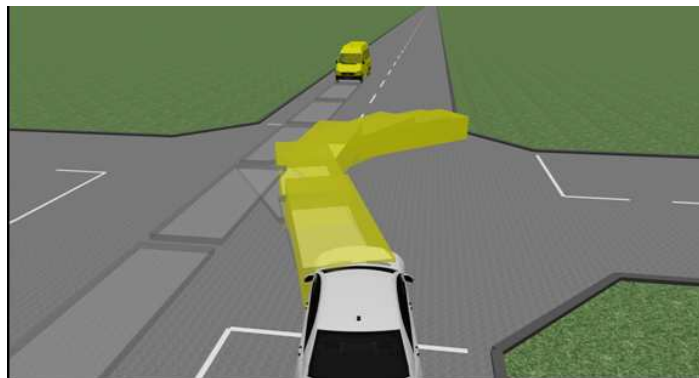


Abbildung 2.1: Die Vorwärtssimulation ist ein iteratives Verfahren, das zur Zeitreservenbestimmung eingesetzt wird

2.4.2 Bewertung der Vorwärtssimulation

Wie erwähnt stellt die Vorwärtssimulation ein sehr flexibles iteratives numerisches Verfahren zur Zeitreservenbestimmung dar.

Die Vorwärtssimulation ist aber auch ein vergleichsweise Rechenaufwendiges Verfahren. Dadurch, dass es ein iteratives Verfahren ist, müssen viele Fahrzeugzustände berechnet und bearbeitet werden. Diese Zustände können auch nicht parallel berechnet werden, denn das numerische Fahrzeugmodell m_N einen Zustand braucht, um den nächsten zu bestimmen. Die Erhöhung des Zeitschritts Δt führt aber auch zu einer Verminderung der Genauigkeit des numerischen Fahrzeugmodells und des Verfahrens insgesamt.

Im nächsten Kapitel wird ein Verfahren vorgestellt, das auf das iterative Fahrzeugmodell verzichtet. Dadurch wird die Recheneffizienz des Verfahrens erhöht.

Kapitel 3

Effiziente Zeitreservenbestimmung

3.1 Mathematische Grundlagen

3.1.1 Aufbau der Distanzfunktion \mathcal{D}

Im Abschnitt 2.2 wird eine Distanz \mathcal{D} erwähnt, die einen Abstandsmaß zwischen das Ego-Fahrzeug und einen Objekt liefert. In diesem Abschnitt wird eine mögliche Realisierung dieser Distanz \mathcal{D} präsentiert. Im Fall dieser Arbeit werden nur die Beziehungen zwischen das Ego-Fahrzeug und die anderen Objekte betrachtet. Das Ego-Fahrzeug wird durch einen Rechteck repräsentiert, während die Objekten (andere Fahrzeug, Leitplanke, Gebäude) durch Polygone dargestellt werden. Dafür wird ein Abstandmaß \mathcal{D}_P zwischen Rechteck (das Ego-Fahrzeug) und Punkt berechnet, anschliessend wird ein Abstandmaß \mathcal{D}_S zwischen Rechteck und Liniensegment aufgezeigt. Das Abstandmaß \mathcal{D}_S wird auf jede Seite der Objekt-Polygone angewendet, um das Abstandmaß \mathcal{D} zwischen dem Ego-Fahrzeug und einem Objekt zu bestimmen.

3.1.1.1 Normen, euklidischer und nichteuklidischer Abstand

Definition 1. Es sei E ein \mathbb{k} -Vektorraum. Ein Abbildung $\|\cdot\| \rightarrow \mathbb{R}_+$ heißt Norm auf E , wenn die folgenden Eigenschaften für alle $(x, y, \lambda) \in E \times E \times \mathbb{k}$ gelten :

$$\|x\| = 0 \Leftrightarrow x = 0_E \quad (\text{Definitheit})$$

$$\|\lambda \cdot x\| = |\lambda| \cdot \|x\| \quad (\text{Homogenität})$$

$$\|x + y\| \leq \|x\| + \|y\| \quad (\text{Dreiecksungleichung})$$

Definition 2. Ein Abstand, auch bekannt als Distanz oder Metrik, ist eine Funktion $d : E \times E \rightarrow \mathbb{R}_+$, die für jedes Paar von Elementen $x, y \in E$ eine reelle positive Zahl $d(x, y)$ zuordnet mit folgenden Eigenschaften :

$$d(x, y) \geq 0, d(x, y) = 0 \Leftrightarrow x = y \quad \forall x, y \in X \quad (\text{Nichtnegativität})$$

$$d(x, y) = d(y, x) \quad (\text{Symmetrie})$$

$$d(x, y) \leq d(x, z) + d(y, z) \quad \forall x, y, z \in X \quad (\text{Dreiecksungleichung})$$

(E, d) wird auch metrisches Raum genannt.

Jede Norm induziert durch $\|x - y\| = d(x, y)$ eine Distanz.

Es existieren auf dem Vektorraum \mathbb{R}^2 mehrere Normen. Die bekannteste ist die euklidische Norm, mit dem von ihr induzierten euklidischen Abstand. Die euklidische Norm ist definiert durch :

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\| = \sqrt{x^2 + y^2}, \quad \forall (x, y) \in \mathbb{R} \times \mathbb{R} \quad (3.1.1)$$

Verallgemeinerung der euklidischen Norm ist die n -Norm. Die n -Norm ist so definiert :

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_n = \sqrt[n]{|x|^n + |y|^n}, \quad \forall (x, y) \in \mathbb{R} \times \mathbb{R}, n \in \mathbb{N}_+^* \quad (3.1.2)$$

Die unendliche Norm, ∞ -Norm, ist der Grenzfall für $n \rightarrow \infty$ und ist wie folgt definiert :

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_\infty = \max(|x|, |y|), \quad \forall (x, y) \in \mathbb{R} \times \mathbb{R} \quad (3.1.3)$$

Die Einheitssphären dieser Normen werden in der Abbildung 3.1 dargestellt. Die Sphären für die 1-Norm und die ∞ -Norm sind nicht kreisförmig sondern quadratisch.

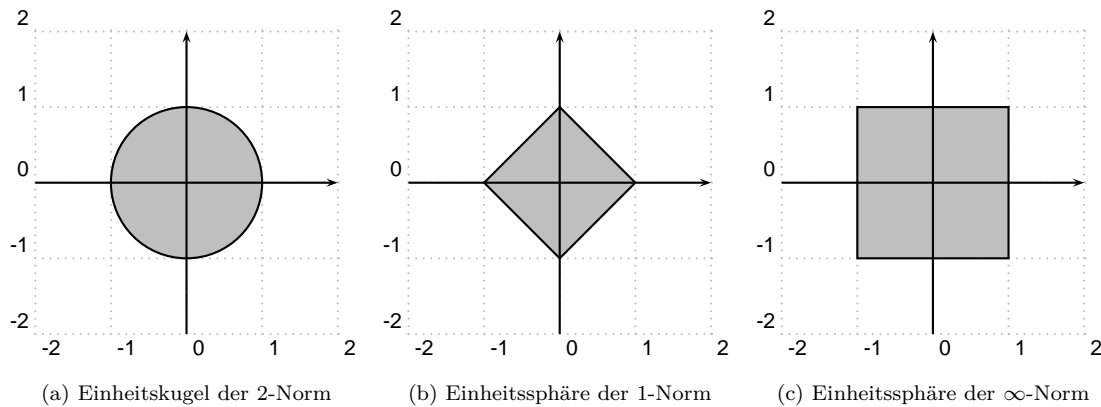


Abbildung 3.1: Einheitssphäre für verschiedene Normen auf \mathbb{R}^2

Wenn $P = \begin{pmatrix} x \\ y \end{pmatrix}$ ein Punkt aus \mathbb{R}^2 ist, dann ist $\left| \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_\infty - 1 \right|$ der Abstand von P zur Kontur der Einheitssphäre. Die Zahl $\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_\infty - 1$ gibt auch eine zusätzliche Information, da sie negativ sein kann :

- sie ist positiv, wenn P außerhalb der Einheitssphäre ist
- sie ist negativ, wenn P innerhalb der Einheitssphäre ist

3.1.1.2 Abstandsmaß zwischen Rechteck und Punkt mit der ∞ -Norm

Um ein Abstandsmaß zwischen Rechteck und Punkt zu haben, braucht man nur den Raum so zu verschieben, zu drehen und zu skalieren, dass das Rechteck der Einheitssphäre der ∞ -Norm entspricht. Im normalen Raum wird ein Rechteck durch die Koordinaten seines Zentrums (x_c, y_c) ,

seine Orientierung ψ , und seine Länge und Breite (l, b) . Diese Transformation \mathcal{T} für einen Punkt $P = \begin{pmatrix} x \\ y \end{pmatrix}$ vom normalen Raum zum transformierten Raum kann so dargestellt werden :

$$\mathcal{T} \begin{pmatrix} x \\ y \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{2}{l} & 0 \\ 0 & \frac{2}{b} \end{pmatrix}}_{\text{Skalierung}} \cdot \underbrace{\begin{pmatrix} \cos(-\psi) & -\sin(-\psi) \\ \sin(-\psi) & \cos(-\psi) \end{pmatrix}}_{\text{Rotation}} \cdot \underbrace{\begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}}_{\text{Verschiebung}} \quad (3.1.4)$$

Die Transformation T ist eine affine Transformation : Geraden bleiben Geraden, die Parallelität bleibt erhalt. Winkel bleiben nicht erhalten. Die Abbildung 3.2 stellt diese Tranformation dar.

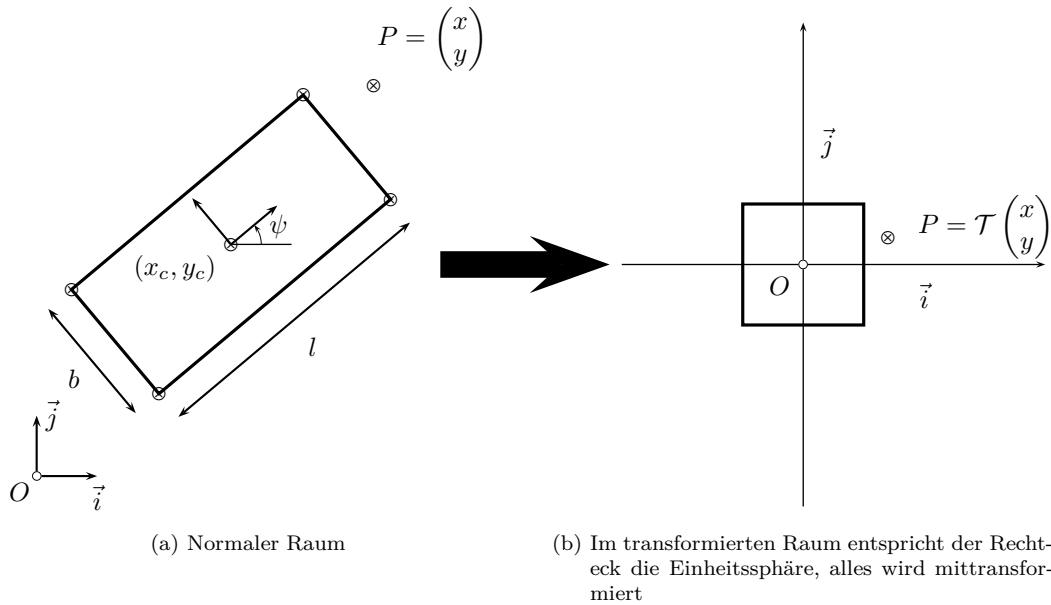


Abbildung 3.2

$$\mathcal{D}_P \begin{pmatrix} x \\ y \end{pmatrix} = \left\| \mathcal{T} \begin{pmatrix} x \\ y \end{pmatrix} \right\|_{\infty} - 1 \text{ bildet ein Abstandsmaß zwischen Rechteck und Punkt.}$$

3.1.1.3 Abstandsmaß zwischen Rechteck und Segment mit der ∞ -Norm

Das Ziel dieses Abschnittes ist eine Methode aufzuzeigen, um ein Abstandsmaß zwischen dem Ego-Fahrzeug und einer Seite eines Objektes zu haben. Dieses Abstandsmaß soll aussagen, ob ein Teil der Seite im Rechteck liegt oder nicht.

Zuerst soll die Seite $[S]$ des Objektes mit der Abbildung \mathcal{T} transformiert werden: der resultierende Segment wird mit $[S']$ bezeichnet. Dann soll der Punkt $\begin{pmatrix} x' \\ y' \end{pmatrix}$ aus $[S']$, der am nächsten (im Sinne der ∞ -Norm) zum Ursprung des transformierten Raums liegt, berechnet werden. Der Abstandsmaß \mathcal{D}_P wird auf diesen Punkt angewendet : $\mathcal{D}_S([S]) = \left\| \begin{pmatrix} x' \\ y' \end{pmatrix} \right\|_{\infty} - 1 = \left\| \mathcal{T} \begin{pmatrix} x \\ y \end{pmatrix} \right\|_{\infty} - 1$.

Drei Fälle sind möglich:

- $\mathcal{D}_S(S) < 0$ die Seite liegt ganz oder teilweise im Rechteck
- $\mathcal{D}_S(S) = 0$ die Seite berührt das Rechteck

- $\mathcal{D}_S(S) > 0$ die Seite liegt außerhalb des Rechteckes

Jede dieser Aussage ist in beiden Räumen (normalen und transformierten) gültig.

Um den Punkt $\begin{pmatrix} x' \\ y' \end{pmatrix}$ aus $[S']$ eindeutig bestimmen zu können, wird der Satz A benutzt :

Satz 1. Sei $[AB]$ ein Segment. Mit der durch die unendliche Norm induzierten Distanz, ist der Abstand des Segments zum Ursprung das Minimum der beiden Abstände :

- der Abstand von A oder B zum Ursprung
- der Abstand des Schnittpunktes zwischen dem Segment und der ersten oder zweiten Diagonale

Der Beweis für diesen Satz kann im Appendix A gefunden werden. Die Abbildung 3.3 stellt den Sachverhalt anhand von zwei Beispielen dar.

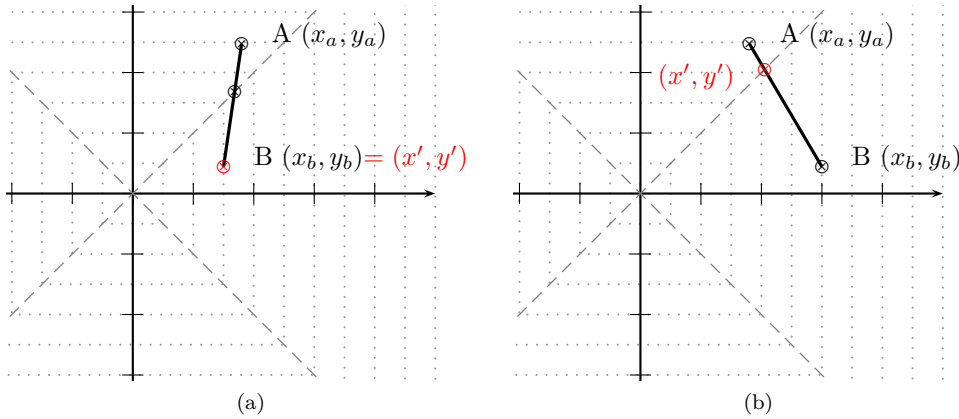


Abbildung 3.3: Der nächste Punkt eines Segmentes zum Ursprung ist entweder ein Endpunkt oder liegt auf der 1. oder 2. Winkelhalbierenden.

Nach dem Satz A, sind nur die Abstände von vier verschiedenen Punkte notwendig, um den durch die ∞ -Norme induzierten Abstand eines Segments zum Ursprung zu bestimmen.

Die Abstände der Endpunkte sind eigentlich schon bekannt. Die Abstände der Schnittpunkte zwischen dem Segment und den Winkelhalbierenden erst berechnet werden, wenn diese Schnittpunkte bestimmt worden sind. Im Folgenden wird die Berechnung dieser Schnittpunkte erklärt.

Ein Segment $[AB]$ ist definiert durch :

$$[AB] = \{P(x, y) \mid x = (x_b - x_a) \cdot t + x_a, y = (y_b - y_a) \cdot t + y_a, \forall t \in [0; 1]\}$$

Wenn es den Schnittpunkt P_1 zwischen dem Segment und der ersten Winkelhalbierende gibt, dann es existiert $t_1 \in [0; 1]$ so, dass :

$$\begin{aligned} x(t_1) = y(t_1) &\iff x_a + (x_b - x_a) \cdot t_1 = y_a + (y_b - y_a) \cdot t_1 \\ &\iff x_a - y_a = t_1 \cdot ((y_b - y_a) - (x_b - x_a)) \\ &\implies t_1 = \frac{x_a - y_a}{(y_b - y_a) - (x_b - x_a)} \text{ wenn } y_b - y_a \neq x_b - x_a \end{aligned}$$

Wenn $y_b - y_a = x_b - x_a$ oder wenn das berechnete $t_1 \notin [0; 1]$, gibt es keinen Schnittpunkt zwischen dem Segment und der ersten Winkelhalbierende. Wenn P_1 existiert und berechnet worden ist, kann sein Abstand berechnet werden.

Wenn es den Schnittpunkt P_2 zwischen dem Segment und der ersten Winkelhalbierende gibt, dann es existiert $t_2 \in [0; 1]$ so, dass :

$$\begin{aligned} x(t_2) = -y(t_2) &\iff x_a + (x_b - x_a) \cdot t_2 = -y_a - (y_b - y_a) \cdot t_2 \\ &\iff x_a + y_a = -t_2 \cdot ((y_b - y_a) + (x_b - x_a)) \\ &\implies t_2 = \frac{-(x_a + y_a)}{(y_b - y_a) + (x_b - x_a)} \text{ wenn } (y_b - y_a) \neq -(x_b - x_a) \end{aligned}$$

Wenn $y_b - y_a = -(x_b - x_a)$ oder wenn das berechnete $t_2 \notin [0; 1]$, gibt es keinen Schnittpunkt zwischen dem Segment und der zweite Winkelhalbierende. Wenn P_2 existiert und berechnet worden ist, kann sein Abstand berechnet werden.

$$\mathcal{D}_S([P_1 P_2]) = \min \left\{ \left\| \begin{pmatrix} x_a \\ y_a \end{pmatrix} \right\|_\infty, \left\| \begin{pmatrix} x_b \\ y_b \end{pmatrix} \right\|_\infty, \left\| \begin{pmatrix} x_{t_1} \\ y_{t_1} \end{pmatrix} \right\|_\infty, \left\| \begin{pmatrix} x_{t_2} \\ y_{t_2} \end{pmatrix} \right\|_\infty \right\} - 1 \quad (3.1.5)$$

3.1.2 Polynom Interpolation

3.1.2.1 Interpolation

Einer der einfachsten Wege eine Funktion zu Approximieren ist die Interpolation. Eine polynomische Interpolation des Grades n einer gegebenen stetigen Funktion f auf dem Intervall $I = [a; b]$, besteht darin einen Polynom zu berechnen, der an den Punkten $x_0 = a, x_1, \dots, x_n = b$ den Wert von $f(x_i)$ nimmt. Ein Weg, die Funktion f an x_0, x_1, \dots, x_n , mit dem Polynom $P(x) = c_0 + c_1 \cdot x + \dots + c_n \cdot x^n$ zu interpolieren, besteht darin das lineare System (3.1.6) zu lösen.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \dots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \dots \\ f(x_n) \end{pmatrix} \iff A \cdot c = b \quad (3.1.6)$$

$$\det(A) = \prod_{0 \leq i < j \leq n} (x_j - x_i) \quad (3.1.7)$$

Die Determinante der Matrix A wird in der Gleichung (3.1.7) ausgedrückt. Die Matrix A wird Vandermonde Matrix benannt. Ist A nicht singular ($\det(A) \neq 0$), dann hat das System (3.1.6) eine einzige Lösung, die mit Hilfe der Inverse der Matrix A berechnet werden kann :

$$c = A^{-1} \cdot b \quad (3.1.8)$$

3.1.2.2 Effiziente Inversion der Vandermonde Matrix

Die Vandermonde-Matrix kommt typischerweise bei der Polynominterpolation vor. In diesem Abschnitt wird eine effiziente Methode zur Inversion dieser Matrix erläutert. Diese Methode wurde von R. Turner entwickelt, und wird in einem technischen Bericht der NASA [4] vorgestellt. Für diesen Abschnitt wird die Matrix umgeschrieben, damit die Indizen der x_i zu den Indizen der Zeilen der

Matrix passen :

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^{n-1} \end{pmatrix} \quad (3.1.9)$$

Für paarweise unterschiedliche x_i ist die Determinante von A ungleich null. Um das System (3.1.6) zu lösen, muss die Matrix A invertiert werden. Dafür wird die LU-Zerlegung benutzt : Die LU-Zerlegung ist ein Verfahren, in dem die Matrix A als Produkt zweier Matrizen L und U dargestellt wird : $A = L \cdot U$. L ist eine untere Dreiecksmatrix (englisch :“lower”) und U eine obere Dreiecksmatrix (‘‘upper’’). Die inverse Matrix von A kann mit Hilfe der Inversen Matrizen L^{-1} und U^{-1} berechnet werden :

$$A^{-1} = U^{-1} \cdot L^{-1} \quad (3.1.10)$$

R. Turner gibt eine explizite Beschreibung der Matrizen L^{-1} und U^{-1} : die Koeffizienten $l_{i,j}$ von L^{-1} sind :

$$\begin{aligned} l_{1,1} &= 1 \\ l_{i,j} &= 0 \quad i < j \\ l_{i,j} &= \sum_{\substack{k=1 \\ k \neq j}}^i \frac{1}{x_i - x_k} \quad \text{sonst} \end{aligned} \quad (3.1.11)$$

Explizite Darstellung von L^{-1} ist :

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 & \dots \\ \frac{1}{x_1 - x_2} & \frac{1}{x_2 - x_1} & 0 & \dots \\ \frac{1}{(x_1 - x_2) \cdot (x_1 - x_3)} & \frac{1}{(x_2 - x_1) \cdot (x_2 - x_3)} & \frac{1}{(x_3 - x_1) \cdot (x_3 - x_2)} & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \quad (3.1.12)$$

Die Koeffizienten $u_{i,j}$ von U^{-1} sind :

$$\begin{aligned} u_{i,i} &= 1 \\ u_{i,1} &= 0 \\ u_{i,j} &= u_{i-1,j-1} - u_{i,j-1} \cdot x_{j-1} \quad \text{sonst, mit } u_{0,j} = 0 \end{aligned} \quad (3.1.13)$$

Explizite Darstellung von U^{-1} ist :

$$U^{-1} = \begin{pmatrix} 1 & -x_1 & x_1 \cdot x_2 & -x_1 \cdot x_2 \cdot x_3 & \dots \\ 0 & 1 & -(x_1 + x_2) & x_1 \cdot x_2 + x_2 \cdot x_3 + x_3 \cdot x_1 & \dots \\ 0 & 0 & 1 & -(x_1 + x_2 + x_3) & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (3.1.14)$$

Durch die einfache Multiplikation von U^{-1} mit L^{-1} ist A^{-1} zu berechnen.

3.1.2.3 Fehler bei der Interpolation

Wenn die Interpolationsstützstellen linear verteilt gewählt werden, kann das Polynom an den Grenzen des Intervalls schwanken und nicht gegen die zu interpolierende Funktion konvergieren. Das wird als Runge Phänomen bezeichnet (Siehe Abbildung 3.4). Damit dieses Phänomen vermieden wird, müssen die Stützstellen anders gewählt werden ([5],[6]).

Theorem 1 (Runge Phänomen, [6]). *Wenn die Stützstellen $\{x_k\}_{k=0}^n$ werden linear verteilt gewählt, dann konvergiert der Interpolationspolynom $p_n(x)$ nicht gleichmäßig auf $[-1; 1]$ mit $n \rightarrow \infty$ gegen die Funktion $f(x) = \frac{1}{1 + 25 \cdot x^2}$.*

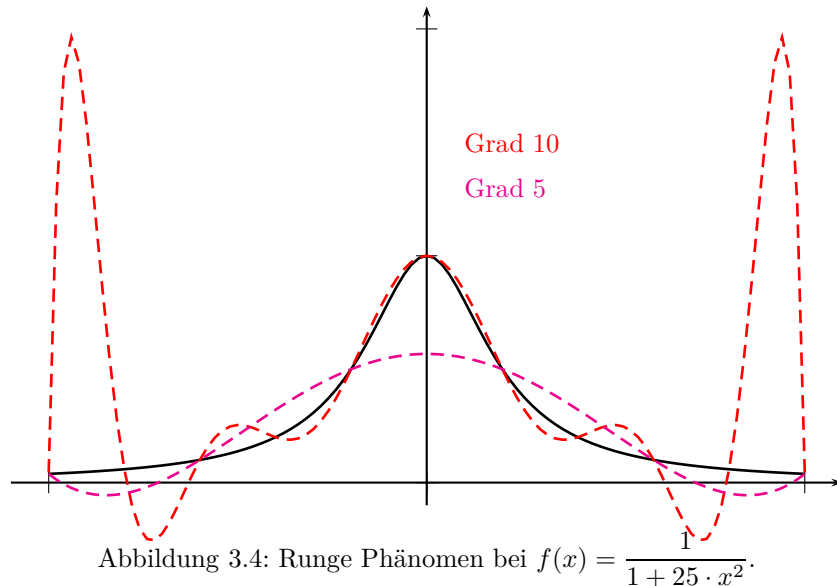


Abbildung 3.4: Runge Phänomen bei $f(x) = \frac{1}{1 + 25 \cdot x^2}$.

Theorem 2 (Theorem von Erdős und Turán, 1937, [6]). *Wenn f auf $[a; b]$ stetig ist, und $\{\phi_i(x), i = 0, 1, \dots\}$ ein System von Polynomen (mit ϕ_i vom Grad i), die bezüglich $w(x)$ orthogonal auf $[a : b]$ sind, und wenn das Polynom $p_n(x)$ $f(x)$ an den Nullstellen von $\phi_n(x)$ interpoliert, dann gilt :*

$$\lim_{n \rightarrow \infty} (\|f(x) - p_n(x)\|_2)^2 = \lim_{n \rightarrow \infty} \int_a^b w(x)(f(x) - p_n(x))^2 dx = 0 \quad (3.1.15)$$

Theorem 3 ([5]). *Sei f eine auf $[a; b]$ stetig ableitbare Funktion, und $p_n(x)$ das Interpolationspolynom an den Stützstellen x_0, \dots, x_n , mit $\min_i(x_i) = a$ und $\max_i(x_i) = b$, dann $\forall x \in [a; b]$:*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (3.1.16)$$

wo $\xi \in [a; b]$ und unabhängig von x ist .

Und mit $M_{n+1} = \min_{x \in [a;b]} |f^{(n+1)}(x)|$:

$$|f(x) - p_n(x)| = E(x) \leq \frac{M_{n+1}}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right| \quad (3.1.17)$$

Nach der Gleichung (3.1.17), ist der einzige Weg den Fehler $E(x)$ zu minimieren den Produkt $\prod_{i=0}^n (x - x_i)$ zu minimieren.

Unter allen normierten Polynome sind die Chebyshev Polynome (1821-1894), die am wenigsten von der x -Achse abweichen ([6] und [5]). Diese Polynome werden $T_n(x)$ bezeichnet ($T_n(x)$ hat den Grad n). Die Chebyshev Polynome sind zu einander orthogonal, was bedeutet, dass laut dem Theorem 2 die Interpolation gleichmässig gegen die zu interpolierende Funktion konvergiert. Deshalb werden die Nullstellen des Polynoms $T_{n+1}(x)$ als Stützstellen für die Interpolation genommen. Normalerweise liegen die Nullstelle in $[-1; 1]$:

$$x_k = \cos\left(\frac{\pi \cdot (2 \cdot k + 1)}{2 \cdot (n + 1)}\right) \quad (3.1.18)$$

Die Nullstellen der Chebyshev Polynome werden auch Chebyshev Knoten genannt. Diese Werte können auf einem beliebigen Intervall $[a; b]$ angepasst werden.

$$x_k = \frac{a + b}{2} - \frac{b - a}{2} \cdot \cos\left(\frac{\pi \cdot (2 \cdot k + 1)}{2 \cdot (n + 1)}\right) \quad (3.1.19)$$

Eine andere Verteilung der Knoten ist $x_k = \frac{a + b}{2} - \frac{b - a}{2} \cdot \cos\left(\frac{\pi \cdot k}{n}\right)$ sein. Der Vorteil dieser Verteilung ist, dass a und b in der Knoten auftauchen([5]).

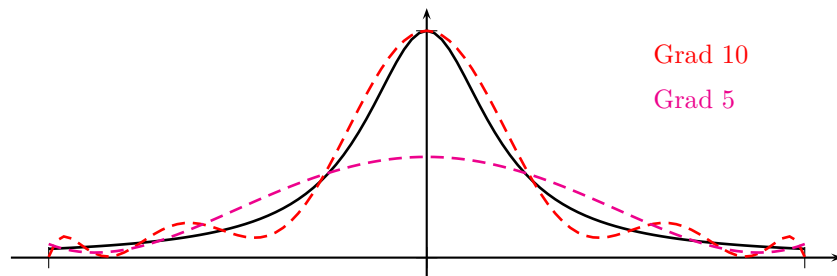


Abbildung 3.5: Die Funktion $f(x) = \frac{1}{1 + 25 \cdot x^2}$ wird mit den Chebyshev Knoten interpoliert.

3.1.3 Sturm-Algorithmus

Die Nullstellen eines Polynoms können geschlossen ermittelt werden, nur wenn der Grad des Polynoms kleiner oder gleich drei ist. Es gibt viele Methode, um die Nullstellen eines Polynoms numerisch durch iterative Verfahren ermitteln zu können. Charles Sturm hat ein Theorem formuliert, das es erlaubt zu berechnen, wie viele Nullstellen ein Polynom in einem gewissen Intervall hat. Dieser Abschnitt stellt dieses Verfahren vor, und leitet daraus dem eine Methode zur Berechnung der ersten Nullstelle in einem Intervall ab.

3.1.3.1 Sturmsche Kette

Definition 3 (Wurzel eines Polynoms). Sei $P(x)$ ein Polynom. Die Lösungen der Gleichung $P(x) = 0$ werden Wurzeln des Polynoms $P(x)$ genannt.

Sei $P(x)$ ein Polynom ohne Mehrfachwurzeln.

Definition 4 (Sturmsche Kette,[7]). Die sturmsche Kette wird wie folgendes gebildet :

$$\begin{aligned}
 P_0(x) &= P(x) \\
 P_1(x) &= P'(x) \\
 P_2(x) &= -\text{rem}(P_0(x), P_1(x)) \\
 &\dots \\
 P_n(x) &= -\text{rem}(P_{n-2}(x), P_{n-1}(x)) = \text{const}
 \end{aligned}
 \tag{3.1.20}$$

wo $\text{rem}(P_{i-2}(x), P_{i-1}(x))$ der Rest der Division ist.

3.1.3.2 Theorem von Sturm

Theorem 4 (Theorem von Sturm, [7]). *Wenn A die Anzahl der Vorzeichenwechsel, d.h. die Anzahl der Übergänge von “+” nach “-” und umgekehrt der Sturmschen Kett für $x = a$ ist, und B die Anzahl der Vorzeichenwechsel in der Folge für $x = b$, dann ist die Differenz $A - B$ gleich der Anzahl der reellen Wurzeln der Gleichung $P(x) = 0$ im Intervall $[a; b]$. Sind in der Zahlenfolge einige Zahlen gleich Null, dann werden diese bei der Abzählung der Vorzeichenwechsel, ausgelassen.*

3.1.3.3 Approximative Bestimmung der Nullstelle eines Polynomes

Da die Nullstellen eines Polynoms nicht exakt bekannt sind, wenn der Grad des Polynoms höher als drei ist, kann das Theorem von Sturm in Kombination mit einer binären Suche verwendet werden, um die Nullstelle mit dem gewünschten Genauigkeit zu approximieren : der Theorem von Sturm gibt nämlich die Anzahl der Nullstelle, die sich in eineme bestimmten Intervall befinden. Bei jeder Iteration der binären Suche wird also die Länge des Suchintervalls halbiert, in dem sich die Nullstelle befindet. Wenn η die gewünschte Genauigkeit des Ergebnisses und $[a; b]$ das betrachtete Intervall bezeichnen, dann kann die Anzahl n der Iterationen bei der binären Suchen vorberechnet werden :

$$\eta > \frac{b-a}{2^n} \implies n = \left\lceil \log_2 \left(\frac{b-a}{\eta} \right) \right\rceil + 1
 \tag{3.1.21}$$

Die Funktion 1 stellt ein einfaches Testverfahren dar, um die Existenz eine Nullstelle festzustellen.

