

Beiträge zum Stuttgarter Maschinenbau

Marcel Huptych

Online-Bahnplanung für mehrere Flugroboter in veränderlicher Umgebung mithilfe der Kurvenflussmethode



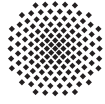
stuttgarter
maschinenbau
interdisziplinär und vielfältig



Universität Stuttgart

Institut für Steuerungstechnik
der Werkzeugmaschinen und
Fertigungseinrichtungen (ISW)





Universität Stuttgart



Beiträge zum Stuttgarter Maschinenbau

Band 2

Herausgeber: Prof. Dr.-Ing. Oliver Riedel
Prof. Dr.-Ing. Alexander Verl
Jun.-Prof. Dr. rer. nat. Andreas Wortmann

Marcel Huptych

**Online-Bahnplanung für mehrere Flugroboter
in veränderlicher Umgebung mithilfe der
Kurvenflussmethode**

Fraunhofer Verlag

Kontaktadresse:

Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen ISW
Seidenstr. 36
70174 Stuttgart
info@isw.uni-stuttgart.de
<https://www.isw.uni-stuttgart.de>

Titelbild: Marcel Huptych, mit freundlicher Genehmigung von KUKA, Schuler und Ascending Technologies A Part of Intel

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.de> abrufbar.

ISSN: 2750-655X

ISBN: 978-3-8396-1800-4

D 93

Zugl.: Stuttgart, Univ., Diss., 2021

Druck und Weiterverarbeitung:
Fraunhofer Verlag, Mediendienstleistungen

Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© Fraunhofer Verlag, 2022

Nobelstraße 12
70569 Stuttgart
verlag@fraunhofer.de
www.verlag.fraunhofer.de

als rechtlich nicht selbständige Einheit der

Fraunhofer-Gesellschaft zur Förderung
der angewandten Forschung e.V.
Hansastraße 27 c
80686 München
www.fraunhofer.de

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften.

Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

Geleitwort

Die deutsche Wirtschaft ist weltweit bekannt für ihren Anlagen- und Maschinenbau. Dabei ist die Universität Stuttgart mit ihren beiden Maschinenbau fakultäten – unter deren Dach sich 42 Institute befinden – die größte universitäre Einrichtung für den Maschinenbau in Deutschland. Unsere wissenschaftliche Exzellenz stützt sich dabei auf unsere zahlreichen Promovierenden und ihre hervorragenden Dissertationen. Viele dieser Dissertationen entstehen in lokaler, nationaler und internationaler Zusammenarbeit mit renommierten Universitäten und außeruniversitären Einrichtungen wie dem Deutschen Zentrum für Luft- und Raumfahrt, der Fraunhofer-Gesellschaft und der Max-Planck-Gesellschaft. Dabei reicht das inhaltliche Spektrum der Dissertationen von Biotechnik, Energietechnik, Fahrzeugtechnik, Kybernetik und Systemtechnik, Produktentwicklung und Konstruktionstechnik, Produktionstechnik bis hin zur Verfahrenstechnik und stützt sich auf die sechs Forschungsschwerpunkte Advanced Systems Engineering, Autonome Produktion, Software-Defined Manufacturing, Resiliente Versorgung, Biointelligenz und Dekarbonisierung der Industrie. Die Ergebnisse aus den Dissertationen zielen darauf ab, kunden-, produkt-, prozess- und mitarbeiterorientierte Technologie zielgerichtet und zeitnah zu entwickeln und anzuwenden.

Viele der im Rahmen der Forschungsarbeiten an den Instituten entstandenen Dissertationen werden in diesen »Beiträgen zum Stuttgarter Maschinenbau« veröffentlicht. Die beiden Fakultäten des Stuttgarter Maschinenbaus wünschen den Promovierenden, dass ihre Dissertationen aus dem Bereich des Maschinenbaus in der breiten Fachwelt als maßgebliche Beiträge wahrgenommen werden und so den Wissensstand auf ein neues Niveau heben.

Für den Stuttgarter Maschinenbau



Stefan Weihe
Prodekan Fakultät 4



Oliver Riedel
Prodekan Fakultät 7

Vorwort der Herausgeber

Innerhalb der Reihe »Beiträge zum Stuttgarter Maschinenbau« berichtet das Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen an der Universität Stuttgart (ISW) über seine Forschungsergebnisse. Das Institut beschäftigt sich in vielfältiger Form mit Steuerungs- und Automatisierungstechnik sowie dem Einsatz von modernen Methoden des Informationsmanagements. Dabei stehen Grundlagenforschung und anwendungsorientierte Entwicklung in einem stetigen Austausch, wodurch ein kontinuierlicher Technologietransfer in die Praxis sichergestellt wird.

Die am ISW entstandenen Dissertationen werden damit unter erweitertem Namen und inzwischen in vierter Generation in der bewährten Konzeption, die der Gründer des ISW Prof. Stute und sein Nachfolger Prof. Pritschow 1972 begonnen haben, durch die heutige Institutsleitung fortgesetzt.

Herrn Dipl.-Ing. Marcel Huptych möchten wir für die geleistete Arbeit danken, dem Verlag für die Aufnahme dieser Schriftenreihe in sein Angebot und der Druckerei für die saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.



Alexander Verl



Oliver Riedel

Online-Bahnplanung für mehrere Flugroboter in veränderlicher Umgebung mithilfe der Kurvenflussmethode

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von

Dipl.-Ing. Marcel Huptych

aus Prag

Hauptberichter: Univ.-Prof. Dr.-Ing. Dr. h. c. mult. Alexander Verl

Mitberichter: Prof. Dr.-Ing. Sascha Röck

Tag der mündlichen Prüfung: 20. April 2021

Institut für Steuerungstechnik der Werkzeugmaschinen und
Fertigungseinrichtungen der Universität Stuttgart

2022

Vorwort des Autors

Die vorliegende Arbeit entstand während meiner Tätigkeit am Virtual Automation Lab (VAL) der Fakultät Maschinenbau an der Hochschule Esslingen in Kooperation mit dem Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik an der Universität Stuttgart sowie dem Stuttgart Research Center for Simulation Technology (SRC SimTech), ebenfalls an der Universität Stuttgart.

Mein herzlicher Dank gilt Herrn Prof. Dr.-Ing. Dr. h. c. mult. Alexander Verl, dem Leiter des ISW, für die Unterstützung und das entgegengebrachte Vertrauen bei der Durchführung dieser Arbeit sowie für die Übernahme des Hauptberichts.

Besonderer Dank gilt Herrn Prof. Dr.-Ing. Sascha Röck, dem Leiter des VAL, für die fachlichen Anregungen sowie die stets motivierenden Diskussionen, welche in hohem Maße zum Gelingen dieser Arbeit beigetragen haben. Außerdem danke ich ihm für die Übernahme des Mitberichts.

Herrn Prof. Dr.-Ing. Hans-Christian Reuss danke ich für die Übernahme des Prüfungsvorsitzes.

Darüber hinaus möchte ich mich bei allen Kollegen für den kontinuierlichen fachlichen Austausch sowie die stets angenehme Zusammenarbeit bedanken.

Marcel Huptych

Inhaltsverzeichnis

Abbildungsverzeichnis	4
Tabellenverzeichnis	7
Abkürzungsverzeichnis	8
Kurzfassung	10
Abstract	12
1 Einleitung	14
1.1 Einführung	14
1.2 Anforderungen	17
1.3 Zielsetzung	19
1.4 Problemstellung	20
1.5 Überblick über die Arbeit	21
2 Stand der Forschung und Technik	23
2.1 Einordnung von UAVs in die Robotik	23
2.2 Das Bahnplanungsproblem	25
2.3 Umgebungs- und Robotermodell	26
2.4 Koordinatenraum	28
2.5 Einordnung der Problemstellung	30
2.6 Bahnplanung in dynamischer Umgebung	32
2.6.1 Globale Methoden	33
2.6.2 Reaktive Methoden	37
2.6.3 Gemischte Methoden	40
2.7 Defizite und Stärken	41
2.8 Bewertung, Zusammenfassung und Fazit	49

3	Voruntersuchung zur Elastic-Bands-Methode (EBM)	53
4	Kurvenflussmethode (KFM)	58
4.1	Prinzip der KFM.....	58
4.2	Nichtlineare Kurvenflüsse als Ursprung der KFM	59
4.3	Ortsdiskretisierung.....	61
4.4	Kurvenfluss für offene Kurven	63
4.5	Dynamik des Kurvenflusses	65
4.6	Kurvenfluss mit Hindernissen	69
4.7	Dynamik mit Hindernissen.....	73
4.8	Stationäre Endlage	80
4.9	Parametrierung.....	82
4.10	Zeitdiskretisierung	86
4.11	Vergleich des Bahnverhaltens von KFM und EBM.....	89
4.12	Vergleich der Rechenzeiten von KFM und EBM.....	93
5	Funktionen für die KFM in komplexen Umgebungen	99
5.1	Stetig differenzierbares Kraftfeld	99
5.2	Räumliche Hinderniskörper mit Durchdringung	103
5.3	Gradientenberechnung.....	106
5.4	Räumliche Bahn mit nichtlinearem Kraftfeld.....	107
5.5	Initialbahn	112
5.6	Interpolation auf dynamischer Bahn	114
5.7	Parallele Ausführung mehrerer Instanzen der KFM.....	118
5.8	Start- und Landevorgänge.....	122
6	Implementierung und Realisierung einer Testumgebung für die KFM	124
6.1	Architektur	124

6.2	Verwendete Soft- und Hardware	125
6.3	Implementierung der KFM in einer SPS	131
6.4	Dynamikmodell der UAVs.....	134
6.5	Evaluierung mit Dynamikmodell	140
6.6	Evaluierung mit realen UAVs.....	147
7	Zusammenfassung	152
8	Literaturverzeichnis.....	158
9	Anhang	192
9.1	Pseudocode für die Berechnung des Parameters K als inverses Problem.....	192
9.2	Pseudocode KFM	194
9.3	Pseudocode Abstoßungskraft.....	196
9.4	Hindernisdaten	199
9.5	Reglereinstellung für das Dynamikmodell	199

Abbildungsverzeichnis

Abbildung 1: Vorgestellte Methoden zur Bahnplanung in dynamischer Umgebung.....	33
Abbildung 2: Die Grundprinzipien von RRT und PRM	34
Abbildung 3: Kollision durch träges Ausweichverhalten der EBM.....	54
Abbildung 4: Dynamisches Ausweichverhalten der EBM durch steife Parametrierung	55
Abbildung 5: Instabiles Lösungsverhalten der EBM für große Schrittweiten.....	56
Abbildung 6: Wirkprinzip der KFM	59
Abbildung 7: Approximation der nichtlinearen Rückstellkraft mit linearer Vektoraddition...	61
Abbildung 8: Normal- und Tangentialwirkung der linearen Rückstellkraft k_i	63
Abbildung 9: Entwicklung einer geschlossenen und einer offenen Kurve	64
Abbildung 10: Eigenwerte der Systemmatrix \mathbf{A} in Abhängigkeit der Diskretisierung n	65
Abbildung 11: Kurvenflussdynamik bei einem Sprung des Zielpunktes y_g	67
Abbildung 12: Entwicklung einer Bahn unter Einfluss dreier Punkthindernisse	71
Abbildung 13: Kraftwirkung von nichtlinearer Punkt- und linearer Streckenlast.....	73
Abbildung 14: Verschiebung der Eigenwerte von \mathbf{A}^* in Abhängigkeit von T und K	75
Abbildung 15: Antworten auf Einheitssprung der Wirkweite r_0 bei Vollbelastung.....	76
Abbildung 16: Gerschgorin-Kreise zur Eingrenzung der Eigenwerte bei Teilbelastung	77
Abbildung 17: Antworten auf Einheitssprung der Wirkweite r_0 im teilbelasteten Fall.....	78
Abbildung 18: Eigenwertverteilung und gemessene Stellzeit T_\emptyset bei Teilbelastung.....	79
Abbildung 19: Stationäre Endlage in Abhängigkeit von K und n	80
Abbildung 20: Stationärer Sicherheitsabstand bemessen mit verschiedenen Normen.....	82
Abbildung 21: Stationäre Endlage als Lösung eines inversen Problems	85
Abbildung 22: Identifikation des Bahnverhaltens der EBM mit linearer Streckenlast.....	90
Abbildung 23: Identisches Bahnverhalten von EBM und KFM.....	91
Abbildung 24: Stabiles Lösungsverhalten der KFM für beliebig große Zeitschrittweiten....	91
Abbildung 25: Unabhängigkeit der KFM von einer Skalierung der Lagegrößen	92
Abbildung 26: Theoretische Rechenkomplexität von KFM und EBM (nur Rückstellung)...	94
Abbildung 27: Lineare Rechenkomplexität der KFM auf allen Plattformen	97
Abbildung 28: Nicht stetige Differenzierbarkeit des bisher verwendeten Kraftfeldes	99
Abbildung 29: Erweiterung der Gewichtungsfunktion $\rho(H)$ um einen Viertelsinus	100

Abbildung 30: Geglättetes Kraftfeld für verschiedene Parametrierungen von r_u und p	102
Abbildung 31: Äquipotentialflächen eines exemplarischen Kraftfeldes	105
Abbildung 32: Auswertungen von p für verschiedene Differenzenquotienten	106
Abbildung 33: Abstoßung und Rückstellung für das einfache Szenario.....	109
Abbildung 34: Auslenkung bei Abstoßung und Rückstellung für das einfache Szenario .	109
Abbildung 35: Abstoßung für das komplexe Szenario	110
Abbildung 36: Orthogonale Ansichten für das komplexe Szenario	111
Abbildung 37: Stützstellendistanzen bei Abstoßung für das komplexe Szenario.....	111
Abbildung 38: Einschwingvorgang einer kollisionsbehafteten Initialbahn	114
Abbildung 39: Verlängerung des Verzögerungsweges um Δs_a durch Bahnkrümmung ...	117
Abbildung 40: Bewegungsprofile einer geraden und einer gekrümmten Bahn	117
Abbildung 41: Sollwerttrajektorien und Abstände für drei UAVs	119
Abbildung 42: Orthogonale Ansichten der Ausweichbewegung.....	120
Abbildung 43: Luftraum innerhalb einer Maschinenhalle	120
Abbildung 44: Abstände für drei UAVs mit statischen Hindernissen	121
Abbildung 45: Sollwerttrajektorien für drei UAVs mit statischen Hindernissen.....	122
Abbildung 46: Vertikaler Start- und Landevorgang	123
Abbildung 47: Architektur des Gesamtsystems zur Realisierung der KFM	124
Abbildung 48: Crazyflie mit LPS-Deck und Crazyradio PA USB Dongle	127
Abbildung 49: Flugzelle und ein einzelner LPS-Anker	128
Abbildung 50: Überblick über die verwendeten Soft- und Hardwarekomponenten	130
Abbildung 51: Gesamtaufbau	130
Abbildung 52: Ein- und Ausgangsvariablen des Funktionsbausteins „FB_KFM“	132
Abbildung 53: Koordinatensysteme zur Modellierung eines UAVs in „H“-Konfiguration ..	135
Abbildung 54: Modelleingänge und Drehzahlen für exemplarische Flugmanöver	136
Abbildung 55: Sprungantworten der Regelgrößen des geregelten Dynamikmodells	139
Abbildung 56: Sprungantworten der Stellgrößen des geregelten Dynamikmodells	139
Abbildung 57: Zustandsgeregeltes Dynamikmodell als Signalflussplan in Virtuos.....	140
Abbildung 58: Online-Visualisierung der Testumgebung im VAL 3D-Webstudio	141
Abbildung 59: Positions- und Distanzverläufe des ersten Szenarios	143
Abbildung 60: Trajektorien des ersten Szenarios	143

Abbildung 61: Online-Visualisierung des ersten Szenarios.....	144
Abbildung 62: Distanzverläufe der Dauertests mit steigender Interpolationsdynamik.....	146
Abbildung 63: Positions- und Distanzverläufe des dritten Szenarios	148
Abbildung 64: Trajektorien des dritten Szenarios	149
Abbildung 65: Bewegungsstudie des dritten Szenarios	150
Abbildung 66: Online-Visualisierung des dritten Szenarios.....	151

Tabellenverzeichnis

Tabelle 1: Klassifikation von Bahnplanungsproblemen.....	30
Tabelle 2: Qualitative Bewertung der untersuchten Bahnplanungsansätze	50
Tabelle 3: Betragkleinster Eigenwert λ_1 und Stellzeit T_s für einige Diskretisierungen n ...	66
Tabelle 4: Übertragungsfunktionen der Stützstellenkoordinaten y_i	68
Tabelle 5: Einfluss der Sicherheits- und Dynamikparameter K und T	72
Tabelle 6: Stellzeiten T_s für einige Parametrierungen T und K und Diskretisierungen n....	76
Tabelle 7: Abhängigkeiten zwischen Parametern und Bahneigenschaften der KFM.....	83
Tabelle 8: Gewichtung verschiedener Rechenoperation gemäß Livermore Loops.....	93
Tabelle 9: Plattformen zur Ermittlung des Rechenaufwands von KFM und EBM	95
Tabelle 10: Rechenaufwand von KFM und EBM auf verschiedenen Plattformen.....	96
Tabelle 11: Für den Reglerentwurf verwendete Parameter für den Crazyflie	138
Tabelle 12: Anzufliegende Positionen des ersten Szenarios	142
Tabelle 13: Zusammenfassung der Ergebnisse aus den Dauertests.....	146
Tabelle 14: Rechenzeiten der KFM bei den Dauertests.....	147
Tabelle 15: Anzufliegende Positionen des dritten Szenarios	148
Tabelle 16: Bahnfehler des dritten Szenarios	149

Abkürzungsverzeichnis

ADS	Automation Device Specification
API	Application Programming Interface
BDF	Backward Differentiation Formula
CAD	Computer-Aided Design
CFM	Curve Shortening Flow Method
CPU	Central Processing Unit
DGL	Differentialgleichung
DLL	Dynamic Link Library
DQ	Differenzenquotient
DRM	Dynamic Roadmap
DWA	Dynamic Window Approach
EBM	Elastic Bands Method
ERM	Elastic Roadmap
ESM	Elastic Strips Method
FLOP	Floating-Point Operation
FPU	Floating-Point Unit
FTS	Fahrerloses Transportsystem
GDWA	Global Dynamic Window Approach
GLS	Gleichungssystem
GPS	Global Positioning System
HMI	Human-Machine Interface
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISG	Industrielle Steuerungstechnik GmbH
ISM	Industrial, Scientific and Medical (Band)
IT	Informationstechnik
KFM	Kurvenflussmethode

KS	Koordinatensystem
LPS	Loco Positioning System
LQ	Linear-quadratische (Optimierung)
LZI	Lineares zeitinvariantes (System)
ND	Nearness Diagram
NRT	Non-Real-Time
OPC-UA	Open Platform Communications Unified Automation
OS	Operating System
PA	Power Amplifier
PFM	Potentialfeldmethode
PID	Proportional-Integral-Derivative (Controller)
PLC	Programmable Logic Controller
PRM	Probabilistic Roadmap
PTP	Point-to-Point (Control)
RAM	Random-Access Memory
RRT	Rapidly-Exploring Random Tree
RT	Real-Time
SP	Service Pack
SPS	Speicherprogrammierbare Steuerung
ST	Structured Text
TCP/IP	Transmission Control Protocol/Internet Protocol
TDoA	Time Difference of Arrival
TSN	Time-Sensitive Networking
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
UWB	Ultra-Wide Band
VAL	Virtual Automation Lab
VFF	Virtual Force Field
VFH	Vector Field Histogram

Kurzfassung

Um für zukünftige Produktionslinien eine möglichst hohe Flexibilität zu ermöglichen, wird für den Kleinteiletransport aktuell der Einsatz unbemannter Flugfahrzeuge (UAVs, engl. unmanned aerial vehicles) untersucht. Gegenüber boden- und liniengebundenen und häufig starren Materialflusseinrichtungen bieten UAVs die Möglichkeit den meist ungenutzten Luftraum innerhalb der Produktionsstätten auszunutzen.

Um dem hohen Kollisionspotenzial, welches dabei durch parallele und unabhängige Flugbewegungen entsteht, zu begegnen, wird im Rahmen dieser Arbeit mit der Kurvenflussmethode (KFM) eine neue Bahnplanungsmethode entworfen.

Auf Basis eines in Echtzeit aktualisierten Umgebungsmodells werden die Flugbahnen durch Aufprägung virtueller Kräfte zur Laufzeit derart an den aktuellen Umgebungszustand angepasst, dass sie stets um statische (z.B. Gebäudeelemente, Versorgungsleitungen) sowie dynamische (z.B. weitere UAVs) Hindernisse herumführen.

Obgleich die in der mobilen Robotik eingesetzte Elastic-Bands-Methode (EBM) ein der KFM ähnliches Funktionsprinzip besitzt, konnten in einer Voruntersuchung einige Schwächen der EBM identifiziert werden. Diese betreffen vorwiegend die begrenzte numerische Stabilität sowie die abstrakte Parametrierung der EBM und werden durch die KFM behoben.

Dazu wird die KFM zunächst als verallgemeinerte Bahnplanungsmethode theoretisch hergeleitet. Eine systemtheoretische Analyse des Bahnverhaltens ermöglicht die Herleitung der Kenngrößen für die Parametrierung der KFM anhand physikalischer Größen. Ein abschließender Vergleich zeigt folgende signifikante Vorteile der KFM gegenüber der EBM: Numerisch stabile Lösbarkeit, analytische Parametrierbarkeit, Skalierungsunabhängigkeit sowie eine etwas höhere Recheneffizienz.

Losgelöst von der EBM werden anschließend weitere Steuerungsfunktionen, welche für die Anwendung der KFM auf die Problemstellung (parallele Bewegungsführung mehrerer UAVs) benötigt werden, erläutert. Dazu zählen z.B. Hindernis- und Arbeitsraummodellierung, Initialisierung, Bahninterpolation sowie Start- und Landemanöver.

Abschließend werden mehrere parallel ausgeführte Instanzen der KFM auf jeweils eigenen Industriesteuerungen, welche über eine lokale Cloud vernetzt sind, in Betrieb genommen. Die KFM wird dabei als eigene (kollisionsfreie) Punkt-zu-Punkt-Interpolationsart implementiert, sodass mithilfe von Ablaufprogrammen nacheinander verschiedene Punkte angefliegen werden können, ohne das Kollisionspotenzial bei der Programmierung explizit berücksichtigen zu müssen.

Verschiedene Validierungsszenarien werden zunächst in Form einer Hardware-in-the-Loop-Simulation mit Dynamikmodellen der UAVs und schließlich mit realen UAVs durchgeführt.

Abstract

To achieve a high degree of flexibility for future production lines, the usage of unmanned aerial vehicles (UAVs) for transportation of small parts is currently being widely studied. In contrast to fixed ground- and line-bound material flow devices, UAVs provide the possibility to exploit the frequently unused airspace inside a manufacturing plant.

To counter the potentially high risk of collision that arises due to parallel and independent flight movements of such UAVs, a new path planning method, the so-called Curve Shortening Flow Method (CFM), is presented as the scope of this work.

Using artificial forces that are derived from a real-time model of the environment, the flight paths are constantly adjusted at runtime to avoid static (e.g. parts of the building or supply lines) and dynamic (e.g. other UAVs) obstacles.

Although the widespread Elastic Bands Method (EBM) has a similar working principle, several shortcomings of the EBM were identified in a preliminary study. These shortcomings are mainly related to the EBM's limited stability and its abstract parameterization. The CFM provides a solution to these shortcomings.

The CFM is derived as a generalized path planning method. A system-theoretical oriented analysis of the path behavior allows for the parameterization of the CFM by means of physical quantities. A conclusive comparison points out the following significant advantages of the CFM over the EBM: Numerically stable solvability, analytical parameterability, independency from space-scaling, and a slightly better computational efficiency.

Additional control functions that are necessary to apply the CFM to the problem statement (parallel flight movements of multiple UAVs in a shared air space) are also addressed separately from the EBM. These include, among other things, modelling of obstacles and workspace, path initialization, path interpolation as well as take-off and landing maneuvers.

Finally, several instances of the CFM are executed in parallel on individual industrial controls that are linked via a local cloud. The CFM is implemented as a new type of (collision-

free) point-to-point interpolation and allows the sequential controller programs to fly the UAVs to consecutive goals without having to explicitly consider the potential for collisions during programming.

Different validation scenarios are performed, namely in the form of a hardware-in-the-loop simulation using dynamic models of the UAVs and finally using real UAVs.

1 Einleitung

1.1 Einführung

Im Zuge der vierten industriellen Revolution verändern sich die Ansprüche an bestehende und zukünftige Produktionsanlagen und -prozesse wesentlich. Das übergeordnete Ziel dieser Entwicklung ist es, die gesamte Wertschöpfungskette derart zu optimieren und zu flexibilisieren, dass dem von zunehmender Individualisierung der Produkte geprägten Kundenwunsch in wirtschaftlicher Art und Weise begegnet werden kann (Dorst 2015; Westkämper et al. 2008). Beispiele für derart personalisierte Produkte lassen sich sowohl bei Konsumprodukten als auch bei industriellen Produkten finden.

Die dafür benötigten Produktions- und Materialflussanlagen werden zu cyberphysischen Systemen (CPS) ausgebaut, welche alle produktionsrelevanten Daten in einem Netzwerk oder einer zentralen Cloud zur Verfügung stellen (Bettenhausen et al. 2013). Diese enge Verschmelzung mit eventuell bereits vorhandener Kommunikations- und IT-Infrastruktur zum sogenannten Internet der Dinge (IdD) bildet die Grundlage einer weitestgehend selbstorganisierten Produktion, welche ein Hauptmerkmal von Industrie 4.0 darstellt.

Um diesen hohen Individualisierungsgrad zu erreichen, sind starre Fertigungslinien für zukünftige Produktionen nicht mehr geeignet. Stattdessen wächst in der Intralogistik die Forderung nach einem flexiblen Teilefluss, um beliebige Abfolgen verschiedener Produktionsschritte zu ermöglichen. Gleichzeitig sollen auch Standort und Anzahl der an der Produktion beteiligten Bearbeitungsmaschinen bedarfsabhängig gestaltet werden können, ohne eine zeitaufwendige und mit einem Stillstand einhergehende manuelle Neuinbetriebnahme durchführen zu müssen. Um diese hohe Flexibilität realisieren zu können, bedarf es eines ebenso flexiblen Förder- oder Materialflusssystems für die zu transportierenden Stückgüter.

Horizontale Bandapplikationen mit Rollen-, Band- oder Kettenförderern gehören zu den sogenannten flurgebundenen Stetigförderern (Günthner et al. 2006) und sind klassischerweise für einen kontinuierlichen Teilefluss konzipiert. Mit ihnen kann die geforderte Flexibilität nur unter dem hohen Aufwand einer häufigen physischen Umorganisation realisiert

werden, weswegen diese Form des Stückguttransports zukünftig an ihre Grenzen stoßen wird (Felix 2017). Gleichzeitig tritt bei diesen eindimensional verketteten Fördersystemen das Problem auf, dass der Ausfall einer einzelnen Maschine oder Fördereinrichtung zwangsläufig einen Stillstand der gesamten Produktionslinie zur Folge hat.

Gegenstand aktueller Forschung sind ebenfalls zweidimensionale Netze von Förderbändern, an deren Knotenpunkten Module mit schwenkbaren und individuell angetriebenen Rollen mehrere Teile individuell ausrichten und umlenken können. Diese sogenannte „entscheidungsfähige Fördermatrix“ kann einen automatisierten und flexiblen Teiletransport zu beliebigen Bearbeitungsstationen realisieren (Stichweh 2015; Thesseling et al. 2015). Solche Systeme bilden damit den Übergang zu den Unstetigförderern, da der Teilefluss hier nicht mehr kontinuierlich, sondern bedarfsabhängig erfolgt.

Allerdings sind diese Systeme nach wie vor an eine durchgehende mechanische Förderverbindung zwischen den Bearbeitungsstationen gebunden, sodass zum einen weitläufig verteilte Bearbeitungsstationen aufwändig zu verbinden sind und zum anderen die zweidimensionale Ausdehnung der Förderwege sehr viel Platz beansprucht.

Als Alternative zum Teiletransport mittels einer stationären Fördereinrichtung werden seit einiger Zeit auch automatisierte Fahrzeugsysteme eingesetzt. Diese fahrerlosen Transportsysteme (FTS) können sich, beauftragt von einer übergeordneten Leitsteuerung, autonom innerhalb festgelegter Korridore in der Maschinenhalle bewegen und Lasten bis zu mehreren Tonnen manövrieren. Die Navigation erfolgt dabei meist anhand optischer, magnetischer oder induktiver Spurführung (Liniennavigation) oder einzelner Stützpunkte (Rasternavigation) und entsprechender Sensorik. Vereinzelt findet auch eine Navigation durch Selbstlokalisierung anhand von Reflektormarken (Lasernavigation) auf virtuellen Leitlinien Anwendung (Bubeck et al. 2014). Da trotz relativ hoher Fahrzeugkosten die zusätzlichen Investitionskosten pro Wegstrecke sehr günstig sind (Köther 2018), eignen sich FTS insbesondere für die Überbrückung größerer Entfernungen.

Obwohl sie eine hohe Flexibilität in der Wegführung und ein hohes Integrationspotential in eine veränderliche Produktionsumgebung aufweisen, sind dennoch signifikante Nachteile der FTS erkennbar (Günter 2014):

- Da FTS meist in Arbeitsumgebungen mit Personenverkehr eingesetzt werden, wird nur mit sehr geringen Geschwindigkeiten (entsprechend der Gehgeschwindigkeit meist 1 m/s) verfahren.
- Die Bindung der FTS an die vorgegebene Linienführung ohne Ausweich- oder Alternativrouten kann Rückstaus und Umwege verursachen.
- Für eine Neukonfigurierung oder Erweiterung der Fahrwege müssen meist aufwendige Bodeninstallationen durchgeführt werden.
- Aufgrund der hohen Anforderungen an Bodenebenheit und -beschaffenheit ist der Betrieb von FTS meist auf einzelne Stockwerke und Gebäude begrenzt.

Um diese Nachteile zu umgehen, wird in dieser Arbeit der Einsatz von unbemannten Flugfahrzeugen (UAVs, engl. unmanned aerial vehicles) für den Kleinteiletransport zukünftiger Produktionsanlagen vorgeschlagen. Durch UAVs lässt sich im meist ungenutzten Luftraum innerhalb der Produktionsstätten eine weitere räumliche Dimension für den flexiblen Materialfluss in einer personenfreien Umgebung erschließen. Zusammen mit der vergleichsweise kostengünstigen Skalierbarkeit in der Anzahl der eingesetzten UAVs und dem geringen mechanischen Zusatzaufwand entsteht so ein hochdynamisches Transportsystem, welches mit entsprechender Steuerungstechnik beliebige und schnell umsetzbare Kombinationsmöglichkeiten im Produktionsprozess ermöglicht. So sind variable Abfolgen und Standorte einzelner Bearbeitungsstationen ebenso möglich wie die Überbrückung von Stockwerks- und Gebäudegrenzen.

Die Verwendung des Begriffs UAV wird in dieser Arbeit auf Drehflügler (engl. rotary wing aircrafts) wie z.B. Quadro- oder verallgemeinert Multicopter eingeschränkt. Entgegen den Starrflüglern (engl. fixed wing aircrafts) erlauben diese durch ihre Fähigkeit zu Vertikalstarts/-landungen und Schwebeflug eine hohe Manövrierbarkeit und kommen daher nur in dieser Bauform für den Einsatz in engen Produktionshallen in Frage.

Für andere industrielle Bereiche wie beispielsweise Vermessungs- und Inspektionsaufgaben, visuelles Monitoring oder die Messdatenaufzeichnung werden UAVs bereits eingesetzt (Landrock et al. 2018), wobei die Steuerung in der Regel noch manuell erfolgt. Im Bereich der Logistik wurden autonom fliegende UAVs bereits für Zustelldienste im Freien

erprobt (Stoller 2014; Lingner 2016; McCauley 2016). Typische industrielle UAVs mit einem Durchmesser von unter einem Meter können dabei Nutzlasten von ca. 1 bis 2 kg befördern (Landrock et al. 2018).

Obgleich im Zuge von Industrie 4.0 UAVs ein ebensolches Potential zur Integration in den Produktionsablauf bescheinigt wird, wie sie Industrieroboter heute vorweisen können, ist eine Integration in die industrielle Intralogistik bis dato allerdings noch nicht gelungen (Landrock et al. 2018).

1.2 Anforderungen

Durch die Integration von UAVs als flexibles Transportsystem in den Produktionsprozess entstehen neue Anforderungen an das zu realisierende Gesamtsystem. Im Folgenden soll auf die signifikantesten Anforderungen eingegangen werden.

Online-Bahnplanung:

Hält man am für FTS gängigen Konzept der Linienführung entlang vorgegebener Bahnen ohne aufwendig zu realisierende Alternativrouten oder Ausweichstrategien fest, birgt die Forderung nach individuellen Bearbeitungsreihenfolgen ein hohes Risiko für Rückstaus oder gar Deadlocks. Diese können in der Regel nur durch manuellen Eingriff aufgelöst werden. Um den zur Verfügung stehenden Luftraum durch die UAVs möglichst parallelisiert, dynamisch und ohne dieses Risiko auszunutzen, soll dieser daher nicht liniengebunden sondern als räumliches Kontinuum genutzt werden.

Zusammen mit der sich stetig verändernden Produktionsumgebung aus variablen Bearbeitungsstationen, baulichen Gegebenheiten der Produktionsstätte und den UAVs selbst ergeben sich parallele Teilebewegungen im Raum, die sich nur noch äußerst schwer vorhersagen lassen. Bezieht man den Menschen als vorab unbekanntes Hindernis mit ein, ist eine Vorhersage gar unmöglich. Dieses nunmehr dreidimensionale Kollisionspotenzial gilt es durch eine geeignete Bahnplanung zu beherrschen.

Es ist offensichtlich, dass es nicht mehr zielführend sein kann, die benötigten Flugbewegungen a priori offline zu programmieren. Stattdessen bedarf es eines Bahnplanungsalgo-

rithmus, welcher in der Lage ist, die kollisionsfreien Flugbahnen online anhand des aktuellen Zustandes der Produktionsumgebung zu planen und anschließend kontinuierlich an Veränderungen anzupassen.

Gleichzeitig muss der Algorithmus für den industriellen Einsatz unabhängig von äußeren Einflüssen und unter Berücksichtigung der durch numerische Verfahren und Maschinengenauigkeit bedingten Näherungs- und Rundungsfehler eines digitalen Rechensystems stets eine robuste Bahn generieren. Dies ist für UAVs umso notwendiger, da gegenüber flurgebundenen Fahrzeugen eine Notabschaltung oder passives Verhalten infolge einer invaliden Lösung nicht zu einem dauerhaft sicheren Halt, sondern zu einem Absturz oder spätestens bei drohender Akkuerschöpfung zu einer unkontrollierten Notlandung führen kann.

Des Weiteren muss die Bahnplanungsmethode für jedes einzelne UAV als autonome Funktionseinheit ohne die Notwendigkeit übergeordneter Steuerungsfunktionalität (Leitsteuerung) implementierbar und ausführbar sein. Wechselwirkungen mit der Umwelt und anderen UAVs erfolgen dabei über geeignete Schnittstellen sowie ein zur Verfügung stehendes digitales Abbild der Produktionsumgebung. Diese Modularität ermöglicht eine Skalierbarkeit des Transportsystems für eine beliebige Anzahl von UAVs bei konstanter Verteilung der Rechenlast. Gleichzeitig erfüllt die Methode dadurch eines der Hauptmerkmale von Industrie 4.0, Entscheidungen möglichst dezentral in selbstorganisierten und reaktionsfähigen Einheiten des Produktionssystems zu treffen.

Zusätzlich muss die Ein- und Auskopplung von UAVs in das Produktionssystem zur Laufzeit möglich sein, um eine kontinuierlich bedarfsabhängige Auslastung ohne Umrüstzeiten zu ermöglichen.

Netzwerkintegration:

Die Vernetzung einzelner „intelligenter“ Produktionskomponenten zu einem Industrie-4.0-Netzwerk wird aktuell noch über anwenderorientierte Internetprotokolle (z.B. TCP/IP) realisiert, welche entgegen industriellen Steuerungsnetzwerken (Feldbussen) nicht für den Echtzeitbetrieb konzipiert sind. Daher ist eine weitere Anforderung, dass die UAVs bzw.

ihre Steuerungen in ein Netzwerk integriert werden, welches in der Lage ist, das digitale Abbild der Produktionsumgebung echtzeitnah und ausfallsicher zur Verfügung zu stellen.

Indoorlokalisierung:

Damit die UAVs die Flugbewegungen des Bahnplanungsalgorithmus ausführen können, ist eine Positionsregelung notwendig. Da UAVs, abgesehen von GPS-geführten Flügen im Freien, standardmäßig lediglich über eine Regelung der Fluglage mit Inertialsensoren verfügen, muss eine geeignete Methode zur absoluten Indoorpositionsbestimmung realisiert werden. Aufgrund des vergleichsweise geringen Arbeitsraums, welcher in der Produktionshalle zur Verfügung steht, muss diese Methode gegenüber GPS eine sehr viel höhere Genauigkeit aufweisen.

Zusätzlich kann bei der Übergabe von und zu den Bearbeitungsstationen gegenüber der eigentlichen Flugbewegung eine noch höhere Präzision erforderlich werden, sodass hier auf eine andere lokal genauere Positionsbestimmungsmethode oder gar auf eine mechanische Führung auszuweichen ist.

Energiemanagement:

Für einen dauerhaften Betrieb der UAVs ist ein geeignetes Energiemanagementkonzept notwendig. Im Akkubetrieb stehen kurze Flug- und lange Ladezeiten der UAVs im direkten Widerspruch zum erstrebenswerten 24/7-Dauerbetrieb der Produktionsanlage. Daher muss untersucht werden, inwieweit die Stillstandzeiten der UAVs effizient für Ladevorgänge genutzt werden können. Gleichzeitig wird ein Sicherheitskonzept benötigt, wie bei drohender Akkuerschöpfung zu reagieren ist.

1.3 Zielsetzung

Ziel dieser Arbeit ist der Entwurf einer industrietauglichen Bahnplanungsmethode für den vernetzten Betrieb von mehreren autonomen UAVs in der Intralogistik. Aus den in Kap. 1.2 aufgezeigten Anforderungen an ein solches Steuerungskonzept soll sich diese Arbeit insbesondere dem damit verbundenen zentralen Thema der kollisionsfreien Online-

Bahnplanung für UAVs widmen. Dies umfasst sowohl die theoretische Herleitung der Methode als auch deren algorithmische Umsetzung und prototypische Realisierung.

Die daran angrenzenden Themenbereiche Netzwerkintegration und Indoorlokalisierung sollen nicht zum Kern dieser Arbeit gehören, sind für eine abschließende Realisierung aber dennoch notwendig. Für deren Umsetzung wird daher auf den aktuellen Stand der Forschung und Technik sowie auf Teillösungen aus weiteren parallelen Forschungsaktivitäten zurückgegriffen.