Die Funktion 2 beschreibt, wie die binäre Suche erfolgt, mit einer Rechenoptimierung, damit die Berechnungen, die von einer Iteration zur anderen noch gültig sind, erhalten bleiben und nicht wieder gemacht werden müssen. Wenn keine Nullstelle vorhanden ist, wird das Ergebnis auf unendlich gesetzt.

Function 1: isRoot(\mathcal{S}, a, b)

Input: \mathcal{S} : Sturm Serie of a polynom t_0 : Start of the interval t_n : End of the interval**Output:** Boolean

```
1  $c_1 = \text{signChanges}(\mathcal{S}, t_0)$  ;
2  $c_2 = \text{signChanges}(\mathcal{S}, t_n)$  ;
3 if  $c_1 - c_2 > 0$  then
4 |   return true;
5 else
6 |   return false;
7 end
```

Function 2: findRoot($\mathcal{P}, t_0, t_n, \eta$)

Input: \mathcal{P} : Polynom t_0 : Start of the interval t_n : End of the interval η : Required precision**Output:** root : first root of polynom \mathcal{P}

```
1  $a = t_0$ ;
2  $b = t_n$ ;
3  $n = \lceil \log_2 \left( \frac{b-a}{\eta} \right) \rceil + 1$  ;
4  $\mathcal{S} = \text{sturmSerie}(\mathcal{P})$  ; /* Hier wird die Reihe von Sturm berechnet */
5  $c_1 = \text{signChanges}(\mathcal{S}, a)$  ; /* zählt die Vorzeichenwechsel der Reihe an einem Punkt */
6  $c_2 = \text{signChanges}(\mathcal{S}, b)$  ;
7 if  $c_1 - c_2 > 0$  then
8 |    $c_2 = \text{signChanges}(\mathcal{S}, \frac{a+b}{2})$  ;
9 |   for  $i = 1$  to  $n$  do
10 | |   if  $c_1 - c_2 > 0$  then
11 | | |    $b = \frac{a+b}{2}$  ;
12 | | |    $c_2 = \text{signChanges}(\mathcal{S}, \frac{a+b}{2})$  ;
13 | |   else
14 | | |    $c_1 = c_2$  ;
15 | | |    $a = \frac{a+b}{2}$  ;
16 | | |    $c_2 = \text{signChanges}(\mathcal{S}, \frac{a+b}{2})$  ;
17 | |   end
18 |   end
19 |   return  $\frac{a+b}{2}$ 
20 else
21 |   return  $\infty$ 
22 end
```

3.2 Das kinematische Einspurmodell

Der Einsatz von mathematischen Fahrzeugmodellen ist unabdingbar für die zuverlässige Simulation von fahrdynamisch relevanten Situationen, für die Bestimmung geeigneter Trajektorien sowie für die Berechnung der Bewegungsgleichungen von Fahrzeugen. Im folgenden Abschnitt wird das Fahrzeugmodell vorgestellt, das für die Arbeit benutzt worden ist. Dieses Fahrzeugmodell wird auch bei der

Vorwärtssimulation in einer iterativen numerischen Form benutzt (siehe Abschnitt 2.4 und 3.2.2). Für diese Arbeit sind die Gleichungen des Modells integriert worden (siehe Abschnitt 3.2.3 und 3.2.3.2). Die Integration der Gleichungen erlaubt es, Zustände des Fahrzeugs an beliebigen Zeitpunkten zu berechnen, ohne dass, Zwischenberechnungen erforderlich sind.

3.2.1 Grundlagen des Einspurmodells

Das Einspurmodell (auch bekannt als Ackermann-Modell) ist ein vereinfachtes Modell ([8],[9],[10]), das die Bewegung eines Fahrzeug beschreiben kann. In diesem Modell wird das gesamte Fahrzeug in einem System zusammengefasst, das nur zwei Räder hat. Nur das vordere Rad ist lenkfähig (Siehe Abbildung 3.6). Das Modell kann dank verschiedener Parameter unterschiedliche Fahrzeugtypen simulieren. Diese Parameter und die wichtigen physikalischen Größen des Modells werden in der Tabelle 3.1 aufgelistet.

Name	Bedeutung
Variablen	
x	x-Koordinate der Mitte-Hinterachse im Weltkoordinatensystem R_1 (Abbildung 3.6a)
y	y-Koordinate der Mitte-Hinterachse im Weltkoordinatensystem R_1 (Abbildung 3.6a)
ψ	Gierwinkel des Fahrzeugs im Weltkoordinatensystem R_1 (Abbildung 3.6a)
$\dot{\psi}$	Gierrate des Fahrzeugs im Weltkoordinatensystem R_1 (Abbildung 3.6a)
v_x	Längsgeschwindigkeit des Fahrzeug im Fahrzeugkoordinatensystem R_2 (Abbildung 3.6a)
a_x	Längsbeschleunigung des Fahrzeug im Fahrzeugkoordinatensystem R_2 (Abbildung 3.6a)
a_y	Querbeschleunigung des Fahrzeug im Fahrzeugkoordinatensystem R_2 (Abbildung 3.6a)
Konstanten	
ELG	Eigenlenkgradient
δ	Radlenkwinkel
δ_l	Lenkradwinkel
l_h	Abstand zwischen der Hinterachse und dem Schwerpunkt
l_v	Abstand zwischen der Vorderachse und dem Schwerpunkt
l	Abstand zwischen der Hinterachse und der Vorderachse

Tabelle 3.1: Wichtige Größen des kinematischen Einspurmodells.

Das Einspurmodell kann mit der Zustandvektor $\mathbf{x}(t)$ beschrieben werden:

$$\mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ \psi(t) \\ v_x(t) \end{pmatrix}$$

Das Einspurmodell wird mit zwei Größen gesteuert : die Längsbeschleunigung $a_x(t)$ und der Radlenkwinkel $\delta(t)$ und sein Verhalten wird mit dem Differentialgleichungssystem (3.2.1) beschrieben :

$$\dot{\mathbf{x}}(t) = m(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.2.1}$$

wo

- $\mathbf{u}(t)$ der Eingabevektor $\mathbf{u}(t) = \begin{pmatrix} \delta(t) \\ a_x(t) \end{pmatrix}$ ist

- $\dot{\mathbf{x}}(t)$ die Ableitung über die Zeit des Zustandsvektors $\dot{\mathbf{x}}(t) = \begin{pmatrix} v_x(t) \cdot \cos(\psi(t)) \\ v_x(t) \cdot \sin(\psi(t)) \\ \frac{\delta(t) \cdot v_x(t)}{l + ELG \cdot v_x^2(t)} \\ a_x(t) \end{pmatrix}$ ist

Der Anfangszustand wird mit $\mathbf{x}(t_0) = \begin{pmatrix} x_0 \\ y_0 \\ \psi_0 \\ v_0 \end{pmatrix}$ bezeichnet.

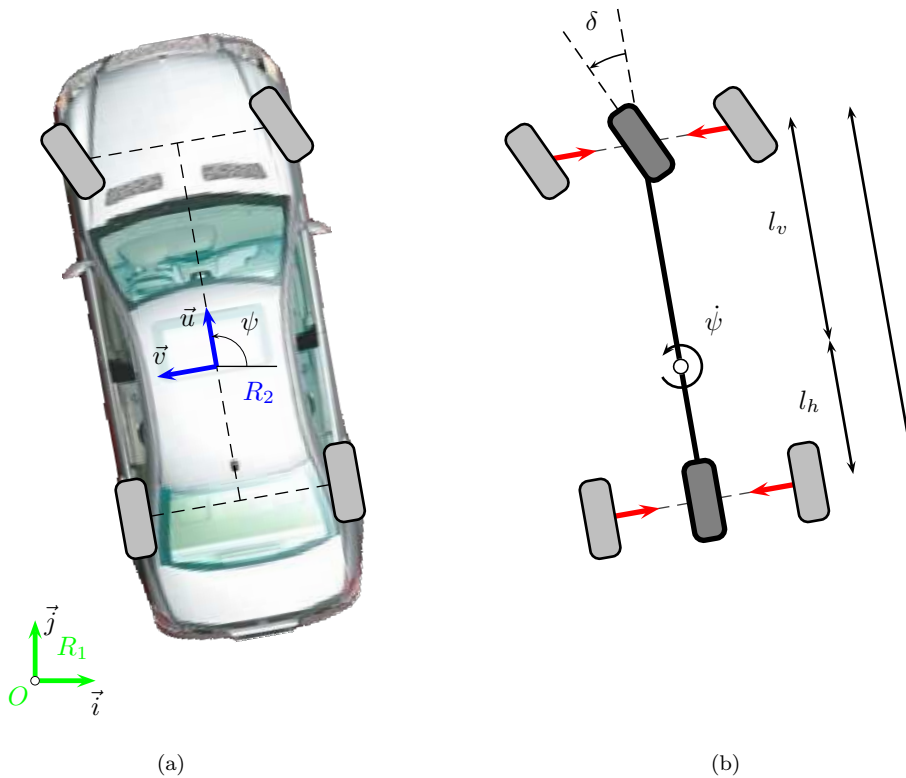


Abbildung 3.6: Das Einspurmodell

Aus $\mathbf{x}(t)$ können andere Größen berechnet werden, wie zum Beispiel die Querbeschleunigung des Fahrzeugs : $a_y(t)$

$$\begin{aligned} a_y(t) &= v_x(t) * \dot{\psi}(t) \\ &= \frac{\delta_{rad}(t) \cdot v_x^2(t)}{l + ELG \cdot v_x^2(t)} \end{aligned} \quad (3.2.2)$$

Das Differentialgleichungssystem (3.2.1) kann numerisch gelöst werden, z.B. mit der Euler-Cauchy Integration mit Zeitschritt Δt .

3.2.2 Euler-Cauchy Integration des kinematischen Einspurmodells

Die einfachste Möglichkeit, um das Differentialgleichungssystem (3.2.1) zu lösen ist eine numerische Integration mit Hilfe des Euler-Cauchy-Verfahrens. Dies erfolgt, indem man die Variablen für konstant

über einen kleinen Zeitschritt hält, und dann linear integriert :

$$\begin{aligned}
 v_{x_{k+1}} &= v_{x_k} + a_{x_k} \cdot \Delta t \\
 \dot{\psi}_{k+1} &= \frac{\delta_{radk} \cdot v_{x_{k+1}}^2}{l + E L G \cdot v_{x_{k+1}}^2} \\
 \psi_{k+1} &= \psi_k + \dot{\psi}_{k+1} \cdot \Delta t \\
 x_{k+1} &= x_k + v_{k+1} \cdot \cos(\psi_{k+1}) \cdot \Delta t \\
 y_{k+1} &= y_k + v_{k+1} \cdot \sin(\psi_{k+1}) \cdot \Delta t
 \end{aligned} \tag{3.2.3}$$

Das Differentialgleichungssystem (3.2.1) könnte auch mit dem Runge-Kutta-Verfahren numerisch gelöst werden. Dieses Verfahren ist stabiler und erlaubt grössere Zeitschritte.

3.2.3 Integration des Ackermann Modells

Unter der Annahme, dass der Eingabevektor $\mathbf{u}(t)$ konstant bleibt : $\mathbf{u}(t) = \mathbf{u}_0 = (\delta, a)^T$ kann das Differentialgleichungssystem (3.2.1) integriert werden.

Die Integration hängt von mehreren Bedingungen ab. Je nachdem ob a , δ , und EG gleich oder ungleich null sind, ergeben sich verschiedene Gleichungen für das Modell.

Nur die Integration im Fall a , δ , und EG ungleich wird in dieser Arbeit vorgestellt. Die anderen Fälle ergeben Gleichungen, die einfacher zu integrieren sind. Alle Fälle und die entsprechende Modellgleichungen werde in der Tabelle 3.2 aufgelistet.

3.2.3.1 Intergration im Fall a , δ und EG ungleich null

Um die Gleichungen zu vereinfachen, wird angenommen, es gilt $t_0 = 0$.

Zuerst wird die Geschwindigkeit berechnet :

$$v(t) - v_0 = \int_0^t a \cdot \tau \quad d\tau \tag{3.2.4}$$

$$\Leftrightarrow v(t) = a \cdot t + v_0$$

Die Gierrate ist mit der Geschwindigkeit zu jedem Zeitpunkt bekannt : $\dot{\psi}(t) = \frac{\delta \cdot v(t)}{l + E L G \cdot v(t)^2}$.

Daher kann der Gierwinkel berechnet werden :

$$\begin{aligned}
 \psi(t) - \psi_0 &= \int_0^t \dot{\psi}(\tau) \, d\tau \\
 \Leftrightarrow \psi(t) - \psi_0 &= \int_0^t \frac{\delta \cdot v(\tau)}{l + EG \cdot v(\tau)^2} \, d\tau \\
 \text{mit } v'(t) &= a : \\
 \Leftrightarrow \psi(t) - \psi_0 &= \frac{\delta}{2 \cdot a \cdot EG} \int_0^t \frac{2 \cdot EG \cdot v'(\tau) \cdot v(\tau)}{l + EG \cdot v(\tau)^2} \, d\tau \tag{3.2.5} \\
 \Leftrightarrow \psi(t) &= \frac{\delta}{2 \cdot a \cdot EG} \cdot \log(l + EG \cdot v(t)^2) - \frac{\delta}{2 \cdot a \cdot EG} \cdot \log(l + EG \cdot v_0^2) + \psi_0 \\
 \text{mit } c_1 &= \frac{\delta}{2 \cdot a \cdot EG} \text{ und } c_2 = -\frac{\delta}{2 \cdot a \cdot EG} \cdot \log(l + EG \cdot v_0^2) + \psi_0 \\
 \Leftrightarrow \psi(t) &= c_1 \cdot \log(l + EG \cdot v(t)^2) + c_2
 \end{aligned}$$

Die Ableitung der Koordinaten des Modells im Weltkoordinatensystem sind bekannt : $\dot{x}(t) = v(t) \cdot \cos(\psi(t))$ und $\dot{y}(t) = v(t) \cdot \sin(\psi(t))$. So werden die x und y -Koordinate berechnet :

$$\begin{aligned}
 x(t) - x_0 &= \int_0^t v(\tau) \cdot \cos(\psi(\tau)) \, d\tau = \int_0^t v(\tau) \cdot \cos(c_1 \cdot \log(l + EG \cdot v(\tau)^2) + c_2) \, d\tau \\
 y(t) - y_0 &= \int_0^t v(\tau) \cdot \sin(\psi(\tau)) \, d\tau = \int_0^t v(\tau) \cdot \sin(c_1 \cdot \log(l + EG \cdot v(\tau)^2) + c_2) \, d\tau
 \end{aligned} \tag{3.2.6}$$

Die Berechnungen beider Koordinate sind sich sehr ähnlich, deshalb wird nur die Berechnung der x -Koordinate erläutert. Um diese Koordinate zu berechnen, wird eine Stammfunktion der Funktion $f(t) = v(t) \cdot \cos(c_1 \cdot \log(l + EG \cdot v(t)^2) + c_2)$ gebraucht :

$$x(t) - x_0 = \int_0^t f(\tau) \, d\tau = \int_0^t v(\tau) \cdot \cos(c_1 \cdot \log(l + EG \cdot v(\tau)^2) + c_2) \, d\tau \tag{3.2.7}$$

Dafür wird zuerst wird die Substitution $u = v(t) = a \cdot t + v_0$ durchgeführt :

$$\begin{aligned}
 u = v(t) &= a \cdot t + v_0 \Rightarrow du = a \cdot dt \\
 x(t) - x_0 &= \int_0^t v(\tau) \cdot \cos(c_1 \cdot \log(l + EG \cdot v(\tau)^2) + c_2) \, d\tau \\
 \Leftrightarrow x(t) - x_0 &= \frac{1}{a} \int_{v_0}^{v(t)} u \cdot \cos(c_1 \cdot \log(l + EG \cdot u^2) + c_2) \, du = \frac{1}{a} \int_{v_0}^{v(t)} g(u) \, du
 \end{aligned} \tag{3.2.8}$$

wo $g(u) = u \cdot \cos(c_1 \cdot \log(l + EG \cdot u^2) + c_2)$

Mit Hilfe der Computeralgebrasystem MAXIMA [11] kann eine Stammfunktion von g berechnet

werden :

$$\int u \cdot \cos(c_1 \cdot \log(l + EG \cdot u^2) + c_2) du = \frac{(u^2 \cdot EG + l) \cdot (c_1 \cdot \sin(c_1 \cdot \log(l + EG \cdot u^2) + c_2) + \cos(c_1 \cdot \log(l + EG \cdot u^2) + c_2))}{2 \cdot (c_1^2 + 1) \cdot EG} \quad (3.2.9)$$

Und daher :

$$x(t) - x_0 = \frac{(v(t)^2 \cdot EG + l) \cdot (c_1 \cdot \sin(c_1 \cdot \log(l + EG \cdot v(t)^2) + c_2) + \cos(c_1 \cdot \log(l + EG \cdot v(t)^2) + c_2))}{2 \cdot (c_1^2 + 1) \cdot EG} - \frac{(v_0^2 \cdot EG + l) \cdot (c_1 \cdot \sin(c_1 \cdot \log(l + EG \cdot v_0^2) + c_2) + \cos(c_1 \cdot \log(l + EG \cdot v_0^2) + c_2))}{2 \cdot (c_1^2 + 1) \cdot EG} \quad (3.2.10)$$

Mit (3.2.5), $c_1 \cdot \log(l + EG \cdot v(t)^2) + c_2 = \psi(t)$ und $c_1 \cdot \log(l + EG \cdot v_0^2) + c_2 = \psi_0$:

$$x(t) = \frac{1}{a} \cdot \frac{(v(t)^2 \cdot EG + l) \cdot (c_1 \cdot \sin(\psi(t)) + \cos(\psi(t)))}{2 \cdot (c_1^2 + 1) \cdot EG} - \frac{1}{a} \cdot \frac{(v_0^2 \cdot EG + l) \cdot (c_1 \cdot \sin(\psi_0) + \cos(\psi_0))}{2 \cdot (c_1^2 + 1) \cdot EG} + x_0 \quad (3.2.11)$$

Gleichermaßen wird die y -Koordinate berechnet :

$$y(t) = \frac{1}{a} \cdot \frac{(v(t)^2 \cdot EG + l) \cdot (-c_1 \cdot \cos(\psi(t)) + \sin(\psi(t)))}{2 \cdot (c_1^2 + 1) \cdot EG} - \frac{1}{a} \cdot \frac{(v_0^2 \cdot EG + l) \cdot (-c_1 \cdot \cos(\psi_0) + \sin(\psi_0))}{2 \cdot (c_1^2 + 1) \cdot EG} + y_0 \quad (3.2.12)$$

3.2.3.2 Intergration für alle Fälle

Fall	Gleichungen
$a = 0$ $\delta = 0$	$\psi(t) = \psi_0$ $x(t) = v_0 \cdot \cos(\psi_0) \cdot t + x_0$ $y(t) = v_0 \cdot \sin(\psi_0) \cdot t + y_0$
$a = 0$ $\delta \neq 0$	$\psi(t) = \frac{\delta \cdot v_0}{l + EG \cdot v_0^2} \cdot t + \psi_0$ $x(t) = \frac{l + EG \cdot v_0^2}{\delta} \cdot (\sin(\psi(t)) - \sin(\psi_0)) + x_0$ $y(t) = -\frac{l + EG \cdot v_0^2}{\delta} \cdot (\cos(\psi(t)) - \cos(\psi_0)) + y_0$
$a = cst \neq 0$ $\delta = 0$	$\psi(t) = \psi_0$ $x(t) = \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t\right) \cdot \cos(\psi_0) + x_0$ $y(t) = \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t\right) \cdot \sin(\psi_0) + y_0$
$a = cst \neq 0$ $EG = 0$ $\delta = cst \neq 0$	$\psi(t) = \frac{\delta}{l} \cdot \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t\right) + \psi_0$ $x(t) = \frac{l}{\delta} \cdot (\sin(\psi(t)) - \sin(\psi_0)) + x_0$ $y(t) = -\frac{l}{\delta} \cdot (\cos(\psi(t)) - \cos(\psi_0)) + y_0$
$a = cst \neq 0$ $EG \neq 0$ $\delta = cst \neq 0$	$c_1 = \frac{\delta}{2 \cdot a \cdot EG}, \quad c_2 = -\frac{\delta}{2 \cdot a \cdot EG} \cdot \log(l + EG \cdot v_0^2) + \psi_0$ $\psi(t) = c_1 \cdot \log(l + EG \cdot v(t)^2) + c_2$ $x(t) = \frac{1}{a} \cdot \frac{(v(t)^2 \cdot EG + l) \cdot (c_1 \cdot \sin(\psi(t)) + \cos(\psi(t)))}{2 \cdot (c_1^2 + 1) \cdot EG}$ $- \frac{1}{a} \cdot \frac{(v_0^2 \cdot EG + l) \cdot (c_1 \cdot \sin(\psi_0) + \cos(\psi_0))}{2 \cdot (c_1^2 + 1) \cdot EG} + x_0$ $y(t) = \frac{1}{a} \cdot \frac{(v(t)^2 \cdot EG + l) \cdot (-c_1 \cdot \cos(\psi(t)) + \sin(\psi(t)))}{2 \cdot (c_1^2 + 1) \cdot EG}$ $- \frac{1}{a} \cdot \frac{(v_0^2 \cdot EG + l) \cdot (-c_1 \cdot \cos(\psi_0) + \sin(\psi_0))}{2 \cdot (c_1^2 + 1) \cdot EG} + y_0$

Tabelle 3.2: Geschlossene Lösung des kinematischen Einspurmodells für unterschiedliche Eingangsgrößenkonstellationen

3.3 Zeitreservenbestimmung mit Interpolation der Distanzfunktion

Wie im Abschnitt 2.4 erklärt stellt die Vorwärtssimulation ein sehr flexibles iteratives numerisches Verfahren zur Zeitreservenbestimmung dar. Sie verfügt über verschiedene Vorteile, aber ihre Nachteile (Rechenaufwand und Speicherplatz) machen diese Methode schwer einsetzbar auf den Steuergeräten, mit den die Serienfahrzeuge ausgerüstet werden sollen.

Einer der Ziele dieser Arbeit ist es, eine weitere Methode zur Zeitreservenberechnung zu entwickeln, die weniger rechenaufwendig ist.

Die Grundideen, die diese neue Methode bilden, sind :

- die Ersetzung des numerischen iterativen Bewegungsmodells mit einem Modell dessen Bewegungsgleichungen integriert worden sind
- die Anwendung einer Distanzfunktion, deren Abbildungsraum kontinuierlich ist
- die Interpolation der Distanzfunktion mit einem Polynom

Die Integration der Bewegungsgleichungen ermöglicht es, Zustände an beliebigen Zeitpunkten direkt und ohne Verlust an Genauigkeit zu berechnen. So können Abstände an beliebigen Zeitpunkten mit dem Abstandmaß berechnet werden, ohne zusätz oder ungebrauchte Berechnung : der Rechenaufwand wird minimiert.

Die mögliche Berechnung des Abstandmaßes an beliebigen Zeitpunkten stellt auch einen Vorteil darin dar, dass die Zeitpunkten, die als Interpolationsstützstellen dienen, so gewählt werden können, dass eine Interpolation einem minimierten Interpolationsfehler möglich ist (siehe Abschnitt 3.1.2.3).

Einst das interpolierende Polynom der Distanzfunktion bekannt ist, können die Zeitreserven, wie TTC ermittelt werden.

3.3.1 Grundalgorithmus

Der Kern dieser Zeitreservenberechnungsmethode, für eine Situation mit N Teilnehmer, kann in verschiedenen Schritten unterteilt werden :

1. die Zustände der verschiedenen Fahrzeuge werden an gewählten Zeitpunkten berechnet :

$$\mathbf{x}_i(t_j) = \mathcal{M}_i(\mathbf{x}_i(t_0), \mathbf{u}_i, t_j), \forall i \in [1; N], \forall j \in [0; n] \quad (3.3.1)$$

Die Funktion \mathcal{M}_i entspricht dem Einspurmodell, der im Abschnitt 3.2 ausführlich beschrieben wird.

Gleichzeitig können die entsprechenden Silhouette bestimmt werden :

$$\mathbf{s}_i(t_j) = \mathbf{s}(x_i(t_j)) \quad (3.3.2)$$

In diesem Ansatz wird die Silhouette des Egofahrzeuges als ein Rechteck dargestellt. Die Eigenschaften des Rechtecks sind in eine Vektor $(x, y, \psi, l, b)^T$ gespeichert. Die Objekten werden als

geschlossene Polygone dargestellt. Diese Darstellung ist für die Anwendung der Distanzfunktion \mathcal{D} sehr geeignet (siehe Abschnitt 3.1.1)

2. die Abstände $\mathcal{D}_e(t_j)$ zwischen dem Egofahrzeug und der Objekte werden mit Hilfe der Distanzfunktion \mathcal{D} berechnet (siehe Abschnitt 3.1.1):

$$\mathcal{D}_e(t_j) = \min_{\forall k \neq e} \{\mathcal{D}(\mathbf{s}_e(t_j), \mathbf{s}_k(t_j))\} \quad (3.3.3)$$

Da die Entwicklungen der Fahrzeugzustände unabhängig von einander berechnet werden, und die Abstände im Nachhinein berechnet werden, entspricht der Abstandsverlauf $\mathcal{D}_e(t_j)$ keine physikalische Realität (Fahrzeuge können nicht kollidieren und weiterfahren). Obwohl der Abstandsverlauf in seinem Ganzen die Realität nicht entsprechen mag, ist er bis zur Kollision der Realität treu. Die Annahme, dass Fahrzeuge kollidieren und gleichzeitig weiterfahren können, erlaubt es den Kollisionszeitpunkt zu isolieren.

3. der Abstandsverlauf wird mit einem Polynom $P_{\mathcal{D}_e}$ interpoliert. $P_{\mathcal{D}_e}$ wird als Vektor seiner Koeffizienten dargestellt :

$$P_{\mathcal{D}_e} = (a_0, a_1, \dots, a_n)^T \iff P_{\mathcal{D}_e}(x) = a_0 + a_1 \cdot x + \dots + a_n \cdot x^n \quad (3.3.4)$$

Die Berechnung von $P_{\mathcal{D}_e}$ erfolgt durch den Produkt der inversen der Vandermonde Matrix V_n^{-1} mit dem Vektor $(\mathcal{D}_e(t_j)_{j \in [0;n]})^T$:

$$P_{\mathcal{D}_e} = V_n^{-1} \cdot \mathcal{D}_e(t_j)_{j \in [0;n]}^T \quad (3.3.5)$$

Da der Grad n des interpolierenden Polynoms $P_{\mathcal{D}_e}$ und das Zeitintervall $[t_0; t_H]$ von Anfang an bekannt sind, kann die Matrix V_n^{-1} vorberechnet werden.

4. je nach Aufgabe werden die Nullstellen des Polynoms $P_{\mathcal{D}_e}$ untersucht, mit Hilfe der im Abschnitt 3.1.3 vorgestellten Methoden

3.3.2 Verwendung mehrere Polynome

Die Anzahl der Interpolationsknoten spielt eine wichtige Rolle : sie gibt die Menge an ursprünglicher Information (die berechneten Zustände) wieder. Aber je mehr der Grad des Interpolationspolynom steigt, desto mehr Operationen gebraucht werden : die Komplexität ist $O(n^2)$. Eine andere Folge des hohen Grades eines Polynom, ist die numerische Ungenauigkeit der Berechnungen durch die sukzessiven Operationen des Rechners. Die Genauigkeit des `double` wird überschritten, einerseits bei der Berechnung der Inverse der Vandermonde Matrix und die Berechnung der Koeffizienten des Polynoms, (siehe Abschnitt 3.1.2) und andererseits bei der Evaluationen der Polynome der sturmschen Reihe (siehe Absatz 3.1.3).

Aus diesen Gründen werden statt einen Polynom hohen Grades, mehrere Polynome niedrigeren Grades benutzt, indem der Interval $[0; T]$ in mehrere Stücke zerlegt wird. Mit zum Beispiel 25 Knoten, können :

- 1 Polynom des Grades 24
- 2 Polynome des Grades 12

- 3 Polynome des Grades 8
- 4 Polynome des Grades 6
- 6 Polynome des Grades 4
- 8 Polynome des Grades 3 (entspricht die Interpolation mit Cubic Splines)
- 12 Polynome des Grades 2 (entspricht die lineare Interpolation)

Je nach Fall, müssen die Interpolationsknoten anders belegt werden. Zu Beispiel im Fall von vier Polynom müssen die Chebychevknoten auf $[0; \frac{T}{4}]$, $[\frac{T}{4}; \frac{T}{2}]$, $[\frac{T}{2}; \frac{3T}{4}]$ und $[\frac{3T}{4}; T]$ verteilt werden.

Jedoch wird in den weiteren Abschnitten angenommen, dass ein einziges interpolierendes Polynom benutzt wird, damit die verschiedenen Algorithmen vereinfacht dargestellt werden können.

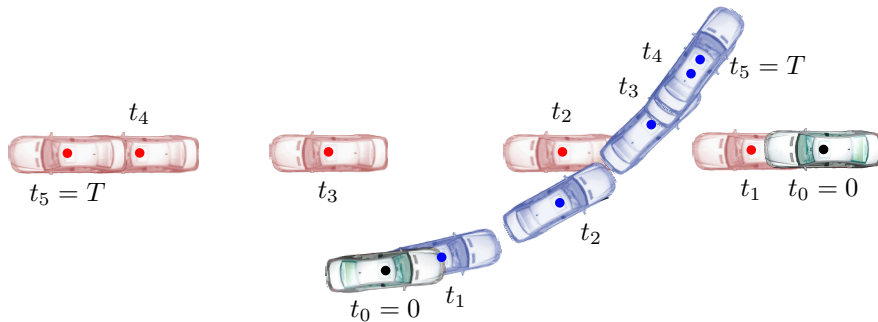
3.3.3 Berechnung der Kollisionszeitreserve

Die Abbildung 3.7 stellt die TTC Berechnung in einem einfachen Szenario mit einem einzigen Objekt Fahrzeug dar :

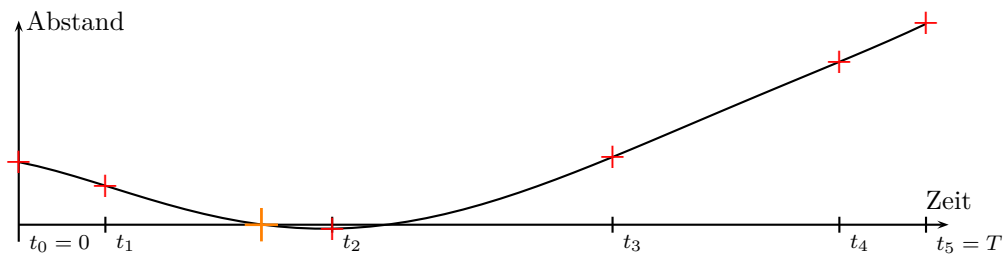
1. die Zustände der Verkehrsteilnehmer an den gewünschten Zeitpunkten werden berechnet
2. die entsprechenden Abstände werden berechnet
3. der Abstandsverlauf wird interpoliert
4. die erste positive Nullstelle wird ermittelt

In verkehrssituationen wo es mehreren dynamische oder statische Objekte gibt müssen alle Objekte in der Berechnung berücksichtigt werden. Zwei unterschiedliche Ansätze zur Berechnung der Zeitreserve TTC sind möglich :

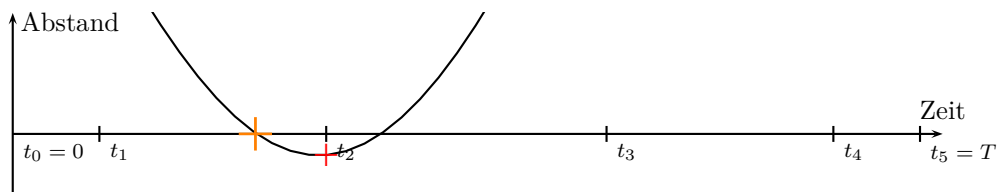
1. Im ersten Ansatz werden die Objekte einzeln betrachtet und das kleinste TTC wird genommen (Algorithmus 3). In diesem Ansatz kann auch sehr einfach herausgefunden werden, welches Objekt in der Kollision mit dem Ego-Fahrzeug verwickelt ist. Aber es muss für jedes Objekt ein Polynom und seine erste positive Nullstelle berechnet werden.
2. Im zweiten Ansatz, werden die Abstände zu jedem Zeitpunkt verglichen, und das Minimum pro Zeitpunkt wird für die Interpolation benutzt. Anschliessend wird TTC gegebenenfalls berechnet (Algorithmus 4). In diesem Ansatz wird ein einziger Polynom berechnet, aber es ist schwierigen herauszufinden, welches Objekt die Kollision verursacht.