Die Handhabung der zu transportierenden Teile sowie deren Übergabe an die Bearbeitungsstationen wird, ebenso wie das für eine erste prototypische Umsetzung noch nicht zwingend erforderliche Energiemanagement, im Rahmen dieser Arbeit nicht behandelt.

1.4 Problemstellung

Aus der Zielsetzung in Kap. 1.3 geht als primäre Problemstellung die Online-Bahnplanung für den vernetzten Betrieb von autonomen UAVs in der Intralogistik hervor. Nachfolgend sollen die Anforderungen an eine solche Bahnplanungsmethode nochmals konkretisiert werden:

- Die Bahnplanungsmethode soll eine kollisionsfreie Bahn für eine sich verändernde Umgebung bestehend aus bewegten (z.B. weitere UAVs oder Menschen) und ruhenden Hindernissen (z.B. Hallenwänden) erzeugen.
- Die Bahnplanungsmethode soll in einem industriellen Umfeld auf industrietypischer Hardware betrieben werden. Aufgrund der Wechselwirkungen mit realen Umgebungsveränderungen ist eine Ausführung der Steuerungssoftware unter harter Echtzeit notwendig. Dies setzt gleichzeitig die zeitdeterministische Implementierbarkeit der Methode voraus.
- Die Bahnplanungsmethode soll möglichst recheneffizient sein. Schnelle Aktualisierungsintervalle verkürzen die Reaktionszeit und erhöhen die Sicherheit. Erstrebenswert ist die Aktualisierung der Bahn in steuerungstypischen Interpolationstakten von wenigen Millisekunden. Damit einhergehende Einbußen bei der

globalen Optimalität der Bahn (z.B. im Sinne einer kürzest möglichen Weglänge oder Ausführungszeit) sollen dabei toleriert werden.

- Die Bahnplanungsmethode soll eine numerisch robuste Bahn garantieren und idealerweise ohne rechenaufwendige Initialisierung auskommen.
- Die Bahnplanungsmethode soll für 3D-Umgebungen geeignet sein. Nur mit räumlichen Bahnen kann der gesamte zur Verfügung stehende Arbeitsraum effizient ausgenutzt werden.
- Die Bahnplanungsmethode muss für beliebig viele UAVs mit jeweils einer eigenen Instanz der Methode skalierbar sein. Auf diese Weise kann eine gleichmäßige Verteilung der Rechenlast sowie eine Gleichberechtigung aller UAVs garantiert werden. Weitere UAVs müssen sich im laufenden Betrieb in das System ein- und ausgliedern lassen. Die kollisionsfreie Bahnplanung und -ausführung soll autonom und ohne Koordination oder Steuerungsfunktionalität einer übergeordneten Leitebene erfolgen. Lediglich die Verwaltung der UAVs, die Verteilung von Informationen über den aktuellen Zustand der Produktionsumgebung sowie die Beauftragung kann von einer übergeordneten Instanz durchgeführt werden.

1.5 Überblick über die Arbeit

In Kap. 2 wird zunächst der für die Problemstellung aus Kap. 1.4 relevante Stand der Forschung und Technik im Bereich der Bahnplanung für Roboter und UAVs dargestellt. Anhand von Defiziten und Stärken wird die Eignung einzelner Methoden für die Lösung der Problemstellung analysiert.

Anschließend werden in Kap. 3 in einer Voruntersuchung die Defizite der aus dem Stand der Technik identifizierten und der Problemlösung am nächsten kommenden Bahnplanungsmethode detailliert analysiert. Diese dienen später als Referenz zur Evaluierung des eigenen Lösungsansatzes.

Dieser Ansatz wird in Kap. 4 als Kurvenflussmethode (KFM) eingeführt. Die Theorie der KFM wird zunächst analytisch hergeleitet und anschließend algorithmisch und numerisch für eine effiziente Ausführung in einem digitalen Rechensystem aufbereitet.

Im Anschluss behandelt Kap. 5 die benötigten Funktionen für die KFM in komplexen Umgebungen und damit die Anwendung der bisher allgemein hergeleiteten KFM auf das eingangs beschriebene Szenario.

Das Kap. 6 behandelt die Implementierung und Realisierung einer Testumgebung für die KFM. Mit dieser wird anhand mehrerer Szenarien die Funktionalität der KFM sowohl virtuell mit simulierten UAVs als auch experimentell mit realen UAVs nachgewiesen.

2 Stand der Forschung und Technik

Bereits seit langer Zeit wird die Forschung auf dem Gebiet der Bahnplanung vorwiegend durch die Robotik vorangetrieben. Dabei werden Roboter unterschiedlichster kinematischer Ausprägung adressiert. Viele klassische Bahnplanungsmethoden lassen sich auf eine Vielzahl von Robotertypen anwenden, da sie das Bahnplanungsproblem unabhängig von einer speziellen Kinematik lediglich für einen stark abstrahierten Bewegungsautomaten formulieren. Dadurch stehen UAVs, welche eine noch vergleichsweise junge Technologie darstellen, ebenfalls eine Vielzahl bereits erprobter Bahnplanungsmethoden zur Verfügung. Diese sollen im Folgenden zusammen mit einigen grundlegenden Konzepten der Bahnplanung nach dem aktuellen Stand der Forschung und Technik und mit Fokus auf die definierte Problemstellung (vgl. Kap. 1.4) vorgestellt werden.

2.1 Einordnung von UAVs in die Robotik

Bei der Klassifizierung von industriell eingesetzten Robotern wird insbesondere zwischen stationären und mobilen Robotern unterschieden.

Stationäre Roboter (z.B. artikulierte oder parallele Manipulatoren) sind fest in ihrer Umgebung verankert. Sie gewinnen ihre Bewegungsfreiheit durch die serielle oder parallele Verbindung einzelner beweglicher Achsen, wodurch ihr Endeffektor, unter Voraussetzung von mindestens sechs Achsen, beliebige Posen, also Vorgabe von Position und Orientierung, innerhalb ihres Arbeitsraums erreichen kann. Sie werden industriell häufig für einzelne wiederkehrende Aufgaben unter vorab bekannten oder kontrollierten Bedingungen eingesetzt und haben prozessbedingt meist hohe Anforderungen an die Bahngenauigkeit. Der Grad der Autonomie bei der Bewegungsausführung ist dabei vergleichsweise gering.

Mobile Roboter hingegen sind ortsungebunden und können sich mittels eines für die Bodenbeschaffenheit geeigneten Antriebs (z.B. gelenkten oder omnidirektionalen Rädern oder Ketten) selbstständig innerhalb ihres meist ebenen Arbeitsraums bewegen. Sie besitzen ein breiter gefächertes Einsatzgebiet und werden neben der industriellen Nutzung (z.B. Transportroboter oder mobile Manipulatoren) ebenfalls als Serviceroboter im kommerziellen (z.B. Inspektionsroboter) oder häuslichen Umfeld (z.B. Staubsaug- oder Ra-

senmäroboter) eingesetzt. Häufig müssen sie dabei autonom und sensorgestützt durch komplexe, unbekannte und sich verändernde Umgebungen navigieren.

Beide Robotertypen sind durch unterschiedliche Eigenschaften und Herausforderungen in der Beschreibung ihrer Kinematik geprägt, was einen direkten Einfluss auf die Komplexität der für die Bahnplanung benötigten Abstandsrechnungen zum Hindernis hat. Während bei stationären Robotern der Zusammenhang zwischen Endeffektor- und Achsstellung sowie die geometrische Beschreibung der Außenhülle mithilfe der sogenannten Vorwärts- und Rückwärtstransformation hergestellt werden muss, können mobile Roboter vereinfacht als freier Starrkörper in der Ebene beschrieben werden. Letztere sind nicht zwingend in der Lage, jede beliebige Pose in Ihrem Arbeitsraum auf direktem Wege zu erreichen. So können sich einige mobile Roboter mit einer fahrzeugähnlichen Front- oder Hecklenkung lediglich auf einer entsprechend dem aktuellen Lenkwinkel gekrümmten Bahn vorwärts und rückwärts bewegen. Diese kinematische Eigenschaft wird als Reeds-Shepps-Car abstrahiert (Reeds et al. 1990). Dadurch können diese Roboter zwar grundsätzlich jede beliebige Pose erreichen, es ist dafür aber eine bestimmte Abfolge mehrerer Fahrmanöver notwendig, was häufig auch als parallele Parkproblem (Batterman 2003) adressiert wird. Diese Problematik tritt dann auf, wenn die Anzahl der Steuereingänge (z.B. Geschwindigkeit und Lenkrate) kleiner als die Anzahl der posebeschreibenden Freiheitsgrade (z.B. zwei ebene Positionskoordinaten und die Orientierung) ist (Laumond et al. 1998). Mobile Roboter, welche beispielsweise durch Verwendung omnidirektionaler Räder fähig sind, ihre Orientierung ohne gleichzeitige Translation zu ändern, sind ebenso wie stationäre Roboter nicht von dieser Einschränkung betroffen. In der Mechanik wird diese Eigenschaft als Holonomität beschrieben (Pin et al. 1994).

UAVs gehören zu den mobilen Robotern. Vereinfacht bilden sie ebenfalls einen einzelnen frei beweglichen Starrkörper in ihrem Arbeitsraum und ihre Umgebung ist meist von signifikanten Unsicherheiten geprägt. Allerdings erhöhen UAVs die Komplexität des Bahnplanungsproblems, da sie den flurgebundenen Arbeitsraum mobiler Roboter um eine Dimension zum dreidimensionalen Luftraum ausdehnen. Dies erhöht die Anzahl der Freiheitsgrade (zur Beschreibung des kinematischen Verhaltens) von drei auf sechs und die

Anzahl der Zustände (zur Beschreibung des dynamischen Verhaltens) von sechs auf zwölf.

Als weitere Analogie lassen sich UAVs entsprechend ihrer Bauform ebenfalls in holonome und nichtholonome Kinematiken unterteilen. Während holonome Drehflügler fähig sind, aus dem Schwebeflug heraus jede beliebige benachbarte Pose anzufliegen, sind nichtholonome Starrflügler an die Einhaltung einer minimalen Fluggeschwindigkeit (Strömungsabrissgeschwindigkeit) gebunden. Damit spezialisieren letztere das zugrundeliegende Kinematikmodell vom Reeds-Shepps-Car zum Dubins Car (Dubins 1957), welches lediglich die Bewegung in eine Richtung (vorwärts) zulässt. Dies erschwert die Bahnplanung zusätzlich, da kollisionsfreie Posen unter Umständen unerreichbar werden.

Da sich diese Arbeit wie in Kap. 1.1 beschrieben lediglich auf den Einsatz von Drehflüglern als UAVs beschränkt, soll der Fokus im Folgenden insbesondere auf Bahnplanungsmethoden liegen, welche für holonome mobile Roboter und damit ebenfalls für holonome UAVs geeignet sind.

2.2 Das Bahnplanungsproblem

Verallgemeinert lässt sich das Bahnplanungsproblem je nach gewähltem Koordinatenraum, in welchem das Problem formuliert wird, als die Überführung eines Roboters in eine neue Pose (euklidischer Raum), Konfiguration (Konfigurationsraum) oder einen neuen Zustand (Zustandsraum) beschreiben. Die Überführung erfolgt entlang einer Bahn und muss insbesondere den beiden Kriterien, kollisionsfrei und ausführbar (z.B. fahrbar oder fliegbare) zu sein, genügen. Um Kollisionsfreiheit zu gewährleisten, müssen ruhende und bewegte Hindernisse berücksichtigt werden. Ausführbarkeit kann durch Einbeziehung kinematischer und dynamischer Randbedingungen, welche die Manövrierbarkeit des Roboters beschreiben, erreicht werden (Tsourdos et al. 2011).

Des Weiteren wird unter Einhaltung dieser beiden Kriterien in der Regel nicht eine beliebige Bahn gesucht, sondern diejenige, welche ein Optimum hinsichtlich einer einzelnen Größe (z.B. Weglänge, Flugdauer oder Energieverbrauch) oder ein Paretooptimum hin-

sichtlich mehrerer sich widersprechender Größen (z.B. kürzester Weg bei gleichzeitiger Maximierung des Abstands zum Hindernis) darstellt.

In diesem Zusammenhang wird strikt zwischen Bahn- und Trajektorienplanung unterschieden. Eine Bahn ist die geometrische Beschreibung einer kollisionsfreien Verbindung zwischen Start- und Zielpose oder -konfiguration und berücksichtigt damit lediglich kinematische Eigenschaften des Roboters. Diese Bahn kann nach einer kontinuierlichen (z.B. Bogenlänge) oder diskreten Größe (z.B. Indizierung bei Abfolge einzelner Wegpunkte) parametrisiert sein.

Eine kontinuierliche und nach der Zeit parametrisierte Bahn wird als Trajektorie bezeichnet. Deren numerische oder analytische Ableitungen enthalten damit zusätzliche Informationen, wie die Bahn bezüglich Geschwindigkeit, Beschleunigung, etc. auszuführen ist. Während eine Bahn also lediglich die Lösung eines geometrischen oder kinematischen Problems darstellt, beschreibt eine Trajektorie die Lösung eines dynamischen Problems unter Berücksichtigung dynamischer Eigenschaften des Roboters (Shiller 2015; Yang et al. 2015).

Die Berücksichtigung dynamischer Eigenschaften bei der Bahnplanung stellt für sich schnell bewegende UAVs, insbesondere für Starrflügler, eine notwendige Bedingung dar. Die typischerweise langsameren Drehflügler sind hingegen, unter Voraussetzung einer entsprechenden Positionsregelung, prinzipiell in der Lage zu schweben und damit auch beliebig stark gekrümmte oder gar eckige Bahnen abzufliegen. So können die Anforderungen an die Berücksichtigung dynamischer Eigenschaften bei der Bahnplanung für Drehflügler oftmals gelockert werden (Filippis et al. 2011).

2.3 Umgebungs- und Robotermodell

Jedes Bahnplanungsproblem setzt voraus, dass ein geometrisches Abbild von Hindernissen und Roboter zur Verfügung steht. Hindernisse stellen physikalische Objekte im Arbeitsraum des Roboters dar, mit welchen dieser bei der Bewegungsausführung kollidieren kann. Die Ausdehnung des Arbeitsraums ist bei stationären Robotern durch Zell- und Achsgrenzen und bei mobilen Robotern durch Gebäude- oder Raumgrenzen sowie die

Kommunikations- und Energiereichweite begrenzt. Digitale Abbilder der Hindernisse werden aus sensorbasierten Punktwolken abgeleitet (z.B. bei Umgebungsdetektion mobiler Roboter) oder auch direkt aus CAD Daten (z.B. bei statischer und kontrollierter Produktionsumgebung, welche häufig bei Fertigungszellen mit stationären Robotern gegeben ist) generiert. Um für die Bahnplanung in einheitlicher und performant weiterzuverarbeitender Form bereitzustehen, werden die Rohdaten als aufbereitete Datenstruktur übergeben. Dabei kommen diskrete (z.B. Zellgitter oder -baum), umrissbeschreibende (z.B. constructive solid geometry, boundary representation) und approximierende (z.B. Polygonnäherung) Repräsentationen zum Einsatz (Hwang et al. 1992).

Je nachdem ob bei der Bahnplanung die Dynamik des Roboters berücksichtigt werden soll, sind unterschiedliche Modelle des Roboters erforderlich. Für eine rein geometrische Bahnplanung ist es ausreichend den Roboter mithilfe seiner Konfiguration zu beschreiben. Die Konfiguration besteht dabei aus einem Satz unabhängiger Parameter, welcher die geometrische Form des Roboters im für die Aufgabe relevanten euklidischen (physikalischen) Arbeitsraum eindeutig beschreibt. Für ein flurgebundenes und als frei beweglicher Starrkörper modelliertes FTS kann diese Konfiguration beispielsweise aus zwei kartesischen Positionen und der Orientierung in der Ebene bestehen. Die Pose eines typischen 6-Achsknickarmroboters im Raum lässt sich hingegen über die sechs Achswinkel eindeutig definieren.

Obgleich ein als Starrkörper im dreidimensionalen euklidischen Raum modelliertes UAV grundsätzlich sechs Freiheitsgrade besitzt, sind diese für einen Drehflügler nicht unabhängig, da Roll- und Nickwinkel ungleich Null im Allgemeinen zu einer Translation (Geschwindigkeit) führen. Für eine rein kinematische Betrachtung ist allerdings lediglich der statische Schwebezustand, bei welchem Roll- und Nickwinkel stets Null betragen, relevant. Dadurch kann die Anzahl der Konfigurationsparameter auf drei kartesische Positionen und eine Orientierung um die Vertikalachse (Gierwinkel) reduziert werden.

Soll bei der Bahnplanung zusätzlich die Dynamik berücksichtigt werden, erfolgt die Beschreibung mobiler Roboter oder UAVs durch deren Zustände auf Basis von Bewegungsgleichungen für einen Starrkörper. Dies erweitert den Problemraum um die zeitlichen

Ableitungen der Konfigurationsparameter zum Zustandsraum mit bis zu zwölf Dimensionen (Goerzen et al. 2010).

Für „grobe“ Bahnplanungsprobleme in Umgebungen, welche nur dünn mit Hindernissen durchsetzt sind, reicht oftmals eine umhüllende Näherung mit geometrischen Primitiven zur Abbildung des Roboters sowie der Hindernisse aus. Grobe Bahnplanungsprobleme zeichnen sich dadurch aus, dass der für die Bahn zur Verfügung stehende freie Raum sehr viel größer als die Ausmaße der beteiligten Roboter einschließlich deren Positioniergenauigkeit und der Hindernisse ist. Dadurch können Kollisionen aufgrund von Regelungsfehlern oder äußeren Störgrößen ausgeschlossen werden (Hwang et al. 1992).

Da UAVs gemessen an ihrer eigenen Größe in vielfach größeren Lufträumen manövrieren, werden Unsicherheiten bei ihrer Positionierung und andere Störgrößen (z.B. Fehler bei der Lokalisierung oder Luftströmungen) häufig durch konservativ ausgelegte Sicherheitskorridore kompensiert. Dadurch kann auch der Detailgrad der geometrischen Approximation von UAVs und Hindernissen sehr viel gröber als bei einem stationären Roboter ausfallen. Häufig werden UAVs lediglich als umhüllende Kugel idealisiert (Goerzen et al. 2010).

2.4 Koordinatenraum

Klassische Bahnplanungsmethoden überführen das Bahnplanungsproblem vom euklidischen Raum, also dem Raum, in welchem Roboter und Hindernisse physikalisch existieren und Bewegungen (Translationen und Rotationen) durchführen können, in einen abstrahierten Koordinatenraum, welcher die Reduktion des Robotermodells zu einem einzelnen Punkt erlaubt. Auf diese Weise können viele Bahnplanungsmethoden entkoppelt von einer spezifischen Roboterkinematik formuliert werden.

Für rein geometrische Probleme ist dies der Konfigurationsraum (Lozano-Perez 1990). Dieser wird von allen Konfigurationsparametern des Roboters aufgespannt und erlaubt es, die geometrische Posenbeschreibung einer beliebigen Kinematik auf einen einzelnen Punkt zu reduzieren. Hindernisse entsprechen darin Regionen aller Konfigurationen, die im euklidischen Raum zu einer Kollision führen würden. Kann der Roboter im euklidischen

Raum als Punkt modelliert werden, so ist der Konfigurationsraum äquivalent zum euklidischen Raum (Latombe 1991).

Die Hauptschwierigkeit dieses Konzepts besteht in der Abbildung von Hindernissen aus dem euklidischen in den Konfigurationsraum, mit dessen häufig hoher Anzahl an Dimensionen die Komplexität des gesamten Bahnplanungsproblems exponentiell wächst (Wise et al. 2000). Diese Transformation wird entweder geometrisch durch explizite Übertragung der tatsächlichen Hindernisumrisse durchgeführt oder es werden lediglich einzelne Zellen des gitterförmig diskretisierten Konfigurationsraums auf Kollisionsfreiheit überprüft. Die Eignung einzelner Methoden ist stark von der Roboterkinematik sowie Hindernisform abhängig. Insbesondere Methoden, welche die Hindernisse geometrisch übertragen, sind häufig nur bei vier oder weniger Konfigurationsparametern anwendbar, während diskrete Methoden einen hohen Rechenaufwand erfordern. Zusammenstellungen gängiger Methoden zur Transformation von Hindernissen in den Konfigurationsraum wurden in verschiedenen Übersichtsstudien veröffentlicht (Hwang et al. 1992; Wise et al. 2000).

Für Roboter, welche als ein einzelner konvexer und nicht rotierender Starrkörper abgebildet werden, sind die Achsen des Konfigurationsraums äquivalent zu denen des euklidischen Raumes. In diesem Fall kann die Konstruktion der Hindernisse im Konfigurationsraum sehr effizient mithilfe der Minowskisumme erfolgen (Varadhan et al. 2004). Dabei entspricht der Konfigurationsraum den um die Robotergeometrie erweiterten Hindernissen (Latombe 1991). Für ein als umhüllende Kugel modelliertes UAV entspricht dies einer Aufweitung der Hindernisse um den Kugelradius (Goerzen et al. 2010; Siciliano et al. 2009).

Viele Bahnplanungsmethoden im Konfigurationsraum sind lediglich für holonome Roboter anwendbar, da die Verbindung einzelner Konfigurationen häufig rein geometrisch ohne Berücksichtigung der Erreichbarkeit für nichtholonome Kinematiken erfolgt (Laumond et al. 1998).

Sollen bei der Bahnplanung zusätzlich dynamische Eigenschaften des Roboters berücksichtigt werden, kann der Konfigurationsraum um die ersten Ableitungen der Freiheitsgrade zum Zustandsraum erweitert werden. Damit verdoppelt sich die Dimension des

Koordinatenraums (LaValle et al. 2001; Frazzoli et al. 2002) und das Bahnplanungsproblem besteht nun aus der Suche nach einer Verbindung von einem Anfangs- zu einem Endzustand des Roboters. Dabei können Zustände nicht beliebig verbunden werden, sondern es muss stets sichergestellt sein, dass ein Steuereingang existiert, welcher einer Überführung des Roboters in den neuen Zustand entspricht (Shkolnik et al. 2009). Hinderisregionen im Zustandsraum beinhalten dabei nicht nur Zustände, welche eine Kollision repräsentieren, sondern zusätzlich diejenigen Zustände, welche eine Kollision bereits unausweichlich machen (Goerzen et al. 2010; Kavraki et al. 2008).

2.5 Einordnung der Problemstellung

Bevor auf einzelne Bahnplanungsmethoden eingegangen wird, soll die Problemstellung anhand der in den Anforderungen (vgl. Kap. 1.4) formulierten veränderlichen Umgebung und der sich daraus ergebenden Hindernischarakteristik einem konkreten Bahnplanungsproblemfall zugeordnet werden.

In Kap. 2.2 wurde bereits die Unterscheidung nach ruhenden und bewegten (bzw. verallgemeinert veränderlichen) Hindernissen getroffen. Sind Hindernisse während der Ausführung der Bahn veränderlich (z.B. in Position, Form oder Größe), handelt es sich um ein zeitvariantes, ansonsten um ein zeitinvariantes Bahnplanungsproblem. Stehen darüber hinaus alle Informationen zu den Hindernissen bereits a priori, also vor Ausführung der Bahn, zur Verfügung, handelt es sich um ein statisches, ansonsten um ein dynamisches Bahnplanungsproblem (Hwang et al. 1992). Somit werden die in Tabelle 1 dargestellten Fälle für Bahnplanungsprobleme unterschieden (Fujimura 2012; Rastgoo et al. 2014).

-	zeitinvariant	zeitvariant
statisch	Fall (1)	Fall (3)
dynamisch	Fall (2)	Fall (4)

Tabelle 1: Klassifikation von Bahnplanungsproblemen

Anhand dieser Klassifizierung kann ein dynamisches Bahnplanungsproblem sowohl davon geprägt sein, dass ein ruhendes oder bewegtes Hindernis erst bei der Ausführung der Bahn erkannt wird (z.B. bei sensorbasierter Hindernisdetektion) als auch davon, dass le-

diglich die Bewegung eines vorab bekannten Hindernisses nicht vorhersehbar ist (z.B. Mensch im Arbeitsraum). Die Kenntnis über den aktuellen Zustand der Umgebung kann somit zudem vollständig oder lediglich partiell sein.

In der Problemstellung (vgl. Kap. 1.4) werden Hindernisse primär in Form weiterer (bewegter) UAVs adressiert. Obgleich die zeitinvarianten Fälle (1) und (2) zu den in der Robotik am häufigsten auftretenden gehören, scheiden Methoden für diese Fälle damit aus. Auf eine detaillierte Beschreibung der ausschließlich für diese Fälle relevanten Methoden wie beispielsweise die Roadmaperzeugung mittels Sicht- und generalisierter Voronoi-Graphen (O'Dunlaing et al. 1985) oder exakter und approximierender Zellzerlegung soll daher verzichtet werden. Die Literatur liefert eine ausführliche Behandlung dieser Methoden sowohl im Kontext der Robotik (Shiller 2015; Hwang et al. 1992; Latombe 1991; Siciliano et al. 2009; LaValle 2006) als auch im Kontext bereits für UAVs adaptierter Methoden (Yang et al. 2015; Goerzen et al. 2010; Radmanesh et al. 2018).

Ob die zeitinvariante Problemstellung als a priori planbarer statischer Fall (3) oder lediglich als zur Laufzeit planbarer dynamischer Fall (4) aufgefasst werden kann, ist eng mit der Vorhersagbarkeit der Bewegungen der Hindernisse bzw. UAVs verbunden. Durch die Forderung nach bedarfsabhängiger Ein- und Auskopplung sowie Beauftragung einzelner Instanzen der Methode ist die Prognose zukünftiger Flugbewegungen nur schwer zu realisieren. Soll zudem auf unerwartete Ereignisse, welche durch die Methode selbst nicht beeinflusst werden können (z.B. Berücksichtigung eines Menschen im Arbeitsraum), reagiert werden können, ist die Vorhersage gar unmöglich.

Die Problemstellung wird daher als rein dynamisches Bahnplanungsproblem nach Fall (4) betrachtet und muss durch eine wechselwirkende Anpassung der Bahnen der einzelnen UAVs zur Laufzeit online gelöst werden. Auf Methoden und Konzepte für die Offline-Bahnplanung mit zeitvarianten Hindernissen nach Fall (3) wie die Entkopplung von Bahn- und Geschwindigkeitsplanung (Kant et al. 1988) oder die Planung im Geschwindigkeits-Zeit-Raum (Fiorini et al. 1998), im Konfigurations-Zeit-Raum (Erdmann et al. 1987) sowie im Zustands-Zeit-Raum (Fraichard 1993) wird hier daher nicht näher eingegangen.

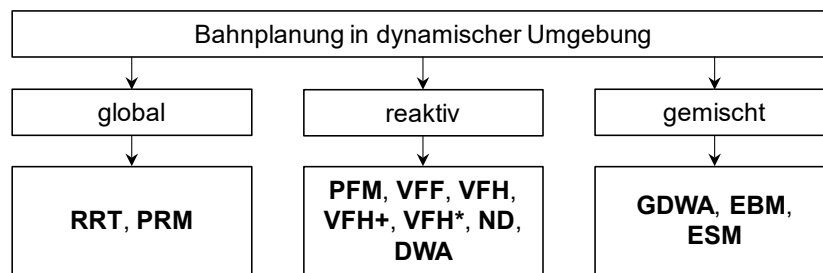
2.6 Bahnplanung in dynamischer Umgebung

Statische Bahnplanungsprobleme erlauben eine vollständige und von der Ausführung entkoppelte Offlineplanung der Roboterbewegung. Diese Bedingungen sind häufig bei stationären Robotern, welche im industriellen Umfeld wiederkehrende Aufgaben unter kontrollierten Bedingungen durchführen, gegeben (Shiller 2015). Dynamische Probleme sind vorwiegend im Bereich mobiler Robotik anzutreffen. Sie stellen den komplexesten Fall des Bahnplanungsproblems dar und können ausschließlich online gelöst werden, da die Bahn kontinuierlich an den sich während der Ausführung unvorhersehbar verändernden oder nur partiell bekannten Umgebungszustand angepasst werden muss (van den Berg 2007). Dabei besteht ein generelles Problem darin, dass bei kontinuierlicher Veränderung der Umgebung die geplante Bahn zum Zeitpunkt der Ausführung bereits veraltet sein kann.

Einige Methoden berücksichtigen dafür bei der Planung Prognosen zukünftiger Hindernisbewegungen durch Extrapolation des aktuellen Umgebungszustandes. Meist wird jedoch versucht, Unsicherheiten durch möglichst kurze Aktualisierungsintervalle auf Basis des aktuellen und als statisch angenommenen Umgebungszustandes zu minimieren (Phillips et al. 2011).

Damit hat die Bahnplanungsmethode lediglich eine reziprok proportional von der Dynamik der Umgebung abhängige Zeit, um auf Veränderungen mit einer Neu-, Um- oder Weiterplanung zu reagieren. Da passives Verhalten oder ein verzögertes Ergebnis die Sicherheit gefährden kann, stellt dieser Fall des Bahnplanungsproblems grundsätzlich ein hartes Echtzeitproblem dar (Petti et al. 2005; Rastgoo et al. 2014). Die Planung muss dabei in vorgegebenen Zeitintervallen parallel zur Ausführung erfolgen und erfordert damit eine recheneffiziente und zeitdeterministische Berechenbarkeit.

Bahnplanungsansätze für dynamische Umgebungen werden nach globalen, reaktiven sowie gemischten Ansätzen unterschieden (Kunz et al. 2010) und sollen im Folgenden näher erläutert werden. Eine Übersicht der vorgestellten Methoden ist in Abbildung 1 gegeben.



RRT=Rapidly-exploring random tree, **PRM**=Probabilistic roadmap,
PFM=Potential field method, **VFF**=Virtual force field, **VFH**=Vector field histogram,
ND=Nearness diagram, **DWA**=Dynamic window approach,
GDWA=Global dynamic window approach, **EBM**=Elastic bands method,
ESM=Elastic strips method

Abbildung 1: Vorgestellte Methoden zur Bahnplanung in dynamischer Umgebung

2.6.1 Globale Methoden

Globale Ansätze für dynamische Umgebungen sind aus Offline-Methoden für statische Umgebungen abgeleitet und berücksichtigen bei der Planung der Bahn die durchgängige Verbindung (Konnektivität) zwischen Start und Ziel. Während für statische Probleme die global optimale Bahn offline in einem einzelnen aufwändigen Berechnungsschritt ermittelt werden kann (Barraquand et al. 1992), ist dies in dynamischer Umgebung nicht möglich. Zum einen kann zum Aktualisierungszeitpunkt jeweils lediglich ein Teilausschnitt der Umgebung bekannt sein und zum anderen lässt die Forderung nach kurzen Aktualisierungintervallen keine allzu rechenintensiven Planungsschritte zu.

Globale Methoden stellen die Umgebung in einem Graphen dar, welcher die Konnektivität einzelner diskreter und kollisionsfreier Punkte (Konfigurationen oder Zustände) innerhalb des zugrundeliegenden Koordinatenraums (Konfigurations- oder Zustandsraum) abbildet. Für die Planung von optimalen Bahnen wird dieser Graph in statischen und zeitinvarianten Umgebungen häufig durch explizite Abbildung von Hindernissen mittels Voronoi-Regionen, Zellzerlegung oder für ebene Probleme mittels Sichtgraphen erzeugt. Für dynamische Umgebungen und hochdimensionale Koordinatenräume sind diese Verfahren zu rechenaufwendig oder gar geometrisch nicht umsetzbar.

Anstatt die vollständige Konnektivität der Umgebung explizit zu ermitteln, werden für Graphen in dynamischer Umgebung lediglich einzelne zufällig ausgewählte Punkte des zugrundeliegenden Koordinatenraums (Samples) sowie deren Verbindungen auf Kollisions-

freiheit überprüft. Durch diese samplebasierte Planung können Koordinatenräume beliebiger Dimension abgebildet werden, wobei die Komplexität nicht exponentiell von der Dimension abhängt (Hsu et al. 1997). Gleichzeitig geht allerdings die dadurch gewonnene Recheneffizienz zu Lasten der Optimalität der Bahn.

Die beiden häufigsten Varianten zufällig samplebasierter Konnektivitätsgraphen sind Rapidly-exploring random trees und probabilistische Roadmaps (Short et al. 2016; Elbanhawi et al. 2014). Deren Grundprinzipien sind in Abbildung 2 dargestellt.

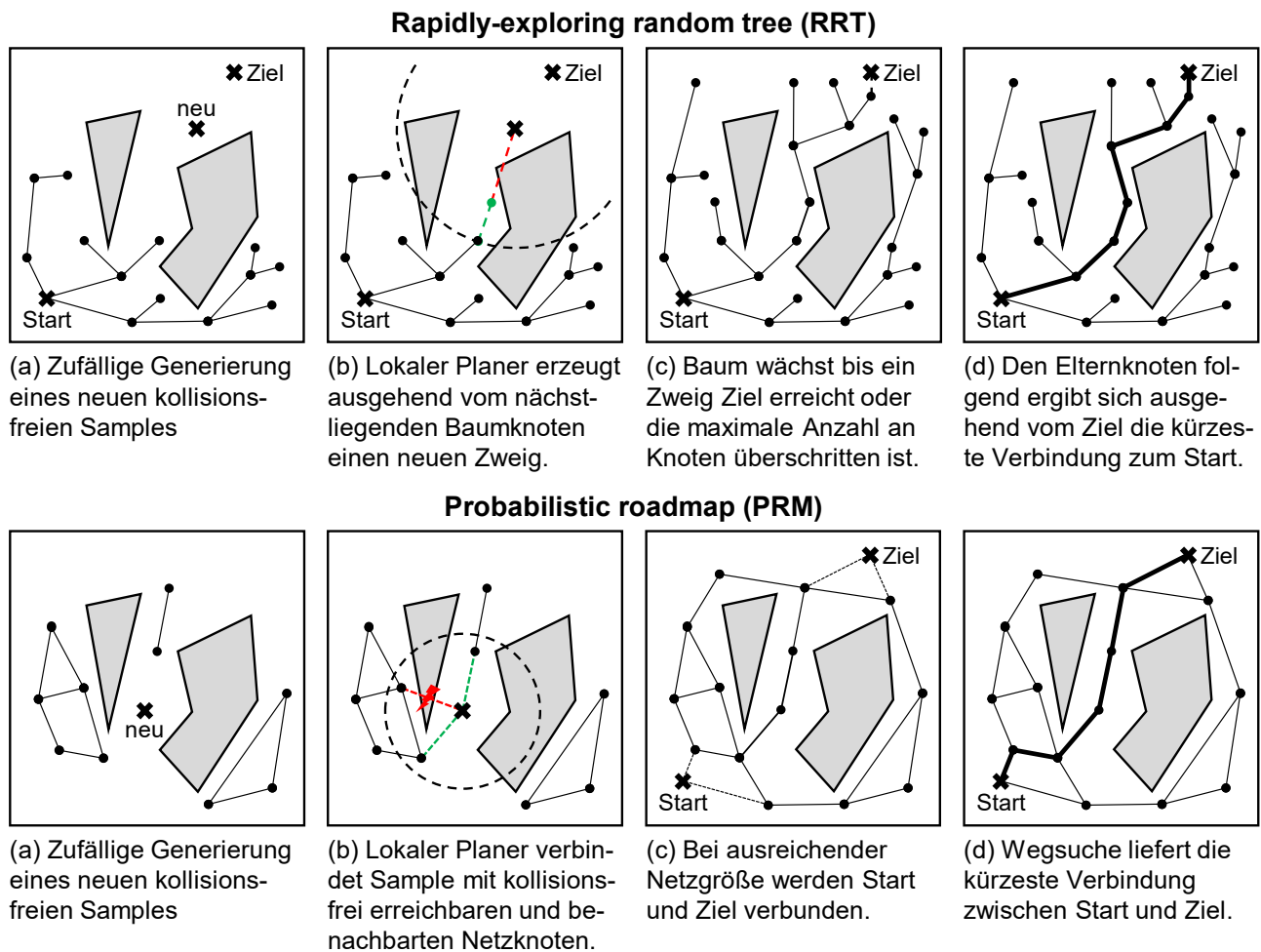


Abbildung 2: Die Grundprinzipien von RRT und PRM

Rapidly-exploring random trees (RRT) wurden ursprünglich für kinodynamische Bahnplanungsprobleme in statischer zeitinvarianter Umgebung entwickelt (LaValle et al. 2001). Die kinodynamische Planung adressiert das Problem unter Berücksichtigung dynamischer Randbedingungen des Roboters direkt in dessen Zustandsraum. Der baumförmig vom

Start- oder Zielzustand ausgehende oder bidirektional verlaufende Graph wird durch einen lokalen Planer aufgebaut. Dabei wird ein Dynamikmodell des Roboters, welchem einzelne zeit- und wertediskrete Steuereingänge (z.B. maximale, minimale oder keine Beschleunigung innerhalb eines Zeitschritts) zufällig aufgeprägt werden, vorwärts- und/oder rückwärts integriert. Alternativ werden auch häufig vorab berechnete Bewegungsprimitive verwendet. Lediglich kollisionsfrei erreichbare Zustände werden dem Graphen als neue Knoten hinzugefügt. Sobald eine Verbindung der Zweige zwischen Start- und Zielzustand existiert, kann die Trajektorie zum Ziel direkt ohne zusätzliche Wegsuche durch Verfolgung des jeweiligen Elternknotens innerhalb des Baumes ermittelt werden.

Mithilfe mehrerer Optimierungen wurden RRTs für eine kinodynamische Bahnplanung in dynamischer Umgebung adaptiert. So wurde RRT zunächst mit RRT* (Karaman et al. 2011) um die Möglichkeit erweitert, einzelne Zweige des Baums zu ersetzen oder umzuvernetzen, sofern diese eine kürzere Verbindung zum Zielzustand ermöglichen. Dadurch strebt RRT* unter der Voraussetzung theoretisch unbegrenzter Rechenzeit asymptotisch gegen die optimale Lösung. Man spricht dabei von einer probabilistisch optimalen Methode. Da RRT* ausgehend von einem einzelnen Zielzustand eine gleichmäßige Erkundung des gesamten Zustandsraums durchführt, findet RRT* zugleich die meist nicht verwertbare optimale Lösung für jeden anderen Startzustand, was mit einer langsamen Konvergenz für die eigentlich gesuchte Start-/Zielkombination einhergeht. Informed RRT* (Gammell et al. 2014) begrenzt die Suche nach kürzeren Verbindungen auf einen Ellipsoiden, welcher die aktuell beste Lösung umhüllt, und verbessert dadurch die Konvergenz. Mit RRT^x (Otte et al. 2015) wurden diese Verbesserungen schließlich für die Berücksichtigung von bewegten, erscheinenden und verschwindenden Hindernissen adaptiert, wodurch Online-Bahnplanung mit kinodynamischen Randbedingungen ermöglicht wird. Mit Realtime-RRT* (Naderi et al. 2015) wurde zeitgleich eine rein geometrische Variante (ohne Berücksichtigung der Dynamik) für die Anwendung in Computerspielen vorgestellt.