(a) Die Zustände beider Fahrzeuge werden an verschiedenen Zeitpunkten berechnet. Diese Zeitpunkte entsprechen die Verteilung der Chebyshev Knoten über $[0; T]$.



(b) Die Abstände zwischen den Fahrzeugen werden berechnet und interpoliert (der Grad des interpolierenden Polynoms ist dabei 5). Anschließend wird die kleinste positive Nullstelle mit Hilfe einer binären Suche und des Theorems von Sturm berechnet.



(c) Vergrößerung der Abbildung (b)

Abbildung 3.7: TTC Berechnung als Bestimmung der kleinsten positiven Nullstelle

Algorithm 3: TTC Algorithmus, erster Ansatz

Input:
 $\mathbf{x}_{0,i}, \mathbf{s}_{0,i}, \mathbf{x}_i^{stat}, \mathbf{s}_{0,i}^{stat}$: startstates and startshapes for dynamic and static objects
 $\mathbf{u}_{P,i}$: input for dynamic objects
 e : ego-vehicle index
 $\Omega = \{t_0, \dots, t_n\}, V_n^{-1}$: interpolation timestamps and inverse Vandermonde Matrix
 η : required precision
Output: $\mathbf{TTC}, \mathbf{O}_{crit}$

```

1 TTC  $\leftarrow \infty$ 
2 foreach  $t_k \in \Omega$  do
3    $\mathbf{x}_e(t_k) \leftarrow \mathcal{M}_e(\mathbf{x}_{0,e}, \mathbf{u}_{P,e}, t_k)$ 
4   foreach dynamic object  $i \neq e$  do
5      $\mathbf{x}_i(t_k) \leftarrow \mathcal{M}_i(\mathbf{x}_{0,i}, \mathbf{u}_{P,i}, t_k)$ 
6      $\mathcal{D}_{e,i}(t_k) \leftarrow \mathcal{D}(\mathbf{s}(\mathbf{x}_e(t_k)), \mathbf{s}(\mathbf{x}_i(t_k)))$ 
7   end
8   foreach static object  $i$  do
9      $\mathcal{D}_{e,i}(t_k) \leftarrow \mathcal{D}(\mathbf{s}(\mathbf{x}_e(t_k)), \mathbf{s}_{0,i}^{stat})$ 
10  end
11 end
12 foreach object  $i \neq e$  do
13    $\mathcal{P}_{\mathcal{D},i} \leftarrow \text{calcPolynom}(d_{e,i}, \Omega, V_n^{-1})$ 
14    $\mathbf{TTC}_i \leftarrow \text{findRoot}(\mathcal{P}_{\mathcal{D},i}, t_0, t_n, \eta)$ 
15   if  $\mathbf{TTC}_i < \mathbf{TTC}$  then
16      $\mathbf{TTC} \leftarrow \mathbf{TTC}_i$ 
17      $\mathbf{O}_{crit} \leftarrow O$ 
18   end
19 end
20 return

```

Algorithm 4: TTC Algorithmus, erster Ansatz

Input:
 $\mathbf{x}_{0,i}, \mathbf{s}_{0,i}, \mathbf{x}_i^{stat}, \mathbf{s}_{0,i}^{stat}$: startstates and startshapes for dynamic and static objects
 $\mathbf{u}_{P,i}$: input for dynamic objects
 e : ego-vehicle index
 $\Omega = \{t_0, \dots, t_n\}, V_n^{-1}$: interpolation timestamps and inverse Vandermonde Matrix
 η : required precision
Output: $\mathbf{TTC}, \mathbf{O}_{crit}$

```

1 TTC  $\leftarrow \infty$ 
2 foreach  $t_k \in \Omega$  do
3    $\mathbf{x}_e(t_k) \leftarrow \mathcal{M}_e(\mathbf{x}_{0,e}, \mathbf{u}_{P,e}, t_k)$ 
4   foreach dynamic object  $i \neq e$  do
5      $\mathbf{x}_i(t_k) \leftarrow \mathcal{M}_i(\mathbf{x}_{0,i}, \mathbf{u}_{P,i}, t_k)$ 
6      $\mathcal{D}_{e,i}(t_k) \leftarrow \mathcal{D}(\mathbf{s}(\mathbf{x}_e(t_k)), \mathbf{s}(\mathbf{x}_i(t_k)))$ 
7   end
8   foreach static object  $i$  do
9      $\mathcal{D}_{e,i}(t_k) \leftarrow \mathcal{D}(\mathbf{s}(\mathbf{x}_e(t_k)), \mathbf{s}_{0,i}^{stat})$ 
10  end
11   $\mathcal{D}_e(t_k) \leftarrow \min_i \mathcal{D}_{e,i}(t_k)$ 
12 end
13  $\mathcal{P}_{\mathcal{D}} \leftarrow \text{calcPolynom}(d_{e,i}, \Omega, V_n^{-1})$ 
14  $\mathbf{TTC} \leftarrow \text{findRoot}(\mathcal{P}_{\mathcal{D}}, t_0, t_n, \eta)$ 
15  $\mathbf{O}_{crit} \leftarrow \text{retrieveObject}(\mathbf{TTC})$ 
16 return

```

3.3.4 Bestimmung von Reaktionszeitreserven

Wie für die Vorwärtsimulation erfolgt die Bestimmung von Reaktionszeitreserven durch wiederholtes testen von Trajektrien auf Kollisionsfreiheit. Bei dieser Suche wird den spätesten Auslösungszeitpunkt des Manövers ermittelt. Dabei werden die potentiellen Folgekollision auch mitberücksichtigt.

Da Suchalgorithmen zeitaufwendig sind, wird einer der Suchalgorithmen gewählt, der die kleinste Komplexität hat : die binäre Suche, deren Komplexität logarithmisch ist $\mathcal{O}(\log n)$.

3.3.4.1 Algorithmus

Die binäre Suche nach einer Reaktionszeitreserve erfolgt in mehrere Schritten. Wie bei der Bestimmung von TTC wird ein Genauigkeitskoeffizient η verwendet :

1. der späteste Manöverauslösungszeitpunkt wird mit einem Fehler kleiner als η gesucht:
 - (a) ein Manöverauslösungszeitpunkt wird gewählt
 - (b) die neuen Zustände der Ego-Fahrzeug werden berechnet
 - (c) es wird überprüft, ob die ursprüngliche Kollision noch vorliegt
 - (d) ein neuer Manöverauslösungszeitpunkt gewählt, früher oder später in der Zukunft, um sich dem optimalen Zeitpunkt anzunähern, dann geht man zurück zum Schritt 1b bis die Genauigkeit η erreicht ist
2. wenn ein Manöverauslösungszeitpunkt für eine kollisionsfreie Manöver gefunden worden ist, wird überprüft ob diese Manöver keine Kollision mit anderen Objekten (Folgekollision) verursacht
 - (a) wenn es keine Folgekollision gibt, dann ist TTM bekannt mit einer Genauigkeit von η
 - (b) wenn es eine Folgekollision gibt, dann geht man zurück zum Schritt 1

Diese Methode wird im Algorithmus 5 ausführlicher erklärt. Man kann sehen, dass bei der Suche nach Folgekollision die anderen Objekten einzeln betrachtet werden, wie bei der TTC Suche im Algorithmus 3. Dieser Ansatz hat den Vorteil, dass der Zeitpunkt der Kollision nicht ermittelt werden soll. Man könnte auch wählen, die anderen Objekten gemeinsam zu betrachten (wie im Algorithmus 4), aber dafür muß anschliessend der Zeitpunkt der Kollision ermittelt werden, um herausfinden zu können, welcher Objekt für die Folgekollision verantwortlich ist.

Im Sinne von Laufzeiten ist der erste Ansatz schneller, wenn es nicht zu viele Objekte gibt.

3.3.4.2 Wichtige Annahme zur Reaktionszeitreservenberechnung

Es wird ein Szenario betrachtet, in dem es nur das Ego-Fahrzeug und einen Objekt gibt. Die Berechnung von TTC sagt eine Kollision vorher. Es wird für einfache Manöver (wie zum Beispiel lenken oder bremsen) angenommen, dass alle Zeitpunkt an dem, das Manöver ausgelöst wird und dadurch die Kollision vermieden wird, ein einziger Intervall bilden. Nach der Gleichung (2.2.8) ist TTM das Maximum dieses Intervalls. Die Abbildung 3.8a macht diese Annahme anschaulich. Mit dieser Annahme kann TTM mit Hilfe einer binären Suche, selbst wenn es mehreren Objekt gibt, ermittelt werden (siehe Abschnitt 3.3.4.1).

Für komplexere Manöver, wie einen Spurwechsel oder ein Überholungsmanöver, ist es sehr wahrscheinlich, dass diese Annahme nicht zutreffend ist. Für solche Fälle können die im Abschnitt 3.3.4.1

Algorithm 5: TTM Algorithmus

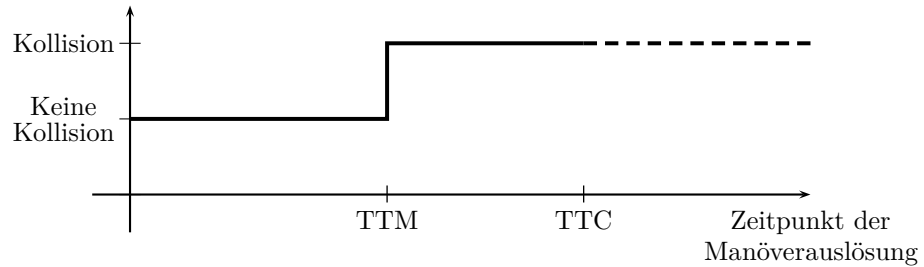
Input: **TTC** : time to collision
 $\mathbf{x}_{0,i}, \mathbf{s}_{0,i}$: startstate and startshapes for dynamic objects
 $\mathbf{u}_{P,e}$: input for ego vehicle
 $\mathbf{x}_i^{stat}, \mathbf{s}_{0,i}^{stat}$: Startstate and startshapes for dynamic objects
 e : ego-vehicle index
 $\Omega = \{t_0, \dots, t_n\}$: interpolation timestamps
 V_n^{-1} : inverse Vandermonde Matrix for the interpolation
 m : Maneuver type
 i_{crit} : Object that causes the collision
Output: **TTM**

```

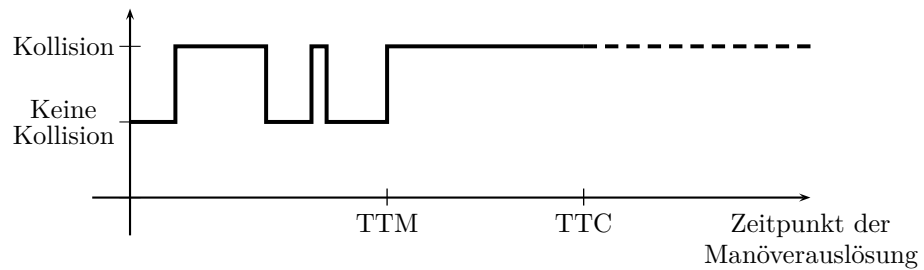
1 TTM  $\leftarrow \infty$ 
2  $t_m \leftarrow \mathbf{TTC}$ 
3  $nocollision \leftarrow 0$ 
4 while ( $nocollision = 0 \parallel t_m > \eta$ ) do
5      $n \leftarrow \lfloor \log_2 \left( \frac{t_m}{\eta} \right) \rfloor + 1$ 
6     for  $k = 1..n$  do
7          $a \leftarrow t_0, \quad b \leftarrow t_m$ 
8         foreach  $t_k \in \Omega$  do
9              $\tau \leftarrow \frac{a+b}{2}$ 
10             $\mathbf{u}_{m,\tau,e} = \mathcal{U}(\mathbf{u}_{P,e}, m, \tau)$ 
11            if  $t_k < \tau$  then
12                 $\mathbf{s}_{e,m}(t_k) \leftarrow \mathbf{s}_e(t_k)$ 
13            else
14                 $\mathbf{s}_{e,m}(t_k) \leftarrow \mathbf{s}(\mathcal{M}_e(\mathbf{x}_{0,e}, \mathbf{u}_{m,\tau,e}, t_k))$ 
15            end
16             $\mathcal{D}_{e,i_{crit}}(t_k) = \mathcal{D}(\mathbf{s}_{e,m}(t_k), \mathbf{s}_{i_{crit}}(t_k))$ 
17        end
18         $\mathcal{P}_{\mathcal{D},i_{crit}} \leftarrow \text{calcPolynom}(d_{E,O_{crit}}, \Omega, V_n^{-1})$ 
19        if  $\text{isRoot}(\mathcal{P}_{\mathcal{D},O}, t_0, t_n)$  then
20             $b \leftarrow \frac{a+b}{2}$ 
21        else
22             $a \leftarrow \frac{a+b}{2}$ 
23             $t_m \leftarrow a$ 
24             $nocollision \leftarrow 1$ 
25        end
26    end
27    if  $nocollision = 1$  then
28        foreach object  $i$  do
29            foreach  $t_k \in \Omega$  do
30                 $\mathcal{D}_{e,i}(t_k) \leftarrow \mathcal{D}(\mathbf{s}_{e,m}(t_k), \mathbf{s}_i(t_k))$ 
31            end
32             $\mathcal{P}_{\mathcal{D},i} \leftarrow \text{calcPolynom}(\mathcal{D}_{e,i}, \Omega)$ 
33            if  $\text{isRoot}(\mathcal{P}_{\mathcal{D},O}, t_0, t_n)$  then
34                 $O_{crit} \leftarrow O$ 
35                 $nocollision \leftarrow 0$ 
36                break
37            end
38        end
39    end
40 end
41 if  $nocollision = 1$  then
42      $\mathbf{TTM} \leftarrow t_m$ 
43 end
44 return TTM

```

beschriebenen Methode nicht verwendet werden, denn die binäre Suche kann ohne diese Annahme nicht eingesetzt werden. (siehe Abbildung 3.8b).



- (a) Es wird angenommen, dass alle Auslösungzeitpunkte für erfolgreiche Kollisionsvermeidungsmanöver zwischen das Ego-Fahrzeug und ein einziges Objekt einen einzigen Intervall bilden



- (b) Im Fall von komplexen Vermeidungsmanöver, kann keine binäre Suche verwendet werden

Abbildung 3.8: Die Manöver müssen einfach sein, damit die Binärsuche anwendbar ist

3.3.5 Direkte Berechnung von TTB

Die geometrische Lage der Fahrzeuge am Zeitpunkt TTC und ihre Eigenschaften kann gleichzeitig mit der Berechnung von TTC bestimmt werden. Die geometrische Lage kann verwendet werden, um TTB direkt zu berechnen.

In diesem Abschnitt wird die Berechnung nur für den Fall “Links-Abbiegen” detailliert, da es als Beispiel für die Diplomarbeit benutzt wird.

In diesem Beispiel können alle Kollisionsfällen unter vier verschiedenen Typen geordnet werden (siehe Abbildung 3.9a bis 3.9d).

Die methode, um TTB direkt bestimmen zu können, besteht darin, die Geometrie der Kollision zum Zeitpunkt TTC zu linearisieren. Die Bahnen werden gerade, und die Kollision der Fahrzeuge hat exakt die selben Eigenschaften wie die echte vorberechnete Kollision.

Mit dieser Methode werden die Trajektorien der Fahrzeuge von den zurückgelegten Strecken $d(t)$ entkoppelt.

Je nach Kollisionsfall muss das Objektfahrzeug eine bestimmte Strecke zurückgelegt haben, bevor das Egofahrzeug die Spur des Objektfahrzeuges befahren darf. Diese Strecke wird als d_{obj} (grün in der Abbildung 3.9) bezeichnet und t_{obj} entspricht die nötige Zeit, um diese Strecke zu fahren. Um die Kollision zu vermeiden, darf der Egofahrzeug entweder vor der Spur stehenbleiben oder die Spur vom Objektfahrzeug erst betreten, nachdem der Objektfahrzeug die Strecke d_{obj} zurückgelegt hat.

$d_B(t)$ bezeichnet die vom Egofahrzeug zurückgelegte Strecke, mit Bremsmanöver, während $d(t)$ die ohne Manöver zurückgelegte Strecke bezeichnet. d_s bezeichnet die vom Egofahrzeug ursprünglich zurückgelegte Strecke, die bei dem Bremsmanöver nicht vor t_{obj} zurückgelegt werden soll.

Die letzt-mögliche Bremsmanöver ist das Manöver, wo der Egofahrzeug nicht vor der Spur stehenbleibt, sondern nur genug bremst, um nach dem Objektfahrzeug zu fahren. Deshalb wird es zuerst nach diesem Manöver gesucht.

$$\begin{aligned}
 d_B(TTC + t_{obj}) &= d(TTC) - d_s \\
 \Leftrightarrow \int_0^{TTB} a_0 \cdot t + v_0 dt + \int_{TTB}^{TTC+t_{obj}} a_B \cdot t + (v_0 + a_0 \cdot TTB) dt &= \int_0^{TTC} a_0 \cdot t + v_0 dt - d_s \\
 \Leftrightarrow \frac{1}{2} \cdot a_0 \cdot TTB^2 + v_0 \cdot TTB + \frac{1}{2} \cdot a_B \cdot (T - TTB)^2 + (v_0 + a_0 \cdot TTB)(T - TTB) \\
 &= \frac{1}{2} \cdot a_0 \cdot TTC^2 + v_0 \cdot TTC - d_s \text{ mit } T = TTC + t_{obj}. \\
 \Leftrightarrow \frac{1}{2} \cdot (a_B - a_0) \cdot TTB^2 - (a_B - a_0) \cdot T \cdot TTB - \frac{1}{2} \cdot a_0 \cdot TTC^2 + v_0 \cdot (T - TTC) + d_s &= 0
 \end{aligned} \tag{3.3.6}$$

Dieses Manöver ist nur möglich, wenn der berechnete TTB kleiner als TTC ist und die Geschwindigkeit zum Zeitpunkt $TTC + t_{obj}$ größer null ist. Die Gleichung (3.3.6) erlaubt nämlich negative

Geschwindigkeiten, was aber die Kollision nicht verhindert.

$$\begin{aligned}
 & d_B \left(TTB - \frac{a_0 \cdot TTB + v_0}{a_B} \right) = d(TTC) - d_s \\
 \Leftrightarrow & \int_0^{TTB} a_0 \cdot t + v_0 \, dt + \int_{TTB}^{TTB - \frac{a_0 \cdot TTB + v_0}{a_B}} a_B \cdot t + (v_0 + a_0 \cdot TTB) \, dt \\
 & = \int_0^{TTC} a_0 \cdot t + v_0 \, dt - d_s \\
 \Leftrightarrow & \frac{1}{2} \cdot a_0 \cdot TTB^2 + v_0 \cdot TTB + \frac{1}{2} \cdot a_B \cdot \left(\frac{a_0 \cdot TTC + v_0}{a_B} \right)^2 - \frac{(a_0 \cdot TTC + v_0)^2}{a_B} \\
 & = \int_0^{TTC} a_0 \cdot t + v_0 \, dt - d_s \\
 \Leftrightarrow & \frac{1}{2} \cdot a_0 \cdot \left(1 - \frac{a_0}{a_B} \right) \cdot TTB^2 + v_0 \cdot \left(1 - \frac{a_0}{a_B} \right) \cdot TTB - \frac{v_0^2}{2 \cdot a_B} - \frac{1}{2} \cdot a_0 \cdot TTC^2 - v_0 \cdot TTC + d_s \\
 & = 0
 \end{aligned}
 \tag{3.3.7}$$

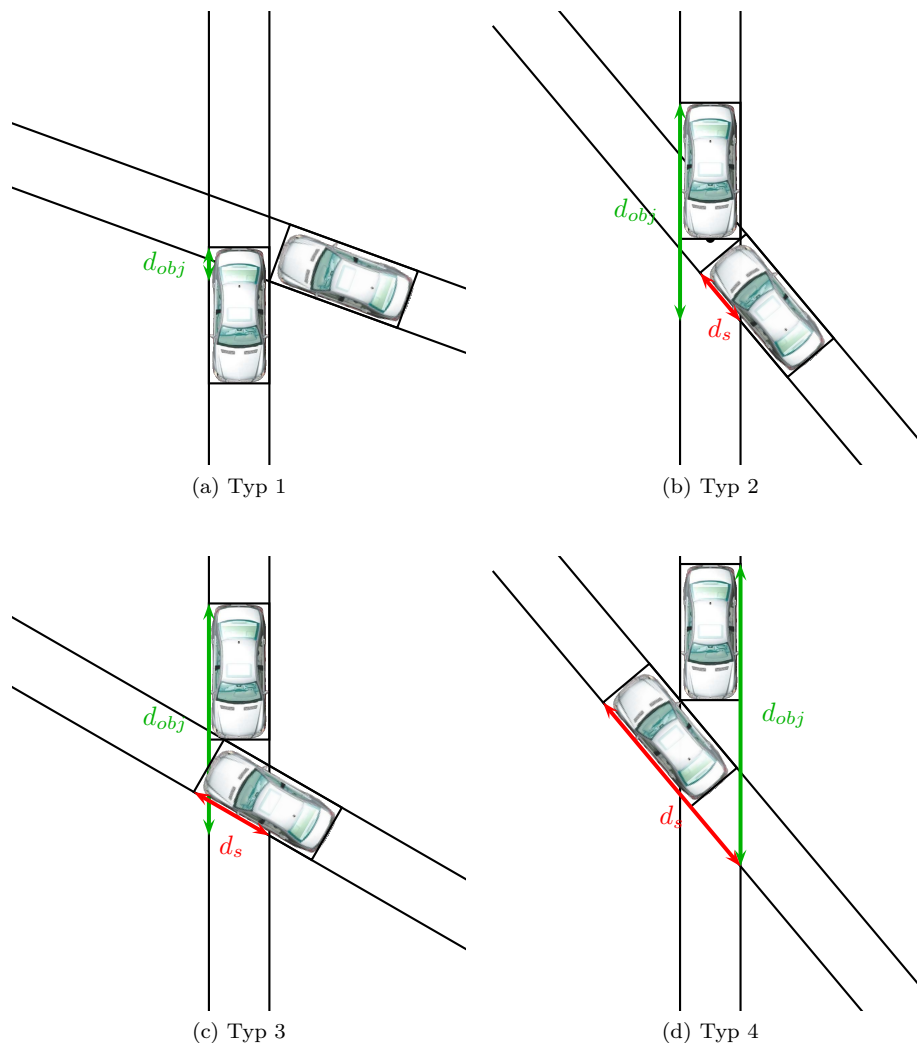


Abbildung 3.9: Die verschiedenen Kollisionstypen, die beim Szenario "Links Abbiegen" möglich sind, können in vier Typen geordnet werden.

3.4 Versuchsaufbau : Auswertung der Zeitreservenbestimmung

3.4.1 Die Simulationsumgebung KSI

Die Simulationsumgebung KSI ¹ [3, 12, 13] wurde für die Situationanalyse bei der Daimler AG entwickelt. Die Simulationsumgebung KSI ermöglicht es, beliebigen Verkehrssituation als Verkehrsszenarien zu modellieren. Die Verkehrsszenarien werden in einer Datei spezifiziert, die von dem Programm gelesen und instanziiert wird. In der Szenariodatei werden die Geometrie des Szenarios (Straße, Spurren , Markierung) sowie die Verkehrsteilnehmer definiert.

Für jeden Verkehrsteilnehmer können verschiedene Attribute definiert werden, zum Beispiel

- das Bewegungsmodell
- die Sensoren, mit den das Fahrzeug ausgerüstet ist

¹Kreuzung Simulation

- die verfügbaren Fahrerassistenzsysteme

Die Simulationsumgebung KSI is so aufgebaut, dass jeder der erwähnten Attribute als Plugin des Programms definiert ist. So kann jeder Benutzer seine eigene Systeme oder Sensormodelle definieren und evaluieren ohne den Kern des Simulationsprogramm bearbeiten zu müssen. Welcher Plugin mit welchen Parametern verwendet werden soll, wird in der Szenariodatei definiert.

Die Abbildung 3.10 zeigt eine Darstellung der Simulationsumgebung KSI.

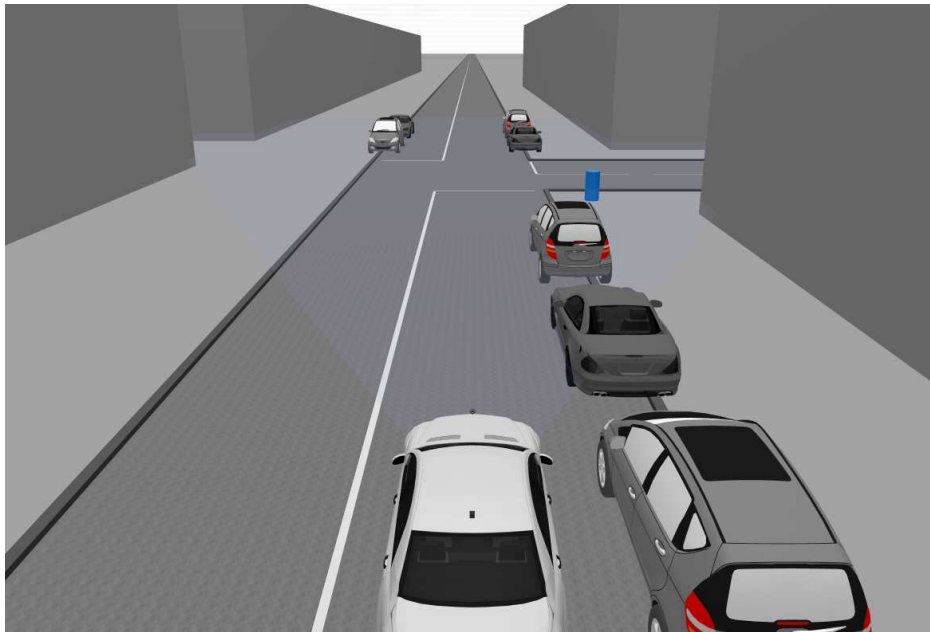


Abbildung 3.10: Szenariodarstellung in KSI

3.4.2 Implementierung der Zeitreservenberechnung

Die im Abschnitt 3.3 beschriebenen Methode zur berechnung der Zeitreserven wurden in C implementiert. So konnte diese Methode einerseits in der Simulationsumgebung KSI als Plugin eingebunden und getestet werden, und andererseits ohne weiteren Aufwand auf den Versuchsträgern verwendet werden.

Die folgende Funktionen wurden implementiert :

- Funktionen zur Berechnung der Inverse der Vandermonde Matrix
- Funktionen zur Berechnung des Interpolationspolynoms
- Funktionen zur Polynomarithmetik (Produkt mit einem Skalar, Ableitung, Integration, Rest einer Polynomdivision ...)
- Funktion zur Bildung der sturmschen Kette
- die im Abschnitt 3.1.3 beschriebene Funktionen zur Ermittlung der Nullstelle eines Polynoms
- die unterschiedlichen Ansätze zur Zeitreserven Berechnung (siehe Abschnitt 3.3)
- Funktionen zur Berechnung der Kollisionseigenschaften (siehe Anhang B)

3.4.3 Integration in KSI

Die Methode zur Zeitreservenberechnung mit Distanzpolynominterpolation sind in KSI als Plugin eingebunden worden. Um die verschiedenen Parameter einstellen zu können, wurde ein neues Plugin definiert, das als System-Fahrzeug Algorithmus dient : FoSi_DPI (FoSi steht für forward simulation und DPI für Distanzpolynominterpolation).

Die Einträge für diesen Algorithmus werden in diesem Szenariodatei Ausschnitt veranschaulicht :

```
[SA_ALGO]
SAALGO_TYPE = FOSI_DPI
DURATION = 4.0
POLYNOM_NUMBER = 2
POLYNOM_DEGREE = 9
PRECISION = 0.05
FOSI_MODE = 0
MODE = 1
BATCH_LOG_MODE = 0
```

- SAALGO_TYPE definiert der Typ des benutzten Algorithmus
- DURATION definiert wie spät in der Zukunft der Zeithorizont T liegt
- POLYNOM_NUMBER definiert die Anzahl der benutzten Polynom zur Interpolation (siehe Abschnitt 3.3.2)
- POLYNOM_DEGREE definiert den Grad der Polynome
- PRECISION definiert die gewünscht Genauigkeit bei der Berechnung der Zeitreserven
- FOSI_MODE erlaubt die gleichzeitige Benutzung des anderen FoSi Algorithmus, um einen direkt Vergleich zu ermöglichen
- MODE definiert welcher Ansatz der Zeitreservenberechnungsalgorithmus benutzt werden soll (siehe Abschnitt 3.3.2 und 3.3.4)
- BATCH_LOG_MODE aktiviert das Speichern von Logdaten für den Batchmodus

Das entwickelte Plugin erlaubt es, die Zeitreservenberechnungen so wie die Zwischenberechnungen darzustellen.

Auf der Hauptdarstellung von KSI werden die Bewegungsprognose, die Kollision sowie die verschiedenen Manöver, die im Rahmen des Algorithmus berechnet werden. Auf der Abbildung 3.11 sieht man, dass die Prognose zu einer Kollision führt. Die Prognose für jedes Fahrzeug wird in grau gezeichnet, und die Lage der Fahrzeug zum Kollisionszeitpunkt in orange. In rot und gelb werden die Wirkungen jeweils einer erfolgreichen Brems- und Lenk-Manöver.

Aber viel mehr Informationen stehen dem Benutzer zur Verfügung mit Hilfe eines Widgets. Dieses Widget stellt alle wichtigen Berechnungsstufen dar. Pro Zeitreserve werden zwei Reiter die verschiedene Information dargestellt :

- im ersten Reiter werden die Distanzverläufe, die Interpolationspolynome und die Stützstellen gezeichnet. Ein farbiges Kreuz stellt die entsprechende Zeitreseve auf dem Graphik dar
- im zeiten Reiter werden in eine Tabelle alle Abstände zwischen dem Ego-Fahrzeug und jedem Objekt-Fahrzeug aufgelistet

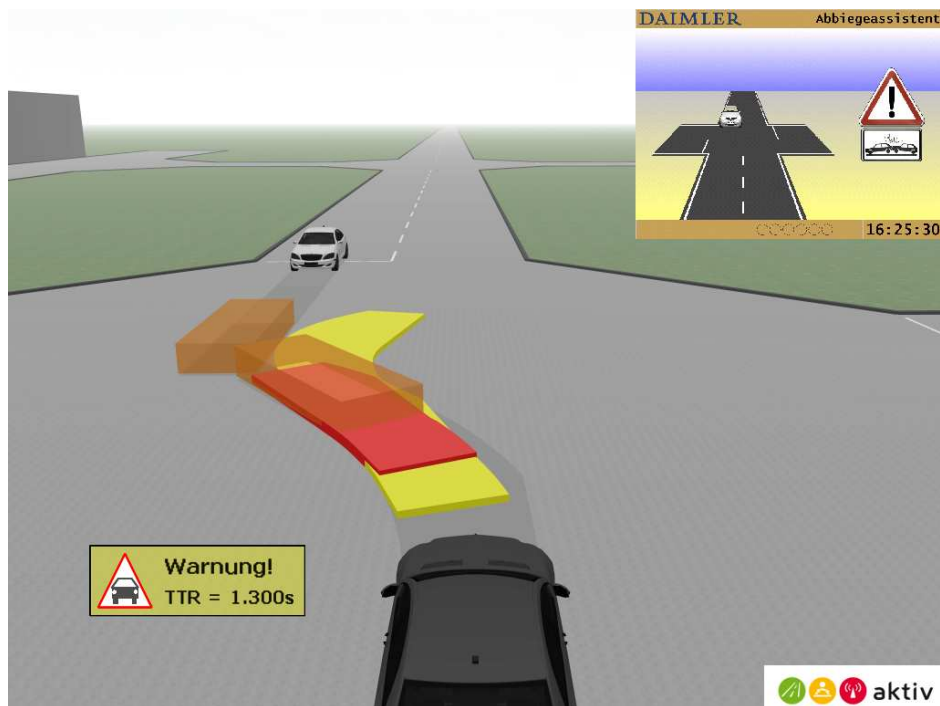


Abbildung 3.11: Main View

Auf der Abbildung 3.12 wird der erste Reiter für TTC dargestellt.