Die Funktionsweise probabilistischer Roadmaps (PRM) (Kavraki et al. 1996) ist sehr ähnlich zu der von RRTs. Allerdings bilden deren Graphen erreichbarer Zustände keine start- oder zielzentrierte Baumstruktur, sondern eine über den gesamten Zustandsraum gleichverteilte Netzstruktur. Gegenüber der auf einzelne Start-/Zielkombinationen beschränkten

Wiederverwendung von RRT-Bäumen (Single-Query Planung) können PRM-Netze damit für beliebige Bahnplanungsaufgaben innerhalb derselben Umgebung wiederverwendet werden (Multi-Query Planung). Im Gegensatz zu RRTs wird der vollständige Graph, dessen Größe während der Ausführung bei den meisten onlinefähigen PRM-Implementierungen nicht mehr verändert wird, vorab berechnet. Die Initialisierungsphase von PRMs ist dadurch gegenüber RRTs grundsätzlich sehr viel rechenintensiver und die Wahl der Sampleddichte (Diskretisierung) beeinflusst die erreichbare Optimalität der Bahnen gegenüber der benötigten Rechenzeit (Short et al. 2016).

Die Adaption des Graphen an die dynamische Umgebung wird durch Dynamic Roadmaps (DRM) (Kallman et al. 2004; Leven et al. 2002) oder seltener mit Elastic Roadmaps (ERM) (Yang et al. 2010) realisiert. DRMs sind in der Lage kollisionsbehaftete Knoten des initial als frei von bewegten Hindernissen angenommenen Graphen während der Ausführung zu deaktivieren und die Kantengewichte entsprechend anzupassen. ERMs binden die Knoten des Graphen an einzelne Features der dynamischen Umgebungen und führen den Graphen unter Aktualisierung der „elastischen“ Kanten mit der Umgebung mit.

Die optimale Bahn innerhalb des kantengewichteten PRM-Graphen muss mit einem Suchalgorithmus ermittelt werden. In statischen und zeitinvarianten Umgebungen wird dafür typischerweise A* (Hart et al. 1968), die heuristische Erweiterung des Dijkstra-Suchalgorithmus (Dijkstra 1959), angewandt. Bei einer heuristischen Suche werden durch Abschätzung der Kosten bis zum Ziel (z.B. mithilfe der verbleibenden Luftliniendistanz) vielversprechende Knoten priorisiert erkundet, wodurch die Rechenzeit in den meisten Fällen stark reduziert werden kann. Es konnte gezeigt werden, dass der A*-Algorithmus hierfür am performantesten ist (Dechter et al. 1985).

Für dynamische Umgebungen existieren zahlreiche onlinefähige Implementierungen von A*, welche auf DRMs oder ERMs angewandt werden. Diese können die Wegsuche nach einer vorgegebenen Zeit unterbrechen und die bis dahin beste Teillösung zurückzugeben. Die am häufigsten eingesetzten Varianten sind Anytime A* (Likhachev et al. 2003), welche ausgehend von einer maximalen Gewichtung der Heuristik zur optimalen Lösung konvergiert, sowie D* Lite (Koenig et al. 2002) als inkrementelle Variante von A*.

Neben den vorgestellten Methoden existieren zahlreiche weitere Varianten onlinefähiger RRTs, PRMs und A*-basierter Suchalgorithmen, welche sich dieselben beschriebenen Funktionsprinzipien teilen und in verschiedenen Übersichtsstudien detailliert erläutert werden (Rastgoo et al. 2014; Short et al. 2016).

2.6.2 Reaktive Methoden

Reaktive Methoden werten zur Bewegungsplanung lediglich lokale Information in direkter Umgebung des Roboters (z.B. auf Basis von Onboard-Sensoren) aus, ohne die globale Konnektivität zu berücksichtigen. Sie dienen primär der sicheren Hindernisvermeidung und generieren direkt Steuersignale, weswegen sie innerhalb der Steuerungsarchitektur meist eng mit der Bewegungsregelung des Roboters verknüpft (Goerzen et al. 2010) und recheneffizient implementierbar sind.

Die Funktionsprinzipien der meisten reaktiven Methoden haben ihren Ursprung in der Potentialfeldmethode (PFM). Entgegen globalen Methoden bildet die PFM die Umgebung nicht in einem diskreten Konnektivitätsgraphen, sondern in einem kontinuierlichen und differenzierbaren Feld ab, welches an jeder beliebigen Stelle des Konfigurationsraums explizit ausgewertet werden kann. In Analogie zum Kraftfeld elektrischer Punktladungen erzeugen Hindernisse künstliche Kraftfelder, deren abstoßende Wirkung mit der Entfernung abnimmt. Gleichzeitig bildet die Zielkonfiguration eine über den ganzen Raum konstant anziehende Quelle. Durch Verfolgung des Feldes in Richtung der größten Potentialabnahme (entspricht dem negativen Feldgradienten) ergibt sich die kollisionsfreie Bahn zur Zielkonfiguration.

Zu den signifikantesten Schwächen der PFM auf Basis von künstlichen Kraftfeldern gehören das Festfahren in lokalen Minima sowie das Nichtpassieren und Oszillieren zwischen engen Hindernissen (Koren et al. 1991; Koditschek 1987). Lokale Minima sind stabile Konfigurationen, an denen sich anziehende und abstoßende Kräfte aufheben (z.B. beim Einfahren in ein U-förmiges Hindernis). Obgleich Strategien wie das zufallsgesteuerte Freifahren (Barraquand et al. 1991) oder Hinzufügen von Hindernissen (Safadi 2007) dieses Problem teilweise algorithmisch umgehen, kann eine beliebige oder gar die global optimale Bahn zum Ziel trotz deren Existenz nicht garantiert werden (Goerzen et al. 2010).

Ein Potentialfeld ohne lokale Minima kann lediglich durch Verwendung einer harmonischen Funktion oder der Distanzfunktion als Potentialfunktion erzeugt werden (Goerzen et al. 2010; Radmanesh et al. 2018). Diese beiden speziellen Potentialfunktionen werden auch als Navigationsfunktionen bezeichnet. Die Distanzfunktion weist durch eine vom Ziel ausgehende und den freien Konfigurationsraum durchdringende Wellenfront (engl. wave propagation) jedem Punkt die zum Erreichen des Ziels zurückzulegende Distanz als Potential zu (Jarvis 1993; Wang et al. 2002). Eine harmonische Potentialfunktion beschreibt die Lösung eines physikalischen Strömungsproblems auf Basis einer Laplace'schen partiellen Differentialgleichung im freien Konfigurationsraum (Connolly et al. 1990). Anschaulich kann dieses Verfahren mit einem initial fluidgefüllten Arbeitsraum verglichen werden, welcher durch einen Auslass am Zielpunkt entleert wird. Die Bahn entspricht dann dem Weg eines vom Startpunkt aus der Strömung folgenden Partikels.

Im Gegensatz zur kraftfeldbasierten PFM muss bei Verwendung einer Navigationsfunktion die Umgebung vollständig bekannt sein (Saravanakumar et al. 2013) und es ist nicht möglich, das Potentialfeld für eine beliebige Konfiguration direkt auszuwerten. Stattdessen muss über den gesamten diskretisierten Konfigurationsraum iteriert (Distanzfunktion) oder eine partielle Differentialgleichung numerisch gelöst werden (harmonische Funktion), was für eine Aktualisierung der Bahn in Echtzeit im Allgemeinen zu rechenaufwendig ist.

Erste Anwendung erfuhr die kraftfeldbasierte PFM bei der direkten Integration in den Lageregelkreis eines stationären Roboters, wobei die abstoßende Wirkung der bewegten Hindernisse in Echtzeit auf die einzelnen Achsmomente transformiert wurde (Khatib 1985).

Aufbauend auf der Idee der kraftfeldbasierten Hindernisvermeidung wurden weitere Ansätze mit stärkerem Fokus auf die mobile Robotik mit Onboardsensorik zur ebenen Abtastung der direkten Umgebung (z.B. 2D-Laserscanner, Ultraschall- oder Infrarotsensor) entwickelt. Eine frühe Realisierung der PFM für die mobile Robotik ist das Virtual Force Field (VFF) (Borenstein et al. 1989). Die Abstoßung durch Hindernisse geht dabei von einzelnen Zellen eines gitterförmigen 2D-Histogramm der Umgebung aus. Die Hinderniswahrscheinlichkeit einzelner Zellen wird kontinuierlich durch neue Sensordaten aktualisiert, wodurch Messunsicherheiten kompensiert werden.

Um das PFM-typische Problem der Durchfahrt in engen Passagen zu umgehen, wurde das VFF zum Vector Field Histogram (VFH) (Borenstein et al. 1991) weiterentwickelt. Das 2D-Histogramm wird dabei zu einem polaren 1D-Histogramm reduziert und die Navigationsrichtung wird unter Berücksichtigung der lokal minimalen Hindernisdichte gewählt. Diese Dichte unterliegt dabei einem empirisch zu ermittelnden Schwellwert, was zu Falschinterpretationen führen kann. Mit VFH+ (Ulrich et al. 1998) wurde diese Methode unter anderem um die Berücksichtigung eines minimalen Kurvenradius sowie einer Kostenfunktion zur optimalen Wahl der vormals lediglich zielorientierten Ausweichrichtung erweitert. VFH* (Ulrich et al. 2000) ergänzt die Wahl der Ausweichrichtung um einen A*-basierten Look-Ahead, wodurch das Risiko für lokale Minima reduziert wird. Für kleine Suchtiefen nähert sich das Verhalten von VFH* dem von VFH+ an.

Beim Nearness Diagramm (ND) (Minguez et al. 2004) basiert die Bestimmung der Ausweichrichtung nicht auf einem Potentialfeld, sondern auf hindernisfrei detektierten Lücken im direkten Umfeld des Roboters. Durch Evaluation von Abfolge und Breite dieser Lücken wird eine situative Navigationsstrategie anhand von Fallunterscheidungen ausgewählt. Insbesondere das Festfahren in lokalen Minima aufgrund U-förmiger Hindernisse (sofern diese vollständig erfasst werden) kann so vermieden sowie Oszillationen in engen Passagen reduziert werden. Eine weitere Stärke dieser Methode ist, dass sie weitestgehend ohne Parametrierung auskommt.

Beim Dynamic Window Approach (DWA) (Fox et al. 1997) werden zusätzlich dynamische Eigenschaften des Roboters berücksichtigt. Im Geschwindigkeitsraum (aufgespannt von Translations- und Rotationsgeschwindigkeit in der Ebene) des Roboters werden für das jeweils nächste Zeitfenster alle unter Berücksichtigung der Beschleunigungsgrenzwerte erreichbaren Geschwindigkeiten betrachtet, welche noch einen sicheren Halt ermöglichen. Unter Einbeziehung der Annäherung zum Ziel, dem Abstand zum nächsten Hindernis sowie der Geschwindigkeitsänderung wird aus diesem Dynamikfenster die optimale Geschwindigkeitspaarung ausgewählt.

Alle hier vorgestellten reaktiven Methoden wurden meist in Umgebungen appliziert, deren Dynamik primär von partieller Kenntnis der räumlichen Gegebenheiten geprägt ist. Ge-

messen an der Robotergeschwindigkeit wurden dabei lediglich langsam bewegte Hindernisse berücksichtigt. Daneben wurden auch Varianten von VFH und DWA mit Fokus auf schnell bewegte Hindernisse vorgestellt (Huiliang et al. 2003; Castro et al. 2002).

2.6.3 Gemischte Methoden

Einige Ansätze stellen eine Mischform der beiden vorangegangenen Kategorien dar, indem sie reaktives Verhalten mit globaler Konnektivität koppeln.

Der Global Dynamic Window Approach (GDWA) (Brock et al. 1999; Brock 2000) baut direkt auf DWA auf. Anstatt die Annäherung zum Ziel lediglich mithilfe der Luftliniendistanz zu bewerten, nutzt GDWA hierfür eine grob aufgelöste Wegsuche auf Basis einer vom Ziel ausgehend propagierten Distanzfunktion. Zur Steigerung der Recheneffizienz wird die Suche auf ein Rechteck begrenzt, welches Roboter und Ziel umschließt. Dessen Breite wird so lange erweitert, bis der Roboter durch die Propagation erreicht werden kann.

Bei der Elastic-Bands-Methode (EBM) (Quinlan et al. 1993) wird eine initial aus einzelnen Stützstellen bestehende globale Bahn als elastisches Band modelliert und kann unter Einfluss von hindernisabhängigen abstoßenden Kräften und internen Kontraktionskräften verformt werden. Mit sogenannten „Bubbles“ (je nach Dimension der Problemstellung ein umhüllender Kreis oder eine Kugel) wird der freie Raum, welcher die einzelnen Stützstellen (Konfigurationen) umgibt, abgeschätzt. Um die Kollisionsfreiheit zwischen den Stützstellen zu gewährleisten, müssen sich deren Bubbles stets überlappen, wodurch die Diskretisierung der Bahn von deren Länge sowie dem Abstand zu den Hindernissen abhängt und kontinuierlich durch Einfügen oder Entfernen von Bubbles verändert werden muss. Dieser Ansatz führt damit eine kontinuierliche lokale Optimierung der gesamten Bahn durch, ohne dass eine vollständige Neuplanung erforderlich ist.

Die Elastic-Strips-Methode (ESM) (Brock et al. 2002b) stellt die direkte Weiterentwicklung der EBM dar und adressiert primär redundante Roboter mit vielen Freiheitsgraden. Die ESM umgeht dabei die komplexe Berechnung der gesamten Bahn im hochdimensionalen Konfigurationsraum, indem sie die Bahn als elastischen Tunnel möglicher Konfigurationen im euklidischen Raum beschreibt. Ausschließlich die aktuelle Pose des Roboters wird da-

bei im Konfigurationsraum abgebildet. So kann beispielsweise der Endeffektor eines redundanten Roboters einer Solltrajektorie folgen, während die Plattform gleichzeitig Hindernissen ausweicht (Brock et al. 2002a).

Grundsätzlich benötigen sowohl die EBM als auch die ESM einen globalen Planer zur Generierung einer kollisionsfreien Initialtrajektorie, was mit einem hohen Initialisierungsaufwand einhergeht. Um dies zu vermeiden, wurde für Experimente mit der EBM und der ESM stets ein initial hindernisfreier Arbeitsraum angenommen.

2.7 Defizite und Stärken

In den vorangegangenen Kapiteln wurden gängige Methoden für die Bahnplanung in dynamischer Umgebung getrennt nach globalen, reaktiven und gemischten Ansätzen vorgestellt. Die Eignung der einzelnen Methoden zur Lösung der Problemstellung soll nun anhand der in Kap. 1.4 formulierten Forderung nach Echtzeitfähigkeit, numerischer Robustheit, Recheneffizienz, Optimalität sowie der Eignung für 3D-Umgebungen auf Defizite und Stärken untersucht und im Anschluss bewertet werden. Da die ebenfalls geforderte Skalierbarkeit primär eine Eigenschaft der Implementierung und nicht der Methode an sich darstellt, wird diese nicht miteinbezogen.

Echtzeitfähigkeit:

Obgleich alle in Kap. 2.6 vorgestellten Methoden als echtzeitfähig gelten, bezieht sich dies insbesondere bei globalen und gemischten Methoden lediglich auf ihre Eigenschaft in für das jeweilige Testszenario ausreichend schnellen Intervallen parallel zur Ausführung berechnet zu werden (weiche Echtzeit). Die geforderte harte Echtzeitfähigkeit im Sinne einer Industriesteuerung setzt hingegen ein streng deterministisches Zeitverhalten der Methode voraus, was insbesondere dynamische Speicherallokationen und unvorhersehbar terminierende Iterationen bei der Implementierung schwierig macht.

Globale Methoden bilden die Umgebung mithilfe baumförmiger (RRT) oder netzförmiger (PRM) Konnektivitätsgraphen ab. Für die Speicherung und Änderung valider Knoten und Kanten sind komplexe verkettete und dynamische Datenstrukturen notwendig, deren speicherdeterministische Umsetzung in einem harten Echtzeitsystem nicht ohne weiteres rea-

lisierbar ist. Globale Methoden unterliegen außerdem starken Schwankungen der Aktualisierungsrate bei Neu- oder Umplanung der Bahn. So lässt die zufallsbasierte, und bei Nichtexistenz einer Lösung theoretisch endlose, Generierung neuer Samples von RRTs keine Prognose zu, nach welcher Zeit wieder eine vollständige Bahn zur Verfügung steht. In der Praxis wird die RRT-Suche mit dem Risiko, eine nicht durchgängig valide Bahn zurückzugeben, nach vorgegebener Zeit oder Anzahl geprüfter Samples abgebrochen. PRMs generieren während der Ausführung keine neuen Samples, sondern modifizieren lediglich einen fein aufgelösten Graphen konstanter Größe. Hier verhindert zwar die heuristische Wegsuche eine konstante Aktualisierungsrate, besitzt aufgrund der endlichen Größe des Graphen entgegen RRTs jedoch eine begrenzte Rechenzeit. Unter Berücksichtigung dieser maximal möglichen Rechenzeit ist eine Implementierung von PRMs unter harter Echtzeit zwar möglich aber ineffizient.

Reaktive Methoden sind eng mit den Steuersignalen des Roboters gekoppelt und zum Teil direkt in deren Lageregelkreis integriert. Die Ausweichstrategien von PFM, VFH, VFH+, ND und DWA nutzen ausschließlich deterministische Berechnungsvorschriften oder Iterationen über eine konstante Anzahl von Sensordaten, womit Zeit- und Speicherdeterminismus unter harter Echtzeit möglich ist. Lediglich VFH* nutzt als Look-Ahead-Funktion eine heuristische A*-Suche, deren Rechenzeit vom Umgebungszustand abhängt. Mit steigender Suchtiefe übersteigt dabei die maßgebende maximale Laufzeit deren Durchschnittswert um ein Vielfaches, wodurch eine Ausführung in harter Echtzeit ebenfalls zwar möglich aber ineffizient ist (Ulrich et al. 2000).

GDWA erweitert DWA um eine globale Wegsuche mit Distanzpropagation auf einem fest diskretisierten Zellgitter. Die gerichtete Propagation wird nur so lange fortgeführt bis eine globale Bahn gefunden wird. Da die Größe des dafür abzusuchenden Gebiets vom aktuellen Umgebungszustand abhängt (Brock 2000), ist eine Ausführung in harter Echtzeit nur unter Berücksichtigung der theoretisch maximalen Rechenzeit möglich und ineffizient. Um die Kollisionsfreiheit der EBM und der ESM zu gewährleisten, muss die Diskretisierung der Bahn kontinuierlich an den Abstand zu den Hindernissen angepasst werden, wodurch diese Methoden in engen Passagen ineffizient werden (Brock 2000). Da lediglich eine Stützstelle pro Interpolationstakt aktualisiert wird, kann dieser zwar unabhängig von der

Diskretisierung gehalten werden, eine zeit- und speicherdeterministische Aktualisierung der gesamten Bahn ist allerdings nur bei konstanter Stützstellenanzahl möglich.

Recheneffizienz:

Um Sicherheit und Kollisionsfreiheit zu garantieren, sollte die Bahn anhand des aktuellen Umgebungszustandes in möglichst kurzen Zyklen vollständig aktualisiert werden. Für Arbeitsräume mit Menschen wurde in Anlehnung an deren Reaktionszeit ein Planungsintervall von etwa 180 Millisekunden als Echtzeit definiert (Kunz et al. 2010). Arbeitsräume mit dynamischen Robotern oder UAVs erfordern entsprechend höhere Zykluszeiten. Gemäß den in der Problemstellung formulierten Anforderungen ist die Aktualisierung daher mindestens im Interpolationstakt gängiger Bewegungssteuerungen von wenigen Millisekunden anzustreben.

Globale Methoden sind häufig theoretisch motiviert, weshalb deren Recheneffizienz meist hardwareunabhängig in Form ihrer Zeitkomplexität (O-Notation) angegeben wird. Dies macht die Vergleichbarkeit mit praxisnahen Implementierungen reaktiver und gemischter Methoden schwierig. Für statische Umgebungen und bei identischer Anzahl von Samples konnte RRTs gegenüber PRMs dieselbe Zeitkomplexität für den Aufbau des Graphen sowie eine geringere Zeitkomplexität für die Wegsuche bescheinigt werden (Karaman et al. 2011). Meist benötigen RRTs allerdings weniger Samples, um eine valide Lösung zu generieren. Online-Implementierungen unterliegen in der Regel einem Abbruchkriterium. Dokumentierte Zykluszeiten liegen hier sowohl bei RRT- als auch bei PRM-basierten Methoden bestenfalls in der Größenordnung von 100 Millisekunden oder mehr und unterliegen hohen Schwankungen (Kunz et al. 2010; Kuwata et al. 2009; Benenson et al. 2006; Hrabar 2011). Informationen über eine Veränderung der Umgebung können darüber hinaus lediglich durch die sukzessive Aktualisierung von Kantengewichten zum Roboter propagiert werden, weshalb dessen Reaktion zusätzlich um ebendiese Informationstransferzeit verzögert wird. Diese Informationstransferzeit ist meist sehr viel höher als das eigentliche Aktualisierungsintervall (Otte et al. 2016). In zahlreichen Fällen reicht die Reaktionsgeschwindigkeit globaler Methoden nicht aus, um Kollisionen zu vermeiden (Hrabar 2008).

Reaktive Methoden sind primär durch kurze und damit sichere Reaktionszeiten motiviert. Dokumentierte Zykluszeiten wie beispielsweise 10 Millisekunden für PFM (Khatib 1985), 27 Millisekunden für VFH (Borenstein et al. 1991), 6 Millisekunden für VFH+ (Ulrich et al. 1998), je nach A*-Suchtiefe zwischen 6 und 242 Millisekunden für VFH* (Ulrich et al. 2000), 125 Millisekunden für ND (Minguez et al. 2004) sowie weniger als 250 Millisekunden für DWA (Fox et al. 1997) sind nur bedingt miteinander vergleichbar, da sie unter verschiedenen Testszenarios und auf unterschiedlicher Hardware ausgeführt wurden. Vergleichende Studien (Zohaib et al. 2013; Portugal 2010; Calisi 2007) haben die Effizienz der Methoden gegenübergestellt und PFM sowie ND die höchste Recheneffizienz bestätigt. Sowohl VFH als auch DWA müssen zur Bestimmung der Ausweichrichtung Funktionen über ein 2D-Gitter auswerten (Reduktion des 2D-Histogramm bei VFH und VFH+ bzw. Auswertung der Optimierungsfunktion über dem Geschwindigkeitsraum bei DWA), was diese Methoden weniger recheneffizient macht. Bei VFH* dominiert die A*-Wegsuche die Rechenzeit, welche gegenüber VFH+ um ein Vielfaches höher ist.

Abhängig von der Komplexität der Navigationsfunktion kann GDWA mit schwankenden Aktualisierungsintervallen zwischen 25 und 200 Millisekunden ausgeführt werden (Brock 2000). Die Rechenzeit wird primär für die zweidimensionale Propagation der Distanzfunktion benötigt. Auch die Ausführungszeiten der EBM und der ESM schwanken stark in Abhängigkeit der umgebungsabhängigen Bahndiskretisierung. Die vollständige Aktualisierung einer in 20 Stützstellen diskretisierten Bahn wurde für die EBM mit Intervallen zwischen 200 bis 1000 Millisekunden dokumentiert (Quinlan 1994; Philippsen 2004). Der Großteil der Rechenzeit wird dabei für die Berechnung des hindernisfreien Raums um die Stützstellen benötigt. Aufgrund der Repräsentation der Stützstellen durch „Bubbles“ im Konfigurationsraum steigt die Komplexität exponentiell mit der Anzahl der Freiheitsgrade des Roboters. Die ESM ist recheneffizienter, da sie die Bahn unabhängig von der Anzahl der Freiheitsgrade ausschließlich im dreidimensionalen euklidischen Raum beschreibt. Für die ESM sind Aktualisierungsintervalle zwischen 10 und 200 Millisekunden dokumentiert (Brock 2000).

Numerische Robustheit:

Online-Bahnplanungsmethoden aktualisieren die Bahn (oder bei Integration der Methode im Regelkreis auch direkt den Sollzustand bzw. die Stellgrößen) des Roboters in Abhängigkeit des aktuellen Zustandes von Roboter, Bahn und Umgebung. In einem Digitalrechner werden diese rückgekoppelten Wechselwirkungen in Form von iterativen Berechnungsvorschriften umgesetzt. Eine iterative Berechnungsvorschrift (die Bahnplanungsmethode) gilt als numerisch robust, wenn sie für beliebige Parametrierungen (z.B. Diskretisierung), Eingänge (z.B. Start- und Zielpunkt oder Hindernisse) und Zustände (z.B. aktuelle Bahn oder Roboterposition) nicht in solch hohem Maße sensitiv auf Rundungs- oder Näherungsfehler reagiert, dass die numerische Lösung (z.B. neue Bahn oder Stellgrößen) nicht mehr gegen die exakte Lösung konvergiert.

Die Iterationsvorschriften globaler Methoden benötigen für die Bahngenerierung keine numerischen Näherungen kontinuierlicher Funktionen, sodass das Risiko von Instabilitäten sehr gering ist. Lediglich die Generierung neuer Samples durch den lokalen Planer kann die numerische Simulation eines Robotermodells erfordern. Dieses Modell ist allerdings frei von äußeren Umgebungseinflüssen und wird lediglich mit vordefinierten Eingängen angeregt. RRT*-basierte Methoden besitzen darüber hinaus eine asymptotisch monotone Konvergenz (Karaman et al. 2011; Otte et al. 2015). Aus diesen Gründen ist die numerische Robustheit globaler Methoden als sehr gut zu bewerten.

Ein weit verbreitetes Problem reaktiver (und häufig in den Regelkreis des Roboters integrierter Methoden) ist ihre Neigung zu Oszillationen in der Nähe von Hindernissen oder in engen Passagen. Diese entstehen durch die Totzeit, welche mit der zeitdiskreten Ausführung der Methode (Berechnung der Steuereingänge) in einem Digitalrechner einhergeht (Koren et al. 1991). Dabei führt eine leichte Abweichung des Roboters von der stabilen potentialminimalen Ideallinie zwischen zwei starken Potentialquellen zu einer dauerhaften wechselseitigen Abstoßung. Dies stellt ein grenz- oder gar instabiles Verhalten dar (Koren et al. 1991), welches lediglich durch die maximalen Steuereingänge begrenzt wird.

Verschiedene Strategien werden angewandt, um die Stabilität reaktiver Methoden zu erhöhen. So verwenden einige PFM-Implementierungen einen Tiefpassfilter, was allerdings

deren Reaktionszeit verlangsamt. Es konnte außerdem gezeigt werden, dass VFH weniger anfällig für Oszillationen als PFM ist (Manz et al. 1993). Bei VFH wirken die Hinderniskräfte nicht direkt auf die Steuereingänge, sondern werden auf befahrbare freie Sektoren im polaren Histogramm abgebildet, welche in Abhängigkeit des gewählten Schwellwertes zwischen befahrbar und nicht befahrbar „springen“ können. Ab VFH+ wurde eine Hysterese für diesen Schwellwert eingeführt (Ulrich et al. 1998), um Oszillationen zusätzlich zu reduzieren. ND navigiert den Roboter in Hindernisnähe ebenfalls nicht potentialfeldbasiert, sondern entlang der geometrischen Mitte zwischen zwei nahestehenden Hindernissen bzw. entlang der Kante bei lediglich einem Hindernis, wodurch die Neigung zu Oszillationen reduziert wird. Für die von DWA verwendete Zielfunktion konnten Schwächen identifiziert werden, welche unter anderem zu Oszillationen in der Orientierung sowie zur Priorisierung einer Bewegungsrichtung weg vom Ziel führen können (Berti et al. 2008). Durch Integration des Lyapunov-Stabilitätskriteriums in die Zielfunktion von DWA konnten die Oszillationen reduziert sowie eine asymptotische Konvergenz in Richtung Ziel erreicht werden.

GDWA verwendet die ursprüngliche Implementierung von DWA. Lediglich die fehlerhafte Navigation weg vom Ziel wurde durch eine Neuinitialisierung der Bewegungsrichtung aus dem Stand abgefangen. Da die EBM (Quinlan et al. 1993) für die Aktualisierung der Bahn eine explizite Transitions Gleichung, welche numerisch auf die stützstellenweise Anwendung des expliziten Eulerverfahrens führt, verwendet, kann die Stabilität der erzeugten Bahn nicht garantiert werden. Die dadurch bedingten kleinen Schrittweiten haben gleichzeitig eine langsame Konvergenz bei der Rückformung der Bahn zur Folge. Diese Stabilitäts- und Oszillationsprobleme des expliziten Eulerverfahrens sollen später in Kap. 3 nochmals genauer untersucht werden. Eine spätere Variante der EBM (Quinlan 1994) umgeht die Stabilitätsproblematik durch Verwendung eines gradientenbasierten Optimierungsverfahrens anstatt der zeitlichen Entwicklung durch numerische Integration. Dadurch wird die näherungsweise energieminimale Gleichgewichtslage der Stützstellen direkt gelöst, was allerdings sehr viel mehr Funktionsauswertungen erfordert und damit rechenaufwendiger ist. Auch Weiterentwicklungen der EBM (Khatib et al. 1997; Rösmann et al. 2015) oder der ESM (Brock et al. 2002b) sowie deren Anwendungen in mobiler Robotik

(Philippsen 2004), Servicerobotik (Graf et al. 2004; Graf et al. 2009) oder dem autonomen Fahren (Brandt et al. 2005) haben die Stabilitätsproblematiken des ursprünglichen Lösungsverfahrens mittels numerischer Integration nicht weiter aufgegriffen oder gelöst.

Optimalität:

Die Optimalität von Bahnplanungsalgorithmen wird meist auf die Bahnlänge oder bei Berücksichtigung der Dynamik auch auf die Ausführungszeit oder den Energieverbrauch bezogen. Sie steht in der Regel in direktem Konflikt mit der Recheneffizienz der Algorithmen. Dies wirkt sich bei der Onlineplanung durch eine verringerte Aktualisierungsrate bzw. Sicherheit aus. Laut Problemstellung ist die Optimalität gegenüber der Recheneffizienz daher nachrangig zu bewerten, wobei das Erreichen des Ziels dennoch garantiert werden soll.

Globale Methoden sind grundsätzlich in der Lage, globale Bahnen in Arbeitsräumen mit komplexen Hindernissen oder labyrinthartigen Begrenzungen zu finden und können dabei gleichzeitig dynamische Randbedingungen des Roboters berücksichtigen. Auf RRT* (Karaman et al. 2011) basierende Methoden sind dabei probabilistisch optimal und konvergieren damit asymptotisch gegen die global optimale Lösung (Elbanhawi et al. 2014). Durch gezielte Lenkung und Eingrenzung des Suchgebiets kann die Konvergenz zusätzlich beschleunigt werden (Gammell et al. 2014). PRM-generierte Bahnen hingegen sind in der Regel lediglich suboptimal, da die erreichbare Qualität von der begrenzten Dichte der initial ermittelten Roadmap abhängt. Eine grobe Abdeckung erlaubt zwar schnelle Aktualisierungsintervalle, gleichzeitig steigt die Wahrscheinlichkeit, dass Hindernisse mögliche oder gar alle Verbindungen zum Ziel unterbrechen (Pomarlan et al. 2013).

Reaktive Methoden planen lediglich lokal auf Basis ihres direkten Umfeldes und sind nicht in der Lage, eine globale oder gar optimale Bahn zu generieren. Bei PFM, VFH und ND bleibt zudem die Ausführbarkeit gänzlich unberücksichtigt. Durch die schwellwertbasierte Wahl der Ausweichrichtung bei VFH und ND sind abrupte Richtungswechsel möglich, weshalb bei diesen Methoden nur sehr langsam vorgefahren werden kann. Die PFM erzeugt durch die Kontinuität und Differenzierbarkeit des Potentialfeldes, welches der Ausweich-

richtung zugrunde liegt, glattere Bahnen. Es konnte allerdings gezeigt werden, dass der Umwegfaktor gegenüber VFH größer ist (Manz et al. 1993).

Den größten Schwachpunkt aller reaktiven Methoden stellt deren Anfälligkeit für lokale Minima dar. Diese kann durch algorithmische Freifahrstrategien (ND, DWA) bzw. Look-Ahead-Funktionalität (VFH*) nur bedingt reduziert werden und geht mit signifikanten Einbußen der Weglänge und Ausführungszeit bzw. Echtzeitfähigkeit einher.

GDWA führt eine in groben Zellen aufgelöste geometrische Wegsuche durch Distanzpropagation durch. Die daraus resultierende kürzestmögliche Bahn zum Ziel kann dabei rechtwinklig und ohne Mindestabstand zu den Hindernissen verlaufen. Obgleich sie damit zwar nicht ausführbar ist, wird sie lediglich als gewichtete Eingangsgröße zur Bestimmung der optimalen Steuersignale herangezogen. Da gleichzeitig der Sicherheitsabstand zu den Hindernissen sowie die Beibehaltung der aktuellen Geschwindigkeiten in diese Gewichtung einfließen, generiert GDWA dennoch sichere Bahnen (Brock 2000) unter Berücksichtigung der Roboterdynamik und mit einem geringen Umwegfaktor.

Sowohl EBM als auch ESM gewährleisten die Konnektivität zwischen Start und Ziel mit lokal paretooptimalen Bahnen, die sich entsprechend einer experimentell zu ermittelnden Gewichtung zwischen Hindernisabstand und Weglänge ergeben. Obgleich sie dadurch resistent gegenüber lokalen Minima sind, können sie aufgrund ihrer lediglich lokalen Optimalität durch Verdrängung ineffizient werden oder gar versagen, wenn die Bahn komplett abgeschnitten wird (z.B. aufgrund einer schließenden Tür). In beiden Fällen ist eine Neuplanung mit einer aktualisierten globalen Initialtrajektorie notwendig. Zusätzlich berechnet die ESM parallel eine zweite Bahn, welche durchlässig für Hindernisse mit starker Verdrängung ist, um bei Bedarf auf eine lokal optimalere Bahn ausweichen zu können (Brock 2000).

3D-Fähigkeit:

Einige der vorgestellten Methoden wurden im Kontext der mobilen Robotik entwickelt. Da dabei meist ebene Bahnplanungsprobleme für flurgebundene Roboter adressiert werden, soll die Eignung der Methoden für räumliche Bahnplanungsprobleme untersucht werden.

Globale Methoden repräsentieren den Arbeitsraum als Konnektivitätsgraphen in einem abstrahierten Koordinatenraum beliebiger Dimension. Häufig berücksichtigen sie nicht nur kinematische (Konfigurationsraum), sondern ebenfalls dynamische (Zustandsraum) Eigenschaften des Roboters und sind damit für räumliche Problemstellungen geeignet.

Durch die praxisnahe Implementierung reaktiver Methoden sind diese sehr stark an die Bedürfnisse der in der Regel flurgebundenen mobilen Roboter angepasst. Die enge Bindung an 2D-Sensordaten macht eine Adaption für eine räumliche Bahnplanung nicht ohne weiteres möglich. So ist unklar, wie die der Ausweichrichtung zugrundeliegenden eindimensionalen Strukturen bei VFH (polares Histogramm) und ND (Lückenfolge) auf räumliche Probleme angewandt werden können. Gleichzeitig steigt der Rechenaufwand exponentiell, da ursprüngliche Funktionsauswertungen auf einem 2D Gitter bei VFH (2D-Histogramm) und DWA (Geschwindigkeitsraum) nun in drei Dimensionen erfolgen müssten. Die PFM hingegen lässt sich für eine beliebige Anzahl an Dimensionen verallgemeinern und die Rechenkomplexität wächst lediglich linear mit Anzahl der Hindernisse und Dimensionen. Sie wird beispielsweise bei Unterwasserfahrzeugen (Saravanakumar et al. 2013) oder auch UAVs (Chen et al. 2013) für räumliche Problemstellungen angewandt.

GDWA ist ebenfalls für dreidimensionale Probleme anwendbar. So wurde GDWA beispielsweise für die Navigation eines nichtholonomen Rollstuhls (Konfigurationsraum beschreibt zwei Translationen und eine Rotation in der Ebene) mit 2-stufiger Distanzpropagation in nicht deterministischen Aktualisierungsintervallen von 200 Millisekunden verwendet (Demeester et al. 2005) oder auch simulativ für die räumliche Bahnplanung eines Unterwasserfahrzeugs adaptiert (Tusseyeva et al. 2013). Obgleich meist für flurgebundene Roboter angewandt, stellt die EBM eine für Koordinatenräume beliebiger Dimensionen abstrahierbare Methode dar und die ESM beschreibt die Bahn bereits als einen den Roboter umhüllenden Tunnel im dreidimensionalen euklidischen Raum.

2.8 Bewertung, Zusammenfassung und Fazit

In Tabelle 2 werden die untersuchten Ansätze für die zeitvariante dynamische Bahnplanung nochmals zusammengefasst und auf Basis der vorangegangenen Untersuchung hinsichtlich der definierten Anforderungen in qualitativer Form bewertet. VFF und VFH

fließen nicht in die Bewertung mit ein, da sie keinen Vorteil gegenüber ihren späteren Varianten VFH+ und VFH* aufweisen. Aufgrund der Varianz von adressiertem Einsatzzweck, verwendeter Hardware sowie der Komplexität der Testszenarien ist eine streng quantitative Bewertung, insbesondere zwischen globalen, reaktiven und gemischten Methoden, nicht möglich.

-		Echtzeitfähigkeit	Recheneffizienz	Numerische Robustheit	Optimalität	3D-Fähigkeit
global	RRT	○ ○ ○	● ○ ○	● ● ●	● ● ●	● ● ●
	PRM	● ○ ○	○ ○ ○	● ● ●	● ● ○	● ● ●
reaktiv	PFM	● ● ●	● ● ●	● ○ ○	○ ○ ○	● ● ●
	VFH+	● ● ●	● ● ○	● ● ○	○ ○ ○	○ ○ ○
	VFH*	● ● ○	● ○ ○	● ● ○	● ○ ○	○ ○ ○
	ND	● ● ●	● ● ●	● ● ○	○ ○ ○	○ ○ ○
	DWA	● ● ●	● ● ○	● ● ○	○ ○ ○	● ○ ○
gemischt	GDWA	● ○ ○	● ○ ○	● ● ○	● ● ●	● ● ○
	EBM	● ○ ○	● ○ ○	● ○ ○	● ● ○	● ● ●
	ESM	● ○ ○	● ● ○	● ○ ○	● ● ○	● ● ●

○ ○ ○ = nicht erfüllt, ● ○ ○ = teilweise erfüllt, ● ● ○ = größtenteils erfüllt, ● ● ● = voll erfüllt

RRT=Rapidly-exploring random tree, **PRM**=Probabilistic roadmap, **PFM**=Potential field method, **VFH**=Vector field histogram, **ND**=Nearness diagram, **DWA**=Dynamic window approach, **GDWA**=Global dynamic window approach, **EBM**=Elastic bands method, **ESM**=Elastic strips method

Tabelle 2: Qualitative Bewertung der untersuchten Bahnplanungsansätze

Aus Tabelle 2 wird ersichtlich, dass keiner der untersuchten Ansätze alle Anforderung der Problemstellung voll erfüllt.