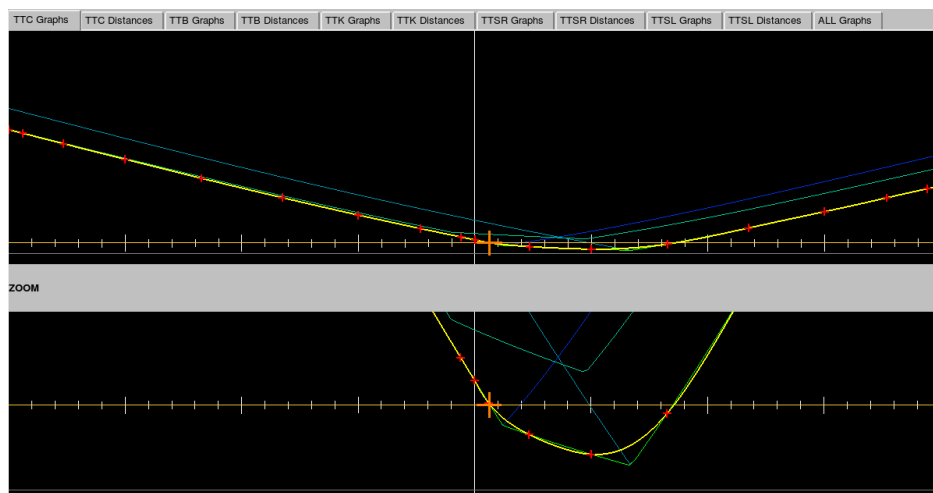


Abbildung 3.12: Die Zwischenberechnungen können dank eines Widgets dargestellt werden

In gelb werden die Interpolationspolynome mit den entsprechenden Interpolationsknoten durch roten Kreuzchen dargestellt. Die Daten sind direkte Ausgaben der Zeitreservenalgorithmen. Auf der x-Achse wird TTC durch einen dickeren orangenen Kreuz bezeichnet. Um diese Darstellung zu ergänzen werden die Abstände mit einem sehr kleinem Zeitschritt vom Plugin berechnet. Die dünnen blauen Linien entsprechen den Distanzverläufe der vier Seiten des betroffenen Objekt-Fahrzeuges. In hell grün wird der Abstand des Objekt-Fahrzeuges zum Ego-Fahrzeug, das Minimum aus den vier Seitenabstände, dargestellt. So kann die Güte des Zwischenschrittes Interpolation mit der Wahrheit

geschätzt werden.

3.4.4 Szenario für die Auswertung : “Abbiegen”

Um die Eigenschaften der neuen Methode zur Zeitreservenberechnung, die Zeitreservenberechnung mit Distanzpolynominterpolation, zu bewerten, wird sie mit der Vorwärtssimulation verglichen. Die Auswertungsdaten werden im Szenarios “Links Abbiegen” in KSI generiert. Die Verkehrssituation dieses Szenarios ist die folgende :

- die Situation betrifft zwei Fahrzeuge an einer Kreuzung
- das Ego-Fahrzeug biegt nach links ab
- das Objekt-Fahrzeug fährt gerade aus, dem Ego-Fahrzeug entgegen

Die Abbildung 3.13 stellt die Linksabbiegesituation dar.

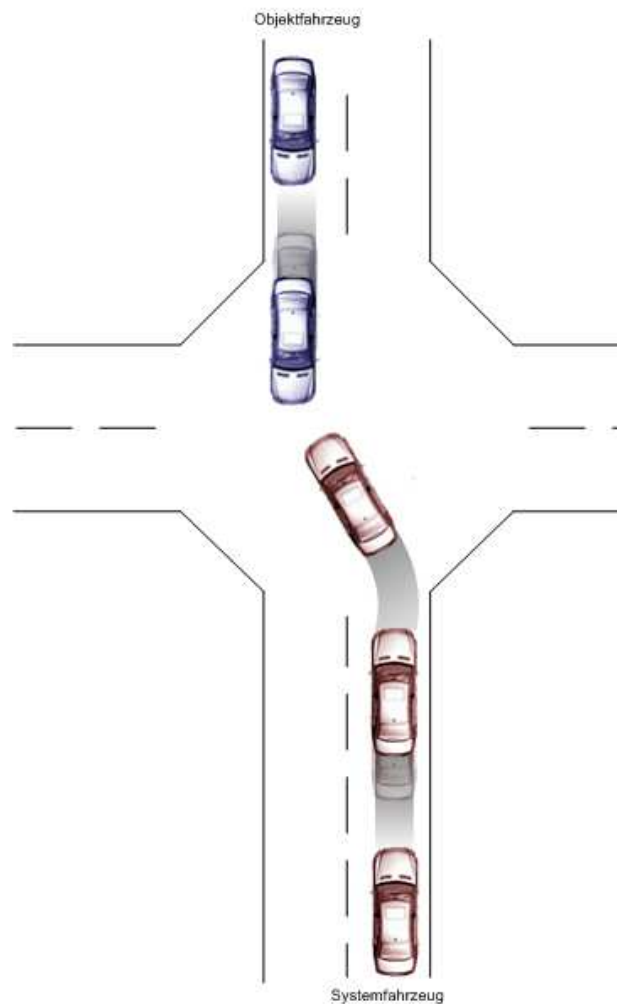


Abbildung 3.13: Das “Links Abbiegen” Szenario

3.4.5 Variationsbreite des Versuchsaufbaus

Dem Objekt-Fahrzeug werden eine Startposition, eine Startgeschwindigkeit und eine Startbeschleunigung zugewiesen. Die Startposition des Objekt-Fahrzeugs wird variiert, damit alle möglichen Kollisionsfällen erzeugt werden können. (40 verschiedenen Startpositionen, jeweils ein Meter Abstand zwischen zwei Startpositionen). Dem Ego-Fahrzeug werden zwanzig verschiedenen gemessenen Abbiegetrajektorien zugewiesen.

Jedes Paar (Startposition des Objektfahrzeugs, Trajektorie Systemfahrzeugs) wird mit jedem Ansatz der Zeitreservenberechnung mit Distanzpolynominterpolation (ein Paar (Anzahl, Grad) bildet einen Ansatz) und mit zwei Ansätzen der Vorwärtssimulation simuliert :

1. die Vorwärtssimulation mit einem Zeitschritt von einer Milisekunde (1 ms) gilt als Referenz für die erzielte Genauigkeit des Verfahrens
2. im Sinne der Laufzeit wird die Vorwärtssimulation mit einem Zeitschritt von fünfzig Milisekunden (50 ms) genommen, denn dieser Zeitschritt wird auf dem Versuchsträger benutzt, um eine genügend schnelle Ausführung zu haben

Jede Konstellation (Startposition, Trajektorie, Zeitreservenberechnungsansatz) wird einmal simuliert und die entsprechenden Ergebnisse werden gespeichert. Eine Simulation entspricht bis 400 verschiedene Prognose und Zeitreservenberechnungen. Dadurch müssen zwei verschiedenen Zeitlinien im Betracht gezogen werden :

1. die reale Zeitachse entspricht dem was in der Messung oder Simulation tatsächlich passiert, diese Zeit entspricht also einer Art Realität
2. die zweite Zeitlinie beschreibt eine rein virtuelle Zeit, die , gegeben eine gegenwärtige Situation, eine virtuelle hypthetische Zukunft darstellt. Jeder Zeitpunkt der reellen Zeitachse, kann der Ursprung einer virtuellen Zeitlinie sein

Die Abbildung 3.14 zeigt die verschiedenen Zeitlinien. In schwarz wird die reelle Zeitachse abgebildet. Diese Zeitachse wird mit dem Buchstaben τ gekennzeichnet. Wie erklärt bildet jeder τ_i einen Ursprung für eine virtuelle Zeitlinie. Diese Zeitlinien und ihre Erlebnisse werden blasser dargestellt. Jede virtuelle Zeitlinie entspricht die Berechnung einer Prognose, einer hypothetischen Zukunft. Für die Bewertung werden diese virtuellen Zeitlinien als Proben bezeichnet : eine Probe entspricht einen Tupel (Startposition, Trajektorie, τ).

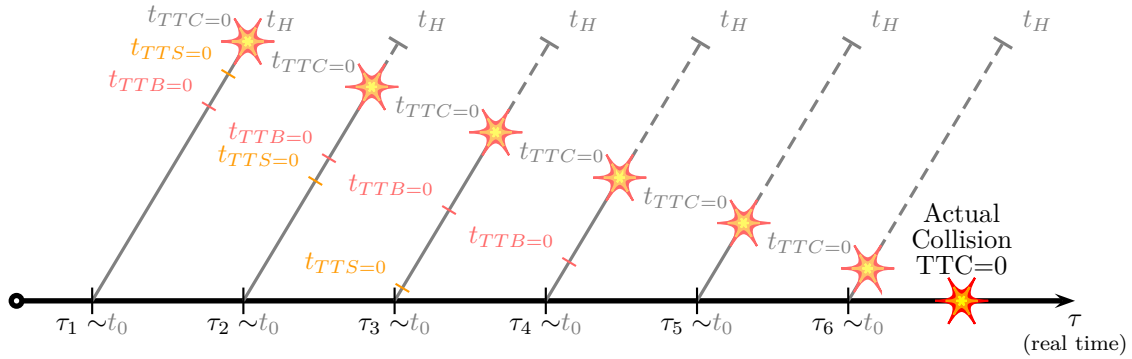


Abbildung 3.14: Die Berechnung von Zeitreserven betrifft eine hypothetische Zukunft, deren Zeitlinie von der der Realität unterschiedlich sein kann.

3.5 Ergebnisse

3.5.1 Relevante Auswertungsgrößen für TTC

Im Folgenden stellt Ω die Menge aller Proben dar. Für ein Verfahren A (es kann sowohl die Vorwärtssimulation (siehe Abschnitt 2.4) als auch die Zeitreservenberechnung mit Distanzpolynominterpolation sein), ist Ω_{P_A} die Untermenge der Proben aus Ω , für die das Verfahren A einen TTC berechnet hat. Analog ist Ω_{N_A} die Untermenge der Proben aus Ω , für die kein TTC berechnet worden ist, weil keine Kollision vorliegt. Daraus folgt :

$$\Omega_{P_A} \cup \Omega_{N_A} = \Omega \quad (3.5.1)$$

$$\Omega_{P_A} \cap \Omega_{N_A} = \emptyset \quad (3.5.2)$$

Die Zeitreservenberechnungsverfahren für TTC werden nach mehreren Größen bewertet :

1. die Laufzeit der Berechnung (der Rechenaufwand ist der größte Nachteil der Vorwärtssimulation)
2. die Erkennungsrate: wie oft wird eine Kollision detektiert im Vergleich zur Referenz :

$$r_R = \frac{|\Omega_{P_A} \cap \Omega_{P_{GT}}|}{|\Omega_{P_{GT}}|} \quad (3.5.3)$$

3. die Fehlerrate:

$$r_E = \frac{|\Omega_{N_A} \cap \Omega_{P_{GT}}|}{|\Omega_{P_{GT}}|} = 1 - r_R \quad (3.5.4)$$

4. die Falsepositive Rate:

$$r_F = \frac{|\Omega_{P_A} \cap \Omega_{N_{GT}}|}{|\Omega_{P_A}|} \quad (3.5.5)$$

5. die Genauigkeit der Berechnung, im Fall einer beidseitigen Erkennung

3.5.2 Ergebnisse für die Berechnung der Zeitreserve TTC

3.5.2.1 Vergleich der Laufzeiten

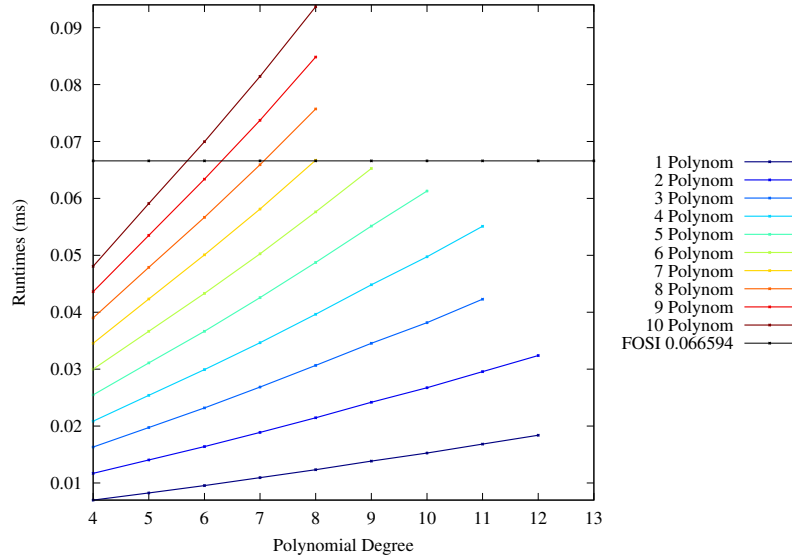


Abbildung 3.15: Vergleich der durchschnittlichen Laufzeiten in Abhängigkeit von der TTC-Berechnung.

In der Abbildung 3.15 werden die Laufzeiten der Berechnung von TTC dargestellt. In Schwarz ist die durchschnittliche Laufzeit der Vorwärtssimulation mit einem Zeitschritt von 50ms gezeichnet : etwa 67 μ s. Die Laufzeit der Zeitreservenberechnung mit Distanzpolynominterpolation wächst mit dem Grad und mit der Anzahl der verwendeten Polynomen, aber die durchschnittliche Laufzeit ist in fast allen Fälle kleiner als für die Vorwärtssimulation.

3.5.2.2 Vergleich der Erkennungs-, Fehler- und Falsepositiverate

Die Abbildung 3.16 stellt eine Partition der Menge Ω dar. Jede Spalte entspricht dabei einem Ansatz der Zeitreservenberechnung mit Distanzpolynominterpolation. Die letzte Spalte stellt die Vorwärtssimulation mit einem Zeitschritt von 50 ms dar.

Etwa 86% aller Proben haben für die Referenz ergeben. Dieser Anteil der Proben wird mit den Farben und dargestellt, und entspricht jeweils die Untermengen $\Omega_{NA} \cap \Omega_{NGT}$ und $\Omega_{PA} \cap \Omega_{NGT}$ von Ω .

Die Untermenge $\Omega_{NA} \cap \Omega_{NGT}$ ist für die Bewertung der Verfahren nicht relevant Es ist offensichtlich, dass in völlig unkritischen Situationen sowohl die Referenz als auch die zu bewertenden Verfahren das gleiche Ergebnis (keine Kollision) liefern. Deswegen wird auch für die Berechnung der Rate der Falsepositiveproben $\Omega_{PA} \cap \Omega_{NGT}$ (in), die Anzahl der positiven Proben des bewerteten Verfahrens $|\Omega_{PA}|$ statt $|\Omega_{NGT}|$, als Vergleichsgröße genommen (siehe Gleichung (3.5.5)). So werden die Erkennungs- und Fehlerraten durch zu vielen irrelevanten Situationen nicht verfälscht.

Die Untermenge in entspricht den fehlerhaften Proben im Vergleich zur Referenz : $\Omega_{NA} \cap \Omega_{PGT}$. In werden alle positiven Proben $\Omega_{PA} \cap \Omega_{PGT}$ dargestellt : das bewertete Verfahren sowie die Referenz haben ein TTC berechnet.

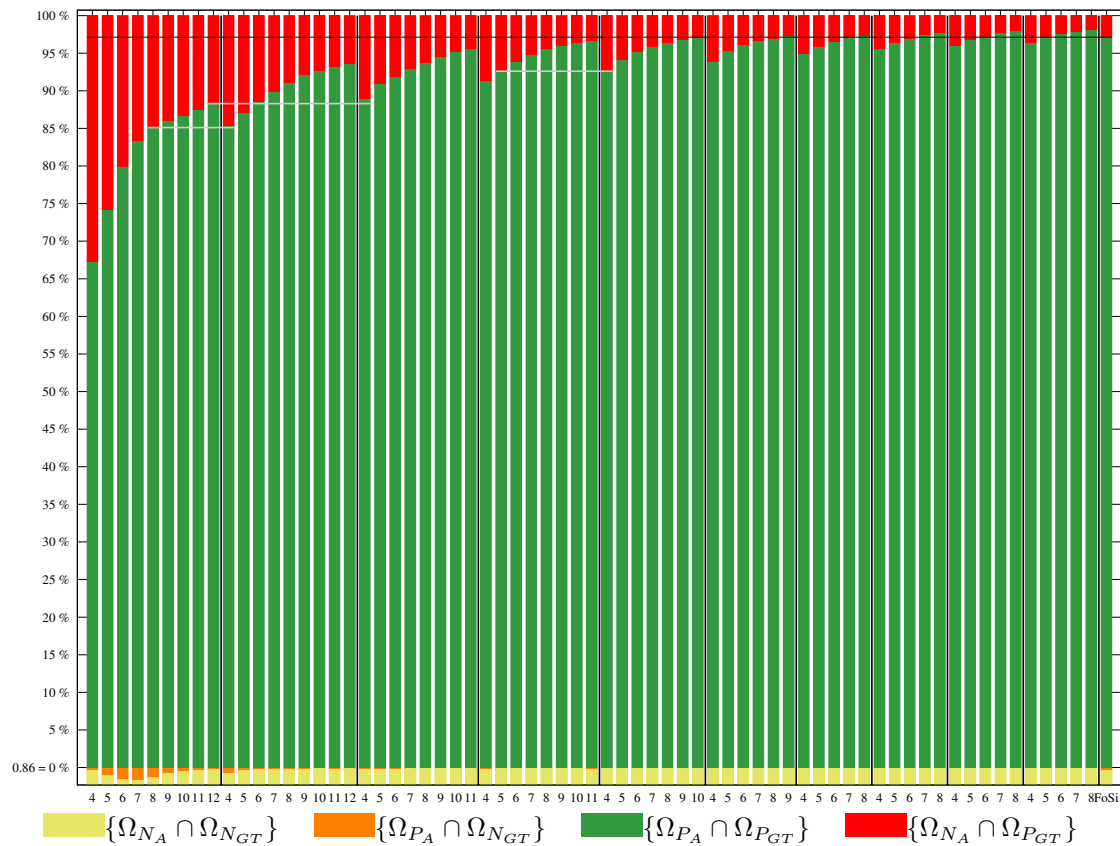


Abbildung 3.16: Darstellung der Menge aller Proben Ω . Jede Spalte entspricht eine Konfiguration von DPI und die letzte Spalte stellt FoSi mit einem Zeitschritt von 50 ms dar

Die Abbildungen 3.17a und 3.17b zeigen eine Darstellung der jeweiligen Fehler- und Falsepositivrate : im Vergleich zum Ansatz der Vorwärtssimulation, der momentan auf dem Versuchsträger läuft, haben fast alle Ansätze der Zeitreservenberechnung mit Distanzpolynominterpolation eine niedrigere Falsepositivrate. In alle Fälle ist die Falsepositivrate extrem niedrig : die Vorwärtssimulation hat zum Beispiel eine Rate von etwa 4 % (Siehe Abbildung 3.17b). Die Fehlerrate ist aber ein bisschen höher : die Vorwärtssimulation hat eine Fehlerrate von ungefähr 2.5 %, während die meisten verschiedenen Ansätze der Zeitreservenberechnung mit Distanzpolynominterpolation eine Fehlerrate von 2.5 bis 10 % haben (siehe Abbildung 3.17a).

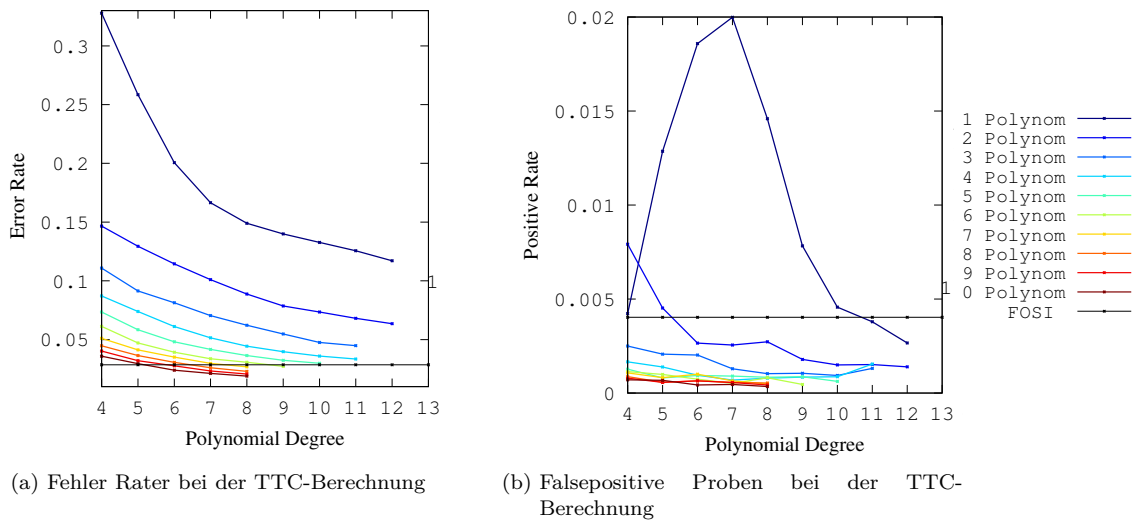


Abbildung 3.17

Die niedrige Falschpositivrate liegt daran, dass die kleinen Unterschiede in den Verfahren, dazu führen, dass die Kollisionen etwas früher oder später im Sinne der Simulationszyklen (reelle Zeitlinie) erkannt werden.

Die grauen Linien in der Abbildung 3.16 zeigen, dass die Menge an Information, die für die Interpolation benutzt wird, großen Einfluß auf das Ergebnis hat. Zum Beispiel zeigt die mittlere graue Linie, dass drei verschiedene Ansätze der Zeitreservenberechnung mit Distanzpolynominterpolation fast die gleiche Fehlerrate erzeugen :

- ein Polynom 12. Grades
- zwei Polynome 6. Grades
- drei Polynome 4. Grade

Alle drei Ansätze brauchen die selbe Anzahl an Interpolationsknoten (13 Knoten) :

- es werden 13 Knoten gebraucht, um ein Polynom 12. Grades zu berechnen
- bei zwei Polynomen 6. Grades werden $2 \times 7 - 1 = 13$ Knoten gebraucht, da einer der Knoten zwei mal benutzt wird
- bei drei Polynomen 4. Grades werden $3 \times 5 - 2 = 13$ Knoten gebraucht, da zwei Knoten zwei mal benutzt werden

Obwohl die Verteilung dieser Knoten von einem Fall zum anderen unterschiedlich ist, bleibt die Menge an ursprüngliche Information ungefähr gleich. Deshalb sind auch die Fehlerraten sehr ähnlich.

3.5.2.3 Vergleich der Genauigkeit in der TTC-Berechnung

Die Abbildung 3.18 zeigt das durchschnittliche Fehler in der Berechnung von TTC aller Proben aus der Menge $\Omega_{PA} \cap \Omega_{P_{GT}}$, für alle Ansätze der Zeitreservenberechnung mit Distanzpolynominterpolation im Vergleich zur Vorwärtssimulation.

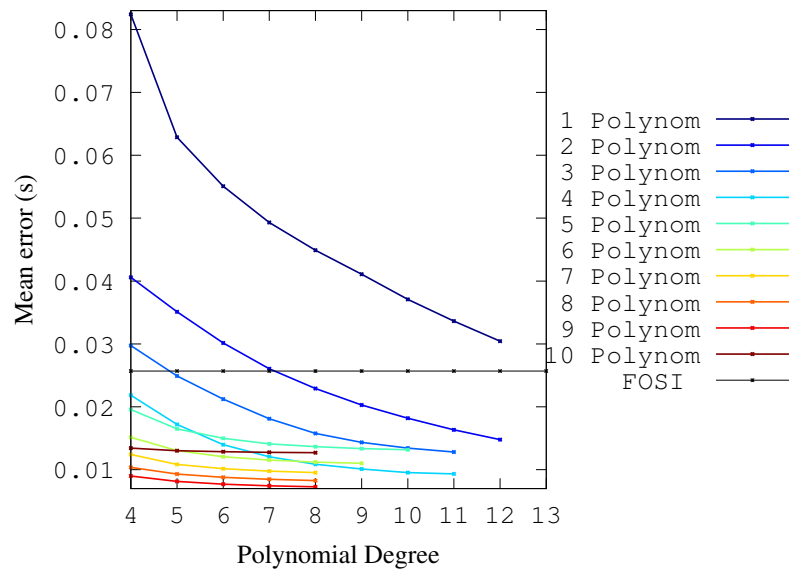


Abbildung 3.18: Durchschnittliche Fehler bei der TTC-Berechnung

3.5.2.4 Fazit

Das Hauptziel der Zeitreservenberechnung mit Distanzpolynominterpolation war die Laufzeitsverminderung, denn die Vorwärtssimulation rechenaufwendig ist. Dieses Ziel ist erfüllt : für bestimmten Parametersätze, zum Beispiel vier Polynome des achten Grades, hat die TTC-Berechnung eine Laufzeit von $38 \mu\text{s}$, gegen $67 \mu\text{s}$ für die Vorwärtssimulation. Dabei ist die Fehlerrate nur um 2% höher, für eine kleinere Rechenungenauigkeit : 12 ms Fehler gegen 25 ms Fehler für die Vorwärtssimulation.

3.5.3 Auswertung der Reaktionszeitreserven

Um die anderen Zeitreserven zu bewerten, wird hier das Beispiel von TTB genommen.

3.5.3.1 Vergleich der Laufzeiten

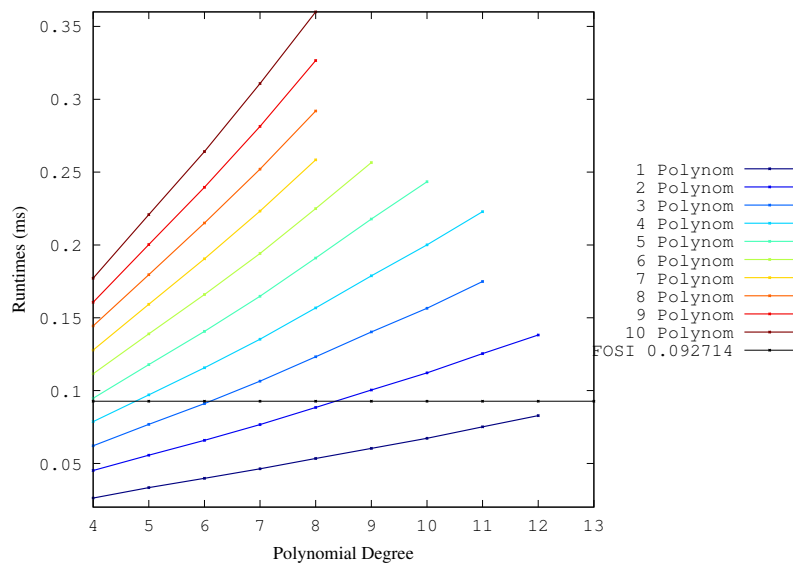


Abbildung 3.19: Durchschnittlichen Laufzeiten für die TTB-Berechnung.

In der Abbildung 3.19 werden die Laufzeiten der Berechnung von TTB dargestellt. In Schwarz ist die durchschnittliche Laufzeit von FoSi mit einem Zeitschritt von 50ms. Für die Berechnung von TTM ist insgesamt die Zeitreservenberechnung mit Distanzpolynominterpolation nicht besser als die Vorwärtssimulation : die Laufzeiten sind sich sehr ähnlich. Die Erklärung liegt daran, dass die Vorgehensweise die gleiche ist. Die Berechnung der Zeitreserven erfolgt nur durch sukzessives Versuchen, was sehr rechenaufwendig ist.

3.5.3.2 Vergleich der Genauigkeit in der TTB-Berechnung

Die Abbildung 3.20 zeigt das durchschnittliche Fehler in der Berechnung von TTB aller Probe aus der Menge $\Omega_{PA} \cap \Omega_{PGT}$, für alle Ansätze der Vorwärtssimulation. In den meisten Fälle bleibt die Genauigkeitsfehler unter 50 ms, was sehr ausreichend ist.

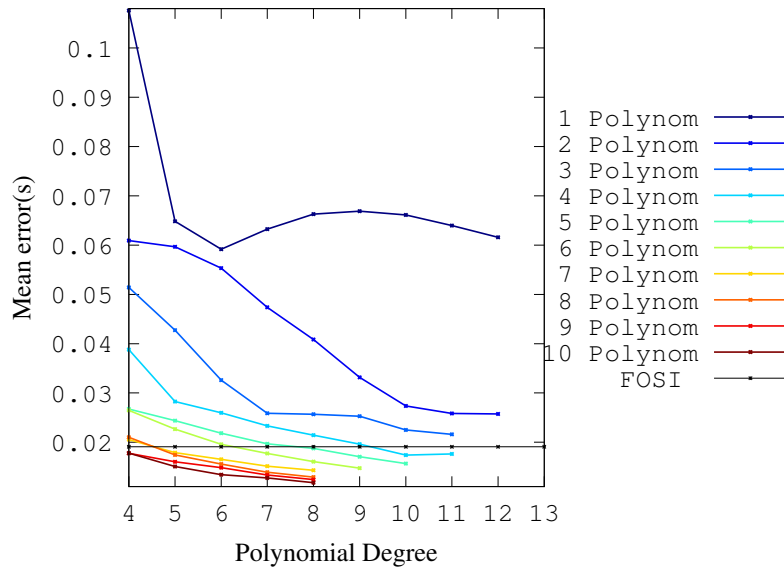


Abbildung 3.20: Durchschnittliche Fehler bei der TTB-Berechnung

3.5.3.3 Fazit

Für die Berechnung der Reaktionszeitreserven hat die Zeitreservenberechnung mit Distanzpolynominterpolation im Vergleich zur Vorwärtssimulation keine besondere Vorteile.

3.5.4 Ergebnisse für die direkte Berechnung der Zeitreserve TTB

In diesem Absatz werden die Ergebnisse der im Abschnitt 3.3.5 beschriebenen direkten Berechnung von TTB präsentiert. Obwohl diese Rechenweise sehr schnell ist, ist sie nur in besonderen Fällen anwendbar, und nur für das Vollbremsmanöver geeignet.

Die direkte Berechnung von TTB hat die folgenden Eigenschaften gezeigt :

- die Laufzeit ist sehr niedrig : $2.4 \mu\text{s}$
- die Fehlerrate ist : $r_E = 0.030$
- die Erkennungsrate ist : $r_R = 0.97$
- die Falsepositivrate ist : $r_F = 0.023$
- das durchschnittliche Fehler bei der Berechnung beträgt : 66 ms

3.6 Fazit

In diesem Kapitel wurde eine neue effiziente Methode zur Zeitreservenberechnung entwickelt und vorgestellt. Diese Methode basiert auf zwei neuen Elementen :

1. die Einführung eines Fahrzeugmodells, deren Zustandsgleichungen integriert worden sind: $\mathbf{x}(t) = \mathcal{M}(\mathbf{x}(t_0), \mathbf{u}, t)$, erspart die Rechenaufwendige Verwendung eines numerisches zeitdiskretes iteratives Fahrzeugmodells wie bei der Vorwärtssimulation: $\mathbf{x}_k = \mathbf{x}_{k-1} + m_N(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \Delta t)$

2. die Einführung einer Distanzfunktion \mathcal{D} , deren Abbildungsraum stetig ist. Die Stetigkeit des Abbildungsraums macht die Funktion für die Interpolation besonders geeignet. So kann die Kollisionszeitreserve als Nullstelle einer Funktion bestimmt werden.

Die Auswertung der Methode mit Hilfe des Verkehrsszenario "Links Abbiegen" hat aufgezeigt, dass für die Bestimmung der Kollisionszeitreserve die Interpolation der Distanzfunktion sich als schneller und genauer als die Vorwärtssimulation herausgestellt hat.

Bei der Berechnung der Reaktionszeitreserven ist der Zeitgewinn nicht so deutlich. Der Grund dafür ist, für beiden Methoden müssen die Manöverauslösezeitpunkte gesucht und getestet werden. Für das spezifische Szenario "Links Abbiegen" konnte jedoch eine Methode entwickelt werden, um die Zeitreserve TTB direkt berechnen zu können.

Die Distanzfunktion stellt auch ein wichtige Grundlage für das im nächsten Kapitel vorgestellte Verfahren dar.

Kapitel 4

Bestimmung der Kollisionswahrscheinlichkeit unter Berücksichtigung von Sensorunsicherheiten

4.1 Motivation

Die Berücksichtigung der Sensorunsicherheit erlaubt es, die Realität korrekter abzubilden. Die korrekte Abbildung der Realität in einem Assistenzsystem gewährleistet eine höhere Zuverlässigkeit des Systems. Die Abbildung 4.1 zeigt eine idealisierte Weltmodellierung, in der die Sensordaten mit keiner Unsicherheit behaftet sind. Eine solche idealisierte Darstellung führt dazu, dass die Systemes mit scharfen Aktivierungsschwellen (z.B. Zeitreserven) arbeiten müssen und, dadurch messfehleranfälliger sind.

Im Gegenteil stellt die Abbildung 4.2 eine realitätsgetreuere Weltdarstellung. Die Systeme, die mit einer solchen Modellierung arbeiten können, verwenden auch stochastische Schwellen. Die Benutzung solcher Schwellen macht die Systeme robuster.

Die Zuverlässigkeit von Fahrerassistenzsysteme ist eine der wichtigsten und eine unumgängliche Eigenschaft für Fahrzeassistenzsysteme. Die Verwendung von Verteilungen, die die Unsicherheiten berücksichtigen, als Basis für Auslöseschwellenn vermeidet Fehlauflösungen von Assistenzsystemen.

4.2 Zielsetzung

Eine frühere Arbeit ([14]) hat sich bereits mit der echtzeitfähigen Approximation von Reaktionszeitreserven beschäftigt. Das Ergebnis stellte ein Verfahren zur Wahrscheinlichkeitsverteilung von Zeitreserven.

Als Ergebnis des entwickelten Verfahren war eine Approximation der bedingten Verteilung der Reaktionszeitreserven bekannt. Wie im Abschnitt 2.2 definiert, werden Reaktionszeitreserven einem reellen Wert zugeordnet, vorausgesetzt, dass die Kollisionszeitreserven endlich ist. Wenn es keine Kollision vorhanden ist, wird die Kollisionszeitreserve als Minimum einer leeren Menge difiniert, was

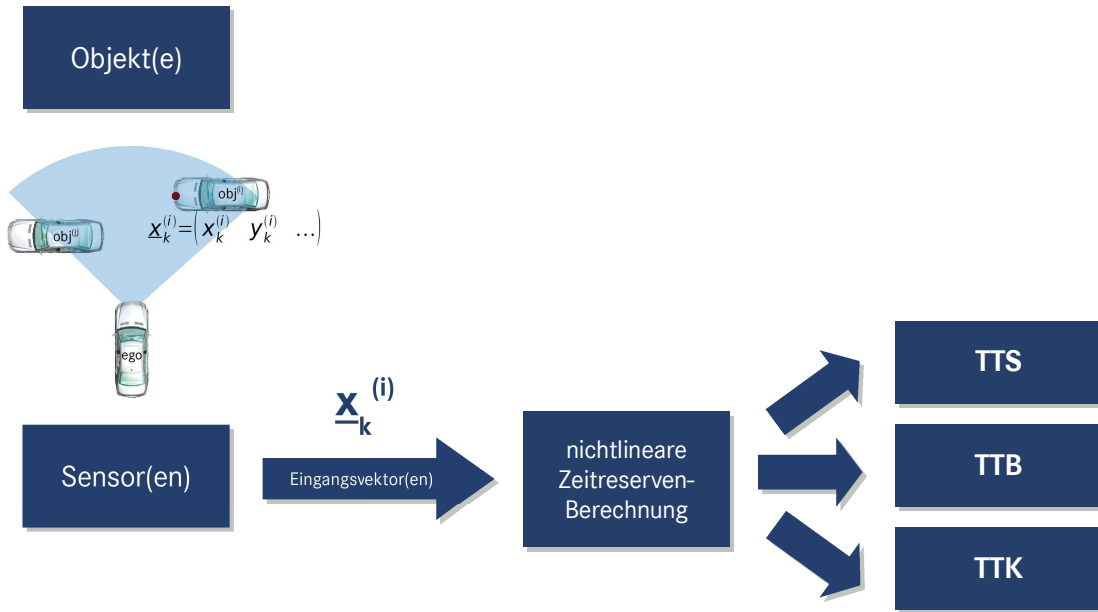


Abbildung 4.1: Idealisierte Annahme unverrauschter Messdaten.

unendlich ist : $TTC = \min\{\emptyset\} = \infty$.

Die Ergebnisverteilungen des Verfahrens sind Reaktionszeitreserven, die durch die Kollisionswahrscheinlichkeit bedingt sind :

$$P(TTM \leq t \mid TTC \in \mathbb{R}) \quad (4.2.1)$$

Wichtig als Auslöseschwellengröße für die Assistenzsysteme ist aber die Verbundwahrscheinlichkeit :

$$P(TTM \leq t, TTC \in \mathbb{R}) \quad (4.2.2)$$

Die Definition der bedingten Wahrscheinlichkeit führt dazu, dass die als Systemschwelle gewünschte Verteilung $P(TTM \leq t, TTC \in \mathbb{R})$ mit Hilfe der bekannten Verteilung $P(TTM \leq t \mid TTC \in \mathbb{R})$ und der Kollisionswahrscheinlichkeit $P(TTC \in \mathbb{R})$ berechnet werden kann :

$$P(TTM \leq t \mid TTC \in \mathbb{R}) = \frac{P(TTM \leq t, TTC \in \mathbb{R})}{P(TTC \in \mathbb{R})} \quad (4.2.3)$$

$$P(TTM \leq t, TTC \in \mathbb{R}) = P(TTM \leq t \mid TTC \in \mathbb{R}) \cdot P(TTC \in \mathbb{R})$$

Die Abbildung 4.3 stellt die Unterschiede zwischen der bedingten Wahrscheinlichkeit und der Verbundwahrscheinlichkeit.

Ziel dieses Kapitel ist es, eine echtzeitfähige Approximation der Kollisionswahrscheinlichkeit $P(TTC \in \mathbb{R})$ vorzustellen, die diese Sensorunsicherheiten berücksichtigen. Dank dieser Wahrscheinlichkeit wird es möglich sein, die Verbundwahrscheinlichkeiten $P(TTM \leq t, TTC \in \mathbb{R})$ zu bestimmen, und bessere Auslöseschwellengröße für Fahrerassistenzsystemen zu definieren.

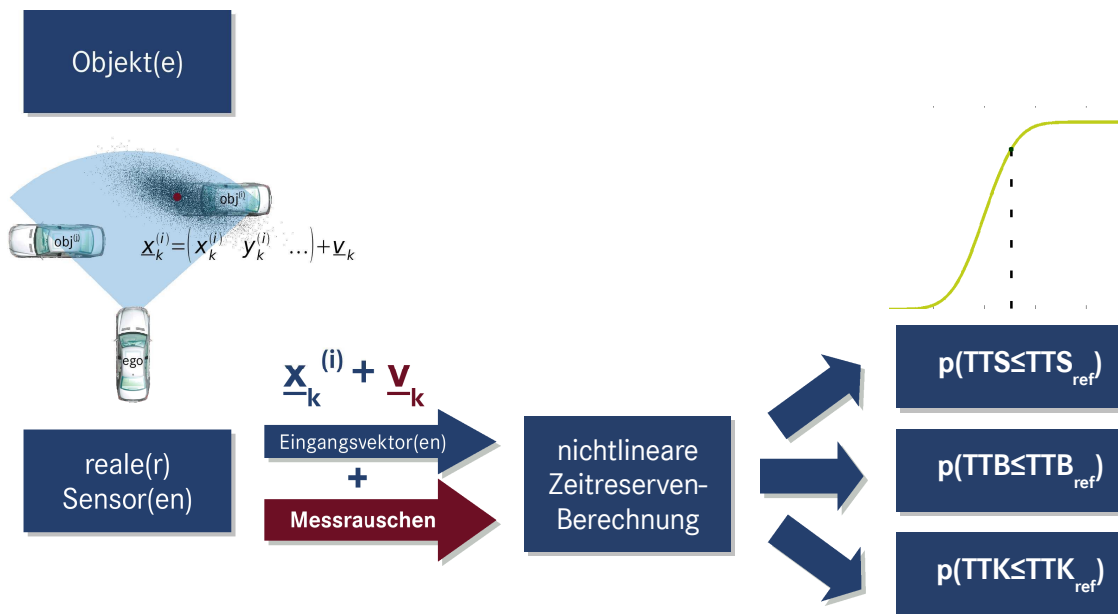


Abbildung 4.2: Realitätsgetreue Annahme verrauschter Messdaten.

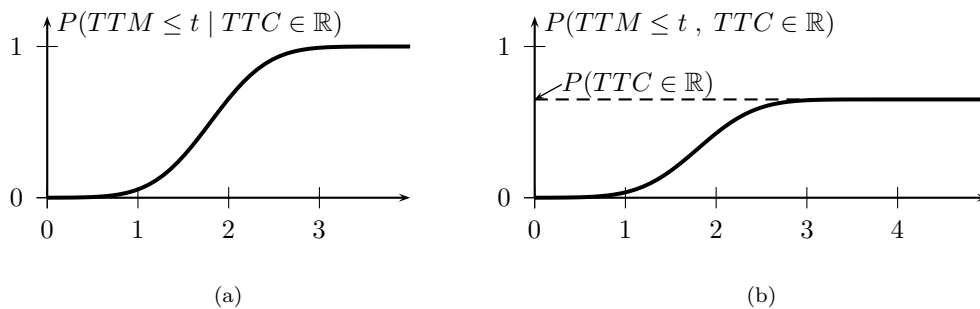


Abbildung 4.3

4.3 Stochastische Grundlagen

4.3.1 Normalverteilung

Es gibt viele unterschiedliche Typen von Standardverteilungen (siehe z.B. [15, 16]). Eine der wichtigsten Verteilungen, die häufig in der Natur bzw. bei technischen Prozessen auftritt, ist die *Normalverteilung* oder auch *Gauß-Verteilung*. Die Dichtefunktion der Normalverteilung ist symmetrisch bezüglich des Erwartungs- oder auch Mittelwerts $\mu = E(X)$ und kann anhand von Mittelwert μ und Standardabweichung σ vollständig parametrisiert werden. Sie ergibt sich zu

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right) \quad (4.3.1)$$

Es wird auch geschrieben

$$X \sim \mathcal{N}(\mu, \sigma) \quad (4.3.2)$$

Dementsprechend kann die zugehörige kumulative Verteilungsfunktion durch Integration über die Dichtefunktion innerhalb der gewünschten Intervallgrenzen erhalten werden. Die entsprechende Verteilungsfunktion lautet

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x \exp\left(-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right) dt \quad (4.3.3)$$

Die Abbildungen 4.4a und 4.4b stellen eine Dichte- und eine Verteilungsfunktion für eine normalverteilte Zufallsvariable exemplarisch dar.

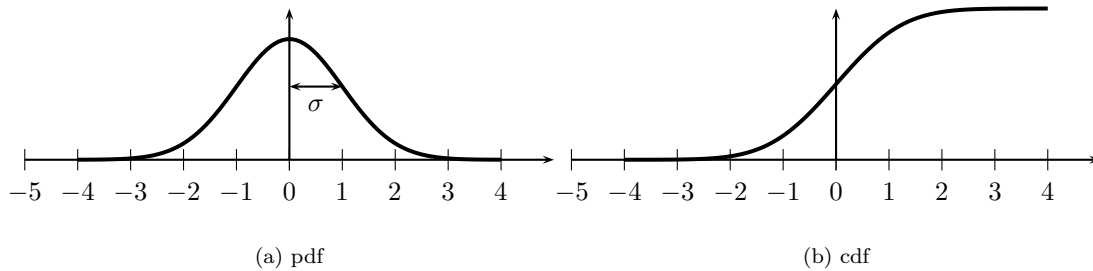


Abbildung 4.4

4.3.2 Multivariate Verallgemeinerung

Gibt es nicht nur eine Zufallsvariable X , sondern einen Vektor $X = (X_1, \dots, X_n)$ von n Zufallsvariablen, so entsteht eine multivariate Normalverteilung, deren Dichtefunktion folgendermaßen bestimmt ist:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mu})^T \Sigma^{-1}(\mathbf{x} - \bar{\mu})\right) \quad (4.3.4)$$

Es wird auch geschrieben

$$X \sim \mathcal{N}(\bar{\mu}, \Sigma) \quad (4.3.5)$$

$\bar{\mu}$ steht für den Mittelwertvektor (Erwartungsvektor) und Σ für die Kovarianzmatrix. Die Kovarianzmatrix wird folgendermaßen gebildet:

$$\Sigma = \begin{pmatrix} \text{var}(X_1) & \text{cov}(X_1, X_2) & \dots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \text{var}(X_2) & & \text{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \text{cov}(X_n, X_2) & \dots & \text{var}(X_n) \end{pmatrix} \quad (4.3.6)$$

Hierbei bezeichnet $\text{var}(X_i) = E[(X_i - \mu_i)^2] = \sigma_{X_i}^2$ die Varianz (quadratische Abweichung vom Erwartungswert) einer Eingangsvariablen X_i und $\text{cov}(X_i, X_j) = E[X_i - \mu_i]E[X_j - \mu_j]$ die Kovarianz zweier Eingangsvariablen X_i und X_j . Je stärker zwei Eingangsvariablen miteinander korreliert sind, desto größer ist deren gemeinsame Kovarianz. Die Kovarianzmatrix Σ ist eine symmetrische,

positiv semidefinite Matrix. Um die multivariate kumulative Verteilungsfunktion zu bilden, muss die Dichtefunktion über jede einzelne Zufallsvariable aus dem Zufallsvariablenvektor \vec{X} integriert werden:

$$F(\mathbf{x}) = P(X^{(1)} \leq x_1, X^{(2)} \leq x_2, \dots, X^{(n)} \leq x_n) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \dots \int_{-\infty}^{x_n} f(\mathbf{t}) dt_1 dt_2 \dots dt_n \quad (4.3.7)$$

4.3.3 Transformation von Wahrscheinlichkeitsverteilungen

4.3.3.1 Varianzfortpflanzungsgesetz für affine Funktionen

Theorem 5 (Varianzfortpflanzungsgesetz). *Gegeben sei X ein Normalverteilungsvektor aus \mathbb{R}^n , dessen Erwartungsvektor μ_X und dessen Kovarianzmatrix Σ_X sind. Gegeben sei A eine Matrix aus $M_{r,n}(\mathbb{R})$ und B ein Vektor aus \mathbb{R}^r . Der Zufallsvektor*

$$Y = A \cdot X + B$$

ist auch ein Normalverteilungsvektor, dessen Erwartungsvektor $\mu_Y = A \cdot \mu_X + B$ und dessen Kovarianzmatrix $\Sigma_Y = A \cdot \Sigma_X \cdot A^T$ sind.

4.3.3.2 Varianzfortpflanzungsgesetz im nicht linearen Fall

Gegeben sei eine ableitbare Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^r$. Die Jacobi-Matrix J_f der Funktion f ist die $r \times n$ -Matrix sämtlicher erster partieller Ableitungen von f :

$$J_f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dots & \dots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_r}{\partial x_1}(\mathbf{x}) & \dots & \dots & \frac{\partial f_r}{\partial x_n}(\mathbf{x}) \end{pmatrix} \quad (4.3.8)$$

Um zu berechnen, wie sich der Fehler des Eingangsvektor $\Delta \mathbf{x}$ auf den Funktionswert $\mathbf{y} = f(\mathbf{x})$ auswirkt, wird zunächst die Funktion $f(\mathbf{x} + \Delta \mathbf{x})$ durch eine Taylorreihenentwicklung bis zum linearen (1.) Glied approximiert:

$$\mathbf{y} = f(\mathbf{x} + \Delta \mathbf{x}) \approx f(\mathbf{x}) + J_f(\mathbf{x}) \cdot \Delta \mathbf{x} \quad (4.3.9)$$

Es folgt, aus dem Abschnitt 4.3.3.1, dass die Kovarianz Matrix Σ_Y des Zufallsvektors $Y = f(X)$ mit Hilfe der Jacobi-Matrix approximiert werden kann :

$$\Sigma_Y \approx J_f(\mu_X) \cdot \Sigma_X \cdot J_f(\mu_X)^T \quad (4.3.10)$$

Der Varianzfortpflanzungsetz bleibt aber nur anwendbar, wenn die Nichtlinearität der Abbildung klein bleibt, im Vergleich zu der Breite der Eingangsdaten. Wenn die Nichtlinearität zu hoch ist, müssen andere Approximationsansätze gesucht werden. Die *Unscndted Transform* stellt einen solchen Ansatz dar.

4.3.4 Unscented Transformation (UT)

Ein weiterer Ansatz zur Approximation der Transformation von Verteilungsfunktionen stellt die von Simon Julier und Jeffrey Uhlmann in [17, 18, ?] vorgestellte Methode zur Approximation der aus einer nichtlinearen Transformationsabbildung resultierenden Wahrscheinlichkeitsverteilung dar. Bei der *Unscented Transformation* werden zunächst anhand einer vorgegebenen Kovarianzmatrix $\Sigma_{\bar{\mathbf{x}}}$ für einen Mittelwert-Eingangsvektor $\bar{\mathbf{x}}$ mit n Eingangsvariablen $2n + 1$ so genannte „Sigma-Punkte“ $\mathcal{S} = \{w_i, \mathcal{X}_i\}$ deterministisch generiert. Diese Sigma-Punkte werden so ausgewählt, dass sie die ersten beiden stochastischen Momente (Mittelwert und Varianz) der jeweiligen Zustandsvariable repräsentieren. Ein Satz \mathcal{S} von $2n + 1$ symmetrischen Sigma-Punkten wird konstruiert durch:

$$\mathcal{X}_0 = \bar{\mathbf{x}} \quad (4.3.11)$$

analog

$$\mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(n + \kappa) \cdot \Sigma_{\bar{\mathbf{x}}}} \right)_i \quad (4.3.12)$$

$$\mathcal{X}_{i+n} = \bar{\mathbf{x}} - \left(\sqrt{(n + \kappa) \cdot \Sigma_{\bar{\mathbf{x}}}} \right)_i \quad (4.3.13)$$

Wobei $(\dots)_i$ der i -ten Zeile der Matrix-Wurzel entspricht. Die Matrix-Wurzel hat folgende Eigenschaften :

$$\sqrt{\Sigma_{\bar{\mathbf{x}}}} \cdot \sqrt{\Sigma_{\bar{\mathbf{x}}}^T} = \Sigma_{\bar{\mathbf{x}}} \quad (4.3.14)$$

$\kappa \in \mathbb{R}$ entspricht einem Skalierungsparameter, der einen direkten Einfluss auf die Position der Sigma-Punkte hat. Der Abstand der Sigma-Punkte vom Mittelwert ist :

$$\|\mathcal{X}_i - \bar{\mathbf{x}}\| = \left\| \left(\sqrt{\Sigma_{\bar{\mathbf{x}}}} \right)_i \right\| \cdot \sqrt{n + \kappa} \quad (4.3.15)$$

Ist $\kappa = 0$, so ist die Entfernung des i -ten Sigma-Punkts vom Mittelwert $\bar{\mathbf{x}}$ proportional zu \sqrt{n} . Der Abstand der Sigma-Punkte vom Mittelwert ist also von der Dimension des Eingangsvektors abhängig. Für $\kappa > 0$ werden die Sigma-Punkte in größeren Abstand zum Mittelwert skaliert, ist $\kappa < 0$, so werden sie näher in Richtung des Mittelwerts skaliert. Für größere Dimensionen n ist es also sinnvoll, mit $\kappa < 0$ zu skalieren, damit die Sigma-Punkte noch repräsentativ genug sind. Wird der Spezialfall $\kappa = 3 - n$ gewählt, so wird der durch die Dimension n hervorgerufene Skalierungseffekt weg vom Mittelwert aufgehoben.

Zu jedem Sigma-Punkt gibt es einen Gewichtungsfaktor. Die Gewichtungsfaktoren sind definiert als

$$w_0 = \frac{\kappa}{n + \kappa} \quad \text{für } i = 0 \quad (4.3.16a)$$

$$w_i = \frac{1}{2(n + \kappa)} \quad \text{für } i > 0 \quad (4.3.16b)$$

Für die Gewichtungsfaktoren muss gelten $w_0 + 2n \cdot w_1 = 1$. Die $2n + 1$ Sigma-Punkte werden dann der jeweiligen Transformationsabbildung g unterworfen und es entsteht ein Satz von transformierten

Sigma-Punkten $\mathcal{S}' = \{w_i, \mathcal{Y}_i\}$ durch:

$$\mathcal{Y}_i = g(\mathcal{X}_i) \quad (4.3.17)$$

Um den Mittelwert zu errechnen, wird der gewichtete Durchschnitt der transformierten Sigma-Punkte gebildet:

$$\bar{\mathcal{Y}} \approx w_0 \cdot \mathcal{Y}_0 + \sum_{i=1}^{2n} w_1 \cdot \mathcal{Y}_i \quad (4.3.18)$$

Nach dem selben Prinzip lässt sich auch die Kovarianzmatrix $\Sigma_{\bar{\mathcal{Y}}}$ des transformierten Mittelwerts über das gewichtete Produkt der transformierten Sigma-Punkte errechnen:

$$\Sigma_{\bar{\mathcal{Y}}} \approx w_0 \cdot (\mathcal{Y}_0 - \bar{\mathcal{Y}}) \cdot (\mathcal{Y}_0 - \bar{\mathcal{Y}})^T + \sum_{i=1}^{2n} w_1 \cdot (\mathcal{Y}_i - \bar{\mathcal{Y}}) \cdot (\mathcal{Y}_i - \bar{\mathcal{Y}})^T \quad (4.3.19)$$

Wenn $\kappa < 0$ besteht die Möglichkeit, dass die berechnete Kovarianzmatrix $\Sigma_{\bar{\mathcal{Y}}}$ nicht positiv semi-definit sein mag. In diesem Fall empfehlen Julier und Uhlmann die Kovarianz wie folgt zu berechnen :

$$\Sigma_{\bar{\mathcal{Y}}} \approx \frac{1}{2(n+1)} \cdot \sum_{i=1}^{2n} (\mathcal{Y}_i - \bar{\mathcal{Y}}) \cdot (\mathcal{Y}_i - \bar{\mathcal{Y}})^T \quad (4.3.20)$$

Die Unscented Transformation ist bis zum zweiten Glied der Taylor-Approximation für die nicht-lineare Funktion g korrekt. Abbildung 4.5 veranschaulicht die Vorgehensweise bei der Unscented Transformation.

4.4 Bestimmung der Kollisionswahrscheinlichkeit unter Berücksichtigung von Sensorunsicherheit für komplexe Verkehrssituationen

In diesem Abschnitt wird eine Methode vorgestellt, die es erlaubt, die Kollisionswahrscheinlichkeit einer Verkehrssituation echtzeitfähig zu approximieren.

Das Ziel der Methode ist es, die Zeitreservenverteilungen zu approximieren, die durch Sensorunsicherheiten verursacht worden sind. Deshalb werden die Größen, die die Objekt-Fahrzeuge und ihre Zustände beschreiben, als mit Unsicherheiten behafteten Größen genommen. Diese Größen sind :

- die Positionskordinaten x und y
- die Ausrichtung des Fahrzeuges ψ
- die Längsgeschwindigkeit und die Längsbeschleunigung v_x und a_x
- der Radlenkwinkel δ
- die Länge und Breite des Fahrzeuges l und b

Der Messrausch wird durch eine Multivariate Normalverteilung dargestellt. Um eine realistische Kovarianzmatrix Σ_M für den Messrausch zu bekommen, wurde eine Kovarianzmatrix aus der Bildverarbeitung genommen, um die gewollte Kovarianzmatrix Σ_M zu berechnen. Wie diese Kovarianzmatrix

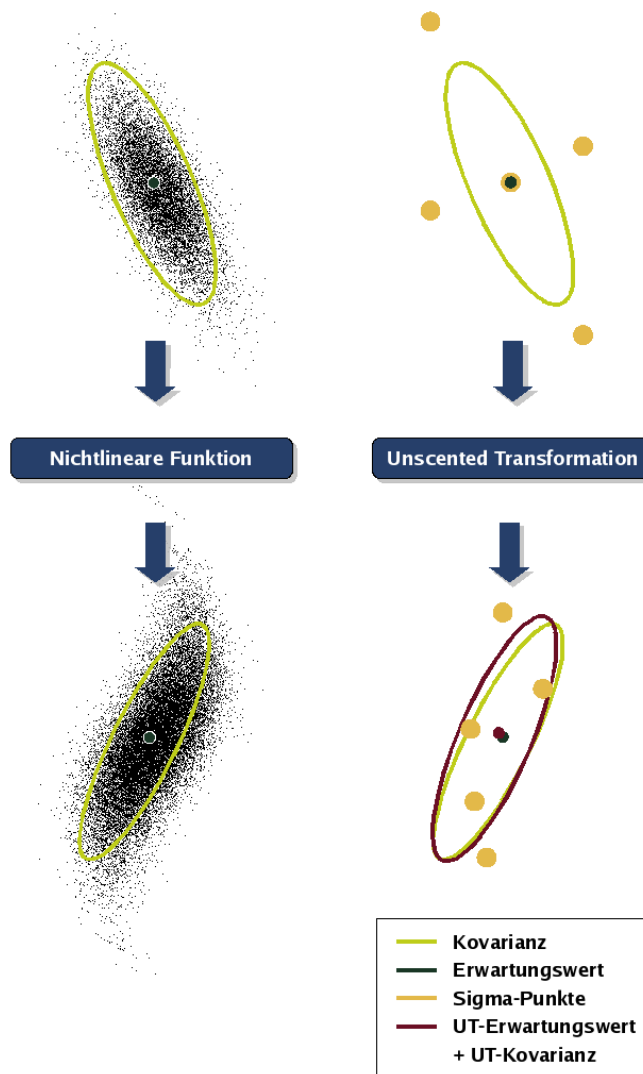


Abbildung 4.5: Unscented Transformation für den zweidimensionalen Fall.

berechnet wird, wird im Anhang C dargestellt. Als Erwartungsvektor $\bar{\mu}_M$ wird der Vektor aus der Simulation genommen. Er stellt eine unsicherheitsfreie Messung dar.

4.4.1 Prinzip der Methode

Die Methode basiert auf der Distanzmaßfunktion, die im Abschnitt 3.1.1 für die effiziente Zeitreservenberechnung vorgestellt worden ist. Im Sinne des Distanzverlaufs entspricht eine Kollision zwischen dem Ego-Fahrzeug und dem Objekt-Fahrzeug dem Nulldurchgang der Distanz :

$$TTC = t \iff \forall \tau < t, \mathcal{D}(\tau) > 0, \mathcal{D}(t) = 0 \text{ und } \dot{\mathcal{D}}(t) < 0 \quad (4.4.1)$$

Es ist zu beachten, daß $\mathcal{D}(t)$ auch negativ werden kann, obwohl dies in der Realität nicht möglich ist, dass reele Fahrzeuge sich durchdringen können. Das Minimum des Distanzverlaufs sagt aus, ob

eine Kollision vorhanden ist. Es gilt :

$$TTC \in \mathbb{R} \iff \min_{\forall t} \mathcal{D}(t) < 0 \tag{4.4.2}$$

Das bedeutet, wenn die Wahrscheinlichkeitsverteilung D_M oder die Wahrscheinlichkeitsdichte d_M des Minimums des Distanzverlaufs \mathcal{D} ermittelt werden kann, kann auch die Kollisionswahrscheinlichkeit ermittelt werden :

$$P(TTC \in \mathbb{R}) = D_M(0) = \int_{-\infty}^0 d_M(v) dv \tag{4.4.3}$$

4.4.2 Untersuchung der Kollisionswahrscheinlichkeit mit Monte-Carlo Sampling

Um die im Abschnitt 4.4.1 beschriebene Methode zu bewerten, müssen zuerst empirischen Referenzwerte, die als Ground Truth dienen werden, ermittelt werden. Die Ermittlung der Ground Truth erfolgt durch rechenaufwendige Monte-Carlo Simulation. Dabei werden Stichproben der Eingangsgrößen entsprechend der Wahrscheinlichkeitsverteilung der Eingangsdaten gesammelt. Diese werden zur Zeitreservenberechnung herangezogen.

Theorem 6 (Kanonische Darstellung von multivariaten Normalverteilungen,[19]). *Gebegen sei eine n-dimensionale Normalverteilung $X \sim \mathcal{N}(\mu, \Sigma)$. Es existiert eine symmetrische positive Matrix A , so dass $A \cdot A^T = \Sigma$. Wenn $Y \sim \mathcal{N}(\mathbf{0}, I_n)$ eine n-dimensionale Standardnormalverteilung ist, dann :*

$$X = A \cdot Y + \mu$$

Wenn Σ_M eine Reguläre Matrix ist, dann ist A die Wurzel-Matrix von Σ_M und kann berechnet werden.

Stichproben zu erzeugen, die eine multivariate Standardnormalverteilung mit unabhängigen Komponenten entsprechen ist einfach. Es wird nur ein Zufallsgenerator benötigt, der Stichproben aus eine 1-dimensionale Standardnormalverteilung erzeugt. Die Methode `randn(0,1)` von Matlab wird für diesen Zweck verwendet. Die Vektoren $(\text{randn}(0,1), \dots, \text{randn}(0,1))^T$ entsprechen dann einer multivariaten Standardnormalverteilung mit unabhängigen Komponenten. Mit Hilfe des Theorems 6 können dann auch Proben, die eine multivariate Normalverteilung mit Kovarianzmatrix Σ_M und Erwartungsvektor $\bar{\mu}_M$ entsprechen, gezogen werden.

Jede einzelne Probe wird als Eingabe für die Zeitreservenberechnung verwendet und das entsprechende Distanzminimum wird berechnet. Der Häufigkeitshistogramm der Distanzminima wird dann verwendet, um die wahre Verteilung des Distanzminimums zu berechnen. Die Abbildung 4.6 zeigt drei verschiedenen Histogramme, die die Häufigkeit der Distanzminima zeigt.