Globale Methoden lassen sich aufgrund ihrer unbestimmten Terminierung und komplexen Datenstrukturen schwer in harter Echtzeit ausführen und sind nicht ausreichend performant, um die Sicherheit dynamischer UAVs zu gewährleisten. Gleichzeitig adressieren sie

Probleme mit sehr viel komplexeren Randbedingungen als sie durch die Problemstellung gegeben sind. So sind sie in der Lage, kinodynamisch optimale Bahnen in labyrinthartigen Umgebungen numerisch robust zu planen, während die Problemstellung lediglich kinematisch einfach abzubildende UAVs in einem vergleichsweise dünn mit Hindernissen besetzten Raum vorsieht.

Reaktive Methoden erfüllen die Anforderungen hinsichtlich einer recheneffizienten Implementierbarkeit in harter Echtzeit, sind aber gleichzeitig anfällig für lokale Minima, wodurch sie eine Bahn zum Ziel nicht garantieren können. Zwar kann durch die Abbildung von UAVs als umhüllende Kugeln das Risiko des Festfahrens in einem großen Arbeitsraum gesenkt werden, dennoch generieren reaktive Methoden suboptimale Bahnen mit hohem Umwegfaktor. Außerdem sind reaktive Methoden durch ihre Bindung an flurgebundene mobile Roboter nicht immer direkt auf räumliche Problemstellungen übertragbar.

Gemischte Methoden mit globaler Konnektivität stellen hier einen guten Kompromiss zwischen Optimalität und Sicherheit dar und sollen als mögliche Kandidaten für die Problemstellung in Betracht gezogen werden.

Die Adaption an eine veränderte Umgebung wird bei gemischten Methoden entweder durch eine entkoppelte globale Wegsuche (GDWA) oder durch kräftebasierte lokale Optimierung (EBM und ESM) realisiert. Ebene Bahnplanungsprobleme bergen die Gefahr, dass dynamische Hindernisse die Konnektivität von Teilen des Arbeitsraums einschnüren oder gänzlich abschneiden können, beispielsweise wenn sich ein Hindernis auf eine Wand zubewegt. In einer solchen Situation liefert die globale Wegsuche eine valide Alternativbahn, während eine lediglich lokal optimierte Bahn durch Verdrängung in Richtung Wand zunächst ineffizient und anschließend invalide oder gar numerisch instabil wird. Räumliche Bahnplanungsprobleme erlauben hingegen Ausweichbewegungen in einer weiteren Dimension. Das Risiko, dass eine Bahn in einem lokalen Minimum „festhängt“, wird reduziert, da eine lokale Optimierung im Raum die verdrängte Bahn über ein als umhüllende Kugel modelliertes UAV „springen“ lässt. Damit erscheint der zusätzliche Rechenaufwand, den eine kontinuierliche globale Wegsuche mittels GDWA mit sich bringt, für die Problemstellung unnötig.

Die EBM scheint den Anforderungen der Problemstellung am nächsten zu kommen, da sie sich gemäß ihrer ursprünglichen Formulierung (Quinlan et al. 1993) effektiv auf die Beschreibung einer eindimensional diskretisierten und krümmungsstetig verformbaren Bahn im Raum mit globaler Konnektivität und schneller Reaktion reduzieren lässt. Die ESM stellt eine Spezialisierung der EBM dar und beschreibt die Bahn als elastischen räumlichen Tunnel, welcher die Geometrie hochdimensionaler redundanter Kinematiken in engen Arbeitsräumen abbildet. Die Berücksichtigung eines solch komplexen Szenarios erscheint im Sinne der Problemstellung (UAVs als Starrkörper in vergleichsweise großen Arbeitsräumen) unnötig.

Die größte Schwäche der EBM stellt deren anfällige numerische Robustheit aufgrund einer expliziten Transitionsleichung zur Aktualisierung der Bahn dar. Diese führt unter starkem Hinderniseinfluss zu unvorhersehbaren Oszillationen oder gar zur Instabilität der Bahn und kann lediglich durch kleinere Schrittweiten sowie eine feinere Diskretisierung kompensiert werden, wodurch wiederum Konvergenz sowie Zeit- und Speicherdeterminismus leiden.

Während weiterführende Arbeiten das Stabilitätsproblem der EBM durch Optimierungsverfahren umgehen, wurde die ursprünglich sehr viel recheneffizientere und elegantere Idee der EBM, die Bahn mithilfe eines numerischen Integrationsverfahrens zu aktualisieren, seitdem nicht wieder aufgegriffen.

3 Voruntersuchung zur Elastic-Bands-Methode (EBM)

Im Stand der Technik wurde die Elastic-Bands-Methode (EBM) in ihrer ursprünglichen Formulierung (Quinlan et al. 1993) aufgrund ihrer Fähigkeit, globale und reaktive Planungselemente besonders effizient zu verbinden, als möglicher Bahnplanungsalgorithmus für die in Kap. 1.4 formulierte Problemstellung vorgestellt. Gleichzeitig konnten aber signifikante Defizite dieser Methode, insbesondere im Hinblick auf ihre numerische Stabilität, identifiziert werden. Diese sollen im Folgenden veranschaulicht werden.

Im Kern beschreibt die EBM (Quinlan et al. 1993) eine Bahn als Abfolge von n diskreten Stützstellen bzw. Konfigurationen \mathbf{p}_i , deren Lage unter Einfluss virtueller Kräfte an den aktuellen Umgebungszustand angepasst wird. Diese Kräfte werden zur Gesamtkraft $\mathbf{f}_{total} = k_c \cdot \mathbf{f}_c + k_r \cdot \mathbf{f}_r$ überlagert. Dabei ist \mathbf{f}_c eine annähernd normal gerichtete und krümmungsproportionale Rückstellkraft, welche die Spannung (Kontraktion) zwischen den Stützstellen modelliert. \mathbf{f}_r stellt die von den Hindernissen ausgehende Abstoßungskraft dar. Sie nimmt proportional mit dem Abstand zum nächsten Hindernis ab und zeigt in Richtung der größten Distanzabnahme. Die Verstärkungsfaktoren k_c und k_r gewichten und begrenzen diese Kräfte.

Die Aktualisierung der Lage erfolgt für jede einzelne Stützstelle gemäß der Berechnungsvorschrift $\mathbf{p}_{i,new} = \mathbf{p}_{i,old} + \Delta t \cdot \mathbf{f}_{total}$. Da die resultierende Gesamtkraft \mathbf{f}_{total} abhängig von der aktuellen Konfiguration $\mathbf{p}_{i,old}$ ist, entspricht dies einer stützstellenweisen numerischen Integration durch das explizite Eulerverfahren mit der Schrittweite Δt . Bekanntermaßen weist das explizite Eulerverfahren ein stark eingeschränktes Stabilitätsverhalten auf (Gipser 1999), welches aufgrund der nichtlinearen Formulierung der Kräfte in \mathbf{f}_{total} nicht analytisch bewertet werden kann.

Um die numerischen Defizite zu demonstrieren, wurden mehrere Experimente mit der gemäß ihrer Urfassung (Quinlan et al. 1993) implementierten EBM durchgeführt. Dabei wurde in einem ebenen Szenario eine mit $n = 10$ freien Stützstellen diskretisierte geradlinige Initialbahn der Länge $L_0 = 10 [m]$ durch zwei punktförmige Hindernisse, welche sich ge-

3 Voruntersuchung zur Elastic-Bands-Methode (EBM)

genläufig mit der Geschwindigkeit $v = 5 \text{ [m/s]}$ auf die Bahn zubewegen, verdrängt. Der Wirkradius der Hindernisse betrug dabei $r_0 = 2.5 \text{ [m]}$.

Abbildung 3 zeigt die Simulation dieses Szenarios zunächst mit einer vereinheitlichten Parametrierung von $k_c = k_r = 1$ und damit ohne Verstärkung der auf die Stützstellen wirkenden Kräfte.

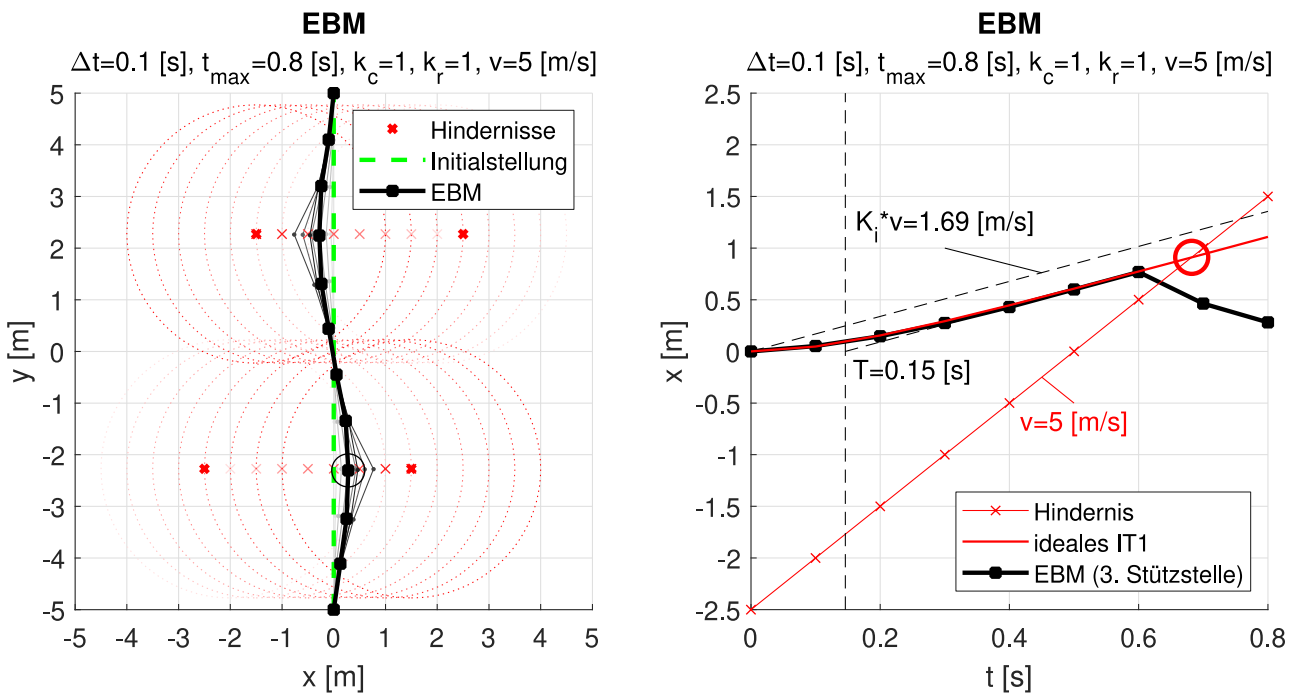


Abbildung 3: Kollision durch träges Ausweichverhalten der EBM

Dass in dem aufgezeigten Beispiel bei einer vergleichsweise großen Schrittweite von $\Delta t = 0.1 \text{ [s]}$ noch monoton stabil gelöst werden kann, ist der „gutmütigen“ Parametrierung ohne Verstärkung der Kräfte geschuldet, wodurch die Steifigkeit des Systems gering gehalten werden kann.

Gleichzeitig führt diese Parametrierung allerdings zu einem trägen Eigenverhalten der Bahn. Diese Eigendynamik kann repräsentativ mithilfe der Rampenantwort der in x -Richtung am stärksten ausgelenkten dritten Stützstelle x_3 (in Abbildung 3 links mit einem Kreis gekennzeichnet) auf die Lage des unteren Hindernisses veranschaulicht werden, was äquivalent zur Betrachtung eines Sprungs der Hindernisgeschwindigkeit v ist.

Der zeitliche Verlauf der Bewegung kann dabei durch ein ideales IT₁-Glied mit der Differentialgleichung $T \cdot \ddot{x}_3 + \dot{x}_3 = K_I \cdot v$ approximiert werden. Als Maß für dessen Trägheit wird die Verzögerungszeitkonstante T (Verzögerungsverhalten 1. Ordnung) sowie die Integrationsverstärkung K_I herangezogen. Diese beiden Kenngrößen bestimmen, wie schnell sich das Hindernis der Stützstelle nähert. Abbildung 3 zeigt, dass die gegebene Parametrierung aufgrund des großen Unterschiedes zwischen stationärer Ausweichgeschwindigkeit der Stützstelle und der Geschwindigkeit des Hindernisses ($K_I \cdot v = 1.69 \text{ [m/s]} \ll v = 5 \text{ [m/s]}$) bereits bei einer sehr geringeren Auslenkung von $x_3 = 0.91 \text{ [m]}$ zu einer Kollision zwischen Hindernis und Bahn (mit rotem Kreis gekennzeichnet) führt.

Diesem Effekt lässt sich nur durch eine dynamischere Parametrierung entgegenwirken. Abbildung 4 zeigt dies exemplarisch für $k_c = k_r = 10$.

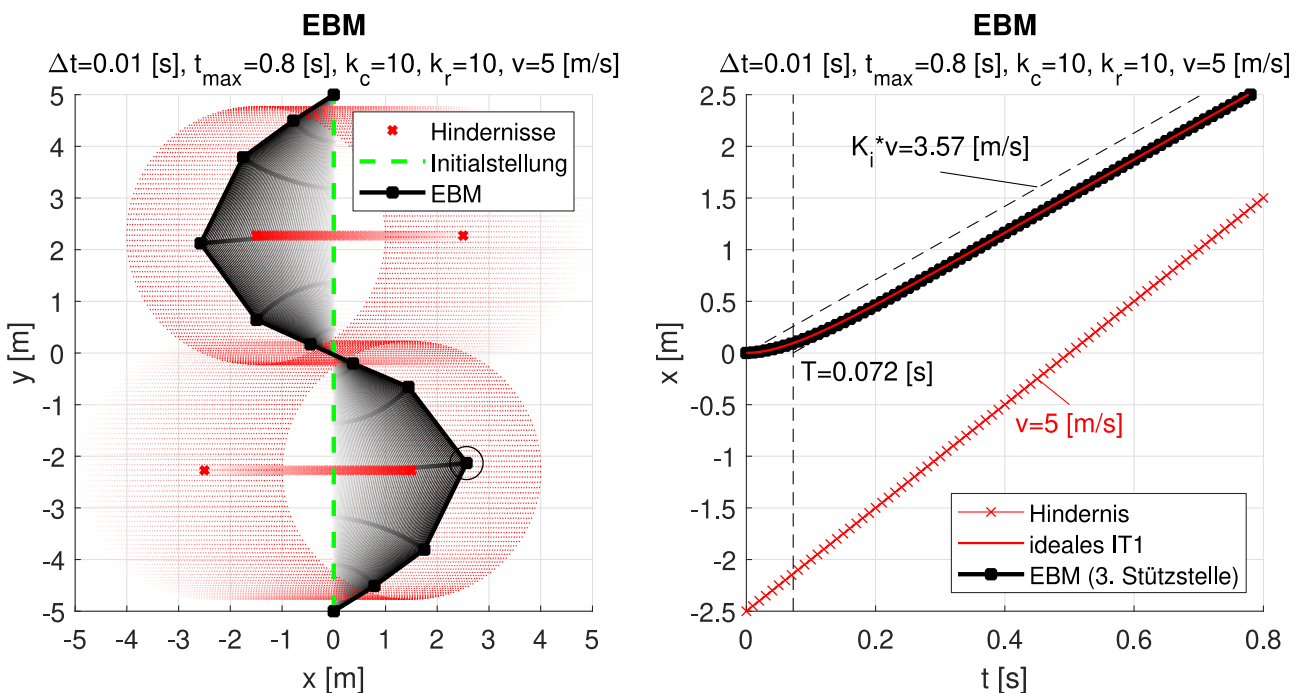


Abbildung 4: Dynamisches Ausweichverhalten der EBM durch steife Parametrierung

Es wird deutlich, dass durch diese Parametrierung der Unterschied der Geschwindigkeiten deutlich verringert werden kann ($K_I \cdot v = 3.57 \text{ [m/s]} < v = 5 \text{ [m/s]}$), sodass es (unter Beibehaltung des linearen IT₁-Verhaltens) erst bei $x_3 = 5.23 \text{ [m]}$ (im Schaubild nicht mehr dargestellt) zu einer Kollision zwischen Hindernis und Bahn kommen würde.

Eine solche dynamische Parametrierung erhöht allerdings gleichzeitig die Steifigkeit des Systems, sodass dieses unter Beibehaltung der ursprünglichen Zeitschrittweite von $\Delta t = 0.1$ [s] nicht mehr numerisch stabil gelöst werden kann und daher auf $\Delta t = 0.01$ [s] reduziert wurde.

Es konnte empirisch ermittelt werden, dass dieses Szenario für die Parametrierung $k_c = k_r = 10$ mithilfe des expliziten Eulerverfahrens lediglich bis zu einer Schrittweite von etwa $\Delta t = 0.02$ [s] monoton stabil gelöst wird. Zwischen $\Delta t = 0.02$ [s] und $\Delta t = 0.08$ [s] stellt sich ein oszillatorisch stabiles und darüber ein instabiles Lösungsverhalten ein. Abbildung 5 zeigt die bereits instabilen Schrittweiten $\Delta t = 0.1$ [s] und $\Delta t = 0.25$ [s].

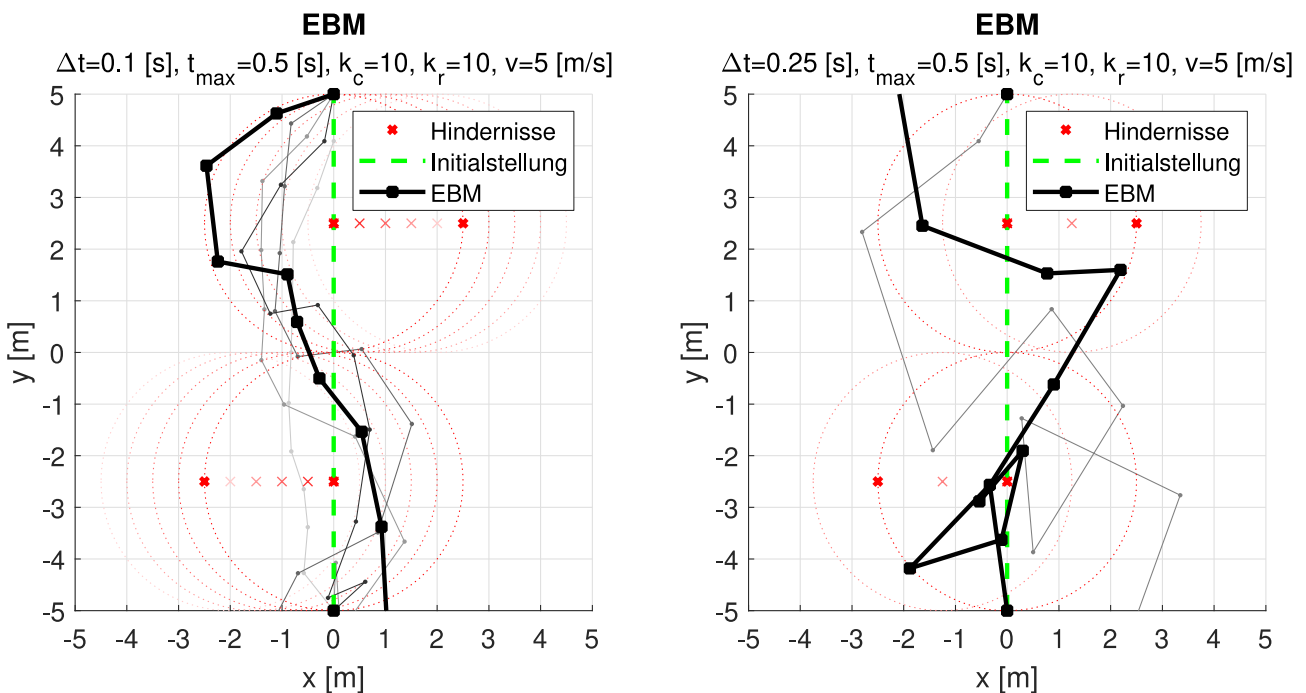


Abbildung 5: Instabiles Lösungsverhalten der EBM für große Schrittweiten

Die vorangegangenen numerischen Experimente zeigen klar die Grenzen der numerischen Robustheit der EBM mit der ursprünglich vorgeschlagenen zeitlichen Entwicklung der Bahn mittels numerischer Zeitintegration auf. Die für eine stabile Lösung erforderlichen Schrittweiten sind aufgrund des eingeschränkten Stabilitätsgebietes des verwendeten expliziten Eulerverfahrens stark abhängig von der erforderlichen Bahndynamik.

Träges Eigenverhalten der Bahn kann zwar noch bei vergleichsweise großen Schrittweiten stabil gelöst werden, es konnte aber gezeigt werden, dass die Reaktionsfähigkeit und damit die Sicherheit vor Kollisionen bei dynamischen Hindernissen mit für UAV typischen Geschwindigkeiten nicht mehr gewährleistet werden kann.

Eine höhere Bahndynamik geht mit einer höheren Systemsteifigkeit einher, was kleine Schrittweiten im Bereich von wenigen Millisekunden erfordert und eine zeitdeterministische Echtzeitimplementierung für Bahnen mit vielen Stützstellen, insbesondere auf UAV-typischer Hardware mit geringer Rechenleistung (z.B. Mikrocontroller), schwierig macht.

Darüber hinaus kann die Stabilität aufgrund der nichtlinearen Formulierung der Kraftterme lediglich empirisch ermittelt werden. Ein allgemeingültiger analytischer Nachweis bzw. die Vorhersage der Stabilität unabhängig von Diskretisierung, Parametrierung und Betriebspunkt ist nicht möglich.

4 Kurvenflussmethode (KFM)

Motiviert durch die Erkenntnisse aus Kap. 3 soll mit der Kurvenflussmethode (KFM) eine neue Bahnplanungsmethode entwickelt werden, welche einerseits das Funktionsprinzip der EBM zum Vorbild nimmt, gleichzeitig aber deren Defizite behebt. Der Name dieser neuen Methode geht dabei auf den geometrischen Kurvenfluss aus der Differentialgeometrie zurück. Dieser bildet den theoretischen Ursprung der Methode und wird später in Kap. 4.2 noch näher erläutert.

Die KFM soll ebenso wie die EBM in der Lage sein, eine beliebig diskretisierte Bahn unter Einfluss virtueller Kräfte online an eine dynamische Umgebung mit bewegten Hindernissen anzupassen und dabei eine parametrierbare Gewichtung zwischen Bahnlänge und Hindernisabstand erlauben. Entsprechend der Anforderungen soll die numerische Berechnung der Bahn sowohl recheneffizient und echtzeitfähig als auch numerisch stabil sein.

Zusätzlich sollen die abstrakten Parameter der EBM mit systemdynamischen Analysemethoden in einen physikalischen Zusammenhang zum Bahnverhalten gebracht werden, sodass die Parametrierung nicht empirisch, sondern auf Basis physikalischer Vorgaben und unabhängig von der Diskretisierung erfolgen kann.

4.1 Prinzip der KFM

Analog zum Funktionsprinzip der EBM beschreibt die KFM die Bahn zwischen einem Start- und einem Zielpunkt als diskrete Abfolge von n Stützstellen p_i , deren Lage unter Einfluss von virtuellen Kräften kontinuierlich an den aktuellen Umgebungszustand angepasst wird. Abbildung 6 illustriert das Wirkprinzip der KFM. Von Hindernissen ausgehende Abstoßungskräfte f_i verdrängen die Bahn, während krümmungsproportionale Rückstellkräfte k_i die Bahn gleichzeitig in einen Zustand minimaler Länge zwingen. Die Bahn konvergiert dabei in Richtung einer Gleichgewichtslage, welche ein Paretooptimum zwischen sicherem Abstand zum Hindernis und geringer Bahnlänge darstellt. T und K stellen dabei die Parameter zur Beeinflussung des Bahnverhaltens (Dynamik und stationäre Endlage) dar.

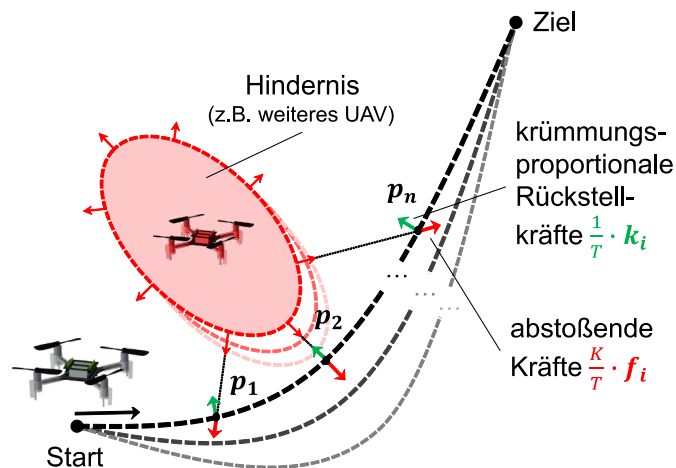


Abbildung 6: Wirkprinzip der KFM

Im Rahmen dieser Arbeit wird die KFM ausschließlich für ebene ($p \in \mathbb{R}^2$) und räumliche ($p \in \mathbb{R}^3$) Probleme im euklidischen Raum ($p \in \mathbb{R}^D$) angewandt. Grundsätzlich ist unter Voraussetzung einer entsprechenden Beschreibung der abstoßenden Kräfte eine Anwendung für Probleme in Konfigurationsräumen beliebiger Dimension D denkbar.

Die KFM wird im Folgenden zunächst ohne Hindernisse ($K = 0$) aus dem geometrischen Kurvenfluss hergeleitet und das Eigenverhalten der Bahn (Rückstellung) wird dynamikneutral ($T = 1$) analysiert. Darauf aufbauend wird das Bahnverhalten mit Hindernissen (Abstoßung für $K > 0$) sowie der Einfluss der (abstrakten) Parameter T und K untersucht, um daraus schließlich eine Methodik für die Parametrierung der KFM anhand physikalischer Kenngrößen abzuleiten. Diese wird angewandt, um abschließend das numerische Experiment mit der EBM aus der Voruntersuchung (vgl. Kap. 3) nochmals vergleichend mit der KFM durchführen zu können.

4.2 Nichtlineare Kurvenflüsse als Ursprung der KFM

Entgegen der EBM oder verwandter Methoden ist die KFM nicht aus den physikalischen Materialeigenschaften eines elastischen Bandes (Quinlan et al. 1993) oder der mechanischen Reihenschaltung von Feder-Masse-Systemen (Bortoff 2000) hergeleitet. Vielmehr geht die KFM auf die aus der Differentialgeometrie bekannten geometrischen Kurvenflüsse zurück, welche die geometrische Entwicklung einer geschlossenen kontinuierlichen Kurve über die Zeit beschreiben (Chou et al. 2001; Devadoss et al. 2011).

Erste ingenieurstechnische Anwendung fanden Kurvenflüsse als Modell für die Reduktion von Kornrenzflächen beim Glühen von dünnen Metallblechen (Mullins 1956; Mantegazza et al. 2018). Weitere anwendungsorientierte Problemstellungen, welche mit Kurvenflüssen im \mathbb{R}^2 gelöst werden, sind beispielsweise in der digitalen Bildverarbeitung bei der Umrissglättung für variierende Abbildungsmaßstäbe (Mokhtarian et al. 1992), der Architektur beim Design von Freiformfassaden (Pottmann et al. 2007) oder auch bei der Lösung des Rendezvous-Problems für mobile Roboter (Smith et al. 2007) zu finden.

Mathematisch kann das Verhalten der parametrischen Kurve \mathbf{p} durch folgende homogene partielle Differentialgleichung (DGL) beschrieben werden:

$$\dot{\mathbf{p}}(s, t) = \mathbf{p}''(s, t) = \kappa(\mathbf{p}(s, t)) \cdot \mathbf{n}(\mathbf{p}(s, t)) \quad (1)$$

Dabei ist \mathbf{p} eine von der Zeit t und dem Bahnparameter $s \in [0,1]$ abhängige geschlossene ($\mathbf{p}(s=0, t) = \mathbf{p}(s=1, t)$) Kurve im \mathbb{R}^2 und stellt zugleich die Zustandsgröße der DGL dar. Deren partielle Zeitableitung $\dot{\mathbf{p}}$ entspricht in jedem Bahnpunkt der partiellen Ortsableitung 2. Ordnung \mathbf{p}'' , wodurch Gl. (1) eine spezielle Form einer parabolischen partiellen DGL zur Beschreibung von Evolutionsproblemen darstellt. Anschaulich entspricht dieser Zusammenhang einer Verschiebung (zeitliche Entwicklung) aller Bahnpunkte in Richtung der nach innen gerichteten Kurvennormalen \mathbf{n} mit der Krümmung κ als Proportionalitätsfaktor (Devadoss et al. 2011). Der Krümmungsterm $\kappa \cdot \mathbf{n}$ wirkt dabei wie eine virtuelle Rückstellkraft, welche die Kurve in Richtung ihres energieminimalen Zustandes treibt.

Obgleich Kurvenflüsse ursprünglich lediglich zur Beschreibung ebener Kurven ($\mathbf{p} \in \mathbb{R}^2$) verwendet wurden, kann die Rückstellkraft (der Krümmungsterm $\kappa \cdot \mathbf{n}$) mithilfe der frenet'schen Formeln zur Beschreibung eines begleitenden Drei- oder D -Beins ebenfalls für räumliche Kurven ($\mathbf{p} \in \mathbb{R}^3$) oder Kurven beliebiger Dimension D ($\mathbf{p} \in \mathbb{R}^D$) verallgemeinert werden (Altschuler 1991; Kühnel 2013). Da die analytische Beschreibung der Rückstellkraft bereits für $D \geq 2$ nichtlineare Operationen der Zustandsgrößen (also der partiellen Ortsableitungen 1. und 2. Ordnung) erfordert, stellt Gl. (1) eine nichtlineare partielle DGL dar. Nachfolgend ist Gl. (1) exemplarisch für eine Kurve in der euklidischen Ebene ($D = 2$) mit $\mathbf{p}(s, t) = (x(s, t), y(s, t))^T$ ausgeschrieben. Die Abhängigkeit der Zustandsgrößen von s und t ist zugunsten der Lesbarkeit nicht mehr explizit dargestellt.

$$\underbrace{\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}}_{\dot{\mathbf{p}}} = \frac{x' \cdot y'' - x'' \cdot y'}{(x'^2 + y'^2)^2} \cdot \underbrace{\begin{pmatrix} -y' \\ x' \end{pmatrix}}_{\mathbf{n}(\mathbf{p})} \quad (2)$$

Einen klassischen Ansatz zur numerischen Lösung einer solchen partiellen DGL stellt die Finite-Differenzen-Methode (FDM) dar. Dabei werden die analytischen Ortsableitungen der kontinuierlichen Kurve $\mathbf{p}(s, t)$ an n diskreten Stützstellen $\mathbf{p}_i(t)$ durch Differenzenquotienten 1. und 2. Ordnung approximiert. Die partielle DGL nach Gl. (2) kann dadurch in ein System von n gekoppelten nichtlinearen gewöhnlichen DGLs 1. Ordnung überführt und durch numerische Zeitintegration gelöst werden. Jede DGL stellt dabei die zeitliche Entwicklung einer Stützstelle $\mathbf{p}_i(t)$ des nunmehr als Polygonzug beschriebenen diskreten Kurvenflusses in Abhängigkeit ihrer Nachbarstützstellen $\mathbf{p}_{i+1}(t)$ und $\mathbf{p}_{i-1}(t)$ dar.

Eine Variante der KFM auf Basis dieses nichtlinearen diskreten Kurvenflusses wurde bereits mehrfach vorgestellt (Huptych et al. 2011a; Röck et al. 2010; Huptych et al. 2011b; Huptych et al. 2012). Die numerisch robuste und gleichzeitig recheneffiziente Lösung sowie die systemtheoretische Analyse stellt sich hier allerdings, wie auch bei der EBM, aufgrund der zugrundeliegenden nichtlinearen DGL schwierig dar. Im Rahmen dieser Arbeit wird diese Variante daher nicht weiterverfolgt. Stattdessen soll der durch die rechte Seite von Gl. (2) exakt beschriebene nichtlineare Krümmungsterm durch ein lineares Ersatzmodell, welches dessen Verhalten noch hinreichend genau wiedergibt, approximiert werden.

4.3 Ortsdiskretisierung

Einen eleganten und häufig verwendeten Ansatz für die lineare Approximation des Krümmungsterms stellt die in Abbildung 7 dargestellte Summe \mathbf{k}_i der beiden zu den Nachbarstützstellen zeigenden Vektoren dar (Chow et al. 2007; Devadoss et al. 2011).

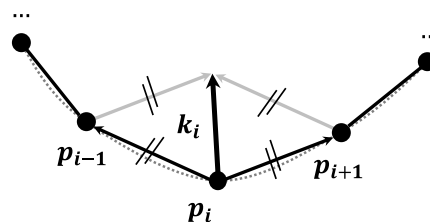


Abbildung 7: Approximation der nichtlinearen Rückstellkraft mit linearer Vektoraddition

Damit lässt sich die nichtlineare diskrete Kurvenfluss-DGL aus Gl. (1) und Gl. (2) an einer einzelnen Stützstelle \mathbf{p}_i wie folgt approximieren:

$$\dot{\mathbf{p}}_i = \mathbf{k}_i = (\mathbf{p}_{i+1} - \mathbf{p}_i) + (\mathbf{p}_{i-1} - \mathbf{p}_i) = \mathbf{p}_{i+1} - 2 \cdot \mathbf{p}_i + \mathbf{p}_{i-1} \approx \kappa(\mathbf{p}_i) \cdot \mathbf{n}(\mathbf{p}_i) \quad (3)$$

Durch die Unabhängigkeit der Zustandsgrößen für die einzelnen Raumrichtungen kann eine Kurve beliebiger Dimension D (hier weiterhin für $D = 2$ dargestellt) entgegen der nichtlinearen Formulierung stets komponentenweise entkoppelt dargestellt werden:

$$\begin{aligned} \dot{x}_i &= x_{i+1} - 2 \cdot x_i + x_{i-1} \\ \dot{y}_i &= y_{i+1} - 2 \cdot y_i + y_{i-1} \end{aligned} \quad (4)$$

Ausgeschrieben lässt sich Gl. (4) somit für eine Kurve $\mathbf{p}(t)$ mit n Stützstellen komponentenweise in Form von D homogenen linearen Zustandsraummodellen in Matrixform $(\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}, \dot{\mathbf{y}} = \mathbf{A} \cdot \mathbf{y}, \dots)^T$ darstellen. Aufgrund der Linearität kann das Eigenverhalten (Wirkung des Zustands auf dessen Änderung) des Systems in der konstanten Systemmatrix \mathbf{A} zusammengefasst werden. Dabei ist zu beachten, dass bei einer geschlossenen Kurve mit n Stützstellen für die Bestimmung von \mathbf{k}_1 und \mathbf{k}_n nach Gl. (3) eine kongruente Überlappung der erweiterten Kurvenränder angenommen werden muss, sodass $\mathbf{p}_0(t) = \mathbf{p}_n(t)$ und $\mathbf{p}_{n+1}(t) = \mathbf{p}_1(t)$ gilt. Wegen der Gleichartigkeit der Systeme für alle Raumrichtungen ist nachfolgend lediglich die x -Komponente dargestellt.

$$\underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-2} \\ \dot{x}_{n-1} \\ \dot{x}_n \end{pmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix}}_{\mathbf{x}} \quad (5)$$

Gegenüber der nichtlinearen DGL, welche den diskreten Kurvenfluss exakt beschreibt, besitzt diese linearisierte Formulierung einige Vorteile. So erlaubt sie die Anwendung der Systemtheorie für lineare zeitinvariante Systeme (LZI-Systeme) zur Analyse des Kurvenverhaltens (vgl. Kap. 4.5 und Kap. 4.7). Zusätzlich kann durch die Unabhängigkeit der einzelnen Raumrichtungen die zur Lösung der linearen Kurvenfluss-DGL notwendige

Systemgröße von einem einzelnen System der Größe $(n \cdot D) \times (n \cdot D)$ zu D entkoppelten Systemen der Größe $n \times n$ reduziert werden.

Außerdem umgeht diese Formulierung ein signifikantes Problem der nichtlinearen Rückstellkraft. Da letztere ausschließlich normal zur Bahn wirkt, verursacht dies eine Häufung von initial äquidistant verteilten Stützstellen in Bereichen hoher Krümmung. Dies ist ein Effekt, von welchem die EBM ebenfalls betroffen ist und welcher durch eine zusätzliche Verschiebung der Stützstellen tangential zur Bahn kompensiert werden muss (Quinlan et al. 1993; Quinlan 1994). Im Zusammenhang mit der nichtlinearen KFM wurde dieses Verhalten ebenfalls untersucht (Huptych et al. 2011a). Da die lineare Rückstellkraft nach Gl. (3) eine Normalen- sowie eine Tangentialwirkung aufweist (vgl. Abbildung 8), konvergieren die Stützstellen automatisch in eine äquidistante Verteilung entlang der Bahn.