4.4.2.1 Diskussion der Ergebnisse

Die Verteilung der Stichproben könnte sich gut mit der Dichtefunktion einer Normalverteilung annähern lassen (siehe Abbildung 4.6a). Die Distanzminima können aber per se nicht kleiner als -1 sein (siehe Abbildungen 4.6b und 4.6b), deshalb soll eine andere Wahrscheinlichkeitsverteilung entwickelt werden, um die Verteilung der Stichproben besser annähern zu können. Diese Verteilung ist eine an -1 geschnittene Normalverteilung. Es wird $X \sim \mathcal{N}_h(\alpha, \mu_h, \sigma_h)$ mit der folgende Dichtefunktion definiert

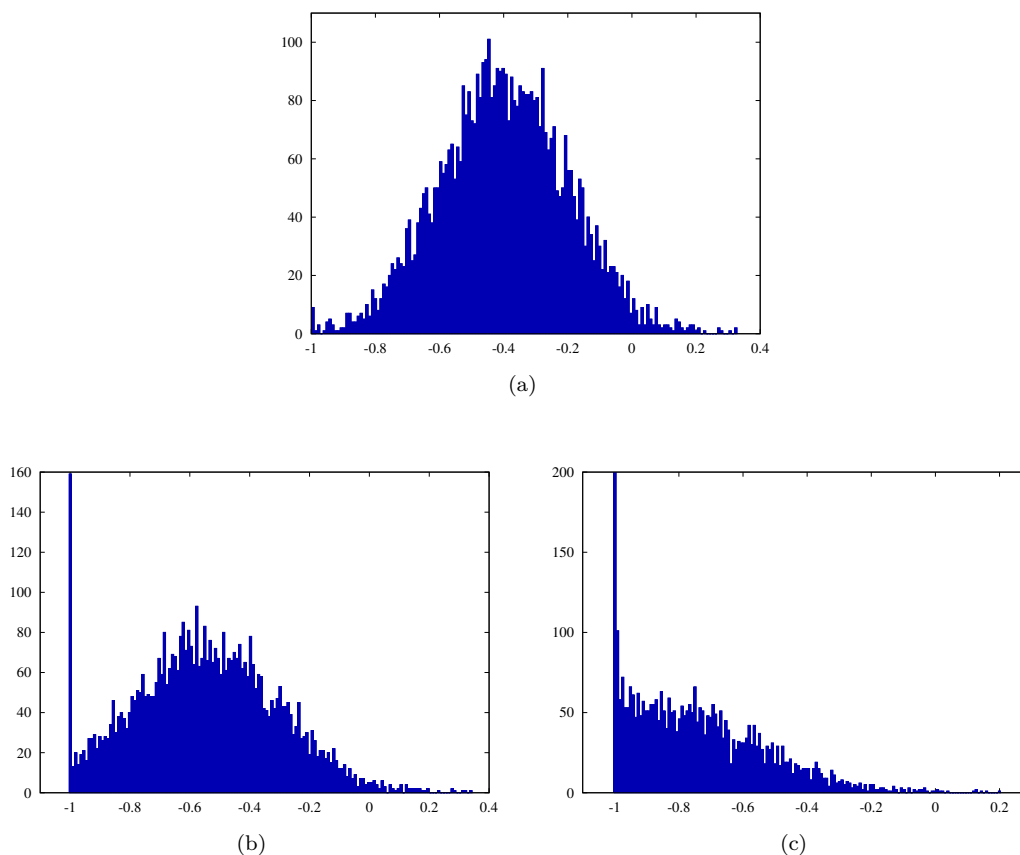


Abbildung 4.6

:

$$p(x) = c \cdot \delta(x - \alpha) + \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} \cdot u(x - \alpha)$$

$$\text{mit } c = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(t-\mu)^2}{2 \cdot \sigma^2}} dt$$

$$\text{und } u(x) = \begin{cases} 1 & \forall x \geq 0 \\ 0 & \forall x < 0 \end{cases}$$
(4.4.4)

Die Dichte- und die Verteilungsfunktion der Zufallsvariable $X \sim \mathcal{N}_h(\alpha, \mu, \sigma)$ werden in den Abbildungen 4.7a und 4.7b beispielhaft dargestellt. Die Werte μ_h und σ_h sind Parameter, die die Verteilung beschreiben. Sie entsprechen weder dem Erwartungswert noch der Standardabweichung der Verteilung.

4.4.3 Approximationsmethode

Da die Stichproben für die Berechnung der empirischen Kollisionswahrscheinlichkeit bekannt sind, können die entsprechenden Erwartungswert μ_{GT} und Standardabweichung σ_{GT} bestimmt werden. Um die Parameter μ_h und σ_h berechnen zu können, wird die Tatsache benutzt, dass beide Verteilungen,

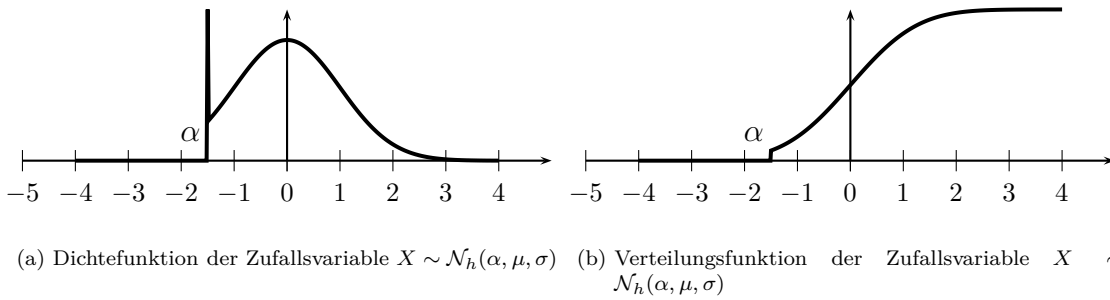


Abbildung 4.7

dasselbe Erwartungswert und dieselbe Varianz haben müssen :

$$\mu_{GT} = E[X] \tag{4.4.5a}$$

$$\sigma_{GT}^2 = E[X^2] - E[X]^2 \tag{4.4.5b}$$

wo $E[X]$ und $E[X^2]$ die Momente der ersten und zweiten Ordnung der Zufallsvariable X sind :

$$E[X] = \int_{-\infty}^{+\infty} x \cdot p(x) dx = \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(\alpha - \mu)^2}{2\sigma^2}} - \frac{\alpha - \mu}{2} \operatorname{erfc}\left(\frac{\alpha - \mu}{\sqrt{2}\sigma}\right) + \alpha \tag{4.4.6a}$$

$$E[X^2] = \int_{-\infty}^{+\infty} x^2 \cdot p(x) dx = \frac{\sigma \cdot (\alpha + \mu)}{\sqrt{2\pi}} e^{-\frac{(\alpha - \mu)^2}{2\sigma^2}} + \frac{\sigma^2 + \mu^2 - \alpha^2}{2} \operatorname{erfc}\left(\frac{\alpha - \mu}{\sqrt{2}\sigma}\right) + \alpha^2 \tag{4.4.6b}$$

mit $\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \cdot \int_x^{+\infty} e^{-t^2} dt$

Die Herleitung der Gleichungen (4.4.6a) und (4.4.6b) wird im Anhang D ausführlich beschrieben.

Mit $\alpha = -1$ werden (4.4.6a) und (4.4.6b) :

$$E[X] = \int_{-\infty}^{+\infty} x \cdot p(x) dx = \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(-1 - \mu)^2}{2\sigma^2}} - \frac{-1 - \mu}{2} \operatorname{erfc}\left(\frac{-1 - \mu}{\sqrt{2}\sigma}\right) - 1 \tag{4.4.7a}$$

$$E[X^2] = \int_{-\infty}^{+\infty} x^2 \cdot p(x) dx = \frac{\sigma \cdot (-1 + \mu)}{\sqrt{2\pi}} e^{-\frac{(-1 - \mu)^2}{2\sigma^2}} + \frac{\sigma^2 + \mu^2 - 1^2}{2} \operatorname{erfc}\left(\frac{-1 - \mu}{\sqrt{2}\sigma}\right) + 1 \tag{4.4.7b}$$

Die zwei Parameter μ_h und σ_h der gesuchten Verteilung $\mathcal{N}_h(-1, \mu, \sigma)$ werden durch die numerische Lösung der Gleichungen des Systems (4.4.5) berechnet. Dafür wird die Matlab Methode `fsolve` benutzt.

Die Abbildungen 4.8a bis 4.8c Zeigen die zwei Verteilungen die, aus den Stichproben berechnet worden sind. In Cyan (■) wird die aus den Stichproben berechnete Normalverteilung dargestellt. In den Abbildungen 4.8a und 4.8b entspricht diese Normalverteilung dem Häufigkeitshistogramm genau. Aber im Fall der Abbildung 4.8c entspricht die Normalverteilung den Häufigkeitshistogramm nicht. In Grün (■) wird die entsprechende linksseitige gesättigte Normalverteilung dargestellt : in den Abbildungen 4.8a und 4.8b ist der Unterschied vernachlässigbar, da die Sättigung sehr gering oder nicht existent ist. In der Abbildung 4.8c, ist es aber unbestreitbar, dass die linksseitige gesättigte

Normalverteilung den Häufigkeitshistogramm besser annähert.

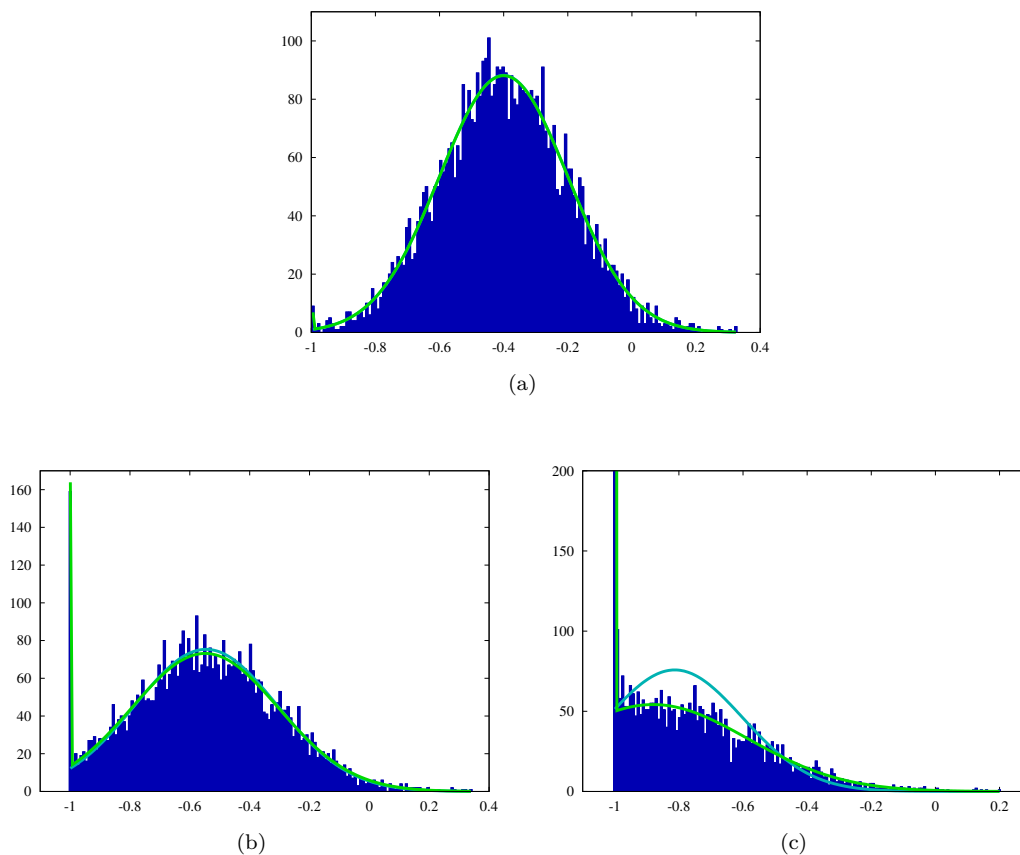


Abbildung 4.8

Der Erwartungswert μ_{GT} und die Standardabweichung σ_{GT} der Stichproben erlauben es, eine Verteilung zu definieren, die das Häufigkeitshistogramm der Stichproben sehr gut approximiert. Die Monte-Carlo Simulation ist aber dafür notwendig, was nicht auf dem Fahrzeug realisierbar ist. Deswegen sollen, wie es im nächsten Abschnitt erklärt wird, das Erwartungswert μ_{GT} und die Standardabweichung σ_{GT} auch zuerst approximiert werden.

4.4.4 Echtzeitfähige Approximationsmethode

Wie im letzten Abschnitt erklärt, braucht eine echtzeitfähige Methode zur Berechnung der Kollisionswahrscheinlichkeit eine Approximation des Erwartungswertes μ_{GT} und der Standardabweichung σ_{GT} . Nur so können auch die Parameter μ und σ in echtzeit bestimmt werden. Die Approximation wird mittels der *Unscented Transform* (siehe Abschnitt 4.3.4) durchgeführt. Die resultierenden Erwartungswert und Standardabweichung werden jeweils mit μ_{UT} und σ_{UT} gekennzeichnet. Die Abbildung 4.9, zeigt das Ergebnis dieser Berechnung : in Magenta (■) werden die Verteilung $\mathcal{N}(\mu_{UT}, \sigma_{UT})$ und die benutzten σ -Punkten (⊕) dargestellt. In Rot (■) wird die entsprechende linksseitige gesättigte Normalverteilung gezeichnet.

In der Abbildung 4.10 werden alle Verteilungen dargestellt, und daraus kann geschlossen werden dass :

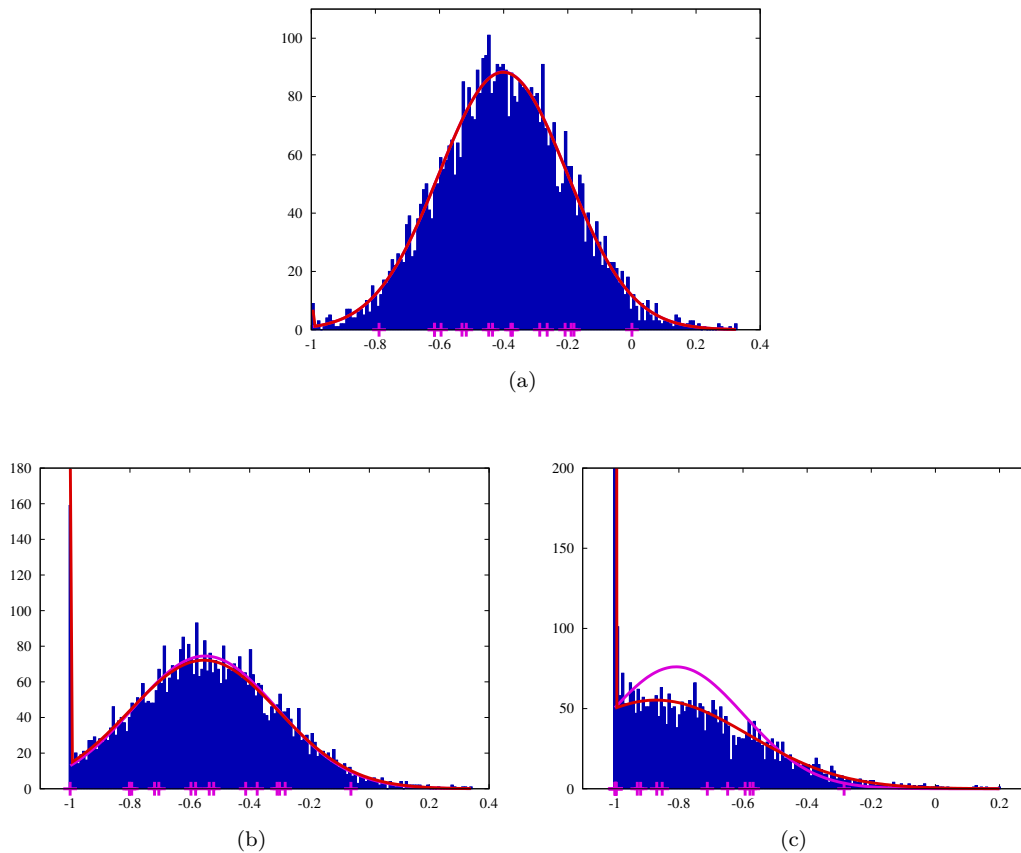


Abbildung 4.9

- die Approximation des Erwartungswerts μ_{GT} und der Standardabweichung σ_{GT} durch die Unscented Transform gut ist
- die darauffolgende berechnete linksseitige besätigte Normalverteilung die Stichprobenverteilung der Ground Truth richtig annähert

Wenn die Parameter μ und σ der Verteilung $\mathcal{N}_h(-1, \mu, \sigma)$ bekannt sind kann die Kollisionswahrscheinlichkeit ermittelt werden :

$$\begin{aligned}
 P(TTC \in \mathbb{R}) &= \int_{-\infty}^0 p_h(x) dx = c + \int_{-1}^0 \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx \\
 &= \int_{-\infty}^{-1} \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx + \int_{-1}^0 \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx \\
 &= \int_{-\infty}^0 \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx
 \end{aligned} \tag{4.4.8}$$

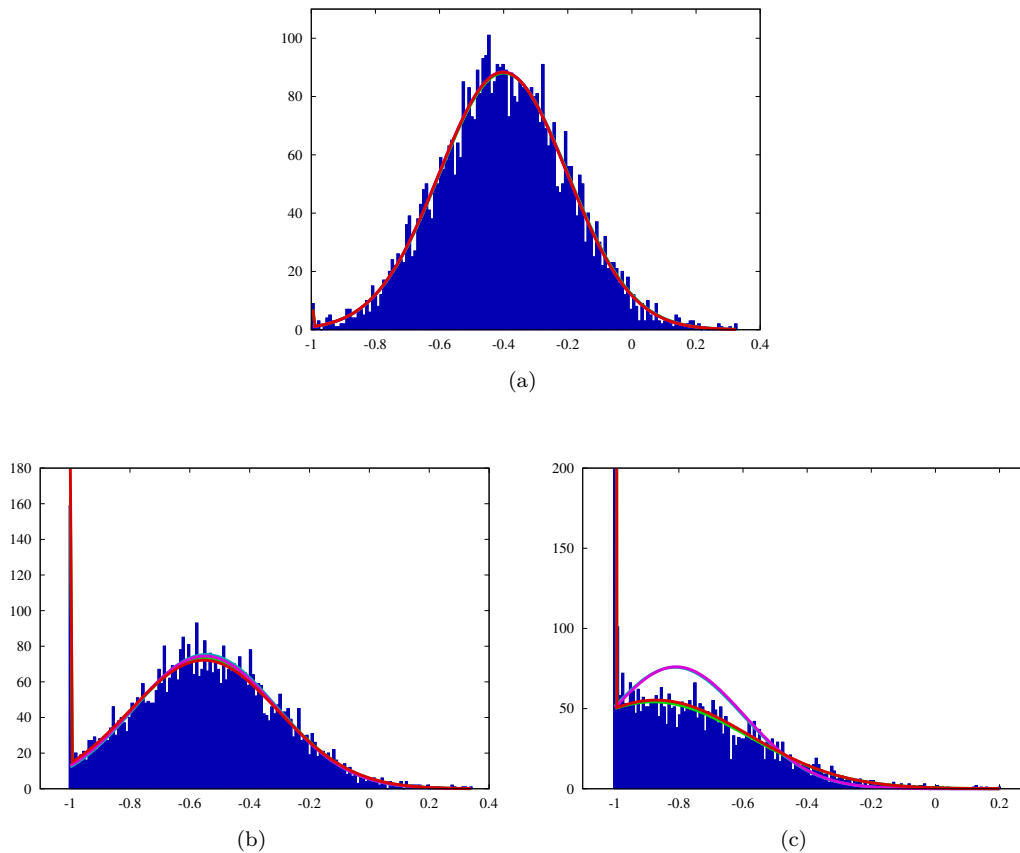


Abbildung 4.10

4.5 Ergebnisse

4.5.1 Szenarien für die Auswertung

Das zur Auswertung verwendete Szenario entspricht, wie im Letzten Kapitel (siehe Abschnitt 3.4.4) dem “Links Abbiegen” Szenario. Da die Berechnung der empirischen Kollisionswahrscheinlichkeiten sehr zeitaufwendig ist, werden im Vergleich zum letzten Kapitel weniger Stichproben verwendet und es werden vor allem nur Ausführungen des Szenarios ausgewertet, in welchen die Kollisionswahrscheinlichkeit nicht eindeutig ist. Das heißt Szenarien in welchen, die Kollisionswahrscheinlichkeit mit der Zeit variiert und nicht eindeutige 1 oder 0 Werte einnimmt.

Für jede Szenarioausführung wird der Zeitpunkt der nächsten Annäherung t_n berechnet. Dieser Zeitpunkt ist

- entweder der Zeitpunkt der Kollision
- oder der Zeitpunkt, wenn es keine Kollision gab, an dem die Fahrzeugen sich am nächsten befunden haben

Es werden dann für die Bewertung der Approximationsmethode Stichproben zwischen 4 und 1 Sekunden vor t_n gezogen (siehe Abbildung 4.11).

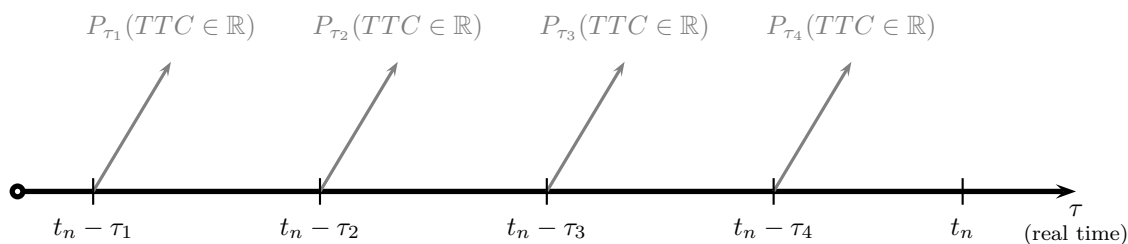


Abbildung 4.11: Für die Bewertung der Approximationsmethode werden Stichproben zwischen 4 und 1 Sekunden vor t_n gezogen

4.5.2 Bewertung des Ansatzes

Die geschätzte Kollisionswahrscheinlichkeit und die entsprechenden empirische Kollisionswahrscheinlichkeit werde für jeden Stichprobe berechnet, und miteinander verglichen. Die Abbildung 4.12 zeigt das Verhältnis der approximativen Wahrscheinlichkeit zur empirischen Wahrscheinlichkeit. Wie man sieht ist die Approximationsmethode in allen Fällen sehr genau. In jedem Wahrscheinlichkeitsbereich bleibt das Approximationsfehler niedrig. Die verschiedenen Farben in der Abbildung 4.12 entsprechen dem zeitlichen Abstand der Stichprobe zu t_n : blau für 4 Sekunden bis rot für 1 Sekunde vor t_n .

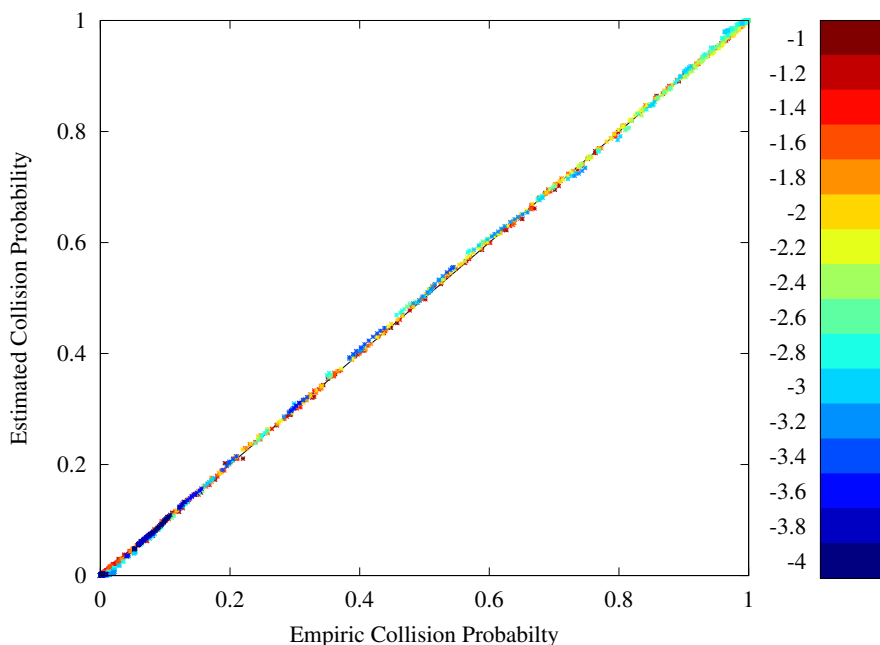


Abbildung 4.12: Verhältnis der approximativen Wahrscheinlichkeit zur empirischen Wahrscheinlichkeit

In der Abbildung 4.13 werden die Häufigkeiten des Approximationsfehlers dargestellt. Das Durchschnittsfehler beträgt 0.0025, und die Standardabweichung 0.0030.

In der Abbildung 4.14 werden die Häufigkeiten des Approximationsfehlers pro Stichprobenklasse dargestellt. Die Abbildung zeigt, dass alle Stichprobenklassen vergleichbare Fehlerklassenverteilungen haben.

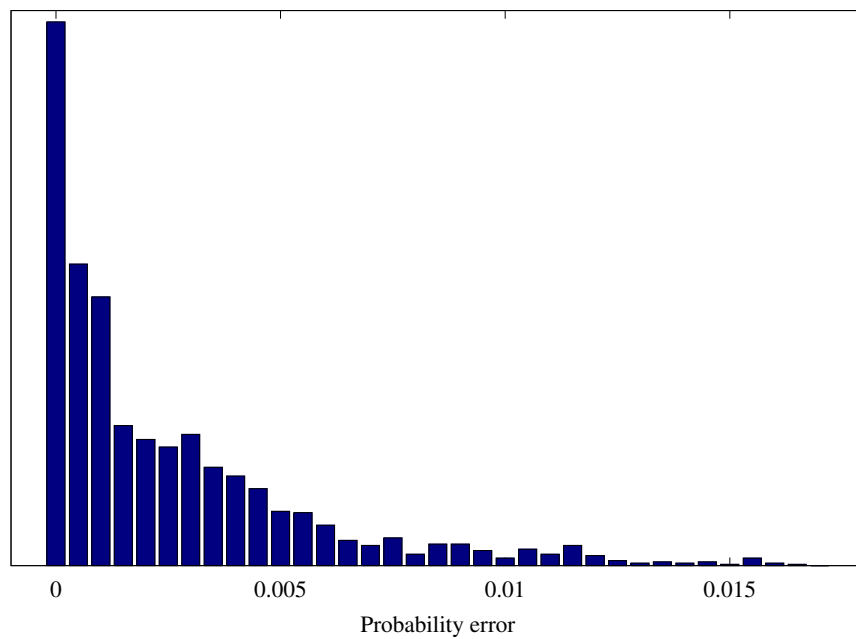


Abbildung 4.13: Häufigkeiten des Approximationsfehlers

Die Auswertung der gewählten Szenarien hat in Matlab ununterbrochen 5 Wochen gedauert.

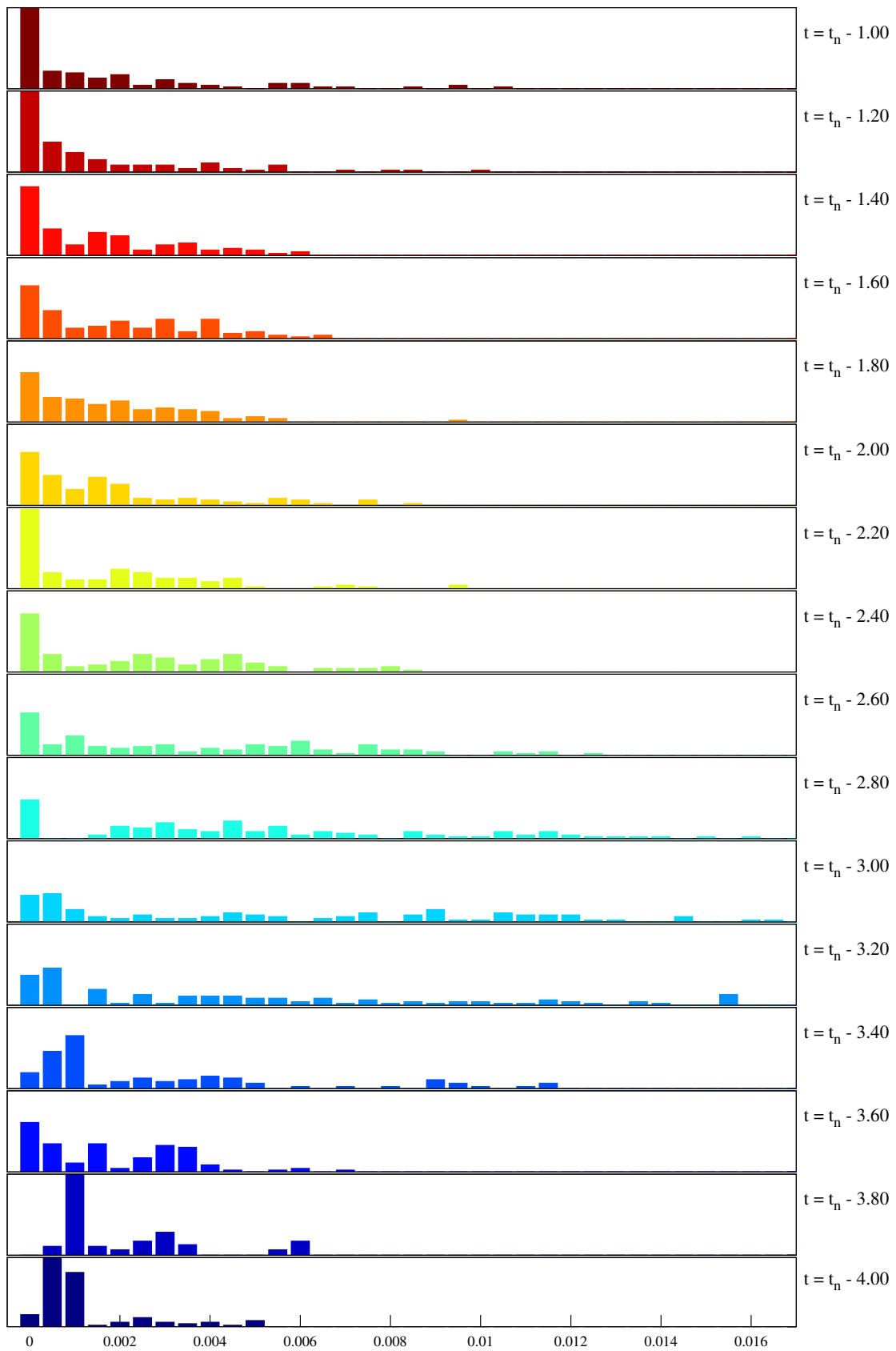


Abbildung 4.14: Häufigkeiten des Approximationsfehlers pro Stichprobenklasse

4.5.3 Betrachtung einzelner Fällen

In diesem Abschnitt werden ein paar detaillierten Fälle der Kollisionswahrscheinlichkeit gezeigt und diskutiert. In den Abbildungen 4.15 bis 4.18 werden verschiedene Häufigkeitshistogramme mit den entsprechenden Approximationen dargestellt.

Die Abbildungen 4.15 und 4.16 entsprechen zwei Situationen in einer 90 Grad Kreuzung : beide Fahrzeug fahren Gerade aus in senkrechten Richtungen. Das Szenario der 90 Grad Kreuzung weist die Eigenschaft auf, dass die getroffene Prognose der tatsächlichen Weiterentwicklung des Szenarios jederzeit entspricht. Die Abbildungen 4.17 und 4.18 entsprechen zwei Ausführungen des “Links Abbiegen” Szenarios.

Die Ausführungen der Abbildungen 4.18 und 4.15 enden mit einer Kollision während die Ausführungen der Abbildungen 4.17 und 4.16 einer knappen Vorbeifahrt entsprechen.

Jeder Histogramm einer Abbildung ist eine Stichprobe der Szenarioverlaufs. Die verschiedenen Farben entsprechen dem zeitlichen Abstand der Stichprobe zum Zeitpunkt der nächsten Annäherung t_n : blau für 4 Sekunden bis rot für 1 Sekunde vor t_n .

Da in den Abbildungen 4.15 und 4.16 die getroffene Prognose der tatsächlichen Weiterentwicklung des Szenarios entspricht, ändert sich das Mittelwert der Häufigkeitshistogramme kaum. Je mehr der Zeitpunkt der nächsten Annäherung t_n zeitlich näher wird, desto kleiner wird die Standardabweichung.

In den Abbildungen 4.17 und 4.18 ist die getroffene Prognose weit von der tatsächlichen Weiterentwicklung des Szenarios entfernt. Als Folge davon, wandert in beiden Fällen der Mittelwert der Häufigkeitshistogramme. Im Fall der Abbildung 4.17 wird die Kollisionsgefahr erst zwei Sekunden vor der tatsächlichen Kollision wahrnehmbar. In der Abbildung 4.18 passiert das Gegenteil : drei Sekunden vor dem Zeitpunkt der nächsten Annäherung wird die Kollisionswahrscheinlichkeit um 90% geschätzt. Eine Sekunde später ist diese Wahrscheinlichkeit nur noch von 0.4 %.