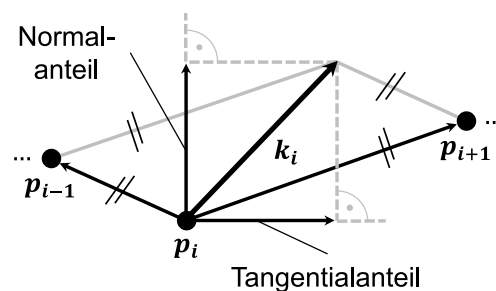


Abbildung 8: Normal- und Tangentialwirkung der linearen Rückstellkraft k_i

4.4 Kurvenfluss für offene Kurven

Ursprünglich wurden mit Kurvenflüssen lediglich geschlossene Kurven beschrieben. Für die Entwicklung einer UAV- bzw. Roboterbahn zwischen zwei beliebigen Endpunkten $\mathbf{p}_s = (x_s, y_s, \dots)^T$ (Start) und $\mathbf{p}_g = (x_g, y_g, \dots)^T$ (Ziel, engl. goal) können Gl. (3) bis Gl. (5) ebenfalls für die Beschreibung einer offenen Kurve mit $\mathbf{p}_0(t) = \mathbf{p}_s$ und $\mathbf{p}_{n+1}(t) = \mathbf{p}_g$ angewandt werden. Obgleich im Rahmen der KFM lediglich statische Endpunkte verwendet werden, können diese grundsätzlich ebenso dynamisch sein. Die Kurve konvergiert dadurch (bei statischen Endpunkten) nunmehr in eine krümmungsfreie stabile Gleichgewichtslage anstatt in einen instabilen singulären Punkt. Die Geschwindigkeit aller Stützstellen beträgt hier $\dot{\mathbf{p}}_i = \mathbf{0}$ und die Kurve beschreibt die geradlinige und damit einhergehend kürzestmögliche Verbindung zwischen den Endpunkten \mathbf{p}_s und \mathbf{p}_g . In Ma-

trixform kann diese Randbedingung komponentenweise mithilfe von D (konstanten) Eingangsvektoren $(\mathbf{b}_x = (x_s, 0, \dots, 0, x_g)^T, \mathbf{b}_y = (y_s, 0, \dots, 0, y_g)^T, \dots)^T$ sowie einer Anpassung der Systemmatrix A berücksichtigt werden. Auch hier ist aufgrund der Gleichartigkeit der Systeme für alle Raumrichtungen lediglich die x -Komponente dargestellt:

$$\underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-2} \\ \dot{x}_{n-1} \\ \dot{x}_n \end{pmatrix}}_x = \underbrace{\begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix}}_x + \underbrace{\begin{pmatrix} x_s \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ x_g \end{pmatrix}}_{b_x} \quad (6)$$

Abbildung 9 zeigt in einer exemplarischen Simulation die ebene ($D = 2$) Entwicklung einer geschlossenen Kurve nach Gl. (5) gegenüber einer offenen Kurve nach Gl. (6). Diese bestätigen das gewünschte Kurvenverhalten und es kann dabei ebenfalls beobachtet werden, wie die zufällig gesetzten Stützstellen der Initialstellung in eine äquidistante Verteilung entlang der Bahn übergehen.

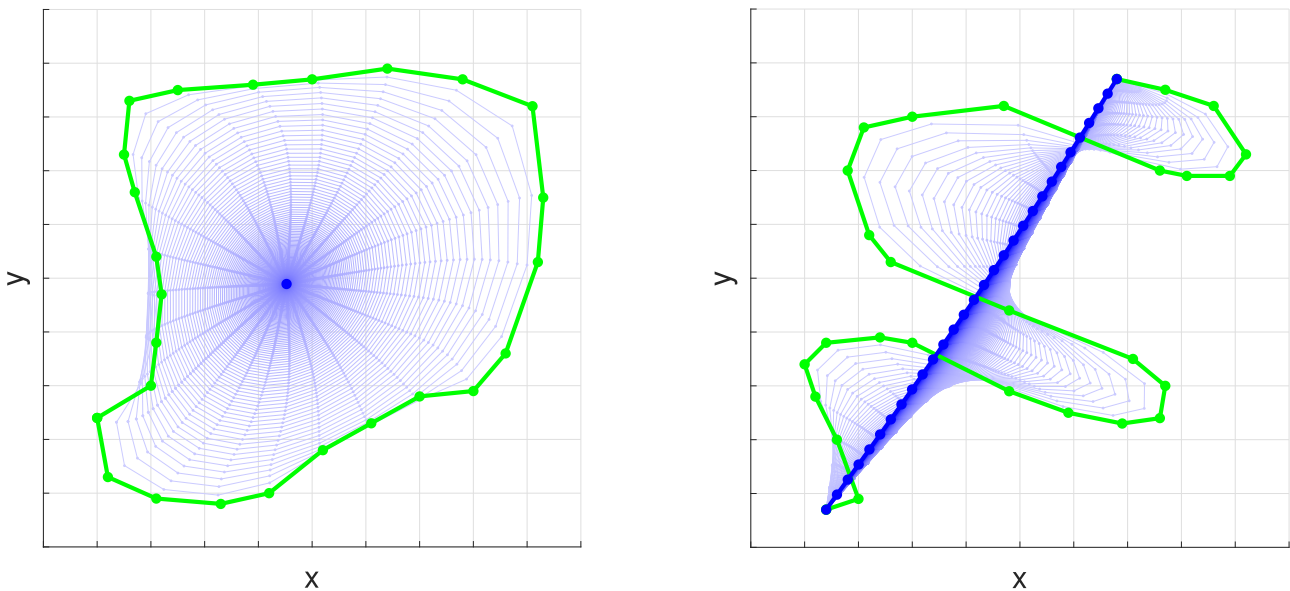


Abbildung 9: Entwicklung einer geschlossenen und einer offenen Kurve

4.5 Dynamik des Kurvenflusses

Anhand der Eigenwerte der Systemmatrix A lassen sich die dynamischen Eigenschaften des Kurvenflusses (Eigenverhalten) analysieren. A besitzt ausschließlich jeweils identische Einträge auf der Hauptdiagonalen ($a_d = -2$) sowie auf der ersten unteren und der ersten oberen Nebendiagonalen ($a_l = a_u = 1$, engl. „lower“ bzw. „upper“). Dadurch stellt sie eine tridiagonale Toeplitz-Matrix dar, deren n von der Diskretisierung abhängigen Eigenwerte λ_i (mit $i = 1, 2, \dots, n$) explizit angegeben werden können (Noschese et al. 2013):

$$\lambda_i = a_d + 2 \cdot \sqrt{a_l \cdot a_u} \cdot \cos\left(\frac{\pi \cdot i}{n+1}\right) = -2 + 2 \cdot \cos\left(\frac{\pi \cdot i}{n+1}\right) \quad (7)$$

Somit liegen alle Eigenwerte von A stets auf der reellen Achse. Deren Wertebereich wird dabei symmetrisch vom betragsmäßig kleinsten Eigenwert λ_1 sowie vom betragsmäßig größten Eigenwert λ_n (entspricht dem Spektralradius) begrenzt. Für $n \rightarrow \infty$ ergeben sich diese Grenzwerte zu $\lambda_1 \rightarrow 0$ und $\lambda_n \rightarrow -4$. In Abbildung 10 ist die Spanne der Eigenwerte in Abhängigkeit der Diskretisierung n dargestellt.

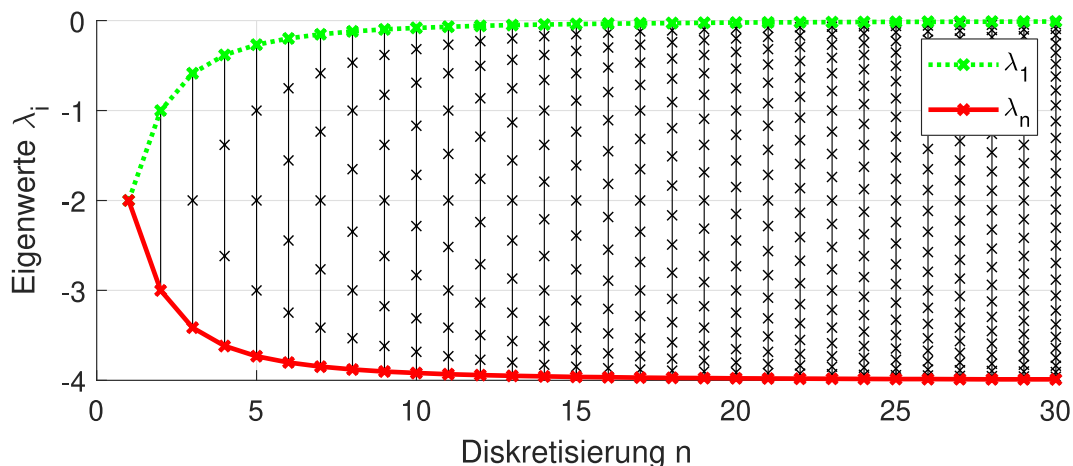


Abbildung 10: Eigenwerte der Systemmatrix A in Abhängigkeit der Diskretisierung n

Daraus lassen sich folgende Eigenschaften für den diskretisierten Kurvenfluss ableiten:

- Das System ist stets stabil, da der Realanteil aller Eigenwerte immer kleiner Null ist.
- Das System bzw. die einzelnen Stützstellen sind nicht eigenschwingfähig, da alle Eigenwerte stets reellwertig sind.

- Das System kann somit als Reihenschaltung von n PT₁-Gliedern aufgefasst werden, wodurch jede einzelne Stützstelle im Kern Verzögerungsverhalten n -ter Ordnung aufweist.

Darüber hinaus kann anhand der Spanne der Eigenwerte auf der Realachse die Dynamik des Eigenverhaltens (Rückstellung zur Geraden) abgeschätzt werden. Der betragsmäßig kleinste Eigenwert λ_1 legt den dominierenden trägsten Eigenvorgang fest, welcher anteilig auf alle n Stützstellenpositionen (Zustände) wirkt.

Die sich für reine PT₁-Glieder aus dem minimalen Eigenwert λ_1 ergebende Beruhigungs- bzw. Stellzeit $T_s = 3/|\lambda_1|$ (Zeit zum Erreichen von ca. 95 Prozent der stationären Endlage) kann in guter Näherung ebenfalls als Maß für die Gesamtdynamik der Bahn herangezogen werden. Tabelle 3 zeigt exemplarisch für einige Bahndiskretisierungen n die sich ergebenden Stellzeiten T_s .

n	1	2	3	4	5	10	20	30
λ_1	-2.000	-1.000	-0.586	-0.382	-0.268	-0.081	-0.022	-0.010
T_s in [s]	1.500	3.000	5.121	7.854	11.196	37.031	134.298	292.359

Tabelle 3: Betragskleinster Eigenwert λ_1 und Stellzeit T_s für einige Diskretisierungen n

In den folgenden numerischen Experimenten soll der Zusammenhang zwischen der Stellzeit T_s und der Diskretisierung n veranschaulicht werden. Aufgrund der Linearität und Unabhängigkeit der Teilsysteme für die einzelnen Raumrichtungen ist es ausreichend die Auslenkung der Stützstellen in lediglich eine Raumrichtung (hier in y -Richtung) zu betrachten.

Um das dynamische Verhalten der einzelnen Stützstellen systemtheoretisch analysieren zu können, wurde in Abbildung 11 einer in Ruhe befindlichen geradlinigen Initialbahn bei verschiedenen Diskretisierungen n ein Einheitssprung der Zielkoordinate y_g aufgebracht. Dabei wurden simulativ die individuellen Stellzeiten der einzelnen Stützstellen $T_{s,i}$ sowie deren Durchschnittswert $T_\emptyset = (1/n) \cdot \sum_{i=1}^n T_{s,i}$ als Maß für die Gesamtdynamik der Bahn bestimmt.

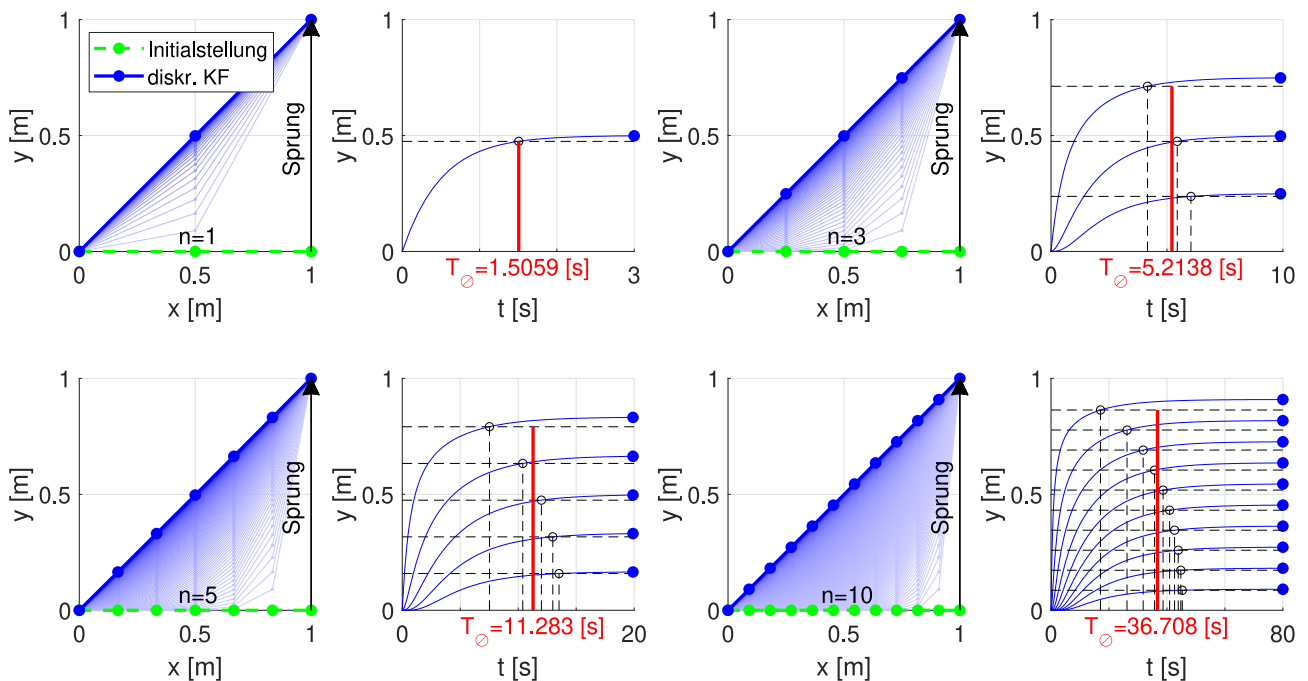


Abbildung 11: Kurvenflusssimulationen bei einem Sprung des Zielpunktes y_g

Aus der mit steigendem Stützstellenindex i zunehmend ausgeprägteren S-Form der Antworten ist zu erkennen, wie scheinbar die Verzögerungsordnung der Stützstellen und damit auch deren individuelle Stellzeit $T_{s,i}$ zunimmt. Die durchschnittliche Stellzeit T_\emptyset stimmt dabei mit den in Tabelle 3 analytisch prognostizierten Werten überein und macht deutlich, dass mit steigender Diskretisierung n die Gesamtdynamik der Bahn sinkt.

Um die individuellen Verläufe der einzelnen Stützstellen besser nachvollziehen zu können, wurde aus dem Zustandsraummodell in Gl. (6) das Übertragungsverhalten der Stützstellenkoordinaten y_i in Abhängigkeit der Zielkoordinate y_g sowie der Diskretisierung n analytisch ermittelt. Tabelle 4 zeigt die sich im Bildraum ergebenden Übertragungsfunktionen $G(s) = y_i(s)/y_g(s)$.

4 Kurvenflussmethode (KFM)

Ausg. y_i	Übertragungsfunktion $G(s) = y_i(s)/y_g(s)$			
	$n = 1$	$n = 2$	$n = 3$	$n = 4$
y_1	$\frac{1}{s+2}$	$\frac{1}{s^2+4s+3}$	$\frac{1}{s^3+6s^2+10s+4}$	$\frac{1}{s^4+8s^3+21s^2+20s+5}$
y_2	-	$\frac{s+2}{s^2+4s+3}$	$\frac{s+2}{s^3+6s^2+10s+4}$	$\frac{s+2}{s^4+8s^3+21s^2+20s+5}$
y_3	-	-	$\frac{s^2+4s+3}{s^3+6s^2+10s+4}$	$\frac{s^2+4s+3}{s^4+8s^3+21s^2+20s+5}$
y_4	-	-	-	$\frac{s^3+6s^2+10s+4}{s^4+8s^3+21s^2+20s+5}$
Polstellen	$s_1 = -2$	$s_{1/2} = -2 \mp 1$	$s_2 = -2$ $s_{1/3} = -2 \mp 1.4142$	$s_{1/4} = -2 \mp 1.6180$ $s_{2/3} = -2 \mp 0.6180$
Nullstellen	Nullstellen für Ausgang y_i sind identisch mit Polstellen für Ausgänge bei $n = i - 1$ (mit $i \geq 2$).			

Tabelle 4: Übertragungsfunktionen der Stützstellenkoordinaten y_i

Die Polstellen (Nullstellen des Nennerpolynoms) entsprechen dabei, wie erwartet, den nach Gl. (7) ermittelten Eigenwerten und bestimmen das für alle Stützstellen gleiche Verzögerungsverhalten n -ter Ordnung. Mit steigendem Stützstellenindex i (Nähe zum Ort der Sprungeinleitung) ergänzen rein reelle Nullstellen das Übertragungsverhalten um differenzierende Anteile bis zur Ordnung $i - 1$. Aufgrund des geringen Abstands zwischen Pol- und Nullstellen auf der Realachse der komplexen Zahlenebene wird das Verzögerungsverhalten zunehmend kompensiert. Für die i -te Stützstelle liegt somit ein $PD_{i-1}T_n$ -Verhalten vor, welches für $i \rightarrow n$ näherungsweise in PT_1 -Verhalten übergeht.

Zusammenfassend lassen sich daraus folgende Erkenntnisse zur Dynamik einer auf dem diskreten Kurvenfluss basierenden Bahn festhalten:

- Die Trägheit der Bahn nimmt mit steigender Diskretisierung n zu.
- „Anregungsnahe“ Stützstellen besitzen aufgrund der engsten Kopplung zum Eingang die höchste Dynamik.
- Anhand des betragsmäßig kleinsten Eigenwerts λ_1 der Systemmatrix A kann die durchschnittliche Stellzeit T_s als Maß für die Gesamtdynamik der Bahn in Abhängigkeit der Diskretisierung n herangezogen werden.

4.6 Kurvenfluss mit Hindernissen

Nun soll der vorgestellte diskrete Kurvenfluss gemäß der Problemstellung für die Beschreibung einer Bahn unter Berücksichtigung der verdrängenden Wirkung von Hindernissen zur Abbildung der Umgebung erweitert werden. Dazu wird die rechte Seite der linearen Kurvenfluss-DGL (3), welche bisher lediglich den Krümmungsterm als rückstellende Kraft beschrieben hat, zusätzlich um eine an jeder Stützstelle \mathbf{p}_i angreifende Abstoßungskraft \mathbf{f}_i ergänzt:

$$\dot{\mathbf{p}}_i = \underbrace{\mathbf{k}_i(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})}_{\text{Rückstellkraft}} + \underbrace{\mathbf{f}_i(\mathbf{p}_i, \mathbf{h})}_{\text{Abstoßungskraft}} \quad (8)$$

Die Kräfte \mathbf{f}_i müssen dabei die sich einstellende Gleichgewichtslage der Bahn derart beeinflussen, dass sie in sicherem Abstand an den in \mathbf{h} definierten Hindernissen vorbeiführt. Typischerweise wird das zugrundeliegende Kraftgesetz derart formuliert, dass der Betrag von \mathbf{f} mit steigender Distanz zu den Hindernissen abnimmt (Abstoßungsstärke) und \mathbf{f} dabei stets in Richtung der größten Distanzabnahme zeigt (Abstoßungsrichtung). Im Rahmen der KFM wird das Kraftgesetz noch weiter verallgemeinert:

$$\mathbf{f}_i(\mathbf{p}_i, \mathbf{h}) = \underbrace{\rho(H(d(\mathbf{p}_i, \mathbf{h})))}_{\text{Abstoßungsstärke}} \cdot \underbrace{-\frac{\nabla\rho(\mathbf{p}_i, \mathbf{h})}{|\nabla\rho(\mathbf{p}_i, \mathbf{h})|}}_{\text{Abstoßungsrichtung}} \quad (9)$$

Dabei ist d eine Metrik zur Beschreibung der Distanzen zwischen \mathbf{p}_i und den m in $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_m)$ definierten Hindernissen. Das aus den Einzeldistanzen d_j resultierende Maß für die an der Stützstelle \mathbf{p}_i repräsentativ wirkende Gesamtdistanz bezüglich aller Hindernisse ist über die Normierungsvorschrift H definiert. Die Gewichtungsfunktion ρ stellt schließlich den Zusammenhang zwischen wachsender Abstoßungsstärke bei abnehmender Normdistanz H her. Während ρ als skalare Größe den Betrag von \mathbf{f}_i und damit die Abstoßungsstärke bestimmt, definiert der auf Einheitslänge normierte und negierte Gradient $\nabla\rho(\mathbf{p}_i, \mathbf{h}) = \partial\rho(\mathbf{p}_i, \mathbf{h})/\partial\mathbf{p}$ die Abstoßungsrichtung. Diese steht stets orthogonal zum von \mathbf{f} erzeugten Kraftfeld und zeigt vom Hindernis weg, was der Richtung der maximalen Distanzzunahme entspricht.

Die Funktionen d , H und ρ beschreiben somit ein problemspezifisches Modell der Umgebung. Entgegen dem Eigenverhalten der Bahn (Rückstellkräfte) stellen sie keinen universell gültigen algorithmischen Bestandteil der KFM (sowie auch der EBM) dar, sondern sind individuell zu implementieren.

In Abhängigkeit von Anzahl und Detailgrad der in \mathbf{h} modellierten Hindernisse kann die Berechnung von \mathbf{f} dabei beliebig komplex werden und den Rechenaufwand für die eigentliche Aktualisierung der Bahn, welcher ausschließlich von der Diskretisierung n und der Anzahl an Dimensionen D abhängt, schnell übersteigen. Für kurze Aktualisierungsintervalle ist daher ein möglichst recheneffizientes Umgebungsmodell anzustreben.

Um später einen umgebungsunabhängigen Vergleich zwischen KFM und EBM herstellen zu können, wird zunächst das auf lediglich punktförmige Hindernisse angewandte Kraftmodell der EBM (Quinlan et al. 1993) aus Kap. 3 beibehalten. Ein Modell für die KFM, welches die Umgebung gemäß Problemstellung präziser abbildet, wird später in Kap. 5.2 vorgestellt. Die formale Separierung dieses einfachen Modells in die Funktionen d , H und ρ erfolgt gemäß Gl. (10) bis Gl. (12). Dabei beschreibt r_0 den Wirkradius der Hindernisse und entspricht gleichzeitig dem maximal möglichen Betrag der Abstoßungskraft.

$$d_j(\mathbf{p}_i, \mathbf{h}_j) = |\mathbf{p}_i - \mathbf{h}_j| \quad (10)$$

$$H(d(\mathbf{p}_i, \mathbf{h})) = \min(d_1(\mathbf{p}_i, \mathbf{h}_1), \dots, d_m(\mathbf{p}_i, \mathbf{h}_m)) \quad (11)$$

$$\rho(H(\mathbf{p}_i, \mathbf{h})) = \begin{cases} r_0 - H, & H < r_0 \\ 0, & H \geq r_0 \end{cases} \quad (12)$$

In Abbildung 12 wurde beispielhaft ein Kraftfeld nach diesem Modell für die Entwicklung (Sprungantwort) einer geraden Initialbahn gemäß Gl. (8) angewandt. Das Feld wird dabei von drei ruhenden Hindernissen erzeugt. Es ist gut zu erkennen, wie initial ausschließlich die in Hindernisnähe hohen Abstoßungskräfte (rot) normal zum Kraftfeld angreifen und die Bahn verdrängen. Die durch die zunehmende Bahnkrümmung entstehenden und normal sowie tangential zur Bahn gerichteten Rückstellkräfte (grün) wirken dieser Verdrängung entgegen. Die sich so einstellende energieminimale Gleichgewichtslage (Kräftegleichgewicht an jeder Stützstelle) der Bahn führt nun um die Hindernisse herum und stellt somit

ein Paretooptimum zwischen minimaler Bahnlänge und maximalem Abstand zu den Hindernissen dar.

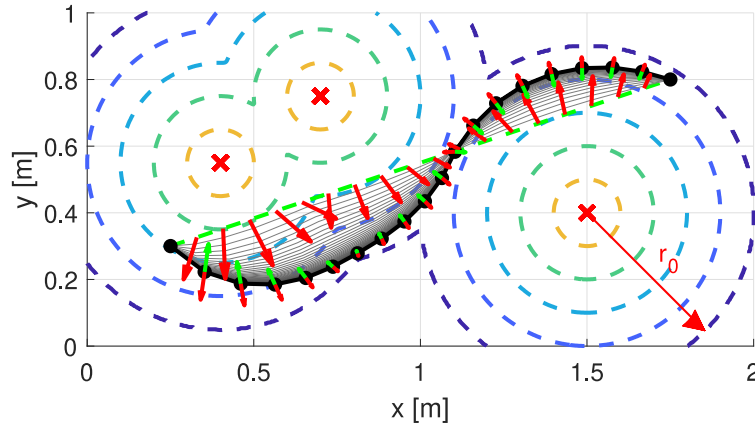


Abbildung 12: Entwicklung einer Bahn unter Einfluss dreier Punkthindernisse

Um das Bahnverhalten unter Einfluss von Hindernissen steuern zu können, wird Gl. (8) zusätzlich um die beiden Parameter T und K ergänzt. Nachfolgend soll diese erweiterte lineare Kurvenfluss-DGL als KFM-DGL bezeichnet werden:

$$T \cdot \dot{\mathbf{p}}_i = \underbrace{\mathbf{k}_i(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})}_{\text{Rückstellkraft}} + K \cdot \underbrace{\mathbf{f}_i(\mathbf{p}_i, \mathbf{h})}_{\text{Abstoßungskraft}} \quad (13)$$

Der Sicherheitsparameter K dient der Verstärkung des abstoßenden Feldes und ermöglicht damit die Steuerung der stationären Endlage der Bahn (Sicherheitsabstand). T stellt einen Dynamikparameter dar und ermöglicht damit eine Vorgabe, wie lange die Bahn zum Erreichen dieser Gleichgewichtslage benötigt (Bahndynamik). Entgegen der direkten Verstärkung von Rückstell- und Abstoßungskraft mit den Faktoren k_c und k_r bei der EBM (vgl. Kap. 3) ist diese geringfügig abweichende Parametrierung intuitiver, da sich der Sicherheitsabstand unabhängig von der Bahndynamik steuern lässt. Tabelle 5 zeigt den qualitativen Einfluss von T und K auf Entwicklung und Gleichgewichtslage der Bahn. Es wird deutlich, wie die Verstärkung des Sicherheitsparameters von K auf $5 \cdot K$ bei konstantem T (spaltenweise) den Abstand zu den Hindernissen erhöht. Eine Verstärkung des Dynamikparameters von T auf $5 \cdot T$ bei konstantem K (zeilenweise) bewirkt eine langsamere Konvergenz der Bahn ohne Einfluss auf deren Gleichgewichtslage. Die dargestellten Entwicklungsschritte sind dabei zeitlich äquidistant.

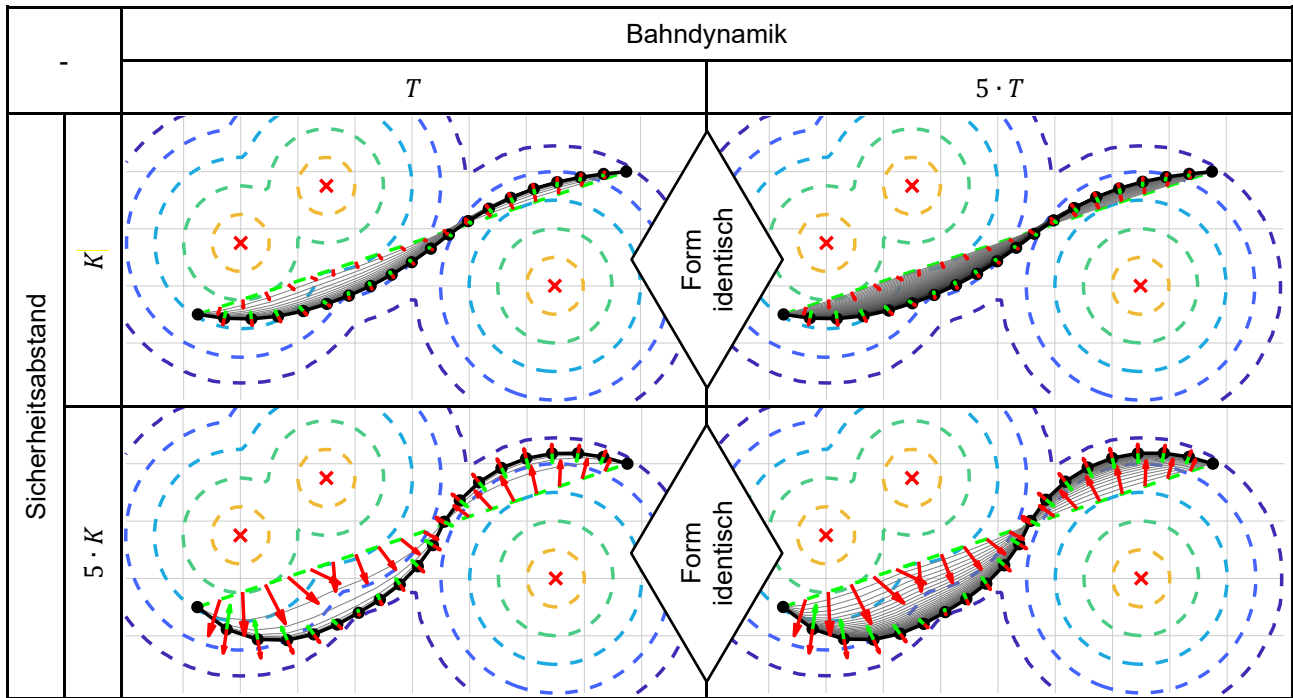


Tabelle 5: Einfluss der Sicherheits- und Dynamikparameter K und T

In Matrixform kann die KFM-DGL in D linearen Zustandsraumssystemen formuliert werden, wovon hier aufgrund der Gleichartigkeit nur die x -Richtung dargestellt ist:

$$T \cdot \underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-2} \\ \dot{x}_{n-1} \\ \dot{x}_n \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix}}_x + \underbrace{\begin{pmatrix} x_s \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{b_x} + K \cdot \underbrace{\begin{pmatrix} f_{1,x}(\mathbf{p}_1, \mathbf{h}) \\ f_{2,x}(\mathbf{p}_2, \mathbf{h}) \\ f_{3,x}(\mathbf{p}_3, \mathbf{h}) \\ \vdots \\ f_{n-2,x}(\mathbf{p}_{n-2}, \mathbf{h}) \\ f_{n-1,x}(\mathbf{p}_{n-1}, \mathbf{h}) \\ f_{n,x}(\mathbf{p}_n, \mathbf{h}) \end{pmatrix}}_{f_x} \quad (14)$$

Es ist zu beachten, dass die Berechnung der Abstoßungskraft f_i für die einzelnen Raumrichtungen entgegen der Rückstellkraft nicht entkoppelt und in linearer Matrixdarstellung abgebildet werden kann, da f_i im Allgemeinen auf nichtlinearen Distanzrechnung zwischen der Stützstelle \mathbf{p}_i und den Hindernissen \mathbf{h} basiert. Dadurch stellt f einen nichtlinearen Kopplungsterm zwischen den einzelnen Teilsystemen dar.

4.7 Dynamik mit Hindernissen

Da die Abstoßungskraft f neben den Hindernisgeometrien h als reinen Eingangsgrößen ebenfalls von den Stützstellenpositionen p als Zustandsgrößen abhängt, beeinflusst dies gegenüber dem reinen Rückstellverhalten des unbelasteten Kurvenflusses die Gesamtdynamik zusätzlich. Allerdings stellt f eine beliebige und bezüglich der Stützstellenpositionen in der Regel nichtlineare Funktion dar. Bereits das für die Voruntersuchung der EBM (vgl. Kap. 3) sowie im vorangegangenen Kap. 4.6 verwendete einfache Umgebungsmodell mit einzelnen Punkthindernissen macht aufgrund der bei $D \geq 2$ nichtlinearen Distanzrechnungen eine systemtheoretische Analyse des Bahnverhaltens in Analogie zum unbelasteten Fall (vgl. Kap. 4.5) unmöglich.

Ersetzt man in Gl. (10) bis Gl. (12) die punktförmigen Hindernisse durch ein einzelnes und parallel zur Bahn verlaufendes Linienhindernis, gleicht dessen Wirkung einer lediglich ein-dimensional wirkenden „Streckenlast“ und die Nichtlinearitäten können dadurch eliminiert werden. Der Wirkradius r_0 wird hierbei zur Wirkweite verallgemeinert. Gleichzeitig bildet dieses lineare Modell der Abstoßungskraft die maximal mögliche Belastung der Bahn („Vollbelastung“) ab und soll daher nachfolgend als Referenzmodell für die Analyse und Auslegung der KFM unter Hinderniseinfluss dienen. Abbildung 13 illustriert die Wirkung der nichtlinearen Punktlast gegenüber der linearen Streckenlast.

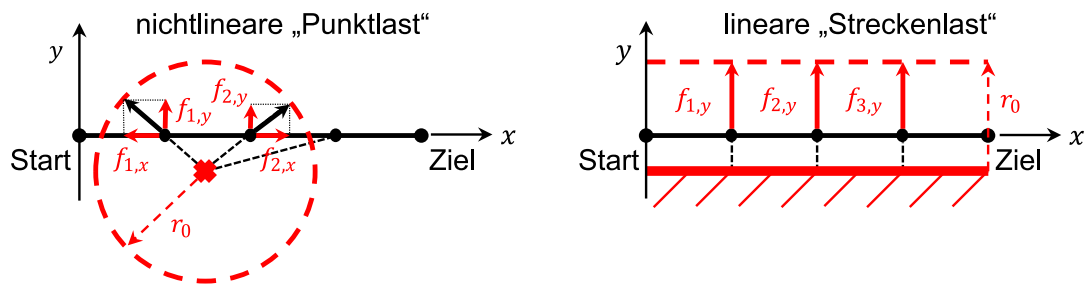


Abbildung 13: Kraftwirkung von nichtlinearer Punkt- und linearer Streckenlast

Da ein Sprung der Hindernisposition äquivalent zu einem Sprung der Wirkweite r_0 ist, kann die Streckenlast vereinfachend direkt auf der Initialbahn (x -Achse) positioniert werden und die Kraftkomponenten ergeben sich zu $f_{i,y} = r_0 - y_i$ (und $f_{i,x} = 0$) mit r_0 als Eingangsgröße. Damit kann die y -Komponente der allgemeinen KFM-DGL nach Gl. (14)

durch das folgende lineare DGL-System beschrieben werden. Der Eingangsvektor für die Randkoordinaten \mathbf{b}_y entfällt, da y_s und y_g als Null angenommen werden:

$$T \cdot \underbrace{\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \vdots \\ \dot{y}_{n-2} \\ \dot{y}_{n-1} \\ \dot{y}_n \end{pmatrix}}_{\dot{\mathbf{y}}} = \underbrace{\begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{pmatrix}}_y + K \cdot \underbrace{\begin{pmatrix} r_0 \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{pmatrix} \end{pmatrix}}_{f_y} \quad (15)$$

Umgestellt ergibt sich daraus folgende normierte Zustandsraumdarstellung mit der nun modifizierten Systemmatrix A^* :

$$\underbrace{\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \vdots \\ \dot{y}_{n-2} \\ \dot{y}_{n-1} \\ \dot{y}_n \end{pmatrix}}_{\dot{\mathbf{y}}} = \underbrace{\begin{pmatrix} -\frac{2+K}{T} & \frac{1}{T} & 0 & \dots & 0 & 0 & 0 \\ \frac{1}{T} & -\frac{2+K}{T} & \frac{1}{T} & \dots & 0 & 0 & 0 \\ 0 & \frac{1}{T} & -\frac{2+K}{T} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\frac{2+K}{T} & \frac{1}{T} & 0 \\ 0 & 0 & 0 & \dots & \frac{1}{T} & -\frac{2+K}{T} & \frac{1}{T} \\ 0 & 0 & 0 & \dots & 0 & \frac{1}{T} & -\frac{2+K}{T} \end{pmatrix}}_{\substack{\text{Systemmatrix} \\ A^* = \frac{1}{T}(A - K \cdot I)}} \cdot \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{pmatrix}}_y + \underbrace{\frac{K}{T} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{pmatrix}}_{\substack{\text{Eingangs-} \\ \text{vektor}}} \cdot \underbrace{r_0}_{\text{Eing.}} \quad (16)$$

Es wird deutlich, dass die Systemmatrix $A^* = 1/T \cdot (A - K \cdot I)$ auch hier eine tridiagonale Toeplitz-Matrix bildet. Mit den Haupt- und Nebendiagonalelementen $a_d = -(2 + K)/T$ und $a_l = a_u = 1/T$ kann die explizite Berechnung der n Eigenwerte λ_i nach Gl. (7) somit verallgemeinert in Abhängigkeit der Parameter T und K beschrieben werden:

$$\lambda_i = \frac{1}{T} \cdot \underbrace{\left(-2 + 2 \cdot \cos\left(\frac{\pi \cdot i}{n + 1}\right) - K\right)}_{\substack{\text{Eigenwerte aus Gl. (7)}}} \quad (17)$$

Aus dieser Darstellung kann direkt abgeleitet werden, wie T und K die Lage der Eigenwerte und damit die Bahndynamik gegenüber dem in Kap. 4.5 untersuchten dynamikneutralen (entspricht $T = 1$) Kurvenfluss ohne Hindernisse (entspricht $K = 0$) beeinflussen:

- Alle Eigenwerte λ_i werden auf der Realachse um denselben Wert K verringert, wodurch die Symmetrieachse der Eigenwertverteilung verschoben wird. Für $K > 0$ steigt damit die Bahndynamik.
- Anschließend werden alle Eigenwerte λ_i mit dem Faktor $1/T$ bezüglich des Nullpunkts der Realachse skaliert, wodurch deren Spanne variiert. Für $T > 1$ sinkt und für $T < 1$ steigt damit die Bahndynamik.

Abbildung 14 zeigt diesen Zusammenhang für einige exemplarische Parametrierungen.

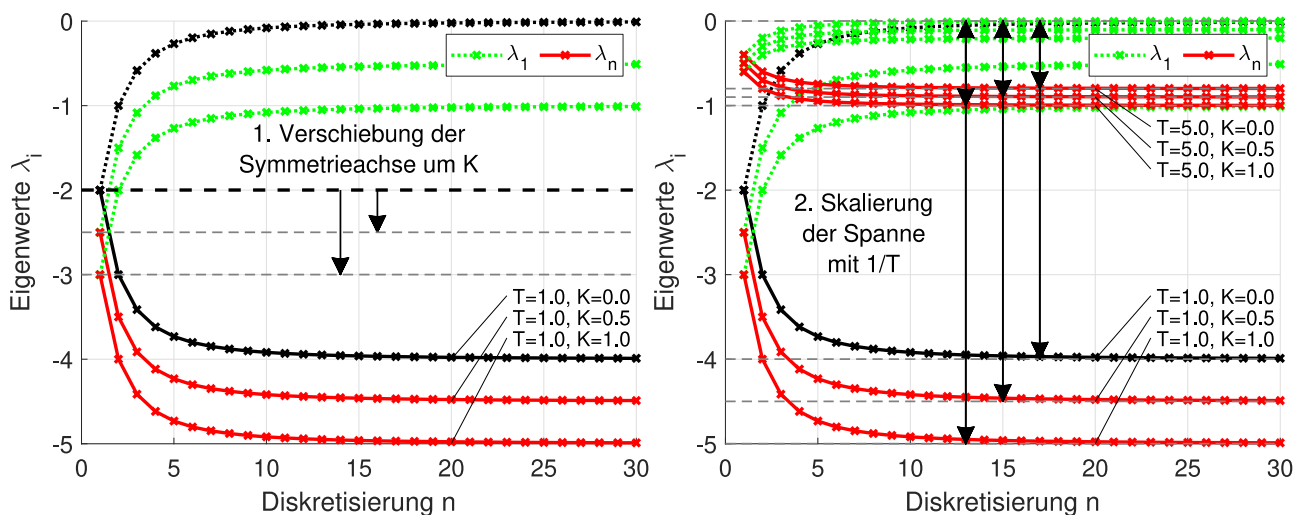


Abbildung 14: Verschiebung der Eigenwerte von A^* in Abhängigkeit von T und K

Die sich aus dem betragsmäßig kleinsten Eigenwert λ_1 ergebende Stellzeit T_s in Abhängigkeit der Parameter T und K und der Diskretisierung n lautet somit wie folgt:

$$T_s(n, T, K) = \frac{3 \cdot T}{2 - 2 \cdot \cos\left(\frac{\pi}{n+1}\right) + K} \quad (18)$$

In Tabelle 6 sind die sich ergebenden Stellzeiten T_s für die Parametrierungen aus Abbildung 14 in Abhängigkeit der Diskretisierung n dargestellt.

4 Kurvenflussmethode (KFM)

-	T	K	n							
			1	2	3	4	5	10	20	30
T_s in [s]	1	0	1.500	3.000	5.121	7.854	11.196	37.031	134.29	292.35
	1	0.5	1.200	2.000	2.763	3.402	3.907	5.163	5.743	5.879
	1	1	1.000	1.500	1.892	2.171	2.366	2.775	2.934	2.970
	5	0	7.500	15.000	25.607	39.271	55.981	185.15	671.49	1461.8
	5	0.5	6.000	10.000	13.815	17.008	19.533	25.817	28.717	29.397
	5	1	5.000	7.500	9.459	10.854	11.830	13.876	14.672	14.848

Tabelle 6: Stellzeiten T_s für einige Parametrierungen T und K und Diskretisierungen n

Abbildung 15 zeigt Simulationen der Sprungantworten der vier in Tabelle 6 markierten exemplarischen Parametrierungen.

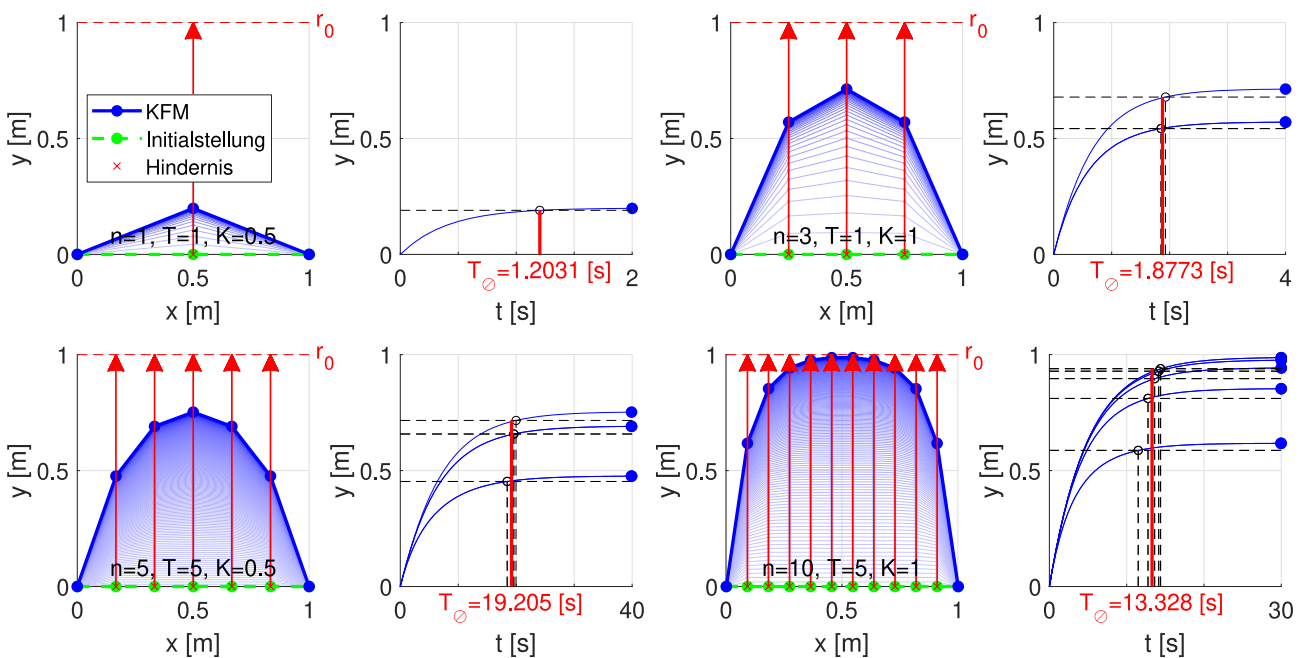


Abbildung 15: Antworten auf Einheitsprung der Wirkweite r_0 bei Vollbelastung

Die Simulationen zeigen, dass alle Stützstellen auch im vollbelasteten Fall PT_1 -ähnlichen Verläufen folgen und die Streuung der individuellen Stellzeiten $T_{s,i}$ nur sehr gering ist. Die durchschnittliche Stellzeit T_θ korreliert auch hier sehr gut mit den analytisch in Abhängigkeit von T und K prognostizierten Stellzeiten T_s aus Tabelle 6.

Da sich in der Praxis häufig nur Teile der Bahn im Wirkungsbereich von Hindernissen befinden, liegt meist eine variierende Mischung aus belasteten und unbelasteten Stützstellen vor. Für diesen „teilbelasteten“ Fall gilt in Gl. (16) für die den unbelasteten Stützstellen entsprechenden Zeilen $K = 0$. Dadurch kann die modifizierte Systemmatrix A^* eine beliebige Verteilung der beiden verschiedenen Hauptdiagonalelemente $a_d = -2/T$ (unbelastet) und $a_d = -(2 + K)/T$ (belastet) enthalten und stellt keine Toeplitz-Matrix mit explizit berechenbaren Eigenwerten mehr dar.

Durch Abschätzung anhand von Gerschgorin-Kreisen (Gerschgorin 1931) sowie der nach wie vor vorhandenen Symmetrie (ausschließlich reelle Eigenwerte) von A^* kann Instabilität sowie Eigenschwingfähigkeit aber auch im teilbelasteten Fall ausgeschlossen werden, da sich alle Eigenwerte λ_i stets auf zwei (für $K < 4$ sich überlappende) Wertespanspannen im Bereich $0 < \lambda_i < K + 4$ auf der Realachse verteilen (vgl. Abbildung 16).

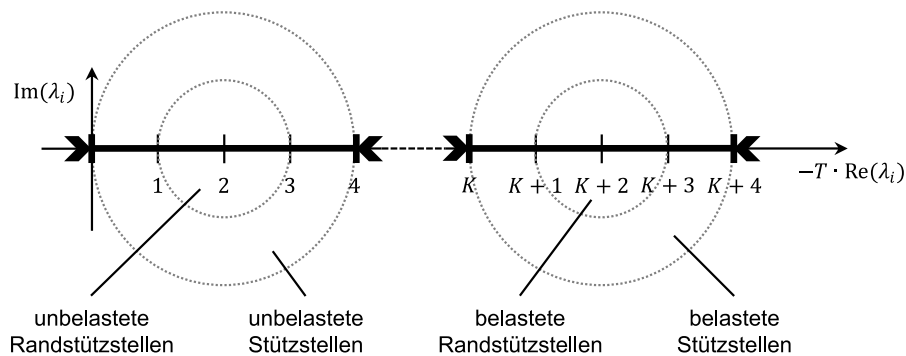


Abbildung 16: Gerschgorin-Kreise zur Eingrenzung der Eigenwerte bei Teilbelastung

Außerdem zeigen Experimente, dass die durchschnittliche Stellzeit T_\emptyset entsprechend der Stützstellenverteilung auf die beiden in Abbildung 16 dargestellten Wertespanspannen im teilbelasteten Fall stets zwischen der minimalen Stellzeit $T_{s,min} = T_s(n, T, K > 0)$ bei vollbelasteter Abstoßung sowie der maximalen Stellzeit $T_{s,max} = T_s(n, T, K = 0)$ bei vollständig unbelasteter Rückstellung liegt. Abbildung 17 zeigt exemplarisch zwei teilbelastete Fälle, bei denen lediglich die erste Stützstelle y_1 bzw. die ersten fünf Stützstellen y_1 bis y_5 unter Einfluss von Hindernissen stehen.

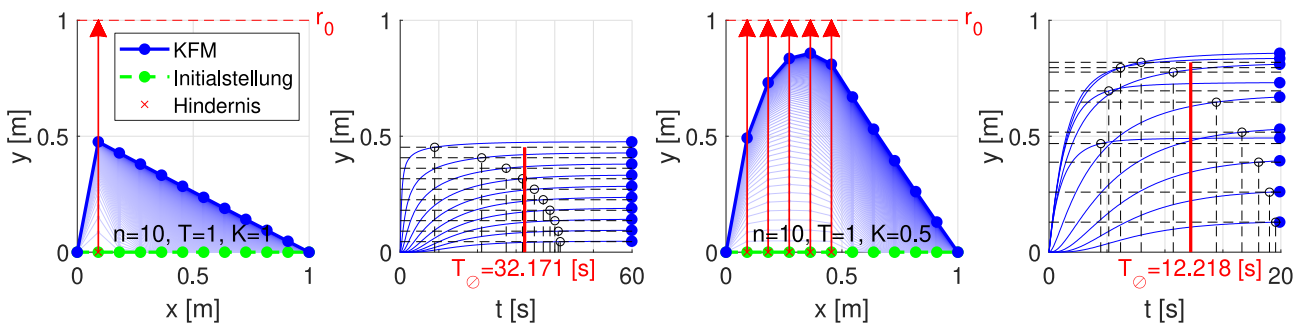


Abbildung 17: Antworten auf Einheitssprung der Wirkweite r_0 im teilbelasteten Fall

Zunächst wird deutlich, dass auch hier (in Analogie zum Sprung der Zielkoordinate des unbelasteten Kurvenflusses) die Verzögerungsordnung (ausgeprägte S-Form) und damit auch die individuelle Stellzeit von unbelasteten Stützstellen sukzessive mit der Entfernung zu den belasteten Stützstellen steigt.

Außerdem liegt die durchschnittliche Stellzeit T_\emptyset in beiden Fällen innerhalb der prognostizierten Schranken $T_{s,min}$ und $T_{s,max}$:

- y_1 belastet: $T_{s,min} = 2.775 [s] < 32.171 [s] < T_{s,max} = 37.031 [s]$
- y_1 bis y_5 belastet: $T_{s,min} = 5.163 [s] < 12.218 [s] < T_{s,max} = 37.031 [s]$

Abbildung 18 zeigt die simulativ gemessene durchschnittliche Stellzeit T_\emptyset aller ab y_1 zusammenhängend teilbelasteten Fälle bei $n = 10$, $T = 1$ und $K = 1$ (Fortführung von Abbildung 17 links) sowie die zugehörige Verteilung der numerisch ermittelten Eigenwerte λ_i . Deren Wertebereich verschiebt sich deutlich sichtbar sukzessive um $K = 1$, wodurch zugleich die Stellzeit monoton abnimmt, dabei aber stets zwischen den analytischen Grenzwerten bleibt. Die Extrapolation aller belasteten Fälle (Ausgleichsrechnung mittels eines Polynoms 10. Ordnung) bestätigt nochmals $T_{s,max}$ als analytische Obergrenze der Stellzeit mit Hindernissen.

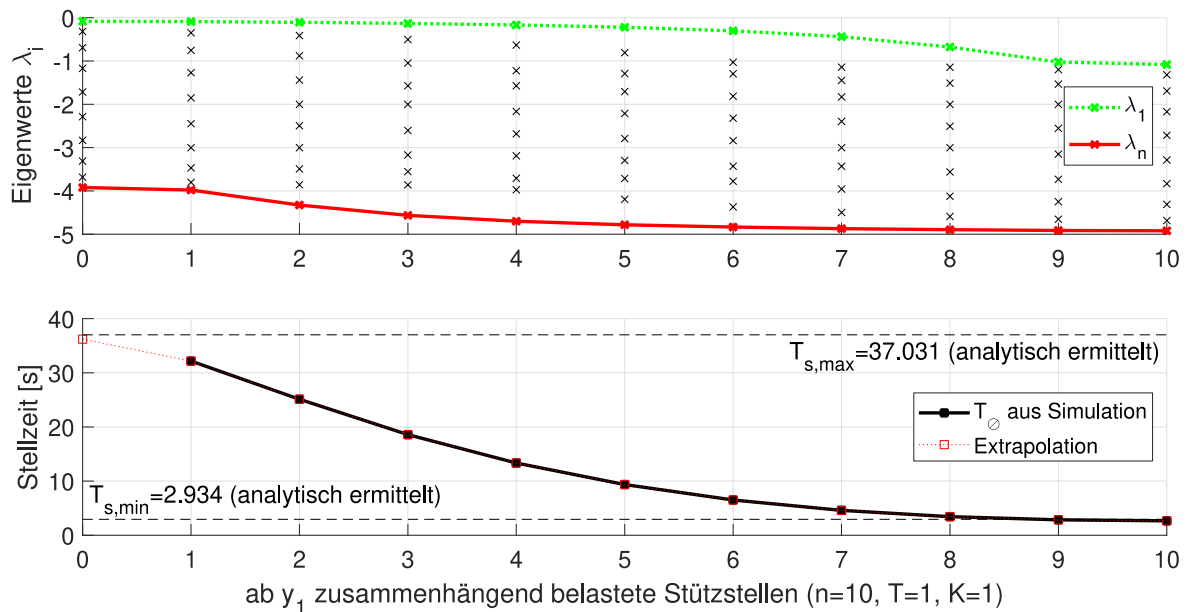


Abbildung 18: Eigenwertverteilung und gemessene Stellzeit T_{\emptyset} bei Teilbelastung

Damit lassen sich folgende Erkenntnisse zur Bahndynamik mit Hindernissen zusammenfassen:

- Unter Einfluss von Hindernissen erhöht sich die Bahndynamik (durchschnittliche Stellzeit der einzelnen Stützstellen) gegenüber der unbelasteten Rückstellung.
- Zusätzlich ist die Bahndynamik umso höher, je mehr Stützstellen unter dem Einfluss von Hindernissen stehen.
- Anhand eines linearen Hindernismodells, welches die maximale Belastung der Bahn (alle Stützstellen durch Streckenlast angeregt) beschreibt, kann die minimal notwendige Stellzeit der Bahn $T_{s,min}$ in Abhängigkeit von Parametrierung und Diskretisierung analytisch berechnet werden.
- Es konnte gezeigt werden, dass für den praxisrelevanten Fall, in denen Hindernisse lediglich auf einzelne Stützstellen wirken, die sich ergebende Stellzeit stets zwischen der analytisch berechenbaren Untergrenze $T_{s,min}$ (vollbelastete Abstoßung) und Obergrenze $T_{s,max}$ (unbelastete Rückstellung) liegt.

4.8 Stationäre Endlage

Unter Einfluss von statischen Hindernissen konvergiert die Bahn für $t \rightarrow \infty$ in eine stationäre Endlage \mathbf{p}_∞ . Diese kann als Maß für den Sicherheitsabstand der Bahn zu den Hindernissen herangezogen werden und soll in Analogie zur Bahndynamik erneut anhand der Streckenlast für die vollbelastete Bahn analysiert werden. Auch hier wird lediglich die y -Komponente untersucht.

Zur Berechnung der stationären Auslenkungen y_∞ kann Gl. (16) durch Nullsetzen der zeitlichen Ableitungen $\dot{\mathbf{y}} = \mathbf{0}$ in das lineare GLS nach Gl. (19) überführt werden. Dabei kann der Dynamikfaktor T herausgekürzt werden, was nochmals die ausschließliche Abhängigkeit der stationären Endlage vom Sicherheitsfaktor K und der Diskretisierung n sowie der Hinderniswirkweite r_0 bestätigt. Da die Lösung y_∞ proportional zur Wirkweite r_0 ist, wird im Folgenden ausschließlich die vereinheitlichte Wirkweite $r_0 = 1 [m]$ verwendet.

$$\underbrace{\begin{pmatrix} -2-K & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2-K & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2-K & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2-K & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -2-K & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2-K \end{pmatrix}}_{\substack{\text{Koeffizientenmatrix} \\ A-K \cdot I}} \cdot \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{pmatrix}}_{\mathbf{y}_\infty} = -K \cdot r_0 \cdot \underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{pmatrix}}_{\substack{\text{Eingangs-} \\ \text{vektor}}} \quad (19)$$

Abbildung 19 zeigt die stationären Endlagen für verschiedene Sicherheitsfaktoren K und Diskretisierungen n , wobei lediglich der vollbelastete Fall nach Gl. (19) dargestellt wurde. Bei Teilbelastung ist in den Zeilen der unbelasteten Stützstellen $K = 0$ zu setzen.

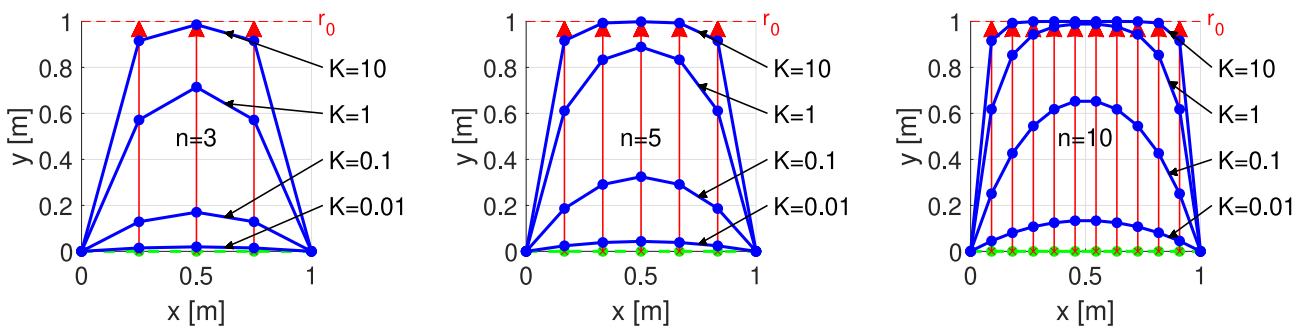


Abbildung 19: Stationäre Endlage in Abhängigkeit von K und n

Es wird ersichtlich, dass die stationäre Auslenkung der Stützstellen y_∞ für $K \rightarrow \infty$ durch die Wirkweite der Hindernisse r_0 begrenzt wird. Bei der hier angewandten Streckenlast ist die stationäre Länge der Bahn L_∞ somit ebenfalls auf $\sim(L_0 + 2 \cdot r_0)$ begrenzt, wobei $L_0 = |\mathbf{p}_g - \mathbf{p}_s|$ die unbelastete Bahnlänge beschreibt. Außerdem ist erkennbar, dass eine höhere Diskretisierung n bei identischem K höhere Stützstellenauslenkungen zulässt.

Um den Sicherheitsabstand der Bahn in einer einzigen Größe bemessen zu können, ist es üblich hierfür eine Norm zu verwenden. Neben der in der Funktionsanalysis häufig verwendeten Maximumsnorm für die maximale Stützstellenauslenkung ($\|\mathbf{y}\|_{max}$) sollen zusätzlich die minimale Stützstellenauslenkung ($\|\mathbf{y}\|_{min}$), die durchschnittliche Stützstellenauslenkung ($\|\mathbf{y}\|_\phi$ als Abwandlung der ebenfalls üblichen Summennorm) sowie die Bahnlänge L ($\|\mathbf{y}\|_L$) als Maß für den Sicherheitsabstand herangezogen werden. Es sei angemerkt, dass lediglich Berechnungsvorschriften, welche die drei Normaxiome der Definitheit, Homogenität und Subadditivität erfüllen, als Normen im mathematischen Sinne gelten (Werner 2011). Obgleich $\|\mathbf{y}\|_{min}$ die Definitheit ($\|\mathbf{y}\|_{min}$ stellt damit eine Halbnorm dar) und $\|\mathbf{y}\|_L$ die Homogenität nicht erfüllen, besitzen beide Berechnungsvorschriften dennoch eine gute Aussagekraft bezüglich des Sicherheitsabstandes der Bahn.

$$\|\mathbf{y}\|_{max} = \max(|y_1|, \dots, |y_n|) \quad (20)$$

$$\|\mathbf{y}\|_{min} = \min(|y_1|, \dots, |y_n|) \quad (21)$$

$$\|\mathbf{y}\|_\phi = \frac{1}{n} \cdot \sum_{i=1}^n |y_i| \quad (22)$$

$$\|\mathbf{y}\|_L = \underbrace{\sqrt{\left(\frac{L_0}{n+1}\right)^2 + y_1^2}}_{|\mathbf{p}_1 - \mathbf{p}_s|} + \sum_{i=2}^n \underbrace{\sqrt{\left(\frac{L_0}{n+1}\right)^2 + (y_i - y_{i-1})^2}}_{|\mathbf{p}_i - \mathbf{p}_{i-1}|} + \underbrace{\sqrt{\left(\frac{L_0}{n+1}\right)^2 + y_n^2}}_{|\mathbf{p}_g - \mathbf{p}_n|} \quad (23)$$

Abbildung 20 zeigt die Anwendung dieser Normen in Abhängigkeit von n und K bei einheitlicher Wirkweite $r_0 = 1$ [m]. Mithilfe der eingezeichneten Isolinien ist zudem bei gegebener Diskretisierung n eine grobe Abschätzung von K unter Vorgabe des gewünschten Sicherheitsabstandes durch die entsprechende Norm möglich.

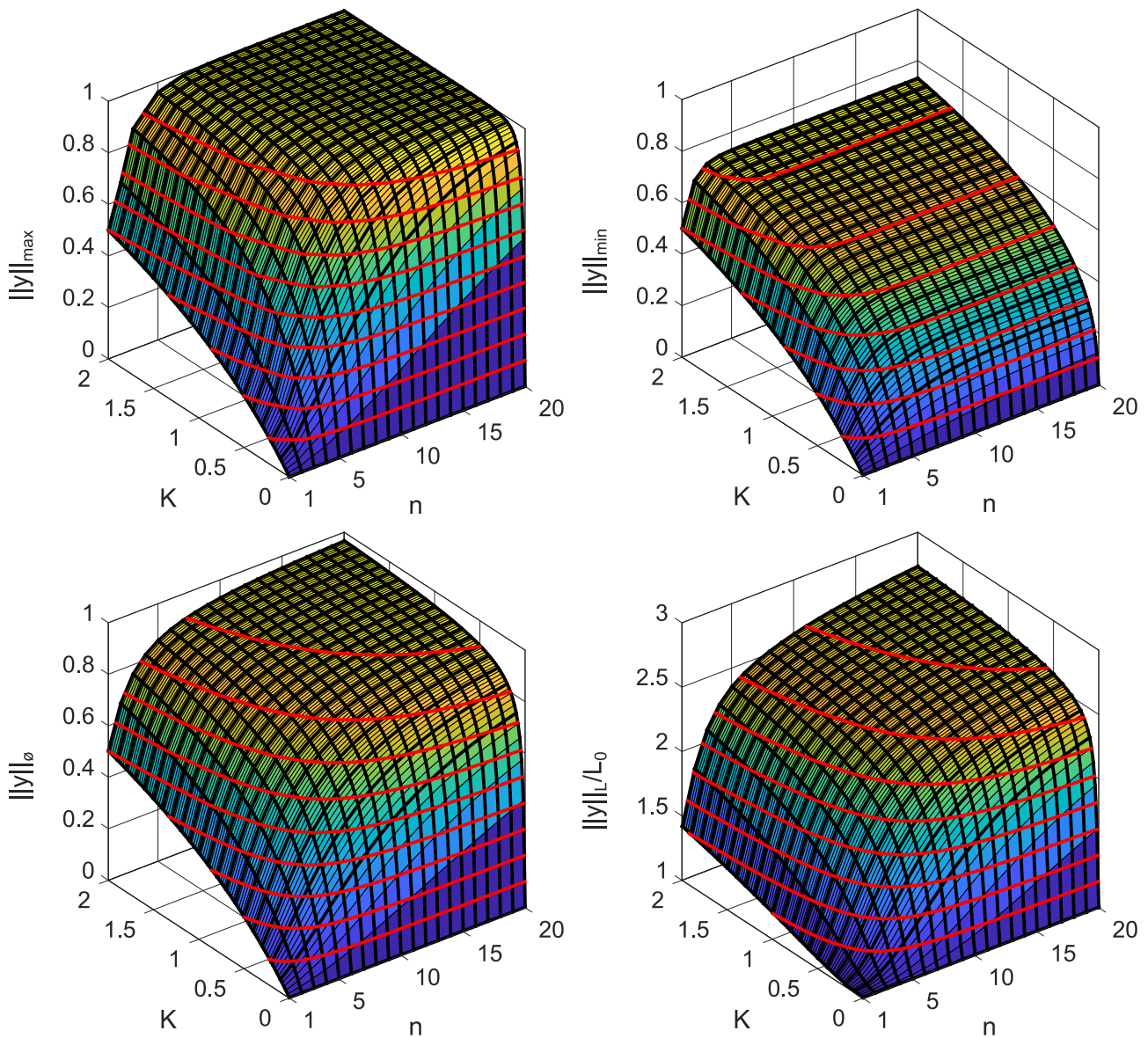


Abbildung 20: Stationärer Sicherheitsabstand bemessen mit verschiedenen Normen

4.9 Parametrierung

In den vorangegangenen Kap. 4.7 und 4.8 wurde aufgezeigt, wie für eine gegebene Parametrierung n , T und K der KFM die sich ergebende Dynamik und stationäre Endlage der Bahn bestimmt werden kann. Dabei bestehen Abhängigkeiten dieser beiden Bahneigenschaften bezüglich nahezu aller Parameter (vgl. Tabelle 7).

		Parameter		
		n	T	K
Einfluss auf ...	Bahndynamik	Ja	Ja	Ja
	stat. Endlage	Ja	Nein	Ja

Tabelle 7: Abhängigkeiten zwischen Parametern und Bahneigenschaften der KFM

Um eine aufwendige empirische Suche nach einer geeigneten Parametrierung für die KFM zu vermeiden, wird der inverse Zusammenhang der abstrakten Parameter T und K zu Bahndynamik und stationärer Endlage bei gegebener Diskretisierung n benötigt. Auf diese Weise können beispielsweise nach einer Änderung der Diskretisierung die ursprünglichen Bahneigenschaften durch eine explizite Neuberechnung von T und K exakt wiederhergestellt werden.

Zunächst ist problemspezifisch eine geeignete Bahndiskretisierung n zu wählen. Feiner aufgelöste Bahnen besitzen eine stetigere Krümmung, sodass deren Stützstellen für einen verhältnismäßig trägen Manipulator wie einem UAV direkt als Lagesollwerte verwendet werden können. Sollen die Stützstellen lediglich als Via-Points für eine nachgeschaltete Feininterpolation dienen, so kann eine gröbere Diskretisierung bereits ausreichend sein. Je nach verwendeter Hardware für die Ausführung der KFM kann auch die zur Verfügung stehende Rechenleistung in Kombination mit der gewünschten Zykluszeit eine Obergrenze für die Wahl von n darstellen. Eine Analyse der Rechenkomplexität der KFM erfolgt später in Kap.4.12.

Anschließend kann der Sicherheitsfaktor K ermittelt werden. Dazu ist notwendig, dass K als weitere Unbekannte in Gl. (19) einfließt, wodurch ein nichtlineares GLS zur Berechnung von K entsteht. Nach Umformulierung in das Nullstellenproblem $F(\mathbf{y}_F) = \mathbf{0}$ der Dimension $n + 1$ mit $\mathbf{y}_F = (y_1, y_2, y_3, \dots, y_n, K)^T = (\mathbf{y}, K)^T$ als Vektor der Unbekannten kann K beispielsweise mit Hilfe des Newton-Raphson-Verfahrens (Plato 2010) für mehrere Unbekannte berechnet werden.

$$\mathbf{F}(\mathbf{y}_F) = \begin{cases} -(2 + K) \cdot y_1 + y_2 + K \cdot r_0 = 0 \\ y_1 - (2 + K) \cdot y_2 + y_3 + K \cdot r_0 = 0 \\ y_2 - (2 + K) \cdot y_3 + y_4 + K \cdot r_0 = 0 \\ \dots = 0 \\ y_{n-3} - (2 + K) \cdot y_{n-2} + y_{n-1} + K \cdot r_0 = 0 \\ y_{n-2} - (2 + K) \cdot y_{n-1} + y_n + K \cdot r_0 = 0 \\ y_{n-1} - (2 + K) \cdot y_n + K \cdot r_0 = 0 \\ R(\mathbf{y}) = 0 \end{cases} \quad (24)$$

Die letzte Zeile $F_{n+1}(\mathbf{y}_F) = R(\mathbf{y})$ ist eine zusätzlich notwendige Randbedingung, durch welche das zu erfüllende Kriterium für die gewünschte stationäre Endlage definiert wird. Erst durch diese Randbedingung ist das GLS bestimmt und kann eindeutig gelöst werden.

Mit den Normen aus Kap. 4.8 können die folgenden vier Randbedingungen formuliert werden, wobei deren (skalare) Sollwerte $y_{max,soll}$, $y_{min,soll}$, $y_{\phi,soll}$ oder $y_{L,soll}$ direkt vorgegeben werden müssen. Es muss dabei berücksichtigt werden, dass die Bahn nicht über die Wirkweite r_0 hinaus ausgelenkt werden kann, sodass $0 < y_{max,soll}, y_{min,soll}, y_{\phi,soll} < r_0$ bzw. $L_0 < L_{soll} < \sim(L_0 + 2 \cdot r_0)$ gelten muss.

$$R_{max}(\mathbf{y}) = \|\mathbf{y}\|_{max} - y_{max,soll} \quad (25)$$

$$R_{min}(\mathbf{y}) = \|\mathbf{y}\|_{min} - y_{min,soll} \quad (26)$$

$$R_{\phi}(\mathbf{y}) = \|\mathbf{y}\|_{\phi} - y_{\phi,soll} \quad (27)$$

$$R_L(\mathbf{y}) = \|\mathbf{y}\|_L - L_{soll} \quad (28)$$

Abbildung 21 zeigt exemplarisch die stationären Endlagen der vollbelasteten Bahn, welche sich durch Anwendung der Randbedingungen R_{max} , R_{min} , R_{ϕ} und R_L aus Gl. (25) bis Gl. (28) mit den folgenden Vorgaben für das inverse Problem ergeben:

- Max. Auslenkung bei $n = 5$ soll $y_{max,soll} = 0.95 \cdot r_0$ betragen $\rightarrow K = 1.7117$
- Min. Auslenkung bei $n = 5$ soll $y_{min,soll} = 0.3 \cdot r_0$ betragen $\rightarrow K = 0.2017$
- Durchschn. Auslenkung bei $n = 10$ soll $y_{\phi,soll} = 0.5 \cdot r_0$ betragen $\rightarrow K = 0.1006$
- Länge der Bahn bei $n = 10$ soll $L_{soll} = 2 \cdot L_0$ betragen $\rightarrow K = 0.1738$

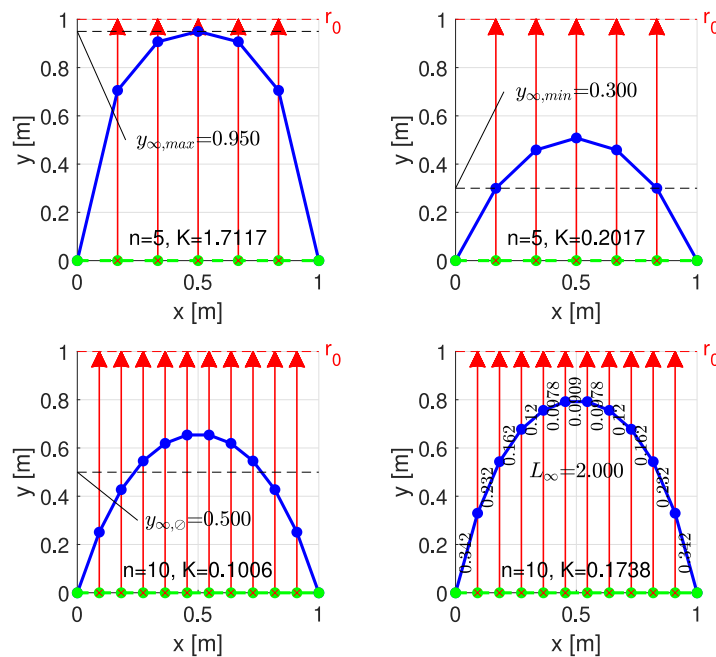


Abbildung 21: Stationäre Endlage als Lösung eines inversen Problems

Im Anhang ist in Code 4 die numerische Berechnung des inversen Problems in Pseudocode dargestellt. Es sei angemerkt, dass K alternativ zum numerischen Lösen des inversen Problems in grober Näherung auch anhand der Isolinien aus Abbildung 20 direkt abgelesen werden kann.

Abschließend erfolgt die Auslegung der Bahndynamik. Da eine langsame Rückstellung der Bahn abgesehen von einem geringen Umweg kein Risiko darstellt, sollte die Auslegung der Bahndynamik anhand der sicherheitsrelevanten vollbelasteten Abstoßung mit $K > 0$ erfolgen. Die inverse Funktion für den Dynamikfaktor T erfolgt direkt durch Auflösen von Gl. (18) nach T unter Vorgabe der gewünschten Stellzeit $T_{s,min}$ sowie des bereits bekannten Sicherheitsfaktors K und der bekannten Diskretisierung n :

$$T(n, K, T_{s,min}) = \frac{1}{3} \cdot (2 - 2 \cdot \cos\left(\frac{\pi}{n+1}\right) + K) \cdot T_{s,min} \quad (29)$$

Anschließend kann bei Bedarf anhand Gl. (18) mit $K = 0$ die sich für T ergebende maximale Stellzeit $T_{s,max}$ ermittelt werden. Alternativ kann T auch umgekehrt unter Vorgabe von $T_{s,max}$ bestimmt werden und $T_{s,min}$ anschließend daraus abgeleitet werden.

4.10 Zeitdiskretisierung

Die zeitliche Entwicklung der KFM-DGL erfolgt mithilfe numerischer Zeitintegration. Dabei wird die kontinuierliche zeitliche Ableitung $\dot{\mathbf{p}}$ durch eine Transitions Gleichung zwischen zwei aufeinander folgenden Zuständen der Bahn $\mathbf{p}^k = \mathbf{p}(t_k)$ und $\mathbf{p}^{k+1} = \mathbf{p}(t_{k+1})$ approximiert. k bezeichnet dabei einen zeitdiskreten Berechnungsschritt zum Zeitpunkt t_k bei konstanter Zeitschrittweite $\Delta t = t_{k+1} - t_k$.

Um die bei der EBM auftretenden Stabilitätsprobleme zu vermeiden, kommen lediglich A-stabile implizite Verfahren (z.B. implizites Eulerverfahren, Trapezregel, Gear-2- bzw. BDF2-Verfahren) zur Lösung der KFM-DGL in Frage. Diese sind (für LZI-Systeme) in der Lage, unabhängig von Zeitschrittweite, Parametrierung oder Betriebspunkt stets eine stabile Lösung zu garantieren. Durch Synchronisierung der Zeitschrittweite Δt mit dem Interpolationstakt der Steuerung kann so außerdem nahezu identisches Zeitverhalten der KFM für beliebige Steuerungstakte gewährleistet werden.

Um die KFM-DGL mithilfe eines impliziten Integrationsverfahrens möglichst effizient lösen zu können, wird die (nichtlineare) Abstoßungskraft \mathbf{f} explizit (also für den aktuellen Zustand \mathbf{p}^k) ausgewertet und fließt anschließend komponentenweise als Systemeingang in die nun als linear betrachteten Teilsysteme ein. Die Integration wird somit lediglich semi-implizit durchgeführt.

Für die Zeitintegration wird das implizite Eulerverfahren gewählt. Als Integrationsverfahren 1. Ordnung weist es gegenüber anderen gängigen A-stabilen Verfahren wie der Trapezregel oder dem Gear-2-Verfahren (beides Verfahren 2. Ordnung) zwar einen größeren numerischen Fehler auf, dieser wirkt sich allerdings nicht auf die Ruhelage der Bahn aus und ist daher tolerabel. Gleichzeitig kompensiert die damit einhergehende hohe numerische Dämpfung Unsicherheiten, welche durch die explizite Berechnung von \mathbf{f} entstehen können. Die zusätzliche L-Stabilität garantiert zudem auch für beliebig steife Systeme eine stets monoton stabile Lösung ohne numerisch bedingte Oszillationen.

Durch die semiimplizite Anwendung (\mathbf{f} wird explizit berechnet) dieses Verfahrens auf die KFM-DGL ergeben sich die folgenden impliziten Transitionsleichungen zur komponentenweisen Aktualisierung der Bahn \mathbf{p}^k mit der Zeitschrittweite Δt :

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k + \frac{\Delta t}{T} \cdot (\mathbf{A} \cdot \mathbf{x}^{k+1} + \mathbf{b}_x + K \cdot \mathbf{f}_x^k) \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + \frac{\Delta t}{T} \cdot (\mathbf{A} \cdot \mathbf{y}^{k+1} + \mathbf{b}_y + K \cdot \mathbf{f}_y^k) \end{aligned} \quad (30)$$

Durch Umformung kann Gl. (30) in ein lineares GLS überführt werden. Dabei stellt \mathbf{I} die $n \times n$ -Einheitsmatrix dar.

$$\begin{aligned} \underbrace{\left(\mathbf{I} - \frac{\Delta t}{T} \cdot \mathbf{A}\right)}_{\text{Koeffizientenmatrix } \mathbf{C}} \cdot \mathbf{x}^{k+1} &= \mathbf{x}^k + \frac{\Delta t}{T} \cdot (\mathbf{b}_x + K \cdot \mathbf{f}_x^k) \\ \underbrace{\left(\mathbf{I} - \frac{\Delta t}{T} \cdot \mathbf{A}\right)}_{\text{Koeffizientenmatrix } \mathbf{C}} \cdot \underbrace{\mathbf{y}^{k+1}}_{\text{Lsg.-vekt.}} &= \underbrace{\mathbf{y}^k + \frac{\Delta t}{T} \cdot (\mathbf{b}_y + K \cdot \mathbf{f}_y^k)}_{\text{Eingangsvektoren } \mathbf{r}} \end{aligned} \quad (31)$$

Ausgeschrieben ergibt sich für jedes Teilsystem die folgende tridiagonale $n \times n$ -Koeffizientenmatrix \mathbf{C} , deren spezielle Eigenschaften für eine besonders recheneffiziente Lösung der linearen GLS aus Gl. (31) ausgenutzt werden können.

$$\mathbf{C} = \left(\mathbf{I} - \frac{\Delta t}{T} \cdot \mathbf{A}\right) = \begin{pmatrix} 1 + 2 \cdot \frac{\Delta t}{T} & -\frac{\Delta t}{T} & 0 & \dots & 0 \\ -\frac{\Delta t}{T} & 1 + 2 \cdot \frac{\Delta t}{T} & -\frac{\Delta t}{T} & \ddots & \vdots \\ 0 & -\frac{\Delta t}{T} & 1 + 2 \cdot \frac{\Delta t}{T} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\frac{\Delta t}{T} \\ 0 & \dots & 0 & -\frac{\Delta t}{T} & 1 + 2 \cdot \frac{\Delta t}{T} \end{pmatrix} \quad (32)$$

Da \mathbf{C} lediglich von den initial festzulegenden Parametern n , Δt und T abhängt und damit zur Laufzeit konstant ist, können die linearen GLS sehr effizient mithilfe der LU-Zerlegung gelöst werden. Dazu wird \mathbf{C} initial in eine untere (engl. „lower“) Dreiecksmatrix \mathbf{L} sowie eine obere (engl. „upper“) Dreiecksmatrix \mathbf{U} zerlegt. Die Lösung der linearen GLS für die sich zur Laufzeit verändernden Eingangsvektoren \mathbf{r} kann anschließend auf die explizite Aus-

wertung einer Vorwärts- sowie einer Rückwärtssubstitutionen reduziert werden, ohne die für eine vollständige Lösung (z.B. mittels des Gauß'schen Eliminationsverfahrens) notwendigen rekursiven Eliminationsschritte wiederholt durchführen zu müssen.