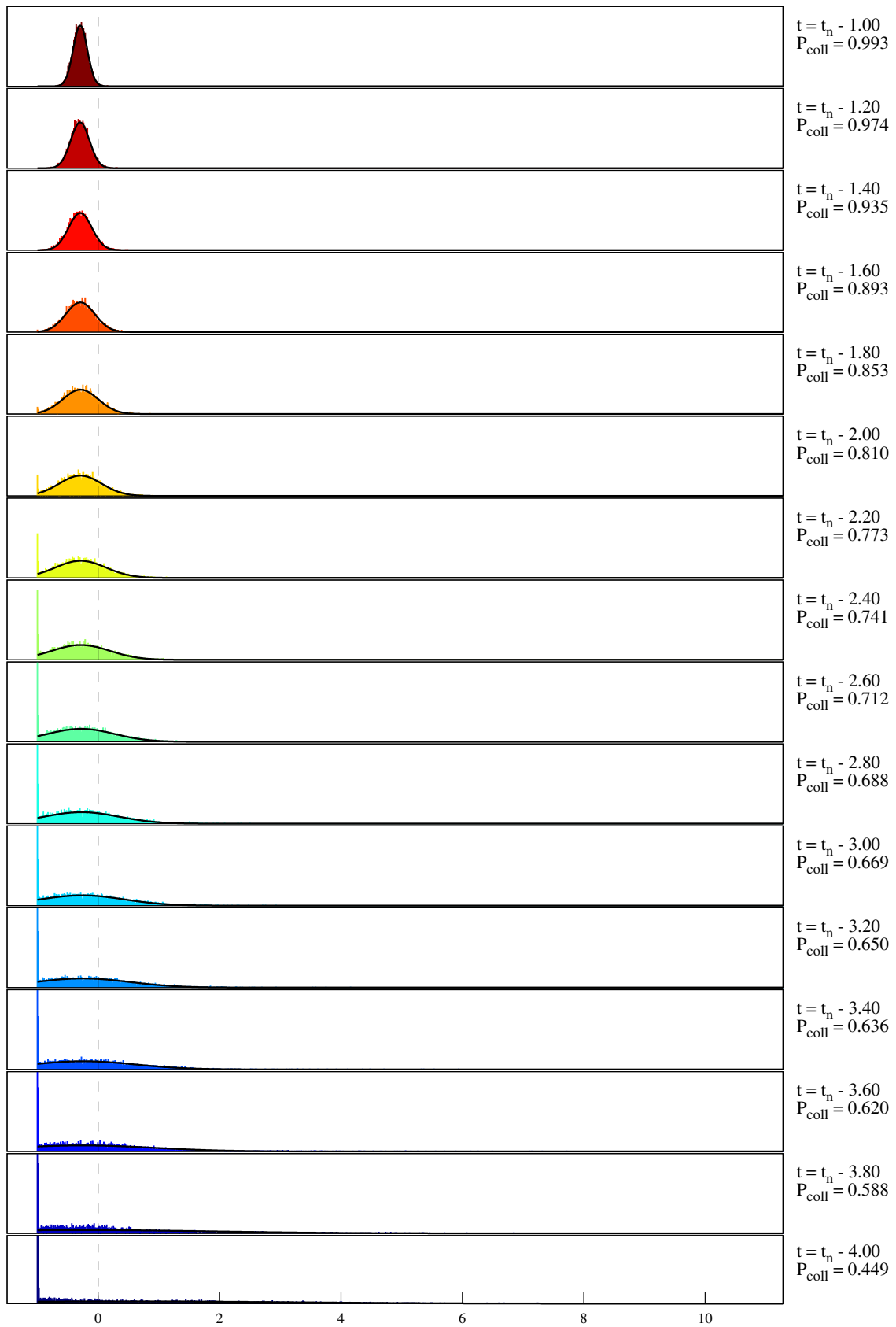


Abbildung 4.15

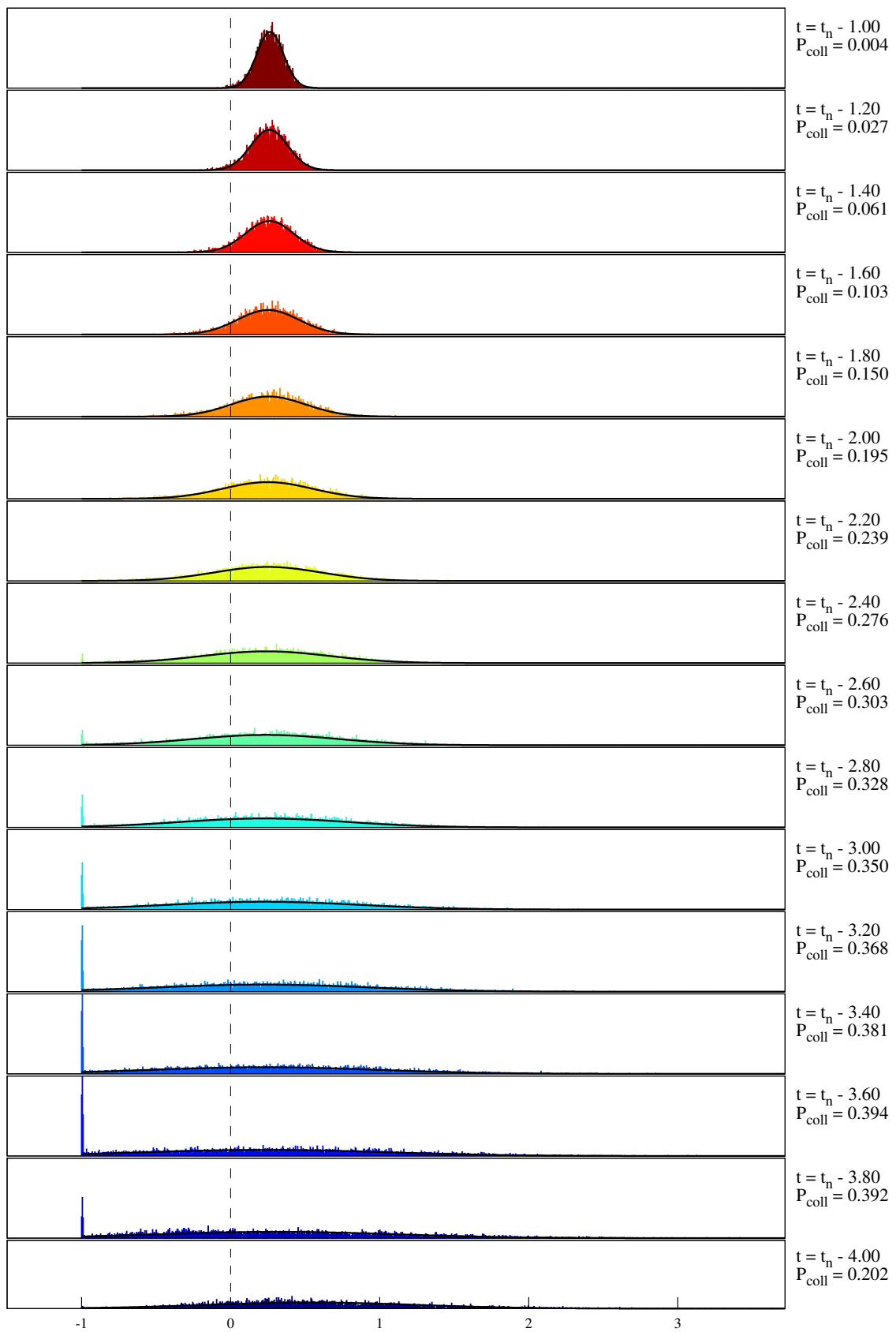


Abbildung 4.16

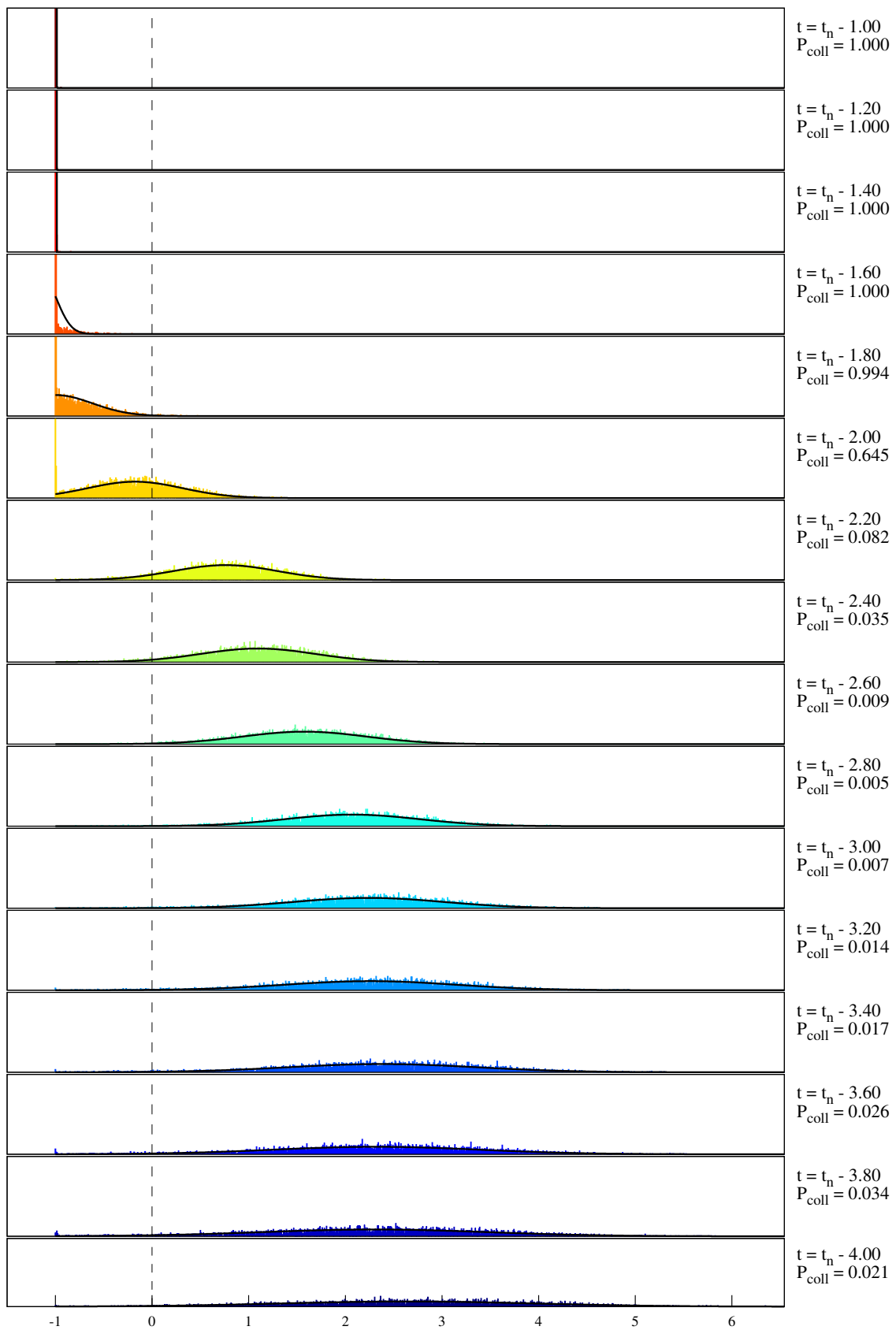


Abbildung 4.17

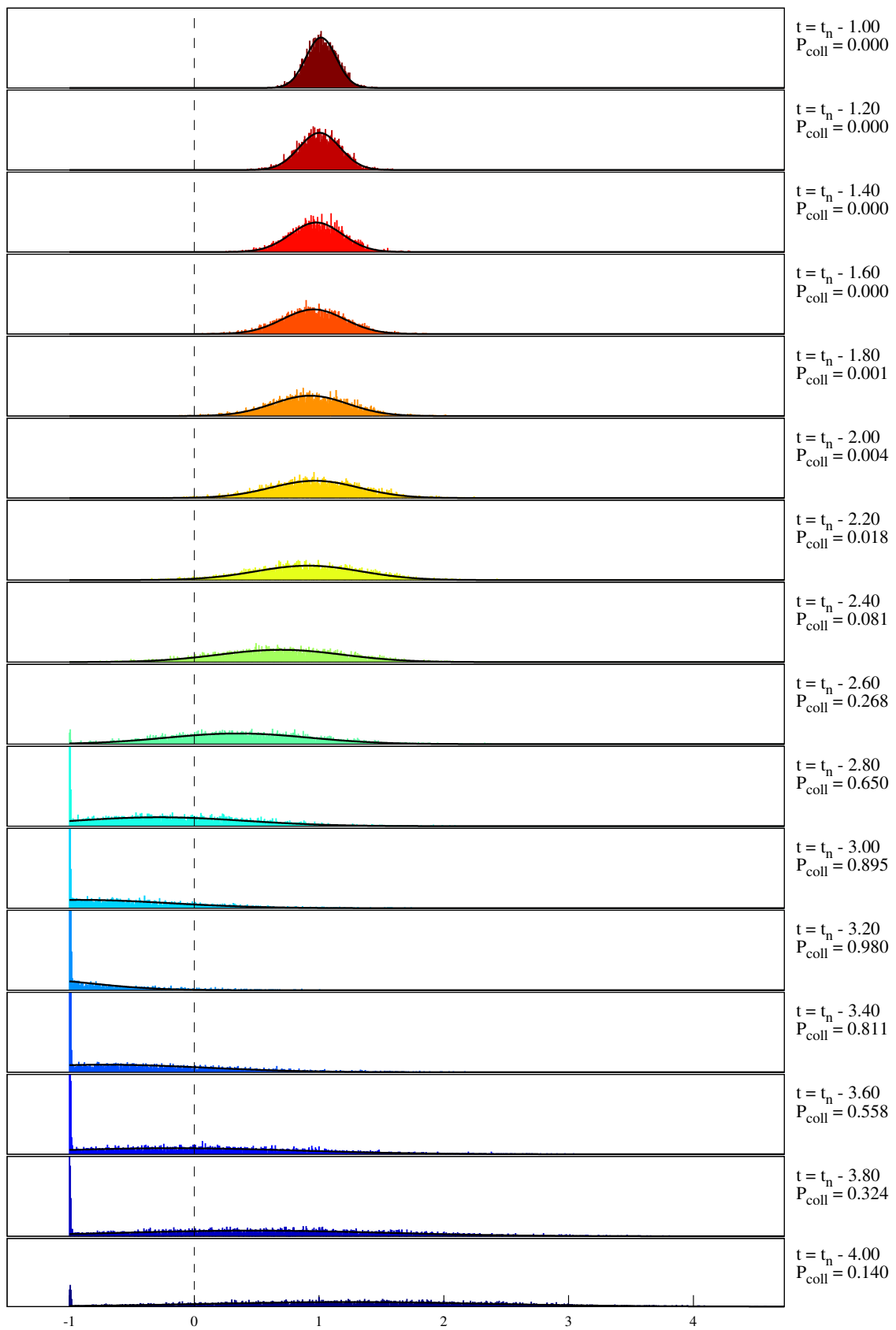


Abbildung 4.18

4.6 Fazit

In diesem Abschnitt wurde eine Methode entwickelt, um die Kollisionswahrscheinlichkeit unter Berücksichtigung der Sensorunsicherheiten zu bestimmen. Die Methode kann in beliebigen Verkehrssituationen angewendet werden, und stellt eine echtzeitfähige Lösung zur Approximation der Kollisionswahrscheinlichkeit dar.

Mit ihrer hohen Genauigkeit (Durschnittsfehler der approximierten Wahrscheinlichkeit : 0.25 %) kann diese Methode ein Grundelement der zukünftigen Fahrerassistenzsystemen darstellen. Die Anwendung der *Unscented Transform* führt dazu, dass diese Methode recheneffizient ist.

Die Untersuchung der Ergebnisse hat aber auch gezeigt, dass die Genauigkeit der Situationprognose eine grosse Rolle bei der Kollisionswahrscheinlichkeit spielt.

Kapitel 5

Fazit und Ausblick

5.1 Fazit

In dieser Arbeit wurden neue Ansätze zur Kritikalitätsbewertung von komplexen Verkehrssituationen entwickelt.

Der im Kapitel 3 eingeführte Ansatz stellt eine effiziente Methode zur Zeitreservenberechnung für beliebigen Verkehrssituation dar. Er basiert auf Interpolation der Distanzfunktion zwischen Objekten der Szene mit Polynomen. Dabei kann die Balance zwischen Geschwindigkeit und Genauigkeit anhand von Parametern, wie der Grad oder die Anzahl der interpolierenden Polynome angepasst werden. Der Ansatz wurde im Rahmen dieser Arbeit vollständig in C implementiert. So konnte er in der Simulation und im Versuchsträger evaluiert werden. Der Ansatz wurde anschließend mit der Vorwärtssimulation und einer Referenz verglichen. Es wurde festgestellt, dass der neue Ansatz schneller und genauer als die Vorwärtssimulation ist.

Im Kapitel 4 wurde der neue Ansatz so erweitert, dass er die Kollisionswahrscheinlichkeit unter Berücksichtigung von Messunsicherheiten echtzeitfähig approximieren kann.

Dazu sind die Referenzwahrscheinlichkeitsverteilungen mit Hilfe von zahlreichen Zufallsproben bestimmt worden. Diese Stichproben sind so erzeugt worden, dass sie die selben stochastischen Eigenschaften wie die Messunsicherheiten aufweisen. Anschließend konnte die erzeugte Referenz erfolgreich mit Hilfe eines echtzeitfähigen Verfahrens mit Matlab approximiert werden.

5.2 Ausblick

Die im Kapitel 3 entwickelte Methode verfügt bereits über die gewünschten positiven Eigenschaften: Recheneffizienz ohne Genauigkeitsverlust. Der Aufbau der Methode, und vor allem die Anwendung eines Fahrzeugmodells, deren Zustandgleichung integriert worden sind (siehe Abschnitt 3.2), erlaubt es allerdings weitere Optimierungen durchzuführen.

Die Zeitreservenberechnung mit Distanzpolynominterpolation kann hochgradig parallelisiert werden. Die Parallelisierung der Berechnungen könnte zum Beispiel auf einem FPGA ¹ durchgeführt werden. Es existieren für Embedded Systems spezialisierte Prozessoren, die über FPGAs als Koprozessoren verfügen. Die Abbildung 5.1 stellt eine mögliche Parallelisierung dar, in der zahlreiche Berechnungen parallel durchgeführt werden können. Die Distanzfunktion weist auch ein hohes Par-

¹Field Programmable Gate Arrays

allelisierbarkeitspotential auf: jede Seite des Objektpolygones kann unabhängig von den anderen Seiten verarbeitet werden.

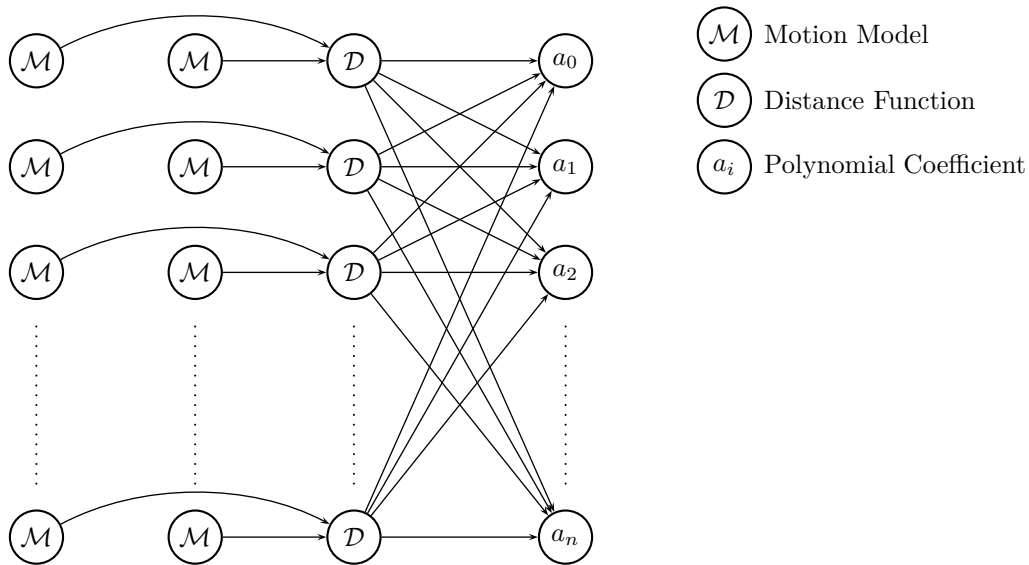


Abbildung 5.1: Mögliche Berechnungsparallelisierung auf einem FPGA

Darüberhinaus könnten die Binärsuchen (siehe Abschnitte 3.1.3 und 3.3.4) mit Hilfe von Multithreading parallelisiert werden.

Weitere interessante Fragestellung, die diese Arbeit aufwirft, betrifft insbesondere die Verteilung des Messrauschens. In der Realität sind auch andere Verteilungen als normalverteilt möglich. Das vorgestellte Verfahren sollte darauf untersucht werden, wie es mit anderen Verteilungen zurechtkommt.

Das entwickelte Verfahren könnte in Zukunft darauf untersucht werden, wie es die Kollisionswahrscheinlichkeit auch unter Berücksichtigung der Prognoseunsicherheit bestimmen kann. Dazu muß aber zunächst untersucht werden, wie Prognoseunsicherheit modelliert werden kann.

Literaturverzeichnis

- [1] J. Hillenbrand. *Fahrerassistenz zur Kollisionsvermeidung*. PhD thesis, Universität Karlsruhe, 2007.
- [2] Jörg Hillenbrand, Kristian Kroschel, and Volker Schmid. Situation assessment algorithm for a collision prevention assistant. Technical report, Daimler AG.
- [3] Andreas Tamke, Thao Dand, and Gabi Breuel. Integrierte simulations- und entwicklungs- umgebung für die ausweichassistenz zum fußgängerschutz. In *11. Braunschweiger Symposium AAET 2010*, 2010.
- [4] L. Richard Turner. Inverser of the vandermonde matrix with applications. Technical report, NASA, Lewis Research Center, 1966.
- [5] Christian Guilpin. *Manuel de calcul numérique appliqué*. EDP Sciences, 1999.
- [6] John C. Mason and David Handscomb. *Chebyshev Polynomials*. Chapman & Hall / CRC, 2003.
- [7] Bronstein, Semendjajew, Musiol, and Mühlig. *Taschenbuch der Mathematik*. Harri Deutsch, 1995.
- [8] Manfred Mitschke. *Dynamik der Kraftfahrzeuge, Band C Fahrverhalten*. Springer, 1990.
- [9] Robin Schubert, Eric Richter, and Gerd Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *The 11th International Conference on Information Fusion*, pages 1–6, 2008.
- [10] Romain Pepy, Alain Lambert, and Hugues Mounier. Reducing navigation errors by planning with realistic vehicle model. In *Intelligent Vehicles Symposium, 2006*, pages 300–307, 2006.
- [11] Maxima, a computer algebra system. <http://maxima.sourceforge.net/>.
- [12] Roman Kreisel. Modellierung und visualisierung von verkehrsszenarien innerhalb eines entwicklungs- werkzeugs für fahrerassistenzsysteme. Master's thesis, Hochschule Esslingen, 2009.
- [13] Enkelejda Tafaj. Situationsbewertung für fahrerassistenzsysteme - entwicklung und evaluation einer simulationsumgebung für kreuzungsszenarien. Master's thesis, Universität Stuttgart, 2001.
- [14] Daniel Trumpp. Untersuchung der auswirkung von messunsicherheiten einzelner sensordaten im kraftfahrzeug auf die nichtlinearen algorithmen zur zeitreservenberechnung durch simulation und approximation anhand eines reduzierten modells für den einsatz in fahrerassistenzsystemen. Master's thesis, Hochschule Esslingen, 2008.

- [15] Lothar Papula. *Mathematik für Ingenieure und Naturwissenschaftler, Band 3 - Vektoranalysis, Mathematische Statistik, Wahrscheinlichkeitsrechnung, Fehler und Ausgleichsrechnung*. Vieweg Verlag, 2001.
- [16] Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1994.
- [17] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. Technical report, The University of Oxford - The Robotics Research Group, 1997.
- [18] Simon J. Julier and Jeffrey K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, University of Oxford, United Kingdom, 1996.
- [19] Laurent Decreasefond. *Cours de probabilités*. TelecomParisTech, 2006.
- [20] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2006.

Anhang A

Beweise

In diesem Anhang wird der Beweis für den Satz A aufgezeigt. Zur Erinnerung, lautet den Satz :

Satz. *Sei $[AB]$ ein Segment. Mit der durch die unendliche Norm induzierten Distanz, ist der Abstand des Segments zum Ursprung das Minimum dieser Abstände :*

- *der Abstand eines der Endpunkte zum Ursprung*
- *wenn es sie gibt, der Abstand des Kreuzpunkt zwischen den Segment und die erste oder zweite Diagonale*

Zuerst wird der generellere Fall mit einer Gerade bewiesen (siehe Lemma 1).

Lemma 1. *Sei Δ eine Gerade. Mit der durch die unendliche Norm induzierte Distanz, ist der Abstand der Gerade zum Ursprung das Minimum der Abstände der Kreuzpunkte zwischen den Segment und die erste und zweite Diagonale.*

$$d(\Delta, O) = \min\{d(A, O) \mid A \in \Delta, |x_A| = |y_A|\} \quad (\text{A.0.1})$$

Beweis des Lemmas 1. Man betrachtet die parametrische Darstellung von Δ :

$$\Delta = \left\{ \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \alpha \cdot t + \gamma \\ \beta \cdot t + \delta \end{pmatrix}, \forall t \in \mathbb{R} \right\} \quad (\text{A.0.2})$$

Die Funktion $f_\Delta(t) = \max(|\alpha \cdot t + \gamma|, |\beta \cdot t + \delta|)$ auf \mathbb{R} ist eine stückweise lineare definierte und stetige Funktion. Die Funktion f_Δ kann auch so definiert werden :

$$f_\Delta(t) = \begin{cases} a_1 \cdot t + b_1 & \forall t \in]-\infty; t_1] \\ a_2 \cdot t + b_2 & \forall t \in [t_1; t_2] \\ \dots & \\ a_n \cdot t + b_n & \forall t \in [t_{n-1}; +\infty[\end{cases} \quad (\text{A.0.3})$$

mit $a_i \cdot t_i + b_i = a_{i+1} \cdot t_i + b_{i+1} \Leftrightarrow |x(t_i)| = |y(t_i)|, \quad \forall i \in [1 : n - 1]$ (f ist stetig)

Da die Funktion f_Δ stückweise linear definiert, und immer positiv ist, ist sein Minimum auf den verschiedenen Intervalle immer an einer (vielleicht nicht nur) den endlichen Intervallgrenze : $t_i, i \in$

[1 : n-1]. Das heißt auch dass, das Minimum der Funktion auch an einer diesen Intervallgrenze erreicht wird, da wo $|x(t)| = |y(t)|$, die Kreuzung der Gerade Δ und der ersten oder zweiten Diagonale.

Die Abbildung A.1 stellt den Gedankengang dar.

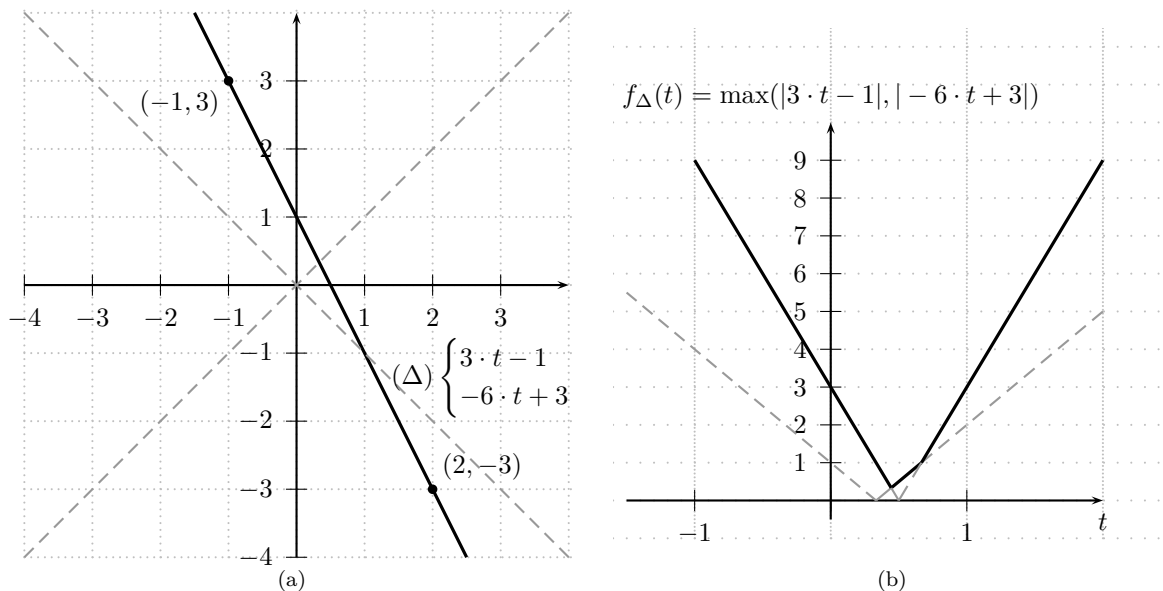


Abbildung A.1: f_{Δ} stellt den durch die unendliche Norm induzierten Abstand der Gerade (Δ) zum Ursprung

□

Satz. Sei $[AB]$ ein Segment. Mit der durch die unendliche Norm induzierten Distanz, ist der Abstand des Segments zum Ursprung das Minimum dieser Abstände :

- der Abstand eines der Endpunkte zum Ursprung
- wenn es sie gibt, der Abstand des Kreuzpunkt zwischen den Segment und die erste oder zweite Diagonale

Beweis der Proposition A. Unter Distanz und Abstand, wird hier die durch die unendliche Norm induzierte Distanz verstanden.

Man betrachtet die parametrische Darstellung von $[AB]$:

$$[AB] = \left\{ \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} (x_B - x_A) \cdot t + x_A \\ (y_B - y_A) \cdot t + y_A \end{pmatrix}, \forall t \in [0; 1] \right\} \quad (\text{A.0.4})$$

Die Funktion $g(t) = \max(|(x_B - x_A) \cdot t + x_A|, |(y_B - y_A) \cdot t + y_A|)$ auf $[0;1]$ ist eine stückweise lineare definierte und stetige Funktion. Die Funktion g kann auch so definiert werden :

$$g(t) = \begin{cases} a_1 \cdot t + b_1 & \forall t \in [0; t_1] \\ a_2 \cdot t + b_2 & \forall t \in [t_1; t_2] \\ \dots & \\ a_n \cdot t + b_n & \forall t \in [t_{n-1}; 1] \end{cases} \quad (\text{A.0.5})$$

mit $a_i \cdot t_i + b_i = a_{i+1} \cdot t_i + b_{i+1} \Leftrightarrow |x(t_i)| = |y(t_i)|, \quad \forall i \in [1 : n - 1] \quad (f \text{ ist stetig})$

Da die Funktion g stückweise linear definiert, und immer positiv ist, ist sein Minimum auf den verschiedenen Intervalle immer an einer (vielleicht nicht nur) den endlichen Intervallgrenze : $0, 1, t_i, i \in [1 : n - 1]$. \square

Anhang B

Bestimmung der Kollisionseigenschaften

Für die Kritikalitätsbewertung einer Verkehrsszene ist es auch von Bedeutung zu wissen, in welcher Konfiguration die Kollision stattfinden wird. Eine wichtige Eigenschaft bei einer solchen Berechnung ist es, die Information so darzustellen oder zu kodieren, dass weiteren Bearbeitungen oder Überprüfungen vereinfacht sind.

Dieser Anhang befasst sich mit der Ermittlung der Kollisionseigenschaften und ihrer Kodierung.

Die Berechnung der Kollisionseigenschaft wird von der im Abschnitt 3.1.1 beschriebenen Distanzfunktion abgeleitet.

Die Zustände der in der Kollision verwickelten Fahrzeuge müssen zur Kollisionszeitpunkt berechnet werden. Aus diesen Zuständen kann für jedes Fahrzeug berechnet werden, welches Element betroffen wird.

Die Beschreibung eines Fahrzeuges, im Sinne der Kollisionseigenschaften, besteht aus acht Elementen (Ecken und Seiten) :

1. vordere linke Ecke
2. hintere linke Ecke
3. hintere rechte Ecke
4. vordere rechte Ecke
5. linke Seite
6. hintere Seite
7. rechte Seite
8. vordere Seite

Um die Barbeitung der Kollisionsfällen, werden diese Elemente mit einer Zahl kodiert. Auf der Abbildung B.1 werden diese Elemente und die entsprechende Kodierung dargestellt.