Da die Koeffizientenmatrix C gemäß des Zeilensummenkriteriums (Betrag der Diagonalelemente ist jeweils größer als die Summe der Beträge aller restlichen Zeilenelemente) außerdem eine streng diagonaldominante Matrix darstellt, kann die LU-Zerlegung auch ohne Pivotisierung gelöst werden (Golub et al. 2013).

Dabei wird anderenfalls zunächst die Reihenfolge der Eliminationsschritte mittels einer Permutationsmatrix so vertauscht, dass iterative Divisionen durch kleine Zahlen möglichst vermieden werden, da dies zu Instabilitäten aufgrund der begrenzten Maschinengenauigkeit führen kann.

Unter Verwendung der Hilfsvektoren \hat{x} und \hat{y} kann die Lösung der KFM-DGL mithilfe der LU-Zerlegung somit wie folgt durchgeführt werden:

$$\begin{aligned} L \cdot \hat{x} &= x^k + \underbrace{\frac{\Delta t}{T} \cdot (b_x + K \cdot f_x^k)}_{r_x} \rightarrow U \cdot x^{k+1} = \hat{x} \\ L \cdot \hat{y} &= y^k + \underbrace{\frac{\Delta t}{T} \cdot (b_y + K \cdot f_y^k)}_{r_y} \rightarrow U \cdot y^{k+1} = \hat{y} \end{aligned} \quad (33)$$

Da die Koeffizientenmatrix C eine Tridiagonalmatrix darstellt, ergeben sich auch L und U lediglich zu einer unteren (L) und einer oberen (U) Bidiagonalmatrix. Einen äußerst effizienten Algorithmus zur Lösung dieser speziellen Form eines linearen GLS mittels LU-Zerlegung stellt der Thomas-Algorithmus dar (Thomas 1949).

Dadurch reduziert sich die Ordnung der bezüglich der Stützstellenanzahl n benötigten Rechenschritte gegenüber einer vollbesetzten Koeffizientenmatrix von quadratisch zu linear. Dies betrifft sowohl die initiale LU-Zerlegung als auch die Vorwärts- und Rückwärtssubstitution (Kanzow 2005).

Zum Vergleich würde die Lösung mittels vollständiger Gauß'scher Elimination einen Aufwand mit kubischer Ordnung erfordern. Auch die Multiplikation des Eingangsvektors mit

der vorab initialisierten Inversen der Koeffizientenmatrix C wäre ineffizienter, da die Inverse einer Tridiagonalmatrix entgegen ihrer LU-Zerlegung voll besetzt ist (Da Fonseca 2007) und somit eine vollständige Matrix-Vektor-Multiplikation mit quadratischer Ordnung zur Lösung der GLS durchzuführen wäre.

Eine ausführliche Analyse der theoretischen und experimentellen Rechenzeiten von sowohl KFM als auch EBM wird später in Kap. 4.12 vorgestellt. Des Weiteren ist im Anhang in Code 5 (vgl. Kap. 9.2) eine in Pseudocode verfasste Implementierung der KFM zu finden.

4.11 Vergleich des Bahnverhaltens von KFM und EBM

Die in den vorangegangenen Kapiteln hergeleitete KFM beschreibt nunmehr das qualitativ identische Verhalten einer diskretisierten Bahn wie die eingangs hinsichtlich ihrer Stabilität untersuchte EBM. Das numerische Experiment aus der Voruntersuchung (vgl. Kap. 3) soll nun nochmals vergleichend mit der KFM durchgeführt werden. Die KFM soll dabei so parametrisiert werden, dass deren Bahnverhalten auch quantitativ mit der EBM übereinstimmt.

Dazu wird die EBM bei $k_c = k_r = 10$ ebenfalls mit der als Referenzmodell dienenden Streckenlast mit $r_0 = 2.5 [m]$ beaufschlagt. Zur Bemessung der stationären Endlage wird die durchschnittliche Stützstellenauslenkung $\|y\|_\phi$ herangezogen, da diese alle Stützstellen miteinbezieht. Aus Abbildung 22 kann entnommen werden, dass die durchschnittliche Stellzeit $T_\phi = 0.225 [s]$ und die durchschnittliche Stützstellenauslenkung $\|y\|_\phi = 2.1136 [m]$ beträgt.

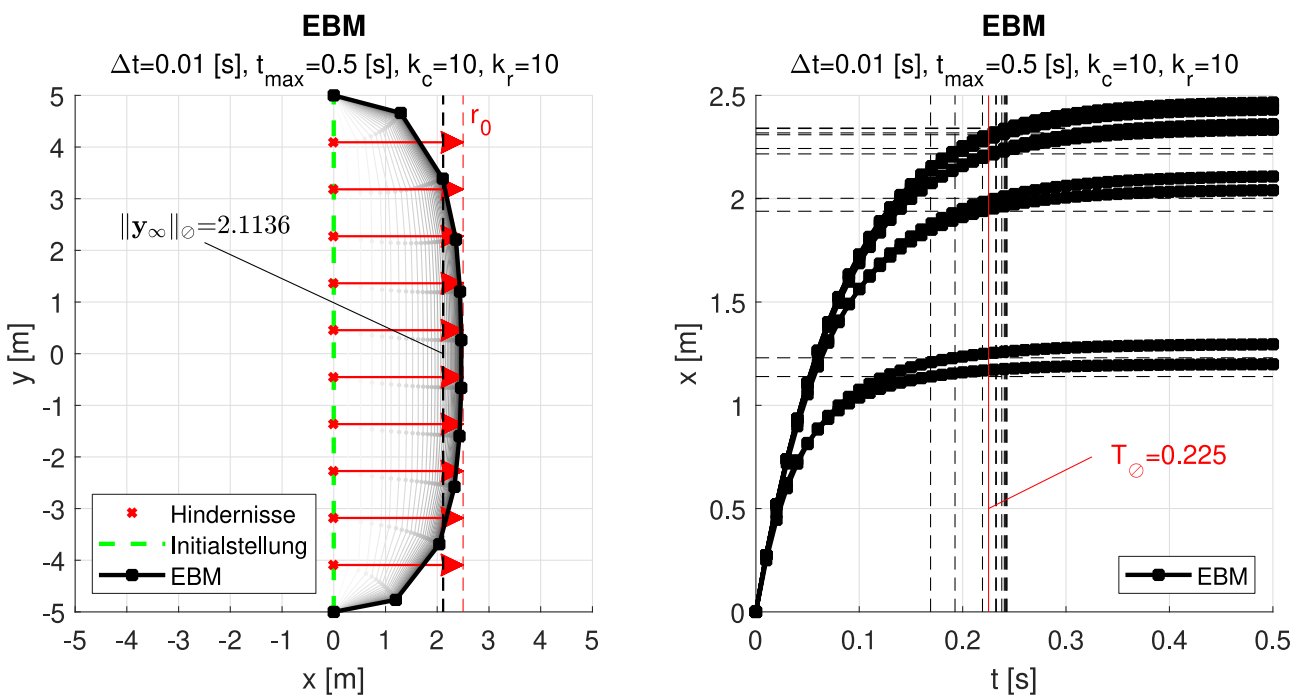


Abbildung 22: Identifikation des Bahnverhaltens der EBM mit linearer Streckenlast

Anhand dieser Werte kann die KFM nun analytisch so parametrisiert werden, dass sie annähernd identisches Bahnverhalten zur EBM aufweist:

- Bei $n = 10$ und $r_0 = 2.5$ [m] ergibt die Anwendung der Randbedingung R_ϕ mit der Vorgabe $y_{\phi, \text{soll}} = 2.1136$ [m] nach Gl. (24) und Gl. (27) den Sicherheitsparameter $K = 0.7295$.
- Anschließend ergibt sich der Dynamikparameter mit $n = 10$, $K = 0.7295$ und $T_{s, \text{min}} = 0.225$ [s] nach Gl. (29) zu $T = 0.0608$.

Abbildung 23 zeigt das Experiment aus der Voruntersuchung zusammen mit der KFM bei der für die EBM noch stabilen Zeitschrittweite $\Delta t = 0.01$ [s]. Anhand der bei Simulationseende jeweils identischen Stützstellenauslenkungen wird deutlich, dass beide Methoden annähernd identisches Zeitverhalten bei der Abstoßung durch die Hindernisse sowie auch bei der unbelasteten Rückstellung aufweisen. Aufgrund der unterschiedlichen Formulierungen der Rückstellkräfte beider Methoden ergibt sich eine unterschiedliche Lage der Stützstellen auf der Bahn. Ebenso wird deutlich, dass die KFM im unbelasteten Fall in eine äquidistante Verteilung der Stützstellen konvergiert.

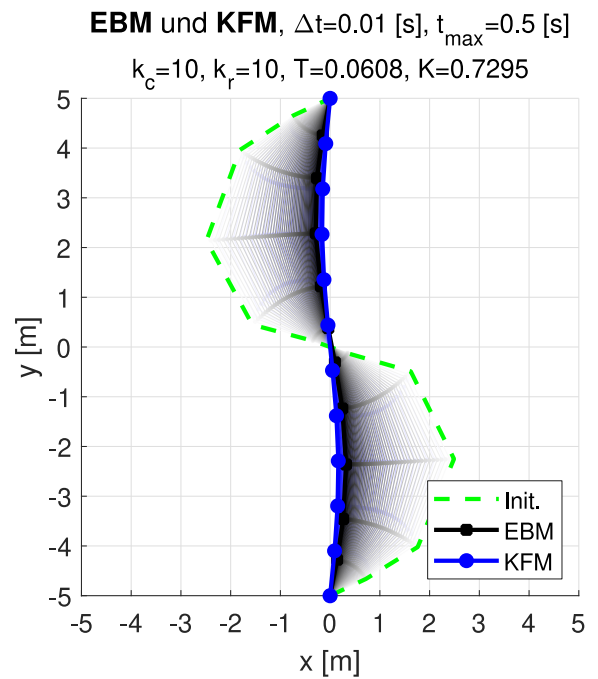
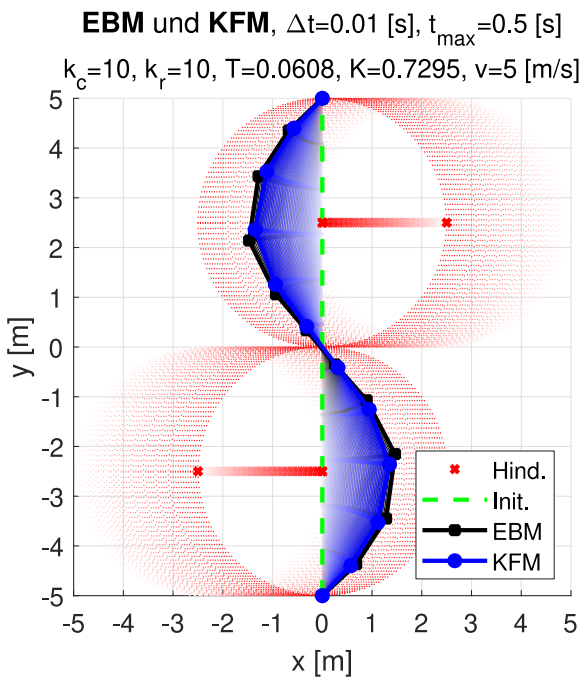


Abbildung 23: Identisches Bahnverhalten von EBM und KFM

Abbildung 24 zeigt die Abstoßung erneut für größere Zeitschrittweiten (exemplarisch dargestellt für $\Delta t = 0.1$ [s] und $\Delta t = 0.25$ [s]) und belegt das robuste Lösungsverhalten der KFM. Entgegen der EBM löst die KFM für beliebige Zeitschrittweiten numerisch stabil.

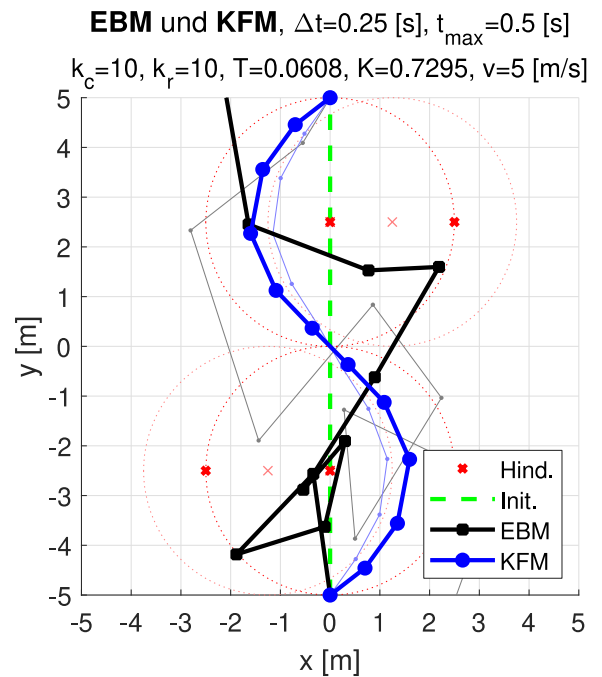
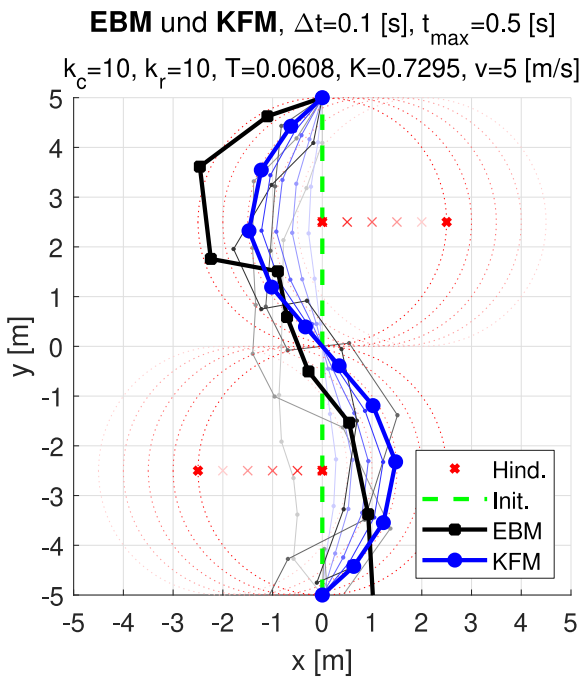


Abbildung 24: Stabiles Lösungsverhalten der KFM für beliebig große Zeitschrittweiten

Eine weitere positive und bisher unberücksichtigte Eigenschaft der KFM stellt deren Unabhängigkeit von einer Skalierung der Lagegrößen (Abstand zwischen Start p_s und Ziel p_g bzw. unbelastete Bahnlänge L_0 sowie die Wirkweite r_0) dar. Abbildung 25 demonstriert diese Eigenschaft anhand einer exemplarischen Skalierung der Lagegrößen mit den Faktoren 10 (links) und 0.1 (rechts). Während die KFM Bahndynamik und die relative stationäre Endlage beibehält, verhält sich die EBM gar kontraintuitiv, da bei kleiner werdendem Maßstab (rechts) der relative Sicherheitsabstand (bezogen auf die Wirkweite der Hindernisse r_0) signifikant abnimmt. Dies liegt darin begründet, dass die Rückstellkraft der EBM auf Basis von normierten Vektoren berechnet wird.

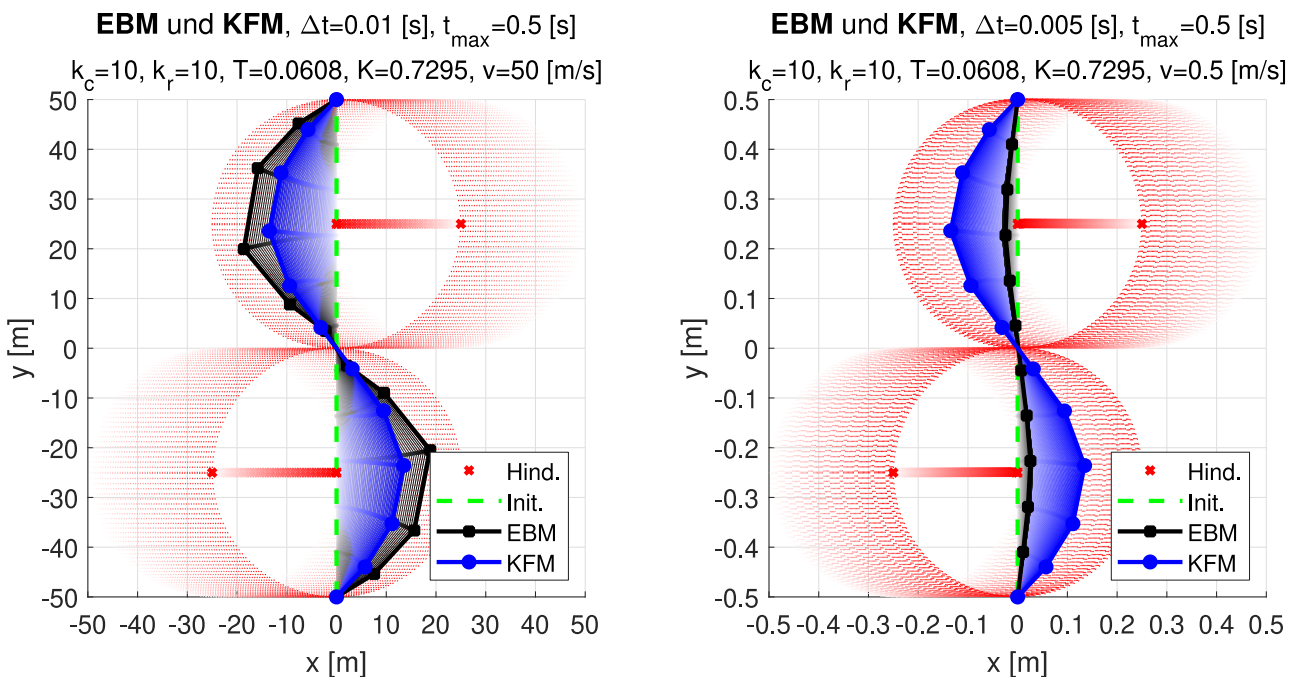


Abbildung 25: Unabhängigkeit der KFM von einer Skalierung der Lagegrößen

Die Skalierungsunabhängigkeit der KFM ist insbesondere dann von Bedeutung, wenn nacheinander mehrere Zielpunkte mit unterschiedlichem Abstand vom jeweiligen Startpunkt erreicht werden sollen. Während die KFM dabei mit konstanter Parametrierung stets ein identisches Verhalten aufweist, würde die EBM mit jedem neuen Zielpunkt jeweils eine (empirische) Neuparametrierung erfordern.

4.12 Vergleich der Rechenzeiten von KFM und EBM

Häufig wird die Rechenkomplexität von Algorithmen neben der Angabe der höchsten Ordnung (vgl. Kap. 4.10) ebenfalls durch die absolute Anzahl der benötigten Gleitkommaoperationen (Additionen und Multiplikationen) bzw. FLOPs (engl. floating-point operations, nicht zu verwechseln mit der Einheit FLOPS oder FLOP/s für das Leistungsmaß FLOPs pro Sekunde) bewertet. Die Ausführung komplexer Rechenoperationen (z.B. Division, Wurzel, etc.) kann plattformabhängig allerdings mehrere FLOPs in Anspruch nehmen, was durch unterschiedliche Gewichtungsverfahren Berücksichtigung findet. Laut einer vergleichenden Studie (Huss et al. 1987) liefert die in Tabelle 8 auszugsweise dargestellte Gewichtung gemäß der Livermore Loops (McMahon 1986), einem Benchmark für parallele Computer, hierbei die umfassendste Abschätzung.

Rechenoperationen	FLOPs
Addition (+), Subtraktion (-), Multiplikation (*)	1
Division (/), Wurzel (SQRT)	4
Exponentialfunktion (EXP), Trigonometrische Funktionen (SIN, COS, etc.)	8

Tabelle 8: Gewichtung verschiedener Rechenoperation gemäß Livermore Loops

Anhand der in Pseudocode verfassten Implementierung der KFM in Code 5 (vgl. Kap. 9.2) kann die Rechenkomplexität (gewichtet nach Tabelle 8) des für die Echtzeit relevanten zyklischen Teils der KFM (exklusive der problemspezifischen Implementierung der Abstoßungskräfte) angegeben werden. Außerdem wurde zum Vergleich die Rechenkomplexität der hier verwendeten eigenen Implementierung der EBM ebenfalls identifiziert.

$$O_{F,KFM}(n, D) = 12 \cdot D \cdot n - 3 \cdot D + 4 \text{ [FLOPs]} \quad (34)$$

$$O_{F,EBM}(n, D) = 23 \cdot D \cdot n + 21 \cdot n \text{ [FLOPs]} \quad (35)$$

Gl. (34) und Gl. (35) bestätigen die lineare Abhängigkeit der Rechenkomplexität von n und von D für sowohl KFM als auch EBM. Zudem bescheinigt die für $n \geq 1$ und $D \geq 1$ positive Differenz $O_{F,EBM}(n, D) - O_{F,KFM}(n, D) = 11 \cdot D \cdot n + 21 \cdot n + 3 \cdot D + 4 \text{ [FLOPs]}$ der KFM stets eine geringere Rechenkomplexität als der EBM. Abbildung 26 veranschaulicht diesen Zusammenhang nochmals graphisch. Hervorgehoben sind dabei die für Bahnplanungs-

probleme in der euklidischen Ebene bzw. im euklidischen Raum relevanten Fälle mit $D = 2$ bzw. $D = 3$.

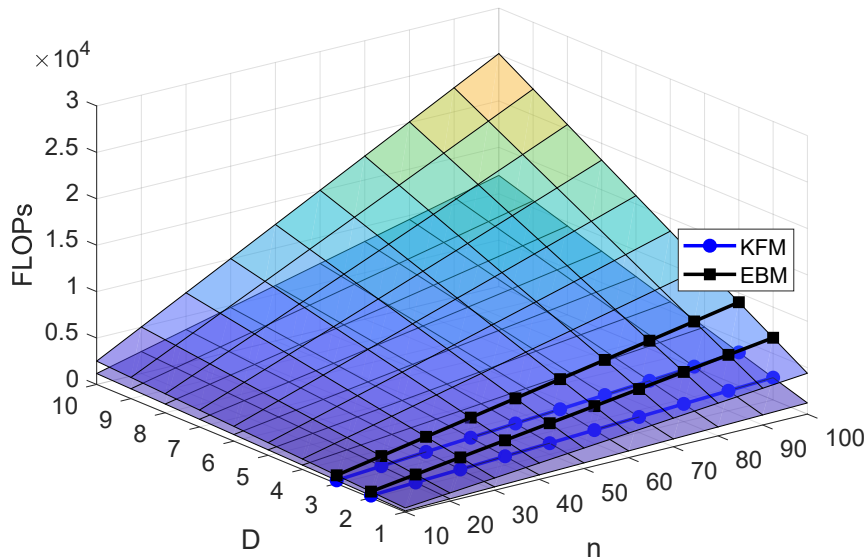


Abbildung 26: Theoretische Rechenkomplexität von KFM und EBM (nur Rückstellung)

Die höhere Rechenkomplexität der EBM ist dabei insbesondere auf zahlreiche Vektornormierungen (Rückstellkraft basiert bei EBM auf normierten Richtungsvektoren) mit nach Tabelle 8 „teuren“ Rechenoperationen (Divisionen und Wurzeln) zurückzuführen. Die KFM hingegen basiert fast ausschließlich auf in einem FLOP ausführbaren Rechenoperationen (Additionen, Subtraktion und Multiplikationen).

Über die absolute Komplexität (in FLOPs) für die Berechnung der Abstoßungskräfte kann keine allgemein gültige Aussage getroffen werden, da sie von weiteren Parametern (z.B. Anzahl der Hindernisse m) und deren problemspezifischer Implementierung abhängt. Dennoch kann die Ordnung der Komplexität bezüglich der Parameter n und D anhand der folgenden Überlegung abgeschätzt werden.

Da die Abstoßungskraft f_i nach Gl. (9) stützstellenweise ausgewertet wird, erfordert dies zugleich n Auswertungen der Gewichtungsfunktion ρ nach Gl. (12). Außerdem wird der Gradient von ρ im Allgemeinen nicht analytisch, sondern numerisch mithilfe eines Differenzenquotienten berechnet, was D (Vorwärts- oder Rückwärtsdifferenzenquotient) bzw. $2 \cdot D$ (zentraler Differenzenquotient) zusätzliche Auswertungen von ρ nach Gl. (12) erfor-

dert. Damit hängt die Rechenkomplexität der Abstoßung (ebenso wie die Rückstellung) lediglich in linearer Ordnung sowohl von der Diskretisierung n als auch von der Dimensionsanzahl D ab.

Um den realen Rechenaufwand der KFM gegenüber der EBM abzuschätzen, wurden Zeitmessungen der vorgestellten numerischen Experimente durchgeführt. Ein Vergleich mit den dokumentierten Rechenzeiten der EBM stellt sich aufgrund der unterschiedlichen und inzwischen sehr viel leistungsstärkeren Hardware schwierig dar. Außerdem beziehen sich diese Zeiten auf eine vollständige Implementierung der EBM mit einem komplexeren Hindernismodell. Aus diesem Grund wurden beide Methoden auf drei unterschiedlichen Plattformen (Standard-PC, Mikrocontroller und Industrie-PC) mit jeweils aktueller Soft- und Hardware ausgeführt, wobei die KFM gemäß Kap. 4.10 und die äquivalente Kernfunktionalität der EBM gemäß ihrer Erstveröffentlichung (Quinlan et al. 1993) implementiert wurden. Tabelle 9 gibt eine Übersicht über die wichtigsten Merkmale der verwendeten Systeme.

Plattform	Standard-PC	Mikrocontroller	Industrie-PC
Modell	Lenovo Thinkpad L540	Arduino Mega 2560	Beckhoff C6930-0050
CPU	Intel® Core™ i5-4200	Atmel ATmega2560	Intel® Core™ i7-4700EQ
Registerbreite	64 Bit	8 Bit	64 Bit
Taktung	2.5 GHz	16 MHz	2.4 GHz
RAM	4 GB	8 KB	8 GB
OS	Win7 (64 Bit) mit SP1	-	TwinCAT v3.1.4020.14
Echtzeit	Nein	Ja	Ja
Bemerkung	Ausführung unter MATLAB R2019a (interpretierter Code)	keine FPU, Umsetzung von FLOPs durch Compiler („Soft-FPU“)	TwinCAT läuft als Echtzeiterweiterung für Win7 Prof. (64 Bit) mit SP1

Tabelle 9: Plattformen zur Ermittlung des Rechenaufwands von KFM und EBM

In Tabelle 10 ist ein Auszug sowie in Abbildung 27 die vollständigen Ergebnisse der Zeitmessung dargestellt. Dabei wurde die anteilige Rechenzeit für Abstoßung und Rückstellung über 5000 (Standard- und Industrie-PC) bzw. 100 (Mikrocontroller) Zeitschritte sowie über drei Wiederholungen der Messung gemittelt. Die Messung beinhaltet ausschließlich

die zyklisch ausgeführten Bestandteile der Methoden. Für die KFM entspricht dies den Zeilen 044 bis 065 in Code 5 (vgl. Anhang in Kap. 9.1). Die für den Echtzeitbetrieb nicht relevante einmalige Initialisierung der Methoden wurde nicht berücksichtigt. Das „interne“ Eigenverhalten (Rückstellung) der Methoden wurde dabei, wie bereits in Kap. 4.6 erläutert, unabhängig von der problemspezifischen „externen“ Abstoßungskraft gemessen. Deren Implementierung teilen sich beide Methoden, was an den nahezu identischen Rechenzeiten für die Abstoßung auf allen Plattformen erkennbar ist.

Pl.-form	n	Rechenaufwand pro Zeitschritt in [ms]					
		Abstoßung		Rückstellung		Gesamt	
		EBM	KFM	EBM	KFM	EBM	KFM
Standard-PC	10	0.0340	0.0340	0.0415	0.0113	0.0755	0.0453
	20	0.0419	0.0418	0.0468	0.0124	0.0888	0.0542
	50	0.0704	0.0684	0.0510	0.0163	0.1214	0.0847
	100	0.1114	0.1119	0.0590	0.0237	0.1705	0.1356
Mikro-controller	10	8.6967	8.6967	2.8833	2.0133	11.5800	10.7100
	20	16.8033	16.8000	5.7400	3.9267	22.5433	20.7267
	50	41.7700	41.7733	14.3400	9.8533	56.1100	51.6267
	100	82.4500	82.4500	28.5600	19.2567	111.0100	101.7067
Industrie-PC	10	0.0017	0.0017	0.0007	0.0003	0.0024	0.0020
	20	0.0035	0.0035	0.0014	0.0005	0.0049	0.0040
	50	0.0088	0.0088	0.0035	0.0014	0.0123	0.0102
	100	0.0173	0.0174	0.0070	0.0028	0.0243	0.0202

Tabelle 10: Rechenaufwand von KFM und EBM auf verschiedenen Plattformen

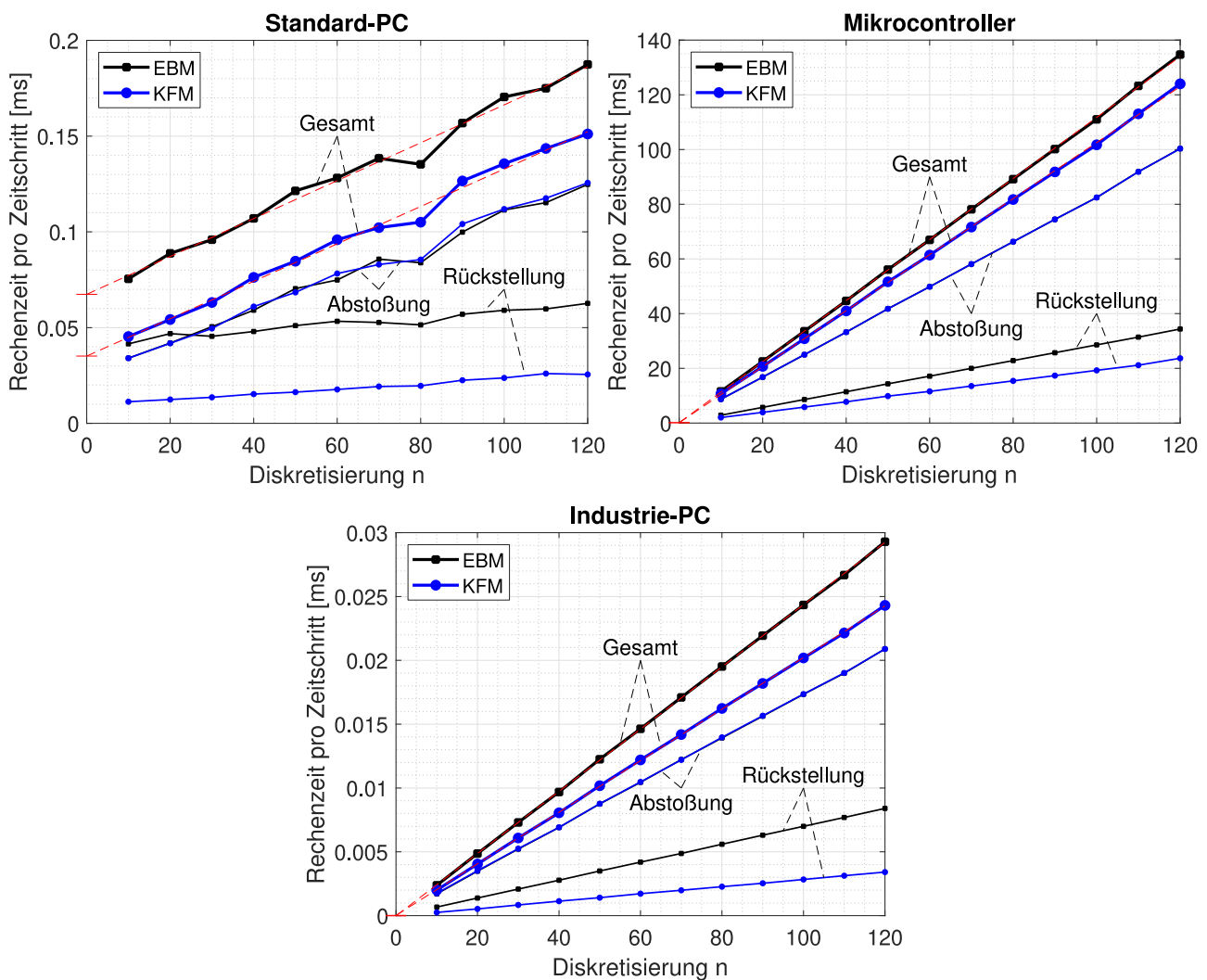


Abbildung 27: Lineare Rechenkomplexität der KFM auf allen Plattformen

Erwartungsgemäß kommt es bei der Nichtechtzeitausführung auf dem Standard-PC zu einer Verfälschung der absoluten Rechenzeiten aufgrund der durch Betriebssystem und Interpreter verursachten und die eigentliche Ausführung der KFM unterbrechenden Grundrechenlast. Dies ist daran erkennbar, dass die Messwerte nicht durch den Ursprung verlaufen sowie an deren hohem Jitter. Dennoch ist bereits hier die lineare Abhängigkeit des Rechenaufwands von der Diskretisierung n erkennbar.

Die Echtzeitausführungen auf Mikrocontroller und Industrie-PC bilden den Rechenaufwand darüber hinaus ohne erkennbare Grundrechenlast (Ursprungsgeraden) und Jitter ab. Das Verhältnis der Rechenzeiten für die Rückstellung weist beim Industrie-PC zudem eine sehr gute Übereinstimmung mit dem Verhältnis der nach Gl. (34) und Gl. (35) prognostizierten

Rechenkomplexität für KFM und EBM auf (nachfolgend exemplarisch für $n = 100$ dargestellt):

$$\frac{0.0028 [ms]}{0.0070 [ms]} = \boxed{0.4} \approx \frac{O_{F,KFM}(n = 100, D = 2)}{O_{F,EBM}(n = 100, D = 2)} = \frac{2398 [FLOPs]}{6700 [FLOPs]} = \boxed{0.358} \quad (36)$$

Aufgrund der fehlenden FPU (Gleitkommaeinheit, engl. floating-point unit) hat die dafür angewandte Gewichtung gemäß der Livermore Loops beim Mikrocontroller keine Gültigkeit.

Generell bestätigen alle drei Plattformen auch experimentell die lineare Rechenkomplexität der KFM in Abhängigkeit der Diskretisierung n . Dies gilt sowohl für die Rückstellung als auch für die Abstoßung. Zudem kann der KFM auch experimentell auf allen Plattformen eine höhere Recheneffizienz als der EBM bescheinigt werden.

Die absolute Rechenzeit der KFM auf dem Industrie-PC, welcher später ebenfalls als Zielplattform für die Realisierung dienen soll, belegt, dass die Aktualisierung der Bahn (Rückstellung) äußerst recheneffizient implementiert werden kann. Die Rechenzeiten liegen hier selbst für hohe Diskretisierungen im Bereich von wenigen Mikrosekunden und beanspruchen damit lediglich Rechenzeit in der Größenordnung eines Tausendstel der in der Problemstellung angestrebten Zykluszeit von wenigen Millisekunden. Damit ist gewährleistet, dass für die Berechnung der Abstoßungskraft mit komplexeren Umgebungsmodellen sowie für weitere Steuerungsfunktionen ausreichend Rechenzeit zur Verfügung steht.

5 Funktionen für die KFM in komplexen Umgebungen

Im vorangegangenen Kapitel konnte mit der KFM eine neue Bahnplanungsmethode hergeleitet werden, die in ihrem Funktionsprinzip der EBM entspricht und zugleich die Schwächen der EBM behebt. Die Beschreibung der KFM erfolgte dabei, soweit möglich, in abstrahierter Form, sodass die KFM als generische Methode zur Lösung einer Vielzahl von Bahnplanungsproblemen eingesetzt werden kann. Losgelöst von einem direkten Vergleich mit der EBM sollen in diesem Kapitel weitere Steuerungsfunktionen beschrieben werden, welche für den in der Problemstellung beschriebenen Anwendungsfall der KFM (simultaner Betrieb einer beliebigen Anzahl von UAVs) notwendig sind.

5.1 Stetig differenzierbares Kraftfeld

Die bisherige Formulierung der Gewichtungsfunktion ρ nach Gl. (12) als auch der Normierungsvorschrift H nach Gl. (11) erzeugen ein Kraftfeld f , welches nicht stetig differenzierbar ist. Abbildung 28 illustriert diese Unstetigkeit an den Übergängen der Wirkbereiche jeweils zweier angrenzender Hindernisse (schwarz) sowie am Übergang zu kräftefreien Bereichen (rot, entspricht Wirkweite $r_0 = 0.5 [m]$) anhand der bereits in Kap. 4.6 verwendeten Hindernispositionen.