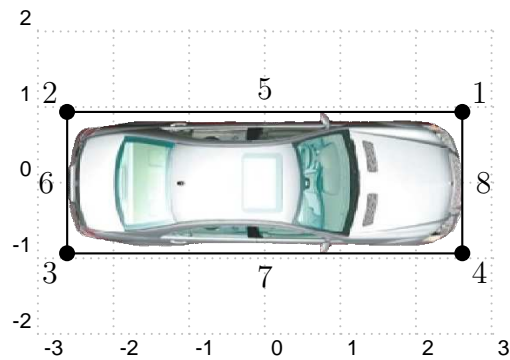
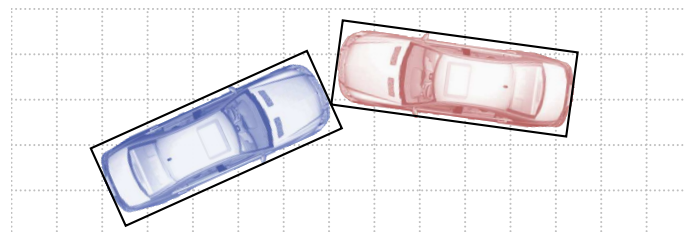


Abbildung B.1: Kodierung der Elemente eines Fahrzeuges

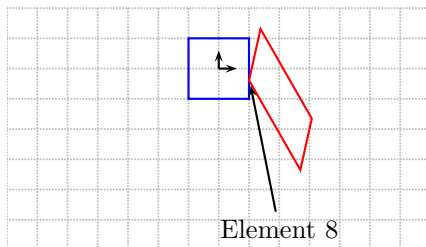
Um herauszufinden welches Element eines Fahrzeuges bei der Kollision betroffen wird, muss der Raum bezüglich dieses Fahrzeuges transformiert werden. Diese Transformation ist die selbe Transformation \mathcal{T} , die im Abschnitt 3.1.1 in der Gleichung (3.1.4) beschrieben wird.

In dem transformierten Raum kann nämlich sehr leicht ermittelt werden, welcher Punkt des anderen Fahrzeuges am nächsten liegt (siehe Abbildungen B.2b, B.2c, und auch den Distanzansatz im Abschnitt 3.1.1). Die im transformierten Raum Koordinaten dieses Punktes geben das betroffene Element des Fahrzeuges. Dieser Punkt liegt nämlich auf dem Rand des Fahrzeuges, und entspricht dadurch ein bestimmtes Element. Wenn es sich um eine Seite handelt, stellt die im Betrag kleinste Koordinate des gefundenen Punktes eine zwischen -1 und 1 normierte Lage der Eintrittsstelle dar. Danach werden die Rollen getauscht, um das betroffene Element des anderen Fahrzeuges herauszufinden.

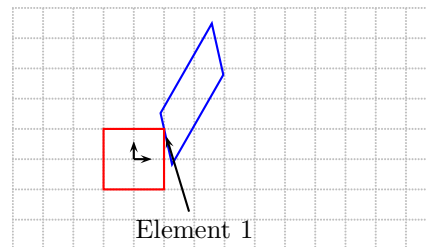
Im Beispiel der Abbildung B.2, wird das blaue Fahrzeug vom roten Fahrzeug an der vorderen Seite getroffen. Im transformierten Raum der Abbildung B.2b hat die Einschussstelle die Koordinaten $(1, -0.3)$. Die y -Koordinate ist die im Betrag kleinste Koordinate und kodiert die Eintrittsstelle auf der betroffenen Seite.



(a) Die Zustände zum Kollisionszeitpunkt werden berechnet.



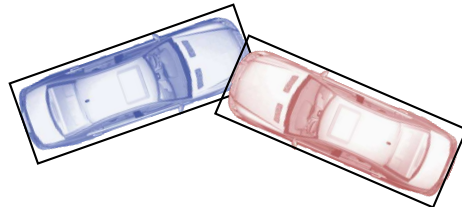
(b) Nach der Raumtransformation : der betroffene Element vom Ego-Fahrzeug ist die vordere Seite. Die genaue Einschussstelle auf dieser Seite wird dabei auch bekannt.



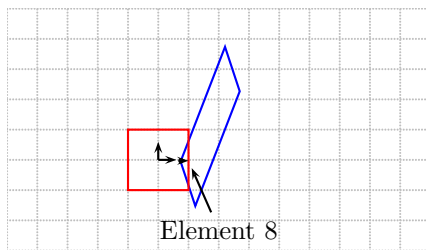
(c) Nach der Raumtransformation : der betroffene Element vom Objekt-Fahrzeug ist die vordere linke Ecke

Abbildung B.2: Berechnung der Kollisionskonfiguration

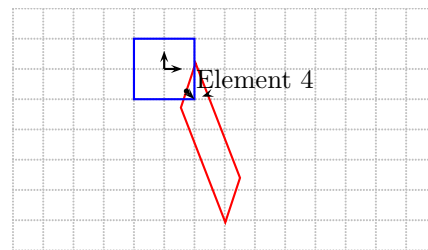
Numerisch betrachtet ist es höchst unwahrscheinlich (unmöglich), dass zwei Seiten oder zwei Ecken in eine Kollision verwickelt werden. Es existiert immer eine Ecke, die als erste in eine Seite eintritt. Dennoch ist TTC nur mit einem Fehler bekannt, das vorherige Beispiel ist ein Idealfall. Je nachdem wie groß das TTC Fehler ist, ist es möglich, dass die berechnete Eintrittsstelle ausserhalb des Fahrzeuges oder im Fahrzeug liegt. Dann liefert die Methode nur eine Approximation der Eintrittsstelle (siehe Abbildung B.3).



(a) Die Zustände zum Kollisionszeitpunkt werden berechnet.



(b) Trotz ein kleines Fehler werden die Elemente richtig erkannt



(c) Trotz ein kleines Fehler werden die Elemente richtig erkannt. Wenn die x - und y -Koordinaten im Betrag gleich sind wird die entsprechende Ecke gewählt

Abbildung B.3: Berechnung der Kollisionskonfiguration bei fehlerhaft TTC

Ein Fehler im dem TTC Wert kann auch dazu führen, dass eine unmögliche Konstellation berechnet wird. Die Abbildung B.4 stellt eine sehr leichte Kollision dar : das blaue Fahrzeug tritt dem roten Fahrzeug in die linke Seite, ganz hinten und für einen sehr kurzen Moment. Aber die Ungenauigkeit von TTC in diesem bestimmten Fall führt dazu, dass eine ganz andere Kollisionskonfiguration erkannt wird. Die Kollisionseigenschaftsberechnung ergibt nämlich : die hintere linke Ecke des roten Fahrzeuges tritt in die linke Seite des blauen Fahrzeuges ein. Diese Situation ist für nach vorne fahrende Fahrzeuge unmöglich.

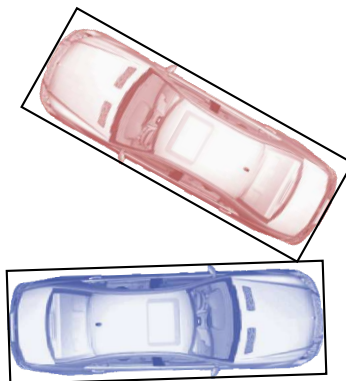


Abbildung B.4: Die Berechnung der Kollisionskonfiguration bei fehlerhaftem TTC kann auch eine unmögliche Konfiguration ergeben.

Die Kodierung, der bei der Kollision betroffenen Elemente, wird in einer zweistelligen kodierten Zahl verwendet :

- die Zehnerstelle stellt das betroffene Element des Ego-Fahrzeuges dar.
- die Einerstelle ist das betroffene Element des Objekt-Fahrzeuges.

Jede Kollisionskonfiguration wird so kodiert, was auch ermöglicht es, in den weiteren Verarbeitungsschritten, die Kollisionskonfigurationen, die normalerweise nicht auftauchen können (denn physikalisch unmöglich), herauszufiltern oder zu korrigieren.

Darüberhinaus erlaubt es die Kodierung, die Kollisionen in Klassen zu ordnen :

- frontale Kollision
- das Ego-Fahrzeug trifft das Objekt-Fahrzeug in die Seite
- das Objekt-Fahrzeug trifft das Ego-Fahrzeug in die Seite
- das Ego-Fahrzeug wird von hinten getroffen
- das Objekt-Fahrzeug wird von hinten getroffen

Der relative Winkel und die Geschwindigkeiten der Fahrzeug gehören auch zu den berücksichtigten Informationen.

Anhang C

Berechnung der Input Kovarianzmatrix

Das Kapitel 4 befasst sich mit der Berechnung der Kollisionswahrscheinlichkeit unter Berücksichtigung von Messunsicherheiten. Dafür wird ein normalverteilter Messrauschen erzeugt, der diese Unsicherheit darstellt. Um eine realistische Kovarianzmatrix des Messrauschen zu haben, wird eine Kovarianzmatrix verwendet, die durch Bildverarbeitung gemessen worden ist. Die Kovarianzmatrix wird entsprechend verändert, um die Eingangsgrößen des Fahrzeugmodells zu beantworten.

Die Eingangsgrößen, die vom Fahrzeugmodell verlangt werden, sind :

- die x-Koordinate x
- die y-Koordinate y
- der Gierwinkel ψ
- die Längsgeschwindigkeit v_x
- die Längsbeschleunigung a_x
- der Radwinkel δ

Zusätzliche gemessene Parameter werden auch benötigt wie Länge, Breite und Radstand.

Die Bildverarbeitung liefert eine Kovarianzmatrix Σ_{bv} für Größen die diese nicht übereinstimmen, denn für die Bildverarbeitung werden andere Koordinatensysteme verwendet, und eine Größe wie der Radwinkel δ kann unmöglich durch Bildverarbeitung gemessen werden. Die Bildverarbeitung liefert eine Kovarianzmatrix für folgende Größe :

- X_{bv} , das $-y$ entspricht
- Z_{bv} , das x entspricht
- ψ_{bv}
- v_{bv} , das v_x entspricht
- $\dot{\psi}_{bv}$
- a_{bv} , das a_x entspricht

- $rotX_{bv}$
- $rotZ_{bv}$
- β_{bv}

Wegen Mangel an Daten werden für das weitere diese Gleichungen angenommen :

$$\dot{\psi} = \dot{\psi}_{bv} \quad (C.0.1)$$

$$\psi = \psi_{bv} + \beta_{bv} \quad (C.0.2)$$

$$\delta = \frac{\dot{\psi}_{bv} \cdot l}{v_{bv}} \quad \text{wo } l \text{ der Radstand ist} \quad (C.0.3)$$

Um die gewünschte Kovarianzmatrix berechnen zu können, werden die Abbildung f der Gleichung (C.0.4) und den Kovarianzfortpflanzungsgesetz (siehe Abschnitt 4.3.3.1) benutzt.

$$f \begin{pmatrix} X_{bv} \\ Z_{bv} \\ \psi_{bv} \\ v_{bv} \\ \dot{\psi}_{bv} \\ a_{bv} \\ rotX_{bv} \\ rotZ_{bv} \\ \beta_{bv} \end{pmatrix} = \begin{pmatrix} x \\ y \\ \psi \\ v_x \\ a_x \\ \delta \end{pmatrix} = \begin{pmatrix} Z_{bv} \\ -X_{bv} \\ \psi_{bv} + \beta_{bv} \\ v_{bv} \\ a_{bv} \\ \frac{\dot{\psi}_{bv} \cdot l}{v_{bv}} \end{pmatrix} \quad (C.0.4)$$

Die Erwartungswerte aus der Simulation werden benutzt, um die Jacobi-Matrix der Abbildung f zu berechnen.

$$J_f = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\dot{\psi} \cdot l}{v^2} & \frac{l}{v} & 0 & 0 & 0 & 0 \end{pmatrix} \quad (C.0.5)$$

Die gewünschte Kovarianzmatrix Σ wird wie folgt berechnet :

$$\Sigma = J_f \cdot \Sigma_{bv} \cdot J_f^T \quad (C.0.6)$$

Anhang D

Berechnung der stochastischen Momente einer linksseitigen gesättigten Normalverteilung

Definition

Im Abschnitt 4.4.3 wurde die Wahrscheinlichkeitsverteilung $\mathcal{N}_h(\alpha, \mu, \sigma)$ eingeführt. Die Verteilung entspricht eine linksseitige gesättigte Normalverteilung :

$$p(x) = c \cdot \delta(x - \alpha) + \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} \cdot u(x - \alpha) \quad (\text{D.0.1})$$

$$c = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(t-\mu)^2}{2 \cdot \sigma^2}} dt \quad (\text{D.0.2})$$

In diesem Anhang wird die ausführliche Berechnung der stochastischen Momente der ersten und zweiten Ordnung vorgeführt.

Mathematische Grundlagen

Diese Gleichungen werden bei der Berechnung benötigt.

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \cdot \int_x^{+\infty} e^{-t^2} dt \quad (\text{D.0.3})$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \cdot \int_0^x e^{-t^2} dt \quad (\text{D.0.4})$$

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) \quad (\text{D.0.5})$$

$$\int_x^{+\infty} e^{-\frac{u^2}{2}} du = \sqrt{\frac{\pi}{2}} \cdot \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) \quad (\text{D.0.6})$$

$$\int_{-\infty}^x e^{-\frac{u^2}{2}} du = \sqrt{2\pi} - \sqrt{\frac{\pi}{2}} \cdot \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) \quad (\text{D.0.7})$$

$$\int_{-\infty}^{+\infty} e^{-\frac{u^2}{2}} du = \sqrt{2\pi} \quad (\text{D.0.8})$$

$$\int t^2 \cdot e^{-t^2} dt = \frac{\sqrt{\pi}}{4} \cdot \operatorname{erf}(t) - \frac{t \cdot e^{-t^2}}{2} \quad (\text{D.0.9})$$

$$\int t^2 \cdot e^{-\frac{t^2}{2}} dt = \frac{\sqrt{2\pi}}{2} \cdot \operatorname{erf}\left(\frac{t}{\sqrt{2}}\right) - t \cdot e^{-\frac{t^2}{2}} \quad (\text{D.0.10})$$

Berechnung des Erwartungswerts (Moment der erster Ordnung)

$$\begin{aligned} E[X] &= \int_{-\infty}^{+\infty} x \cdot P(x) dx \\ &= \alpha \cdot c + \int_{\alpha}^{+\infty} \frac{x}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx \\ &= \alpha \cdot \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx + \int_{\alpha}^{+\infty} \frac{x}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx \\ &\text{mit } u = \frac{x-\mu}{\sigma}, \quad du = \frac{dx}{\sigma} \\ &= \alpha \cdot \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\alpha-\mu}{\sigma}} e^{-\frac{u^2}{2}} du + \frac{\sigma}{\sqrt{2\pi}} \int_{\frac{\alpha-\mu}{\sigma}}^{+\infty} u \cdot e^{-\frac{u^2}{2}} du + \frac{\mu}{\sqrt{2\pi}} \int_{\frac{\alpha-\mu}{\sigma}}^{+\infty} e^{-\frac{u^2}{2}} du \\ &= \alpha \cdot \frac{1}{\sqrt{2\pi}} \left[\sqrt{2\pi} - \sqrt{\frac{\pi}{2}} \cdot \operatorname{erfc}\left(\frac{\alpha-\mu}{\sqrt{2}\sigma}\right) \right] + \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} + \frac{\mu}{\sqrt{2\pi}} \left[\sqrt{\frac{\pi}{2}} \cdot \operatorname{erfc}\left(\frac{\alpha-\mu}{\sqrt{2}\sigma}\right) \right] \\ &= \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} - \frac{\alpha-\mu}{2} \operatorname{erfc}\left(\frac{\alpha-\mu}{\sqrt{2}\sigma}\right) - \alpha \end{aligned} \quad (\text{D.0.11})$$

Berechnung des Moments der 2er Ordnung

$$\begin{aligned}
 E[X^2] &= \int_{-\infty}^{+\infty} x^2 \cdot P(x) dx \\
 &= \alpha^2 \cdot c + \int_{\alpha}^{+\infty} \frac{x^2}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} dx \\
 &\text{mit } u = \frac{x-\mu}{\sigma}, \quad du = \frac{dx}{\sigma} \\
 &= \alpha^2 \cdot \left[1 - \frac{1}{2} \cdot \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \right] + \frac{\sigma^2}{\sqrt{2\pi}} \int_{\frac{\alpha-\mu}{\sigma}}^{+\infty} u^2 e^{-\frac{u^2}{2}} du + \frac{2\mu\sigma}{\sqrt{2\pi}} \int_{\frac{\alpha-\mu}{\sigma}}^{+\infty} u e^{-\frac{u^2}{2}} du + \frac{\mu^2}{\sqrt{2\pi}} \int_{\frac{\alpha-\mu}{\sigma}}^{+\infty} e^{-\frac{u^2}{2}} du \\
 &= \alpha^2 \cdot \left[1 - \frac{1}{2} \cdot \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \right] + \frac{\sigma^2}{\sqrt{2\pi}} \left[\frac{\sqrt{2\pi}}{2} \cdot \operatorname{erf} \left(\frac{t}{\sqrt{2}} \right) - t \cdot e^{-\frac{t^2}{2}} \right]_{\frac{\alpha-\mu}{\sigma}}^{+\infty} \\
 &\quad + \frac{2\mu\sigma}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} + \frac{\mu^2}{\sqrt{2\pi}} \sqrt{\frac{\pi}{2}} \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \\
 &= \alpha^2 \cdot \left[1 - \frac{1}{2} \cdot \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \right] + \frac{\sigma^2}{\sqrt{2\pi}} \left[\frac{\sqrt{2\pi}}{2} - \left(\frac{\sqrt{2\pi}}{2} \left[1 - \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \right] \right) \right] + \frac{\alpha-\mu}{\sigma} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} \\
 &\quad + \frac{2\mu\sigma}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} + \frac{\mu^2}{2} \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \\
 &= \alpha^2 \cdot \left[1 - \frac{1}{2} \cdot \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \right] + \left[\frac{\sigma^2}{2} \cdot \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) + \frac{\sigma(\alpha-\mu)}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} \right] \\
 &\quad + \frac{2\mu\sigma}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} + \frac{\mu^2}{2} \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) \\
 &= \frac{\sigma \cdot (\alpha + \mu)}{\sqrt{2\pi}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}} + \frac{\sigma^2 + \mu^2 - \alpha^2}{2} \operatorname{erfc} \left(\frac{\alpha-\mu}{\sqrt{2}\sigma} \right) + \alpha^2
 \end{aligned} \tag{D.0.12}$$

Anhang E

Bestimmung der Aufenthaltswahrscheinlichkeit

Im Rahmen der Arbeit wurde die Berechnung der Aufenthaltswahrscheinlichkeit untersucht. Obwohl diese Untersuchung nicht Teil des finalen Verfahrens zur Kollisionswahrscheinlichkeitsberechnung ist, soll die Methode für späteren möglichen Anwendungen dokumentiert werden.

Die Berechnung der Aufenthaltswahrscheinlichkeit der Objektfahrzeuge erfolgt durch die Benutzung des Kovarianzfortpflanzungsgesetzes (siehe Abschnitt 4.3.3) : die Bewegungsgleichungen werden linearisiert, um die Verteilungen der resultierenden Grössen , Position und Orientierung, bestimmen zu können

Berechnung der Jacobi-Matrix für die Bewegungsgleichungen

Die Linearisierung der Bewegungsgleichungen erfolgt durch die Berechnung der Jacobi-Matrix $J(t)$. Je nachdem, ob der Radlenkwinkel δ null oder nicht null ist, gibt es zwei Sätzen von Bewegungsgleichungen (siehe Abschnitt 3.2.3.2).

Im ersten Satz ist δ nicht null :

$$\begin{aligned}\psi(t) &= \frac{\delta}{l} \cdot \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t \right) + \psi_0 \\ x(t) &= \frac{l}{\delta} \cdot (\sin(\psi(t)) - \sin(\psi_0)) + x_0 \\ y(t) &= -\frac{l}{\delta} \cdot (\cos(\psi(t)) - \cos(\psi_0)) + y_0\end{aligned}\tag{E.0.1}$$

Die partiellen Ableitungen dieser Bewegungsgleichungen sind :

$$\begin{aligned}
 \left. \frac{\partial \psi}{\partial \psi_0} \right|_t &= 1 \\
 \left. \frac{\partial \psi}{\partial v_0} \right|_t &= \frac{\delta}{l} \cdot t \\
 \left. \frac{\partial \psi}{\partial a} \right|_t &= \frac{\delta}{l} \cdot \frac{1}{2} \cdot t^2 \\
 \left. \frac{\partial \psi}{\partial \delta} \right|_t &= \frac{1}{l} \cdot \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t \right)
 \end{aligned} \tag{E.0.2}$$

$$\begin{aligned}
 \left. \frac{\partial x}{\partial x_0} \right|_t &= 1 \\
 \left. \frac{\partial x}{\partial \psi_0} \right|_t &= \frac{l}{\delta} \cdot (\cos(\psi(t)) - \cos(\psi_0)) \\
 \left. \frac{\partial x}{\partial a} \right|_t &= \frac{1}{2} \cdot t^2 \cdot \cos(\psi(t)) \\
 \left. \frac{\partial x}{\partial v_0} \right|_t &= t \cdot \cos(\psi(t)) \\
 \left. \frac{\partial x}{\partial \delta} \right|_t &= -\frac{l}{\delta^2} \cdot (\sin(\psi(t)) - \sin(\psi_0)) + \frac{1}{\delta} \cdot \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t \right) \cdot \cos(\psi(t))
 \end{aligned} \tag{E.0.3}$$

$$\begin{aligned}
 \left. \frac{\partial y}{\partial y_0} \right|_t &= 1 \\
 \left. \frac{\partial y}{\partial \psi_0} \right|_t &= \frac{l}{\delta} \cdot (\sin(\psi(t)) - \sin(\psi_0)) \\
 \left. \frac{\partial y}{\partial a} \right|_t &= \frac{1}{2} \cdot t^2 \cdot \sin(\psi(t)) \\
 \left. \frac{\partial y}{\partial v_0} \right|_t &= t \cdot \sin(\psi(t)) \\
 \left. \frac{\partial y}{\partial \delta} \right|_t &= \frac{l}{\delta^2} \cdot (\cos(\psi(t)) - \cos(\psi_0)) + \frac{1}{\delta} \cdot \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t \right) \cdot \sin(\psi(t))
 \end{aligned} \tag{E.0.4}$$

Im zweiten Satz ist δ null, die entsprechenden Bewegungsgleichungen sind :

$$\begin{aligned}
 \psi(t) &= \psi_0 \\
 x(t) &= \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t \right) \cdot \cos(\psi_0) + x_0 \\
 y(t) &= \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t \right) \cdot \sin(\psi_0) + y_0
 \end{aligned} \tag{E.0.5}$$

Die partiellen Ableitungen dieser Bewegungsgleichungen sind :

$$\left. \frac{\partial \psi}{\partial \psi_0} \right|_t = 1 \quad (\text{E.0.6})$$

$$\begin{aligned} \left. \frac{\partial x}{\partial x_0} \right|_t &= 1 \\ \left. \frac{\partial x}{\partial \psi_0} \right|_t &= -\left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t\right) \cdot \sin(\psi_0) \\ \left. \frac{\partial x}{\partial a} \right|_t &= \frac{1}{2} \cdot t^2 \cdot \cos(\psi_0) \end{aligned} \quad (\text{E.0.7})$$

$$\begin{aligned} \left. \frac{\partial x}{\partial v_0} \right|_t &= t \cdot \cos(\psi_0) \\ \left. \frac{\partial y}{\partial y_0} \right|_t &= 1 \\ \left. \frac{\partial y}{\partial \psi_0} \right|_t &= \left(\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t\right) \cdot \cos(\psi_0) \end{aligned} \quad (\text{E.0.8})$$

$$\begin{aligned} \left. \frac{\partial y}{\partial a} \right|_t &= \frac{1}{2} \cdot t^2 \cdot \sin(\psi_0) \\ \left. \frac{\partial y}{\partial v_0} \right|_t &= t \cdot \sin(\psi_0) \end{aligned}$$

Allgemein wird die Jakobie Matrix $J(t)$ ausgedrückt, wie folgt :

$$J(t) = \begin{pmatrix} \left. \frac{\partial x}{\partial x_0} \right|_t & \left. \frac{\partial x}{\partial y_0} \right|_t & \left. \frac{\partial x}{\partial \psi_0} \right|_t & \left. \frac{\partial x}{\partial a_0} \right|_t & \left. \frac{\partial x}{\partial v_0} \right|_t & \left. \frac{\partial x}{\partial \delta_0} \right|_t \\ \left. \frac{\partial y}{\partial x_0} \right|_t & \left. \frac{\partial y}{\partial y_0} \right|_t & \left. \frac{\partial y}{\partial \psi_0} \right|_t & \left. \frac{\partial y}{\partial a_0} \right|_t & \left. \frac{\partial y}{\partial v_0} \right|_t & \left. \frac{\partial y}{\partial \delta_0} \right|_t \\ \left. \frac{\partial \psi}{\partial x_0} \right|_t & \left. \frac{\partial \psi}{\partial y_0} \right|_t & \left. \frac{\partial \psi}{\partial \psi_0} \right|_t & \left. \frac{\partial \psi}{\partial a_0} \right|_t & \left. \frac{\partial \psi}{\partial v_0} \right|_t & \left. \frac{\partial \psi}{\partial \delta_0} \right|_t \end{pmatrix} \quad (\text{E.0.9})$$

Die Kovarianzmatrix $\Sigma_{loc}(t)$ der Aufenthaltswahrscheinlichkeit wird Hilfe des Kovarianzfortpflanzungsgesetzes und der Kovarianzmatrix der Eingangsdaten berechnet :

$$\Sigma_{loc}(t) = J(t) \cdot \Sigma_{bv} \cdot J(t)^T \quad (\text{E.0.10})$$

Ergebnisse

Die Ergebnisse dieses Verfahrens werden in der Abbildungen E.1 bis E.3 dargestellt. Die verschiedenen farbigen Histogramme entsprechen den simulativen Referenzdaten an verschiedenen Zeitpunkten des Intervalls $[t_0; t_H]$ (dunkel-blau für t_0 bis dunkel-rot für t_H). Das Histogramm für t_0 entspricht eigentlich der Streuung der Eingangsdaten.

Die mit dem Kovarianzfortpflanzungsgesetz berechneten Verteilungen werden in schwarz darge-

stellt.

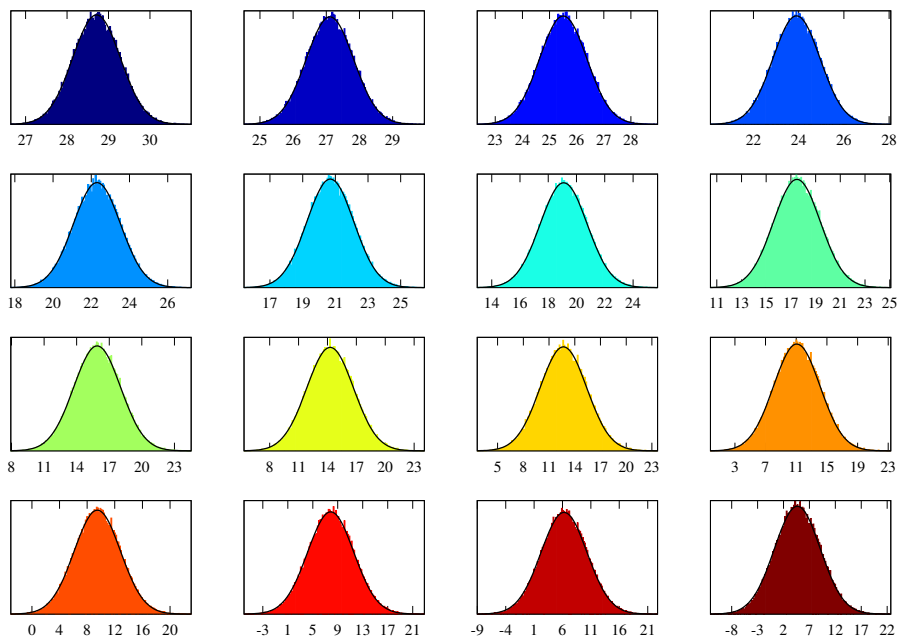


Abbildung E.1: Vergleich zwischen den tatsächlichen Aufenthaltshäufigkeitshistogramm und die Approximation der Aufenthaltswahrscheinlichkeit für die x -Koordinate.

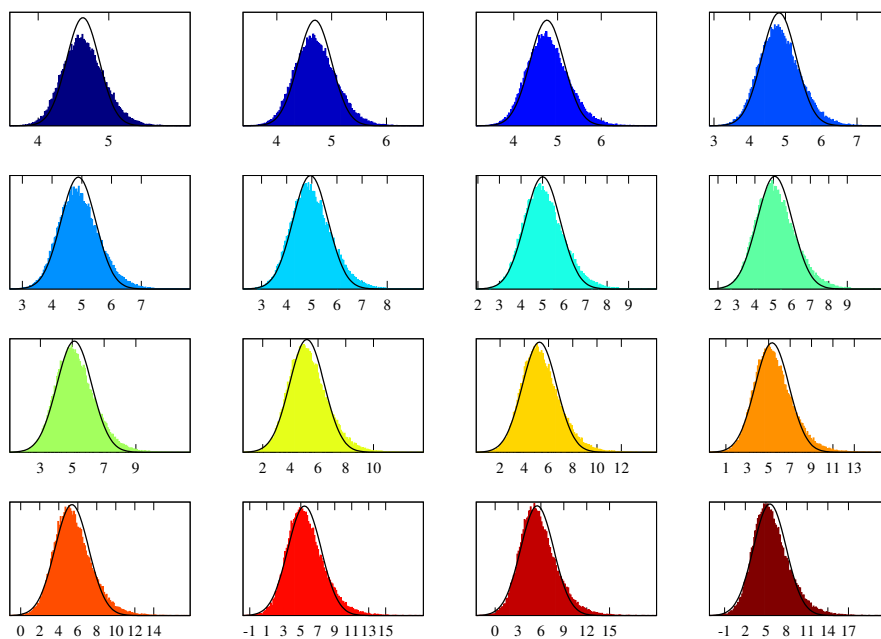


Abbildung E.2: Vergleich zwischen den tatsächlichen Aufenthaltshistogramm und die Approximation der Aufenthaltswahrscheinlichkeit für die y -Koordinate.

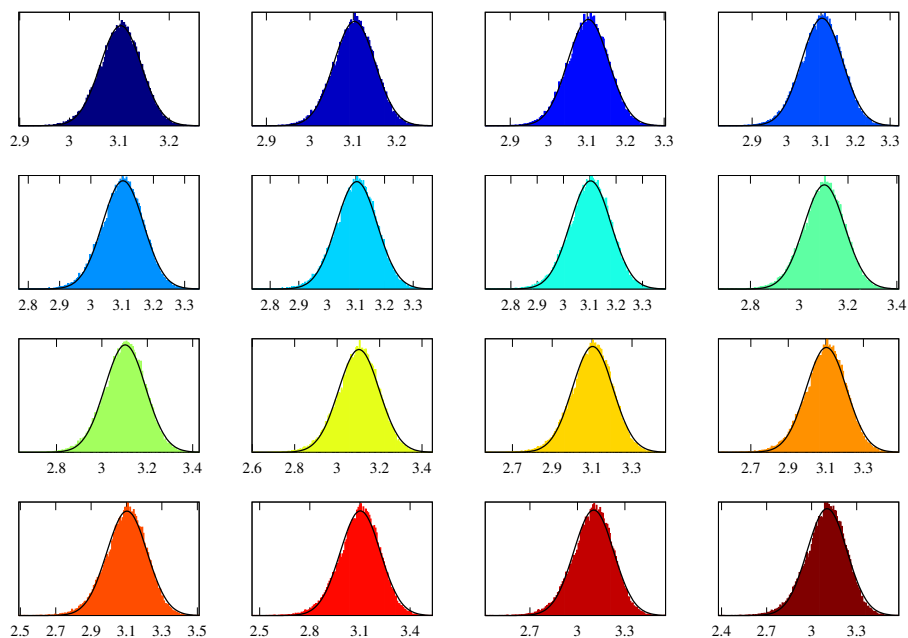


Abbildung E.3: Vergleich zwischen den tatsächlichen Aufenthaltshistogramm und die Approximation der Aufenthaltswahrscheinlichkeit für das Gierwinkel ψ .