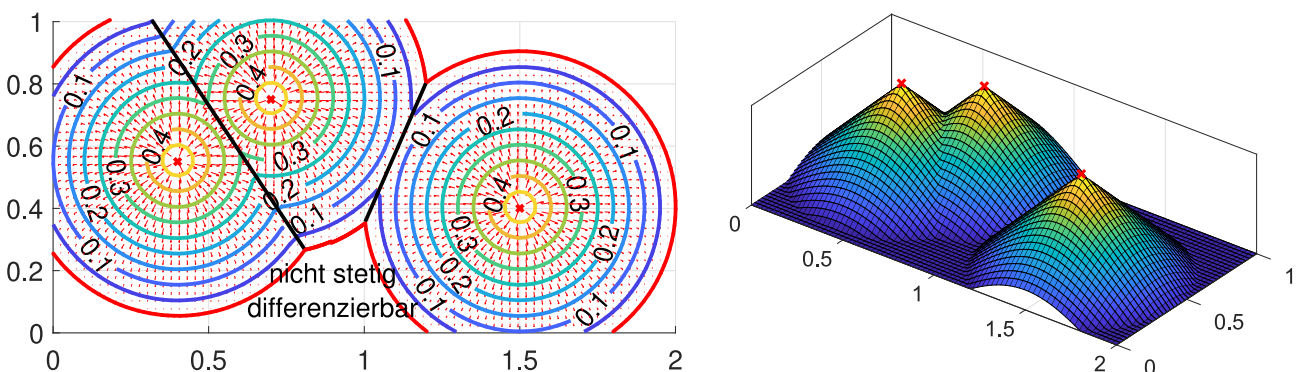


Abbildung 28: Nicht stetige Differenzierbarkeit des bisher verwendeten Kraftfeldes

Es wird deutlich, dass dadurch abrupte Richtungsänderungen des Gradienten von H und damit auch der auf die Stützstellen wirkenden abstoßenden Kräfte f möglich sind. Dies kann zu einem unerwünschten Bahnverhalten im Bereich der Übergänge führen. So kann

etwa die stationäre Endlage der Bahn hier ebenfalls Knicke aufweisen oder es kann aufgrund der Bahndiskretisierung zu einem „Nachziehen“ aufeinanderfolgender Stützstellen über die Unstetigkeitsgrenze kommen.

Um diese Effekte zu vermeiden, sollen die Unstetigkeiten des Kraftfeldes im Folgenden durch verallgemeinerte Definitionen der Gewichtungsfunktion ρ sowie der Normierungsvorschrift H geglättet werden. Damit die Parametrierung der KFM gemäß dem vorangegangenen Kapitel dabei nach wie vor ihre Gültigkeit behalten kann, soll das Maß der Glättung und damit die Abweichung vom ursprünglichen Kraftfeld einstellbar gestaltet werden.

Abbildung 29 zeigt die Erweiterung der Gewichtungsfunktion ρ um einen Viertelsinus (Phase 2) als stetig differenzierbaren Übergang zwischen linearem (Phase 1) und kräftefreiem (Phase 3) Bereich.

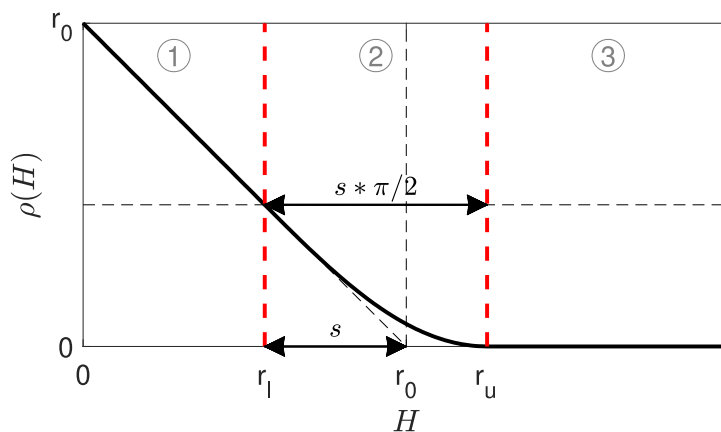


Abbildung 29: Erweiterung der Gewichtungsfunktion $\rho(H)$ um einen Viertelsinus

Unter Vorgabe der Obergrenze r_u (mit $r_0 \leq r_u \leq (\pi/2) \cdot r_0$) können der Skalierungsfaktor $s = (r_u - r_0)/(\pi/2 - 1)$ sowie die Untergrenze $r_l = r_0 - s$ als Hilfsvariablen für diese neue 3-phasige Gewichtungsfunktion bestimmt werden:

$$\rho(H(\mathbf{p}_i, \mathbf{h})) = \begin{cases} r_0 - H, & H \leq r_l \\ s \cdot \left(1 - \sin\left(\frac{H - r_l}{s}\right)\right), & r_l < H < r_u \\ 0, & H \geq r_u \end{cases} \quad (37)$$

Für $r_u = r_0$ verschwindet die Übergangsphase und die Gewichtungsfunktion entspricht wieder der ursprünglichen 2-phasigen (Phasen 1 und 3) Formulierung. Es ist zu berücksichtigen, dass bei geglätteter Gewichtungsfunktion (Phase 2 vorhanden) die tatsächliche Auslenkung von Stützstellen, deren stationäre Endlagen in Phase 2 liegen, geringfügig größer ist, als nach Kap. 4.9 anhand der rein linearen Gewichtungsfunktion ausgelegt.

Es hat sich weiter gezeigt, dass durch die Anwendung der p -Norm (Plato 2010) (Verallgemeinerung von u.a. Summen-, euklidischer und Maximumsnorm) mit $p \leq -1$ (anstatt üblicherweise $p \geq 1$) zur Gewichtung der Einzeldistanzen d_j in der Normierungsvorschrift H die gewünschte Glättung des Kraftfeldes zwischen angrenzenden Hindernissen erzielt werden kann. Es sei angemerkt, dass die p -Norm nach Gl. (38) für $p < 1$ keine Norm im mathematischen Sinne (vgl. Kap. 4.8) darstellt (Werner 2011).

$$H(d(\mathbf{p}_i, \mathbf{h})) = \left(\sum_{j=1}^m |d_j(\mathbf{p}_i, \mathbf{h}_j)|^p \right)^{\frac{1}{p}} \quad (38)$$

Durch diese „inverse“ Anwendung fließen nahe Hindernisse (entspricht kleinen Distanzen d_j) mit höherem Einfluss in H ein als weit entfernte Hindernisse. Für $p \rightarrow -\infty$ wird H gar ausschließlich von der kleinsten Einzeldistanz d_j dominiert und entspricht damit der „Minimumsnorm“ der ursprünglichen Formulierung (Mitrinovic et al. 1970).

Abbildung 30 zeigt das Kraftfeld bei nahezu keiner (oben), bei moderater (Mitte) sowie bei starker (unten) Glättung durch die Parameter r_u und p sowie die sich jeweils ergebende stationäre Endlage einer exemplarischen Bahn (mit $n = 20$ und $K = 0.1665$). Zum Vergleich mit der ursprünglichen ungeglätteten Formulierung ist zusätzlich die Äquidistanzlinie bei $r_0 = 0.5 [m]$ (rot) eingezeichnet. Anhand der Äquipotentiallinien (sowie der Gradientenvektoren) wird deutlich, wie die Glättung ausschließlich in Bereichen nahe der ursprünglichen Unstetigkeiten wirkt. Durch eine moderate Parametrierung bleiben Wirkbereich und Form des ursprünglichen Kraftfeldes weitestgehend erhalten, gleichzeitig weist das Kraftfeld nun aber eine allorts kontinuierliche Gradientenänderung auf.

5 Funktionen für die KFM in komplexen Umgebungen

Es sei angemerkt, dass durch die Anwendung der p -Norm und des Viertelsinus die H zugrundeliegende Metrik nun ein von der ursprünglich angesetzten minimalen Distanz zum nächstgelegenen Hindernis leicht abweichendes Maß beschreibt. Die Vorgaben für r_0 , r_u und r_l sind nun ebenfalls auf dieses Maß bezogen. Da H aber stets mindestens die euklidische Minimaldistanz beträgt, wird der Sicherheitsabstand der Bahn dadurch ebenfalls geringfügig vergrößert, niemals aber verringert. Da H hier lediglich als Maß für die Abstoßung und nicht etwa für eine quantitativ exakte Kontakt- oder Kollisionsrechnung herangezogen wird, ist diese geringe Abweichung tolerabel.

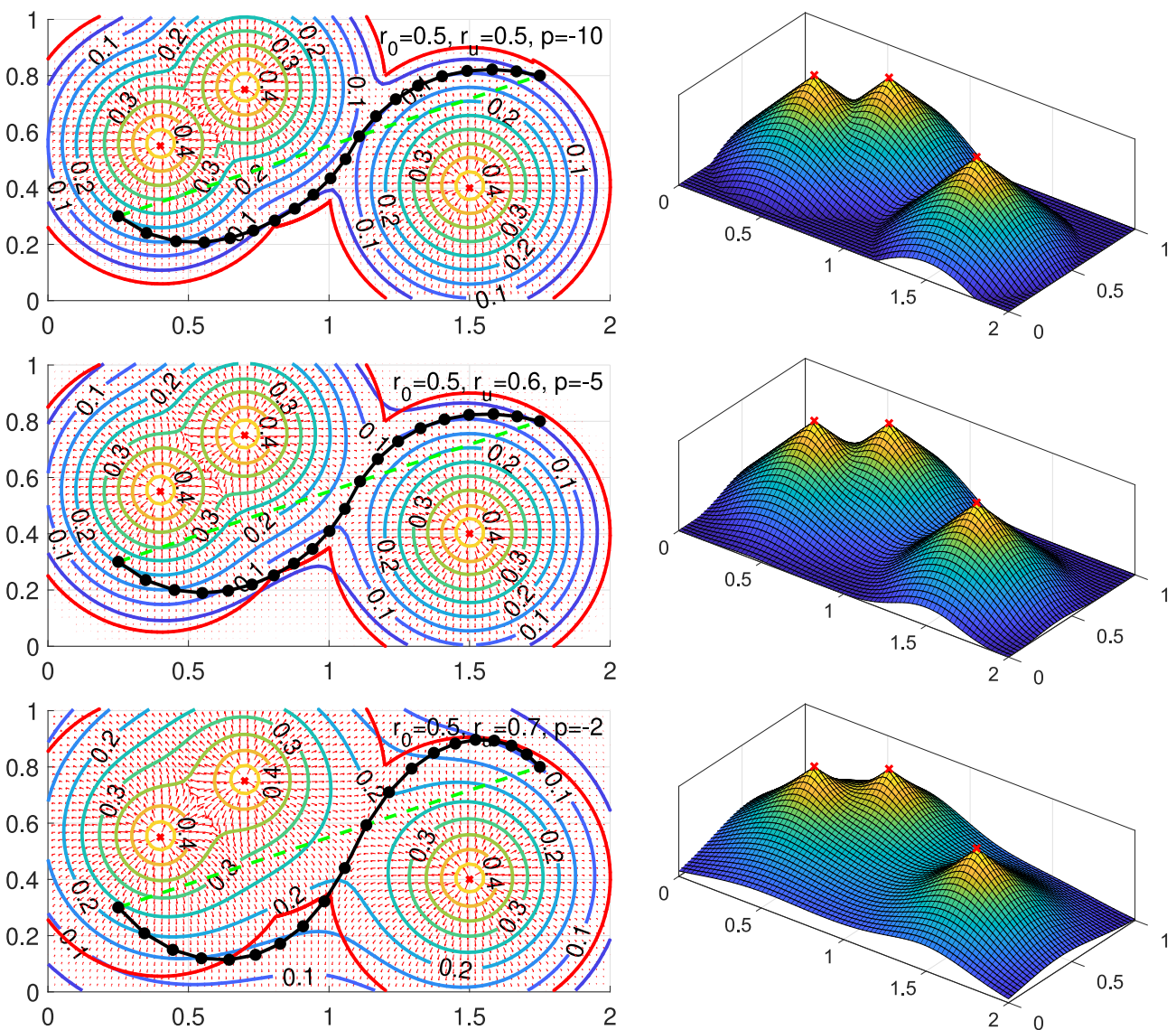


Abbildung 30: Geglättetes Kraftfeld für verschiedene Parametrierungen von r_u und p

5.2 Räumliche Hinderniskörper mit Durchdringung

Die Grundlage für die Berechnung der Abstoßungskräfte bildet ein geometrisches Abbild des aktuellen Umgebungszustandes mit allen relevanten bewegten und ruhenden Hindernissen. In den vorangegangenen Kapiteln beschränkte sich dieses Abbild auf lediglich punktförmige Hindernisse in der Ebene. Für die gemäß der Problemstellung angestrebte Modellierung des Arbeits- bzw. Luftraums für UAVs innerhalb von Fabrikhallen ist dies allerdings unzureichend. Zum einen ist generell eine räumliche Modellierung der Hindernisobjekte ($D = 3$) erforderlich und zum anderen müssen neben den UAVs selbst noch weitere Objekte, welche vereinfachend nicht mehr als Punkthindernis approximiert werden können, abgebildet werden. Dazu gehören insbesondere Wände als Arbeitsraumbegrenzungen sowie Versorgungsleitungen oder statische Strukturelemente des Gebäudes.

Neben der äußerst rechenintensiven Abbildung komplexer Hindernisgeometrien (z.B. in Form von Punktwolken aus bildgebenden Verfahren) mithilfe diskreter Belegungsgitter oder -bäume (Elfes 1989; Hwang et al. 1992) erfolgt die Beschreibung einzelner und einfacherer Objekte, deren Lage und Form bekannt ist, häufig durch eine Approximation mit Polyedern oder Primitiven als umhüllende Geometrien. Da letztere analytisch mit wenigen Parametern beschrieben werden, können Distanz- und Durchdringungsrechnungen explizit und ohne den zu D exponentiellen Rechenaufwand von Belegungsgittern erfolgen.

Obgleich die KFM grundsätzlich auf Basis beliebig komplexer Umgebungsmodelle verwendet werden kann, wird die Berechnung des abstoßenden Kraftfeldes nachfolgend auf die aus der Problemstellung hervorgehenden Anforderungen an die abzubildenden Objekte mit folgenden geometrischen Primitiven reduziert:

- Kugeln (engl. sphere) werden als umhüllende Geometrie für UAVs und weitere Hindernisse mit gleichmäßiger Ausdehnung verwendet. Die Beschreibung eines Kugelhindernisses $\mathbf{h}_{sph} = (\mathbf{m}_0, r)^T$ erfolgt über einen Mittelpunkt \mathbf{m}_0 sowie den Radius r (insgesamt 4 skalare Parameter). Die Distanz zum Punkt \mathbf{p}_i beschreibt die zugehörige Distanzfunktion $d_{sph}(\mathbf{p}_i, \mathbf{h}_{sph}) = |\mathbf{p}_i - \mathbf{m}_0| - r$.
- Zylinder (engl. cylinder) mit halbkugelförmigen Enden werden als umhüllende Geometrien für gesicherte Start- und Landekorridore und weitere Hindernisse mit läng-

licher Ausdehnung (z.B. Rohrleitungen oder Hallenstützen) verwendet. Die Beschreibung eines Zylinderhindernisses $\mathbf{h}_{cyl} = (\mathbf{m}_0, \mathbf{m}_1, r)^T$ erfolgt über ein Liniensegment mit den Endpunkten \mathbf{m}_0 und \mathbf{m}_1 sowie dem Radius r (insgesamt 7 skalare Parameter). Die zugehörige Distanzfunktion $d_{cyl}(\mathbf{p}_i, \mathbf{h}_{cyl})$ stellt somit die um den Radius r reduzierte Distanz des Punktes \mathbf{p}_i zu diesem Liniensegment dar und kann sehr recheneffizient implementiert werden (Ericson 2004).

- Ebenen (engl. plane) werden als Arbeitsraumbegrenzungen des zur Verfügung stehenden Luftraums verwendet. Die Beschreibung eines Ebenenhindernisses $\mathbf{h}_{pln} = (\mathbf{m}_0, \mathbf{n})^T$ erfolgt über einen beliebigen Ebenenpunkt \mathbf{m}_0 sowie einen Normalenvektor \mathbf{n} . Auch hier ist eine äußerst recheneffiziente Berechnung der Distanz $d_{pln}(\mathbf{p}_i, \mathbf{h}_{pln})$ zwischen Ebene und dem Punkt \mathbf{p}_i möglich (Ericson 2004).

Entgegen der bisherigen Betrachtung von lediglich punktförmigen Hindernissen ist bei den hier beschriebenen Hinderniskörpern ebenfalls der Fall der „Durchdringung“ (\mathbf{p}_i befindet sich innerhalb eines Kugel- oder Zylinderhindernisses bzw. im dem Normalenvektor \mathbf{n} abgewandten Halbraum eines Ebenenhindernisses) zu berücksichtigen.

Alle Distanzfunktionen werden so implementiert, dass der Abstand zur Körperhülle bei Durchdringung negativ und ansonsten positiv ist, wodurch die Stützstellen auch im Inneren der Hindernisse eine stets nach außen gerichtete Abstoßung erfahren. Andernfalls würden sie nach Überschreitung der Körperhülle in lokale Minima im Innern der Hindernisse gesogen. Diese Eigenschaft wird später für die Initialisierung (vgl. Kap. 5.5) der Bahn noch eine wichtige Rolle spielen.

Da die p-Norm nach Gl. (38) durch Verwendung der Absolutbeträge $|d_j(\mathbf{p}_i, \mathbf{h}_j)|$ die Vorzeichen der Distanzen neutralisiert wird die Normierungsvorschrift H „vorzeichenkonservierend“ angepasst (sgn beschreibt dabei die Signum- oder Vorzeichenfunktion). Zu Gunsten der Lesbarkeit wurde H_Σ als Hilfsvariable eingeführt.

$$H_{\Sigma}(d(\mathbf{p}_i, \mathbf{h})) = \sum_{j=1}^m \operatorname{sgn}(d_j(\mathbf{p}_i, \mathbf{h}_j)) \cdot |d_j(\mathbf{p}_i, \mathbf{h}_j)|^p \quad (39)$$

$$H(d(\mathbf{p}_i, \mathbf{h})) = \operatorname{sgn}(H_{\Sigma}(d(\mathbf{p}_i, \mathbf{h}))) \cdot |H_{\Sigma}(d(\mathbf{p}_i, \mathbf{h}))|^{\frac{1}{p}} \quad (40)$$

In Abbildung 31 sind die Äquipotentialflächen bei $|\mathbf{f}| = \{0.1; 0.2; 0.3; 0.4\}$ des räumlichen Kraftfeldes (mit $r_0 = 0.5$ [m], $r_u = 0.6$ [m] und $p = -5$) in einem exemplarischen Szenario bestehend aus jeweils zwei Ebenen- und Zylinderhindernissen sowie drei Kugelhindernissen dargestellt. Der Radius der Kugel- und Zylinderhindernisse beträgt dabei jeweils $r = 0.075$ [m]. Die vollständigen Koordinaten der einzelnen Hindernisse können dem Anhang (vgl. Kap. 9.4) entnommen werden.

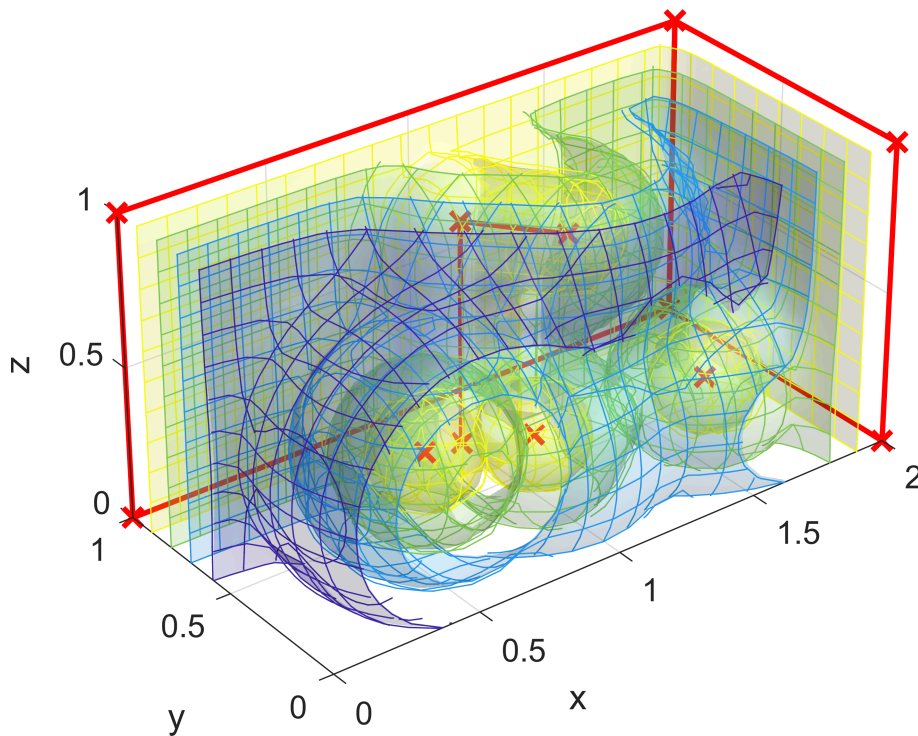


Abbildung 31: Äquipotentialflächen eines exemplarischen Kraftfeldes

Aus diesem skalaren Kraftfeld werden die Abstoßungskräfte, welche auf die einzelnen Stützstellen wirken, abgeleitet. Gemäß Gl. (9) entspricht die Abstoßungsstärke dabei dem Betrag des Feldes und die Abstoßungsrichtung dessen normiertem sowie negiertem Gradienten, dessen Berechnung im nachfolgenden Kapitel näher erläutert wird.

5.3 Gradientenberechnung

Da der Gradient $\nabla\rho(\mathbf{p}_i, \mathbf{h}) = \partial\rho(\mathbf{p}_i, \mathbf{h})/\partial\mathbf{p}$ (vgl. Kap. 4.6) zur Bestimmung der Abstoßungsrichtung einer Stützstelle \mathbf{p}_i im Allgemeinen nicht analytisch beschrieben werden kann, ist dessen numerische Berechnung mittels Differenzenquotienten (DQ) erforderlich. Je nach Wahl des zu verwendenden DQ (zentraler bzw. symmetrischer DQ oder Vorwärts- bzw. Rückwärts-DQ) sowie der Dimensionsanzahl D erhöht sich die erforderliche Anzahl an Auswertungen der Gewichtungsfunktion ρ . Abbildung 32 illustriert für $D = 3$ unter Verwendung der Einheitsvektoren $\mathbf{x}_e = (1,0,0)^T$, $\mathbf{y}_e = (0,1,0)^T$ und $\mathbf{z}_e = (0,0,1)^T$ sowie der Stör-schrittweite ε die für beide Varianten benötigten Auswertungen von ρ .

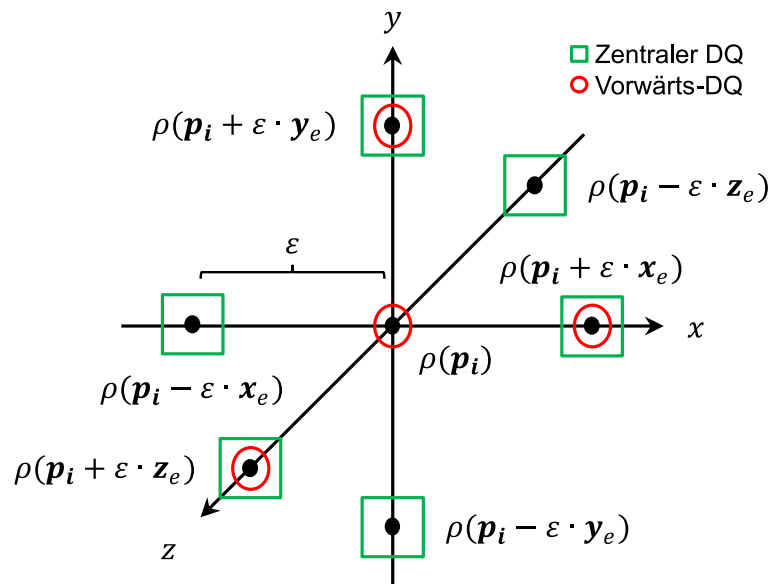


Abbildung 32: Auswertungen von ρ für verschiedene Differenzenquotienten

Verallgemeinert erfordert die Gradientenberechnung bei Anwendung des zentralen DQ $2 \cdot D$ und bei Anwendung des Vorwärts-DQ $D + 1$ Auswertungen von ρ . Da $\rho(\mathbf{p}_i)$ ohnehin zur Bestimmung der Abstoßungsstärke bestimmt werden muss, ergibt sich die Anzahl von Auswertungen von ρ zur Bestimmung der Abstoßungskräfte entlang der gesamten Bahn innerhalb eines Rechenzyklus zu $n \cdot (2 \cdot D + 1)$ bei zentralem DQ bzw. $n \cdot (D + 1)$ bei Vorwärts-DQ. Dies bekräftigt nochmals die in Kap. 4.12 experimentell nachgewiesene lineare Rechenkomplexität der Abstoßung sowohl in Abhängigkeit von n als auch von D .

Aufgrund des geringeren Rechenaufwands findet für die KFM im Folgenden ausschließlich der Vorwärts-DQ Anwendung. Auf dessen Basis kann der Gradient für $D = 3$ gemäß Gl. (41) numerisch approximiert werden. Als Störschrittweite wird $\varepsilon = 1 \cdot 10^{-3} [m]$ verwendet. Diese Wahl von ε liegt einerseits in einer Größenordnung, welche gemessen an der Positioniergenauigkeit von UAVs ausreichend erscheint, gleichzeitig aber auch noch für rechenschwache Plattformen mit geringer Maschinengenauigkeit geeignet ist.

$$\nabla\rho(\mathbf{p}_i, \mathbf{h}) = \partial\rho(\mathbf{p}_i, \mathbf{h})/\partial\mathbf{p} \approx \frac{1}{\varepsilon} \cdot \left(\begin{pmatrix} \rho(\mathbf{p}_i + \varepsilon \cdot \mathbf{x}_e, \mathbf{h}) \\ \rho(\mathbf{p}_i + \varepsilon \cdot \mathbf{y}_e, \mathbf{h}) \\ \rho(\mathbf{p}_i + \varepsilon \cdot \mathbf{z}_e, \mathbf{h}) \end{pmatrix} - \rho(\mathbf{p}_i, \mathbf{h}) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right) \quad (41)$$

Damit ist die Berechnung der Abstoßungskraft für die KFM vollständig beschrieben. Im Anhang ist in Code 6 (vgl. Kap. 9.3) eine in Pseudocode verfasste Implementierung der Berechnung der Abstoßungskraft \mathbf{f} zu finden.

5.4 Räumliche Bahn mit nichtlinearem Kraftfeld

In Kap. 4 wurde die KFM bisher lediglich für eine ebene ($D = 2$) Bahn mit $\mathbf{p} = (x, y)^T$ betrachtet. Die Erweiterung der KFM-DGL für räumliche ($D = 3$) Bahnen mit $\mathbf{p} = (x, y, z)^T$ erfordert aufgrund des entkoppelt beschriebenen Rückstellverhaltens lediglich die Ergänzung aller komponentenweise durchgeführten Berechnungsschritte um die zusätzliche z -Koordinate (vgl. Code 5 in Kap. 9.2).

Im Folgenden soll dargestellt werden, dass die theoretisch hergeleitete Parametrierung der KFM anhand einer eindimensionalen linearen Streckenlast auch für räumliche Bahnen unter Einfluss des erweiterten nichtlinearen Kraftmodells (vgl. Kap. 5.1 bis Kap. 5.3) aussagekräftig ist.

Dazu werden nachfolgend die Abstoßung und die Rückstellung einer räumlichen Bahn für ein einfaches und ein komplexes statisches Hindernisszenario in Form von Sprungantworten untersucht. Die Initialbahn \mathbf{p}_0 wird dabei so gewählt, dass stets alle Stützstellen unter Hinderniseinfluss stehen, sodass für die Abstoßung der vollbelastete Fall vorliegt.

Das einfache Szenario bildet die räumliche Analogie zur eindimensionalen Streckenlast anhand eines einzelnen und (zu den Koordinatenachsen) verdreht positionierten Zylinderhindernisses ab. Für das komplexe Szenario wird die Hindernisanordnung aus Abbildung 31 erneut aufgegriffen und soll die Wirkung eines Kraftfeldes mit ausgeprägten Nichtlinearitäten, welche bei hoher Hindernisdichte durch die starke Überlappung der einzelnen Wirkbereiche entstehen können, demonstrieren.

Für beide Szenarien erfolgt die Parametrierung der KFM anhand der folgenden Vorgaben:

- Die Diskretisierung der Bahn beträgt $n = 20$ (freie) Stützstellen.
- Das Kraftfeld wird durch die lineare Wirkweite $r_0 = 0.5 [m]$, die Obergrenze $r_u = 0.6 [m]$ sowie den Glättungsexponenten $p = -5$ beschrieben.
- Der Sicherheitsparameter K wird so ausgelegt, dass die durchschnittliche Stützstellenauslenkung bei linearer Streckenlast 75 Prozent der Wirkweite r_0 beträgt. Die Lösung des inversen Problems nach Kap. 4.9 liefert $K = 0.1139$ für diese Vorgabe.
- Der Dynamikparameter T wird so ausgelegt, dass die minimale Stellzeit zur Erreichung von 95 Prozent der stationären Endlage bei linearer Streckenlast $T_{s,min} = 1 [s]$ beträgt. Gl. (29) liefert $T = 0.0454$ für diese Vorgabe. Zusätzlich kann nach Gl. (18) die maximale Stellzeit bei unbelasteter Rückstellung mit $T_{s,max} = 6.0971 [s]$ angegeben werden.

Für das einfache Szenario wurden $\mathbf{p}_s = (0.25, 0.35, 0.25)^T [m]$ als Start- und $\mathbf{p}_g = (1.75, 0.85, 0.75)^T [m]$ als Zielpunkt der Bahn gewählt. Abbildung 33 zeigt die sich einstellende stationäre Endlage der Bahn (links) sowie die unbelastete Rückstellung aus dieser Endlage (rechts). Zusätzlich sind die Trajektorien der einzelnen Stützstellen und die an diesen initial ($t = 0$) und stationär ($t \rightarrow \infty$) wirkenden Kräfte dargestellt.

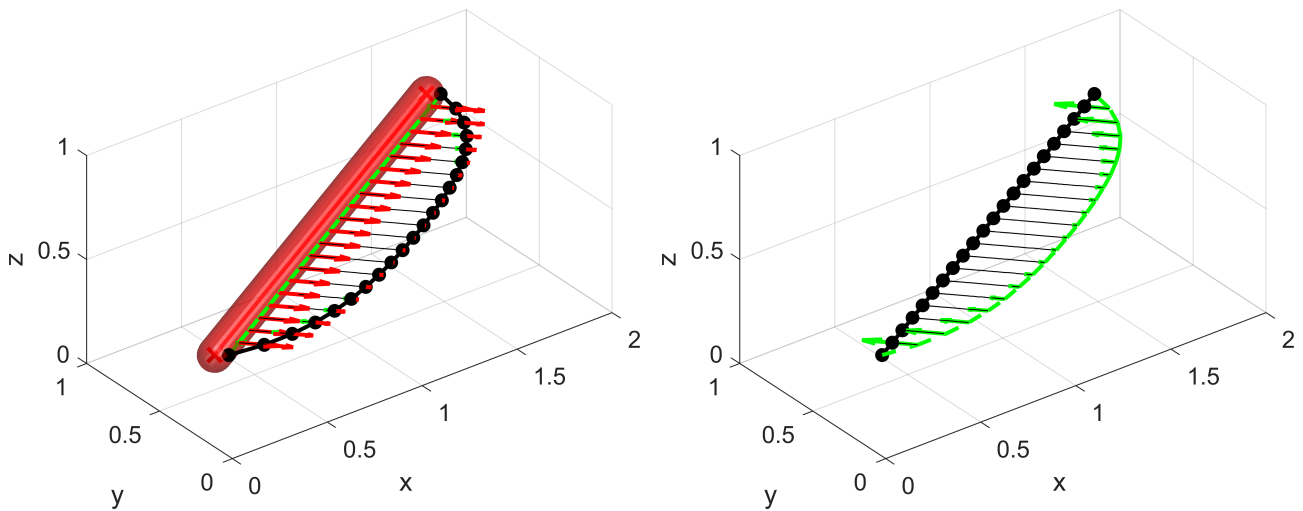


Abbildung 33: Abstoßung und Rückstellung für das einfache Szenario

Abbildung 34 zeigt die Verläufe der einzelnen (schwarz) sowie der durchschnittlichen (rot) räumlichen Stützstellenauslenkungen für die Abstoßung (links) und Rückstellung (rechts). Analog zum eindimensionalen Belastungsfall sind die Auslenkungen dabei relativ zur Initialstellung \mathbf{p}_0 beschrieben und entsprechen dadurch gleichzeitig der (euklidischen) Distanz zum Hindernis.

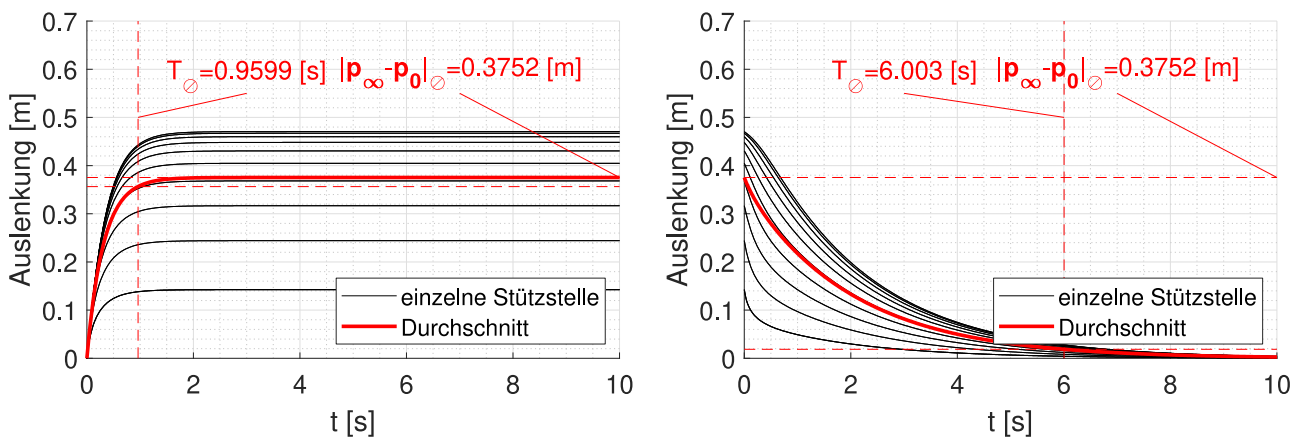


Abbildung 34: Auslenkung bei Abstoßung und Rückstellung für das einfache Szenario

Die Verläufe belegen, dass auch für den räumlichen Belastungsfall mit nichtlinearem Kraftmodell eine sehr gute Übereinstimmung der stationären Endlage ($|\mathbf{p}_\infty - \mathbf{p}_0|_\emptyset = 0.3752 \text{ [m]} \approx 0.75 \cdot r_0 = 0.375 \text{ [m]}$) sowie der Stellzeit ($T_\emptyset = 0.9599 \text{ [s]} \approx T_{s,min} = 1 \text{ [s]}$ bei Abstoßung und $T_\emptyset = 6.0032 \text{ [s]} \approx T_{s,max} = 6.0971 \text{ [s]}$ bei Rückstellung) mit den analytisch prognostizierten Werten vorliegt.

Für das komplexe Szenario wurde eine Initialbahn mit dem Startpunkt $\mathbf{p}_s = (0.25, 0.5, 0.25)^T [m]$ und dem Zielpunkt $\mathbf{p}_g = (1.65, 0.75, 0.85)^T [m]$ gewählt. Diese verläuft kollisionsfrei und vollständig innerhalb des Wirkungsbereichs der Hindernisse aus Abbildung 31, sodass auch hier eine Vollbelastung der Bahn vorliegt. Abbildung 35 zeigt die sich hierbei einstellende stationäre Endlage der räumlichen Bahn bei Abstoßung. Zusätzlich sind die Trajektorien der einzelnen Stützstellen und die an diesen initial ($t = 0$) und stationär ($t \rightarrow \infty$) wirkenden Kräfte dargestellt.

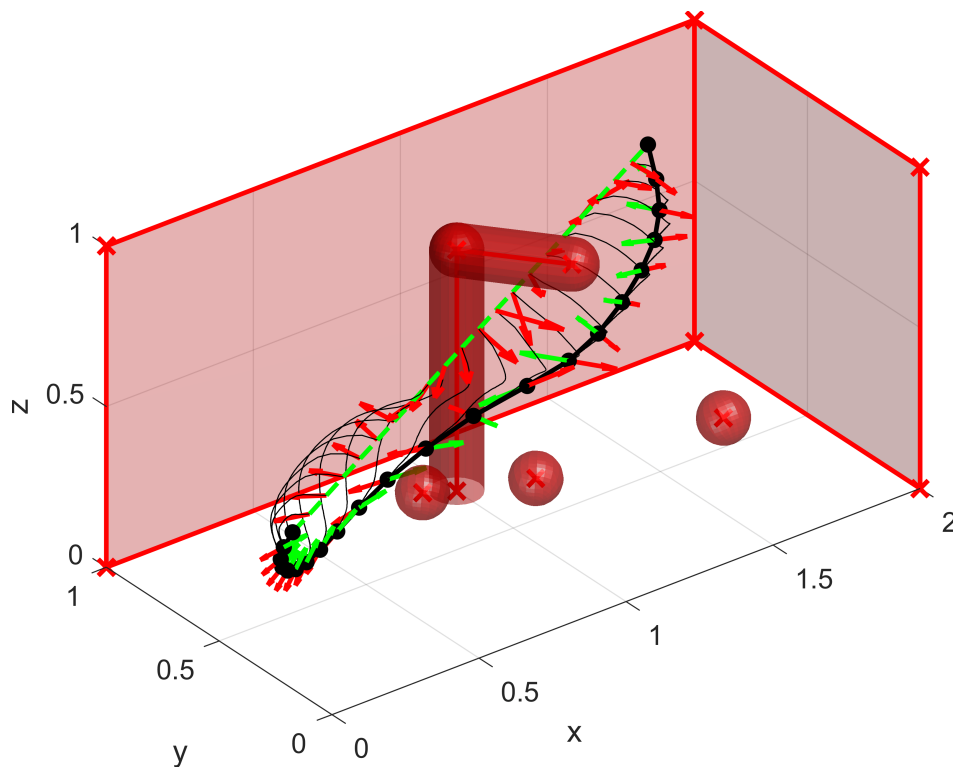


Abbildung 35: Abstoßung für das komplexe Szenario

Für ein besseres Verständnis der räumlichen Lage der Bahn sind in Abbildung 36 zusätzlich die in die drei Koordinatenebenen projizierten (orthogonalen) Ansichten dargestellt.

Da in diesem Szenario die (relativ zu \mathbf{p}_0 gemessenen) Auslenkungen der Stützstellen nicht mehr äquivalent zu deren minimaler Distanz zum nächsten Hindernis sind, zeigt Abbildung 37 die tatsächlichen Distanzen sowie deren Durchschnittswert. Zusätzlich sind die Zeitpunkte markiert, an welchen ein Wechsel des der jeweiligen Stützstelle nächstgelegenen Hindernisses erfolgt.

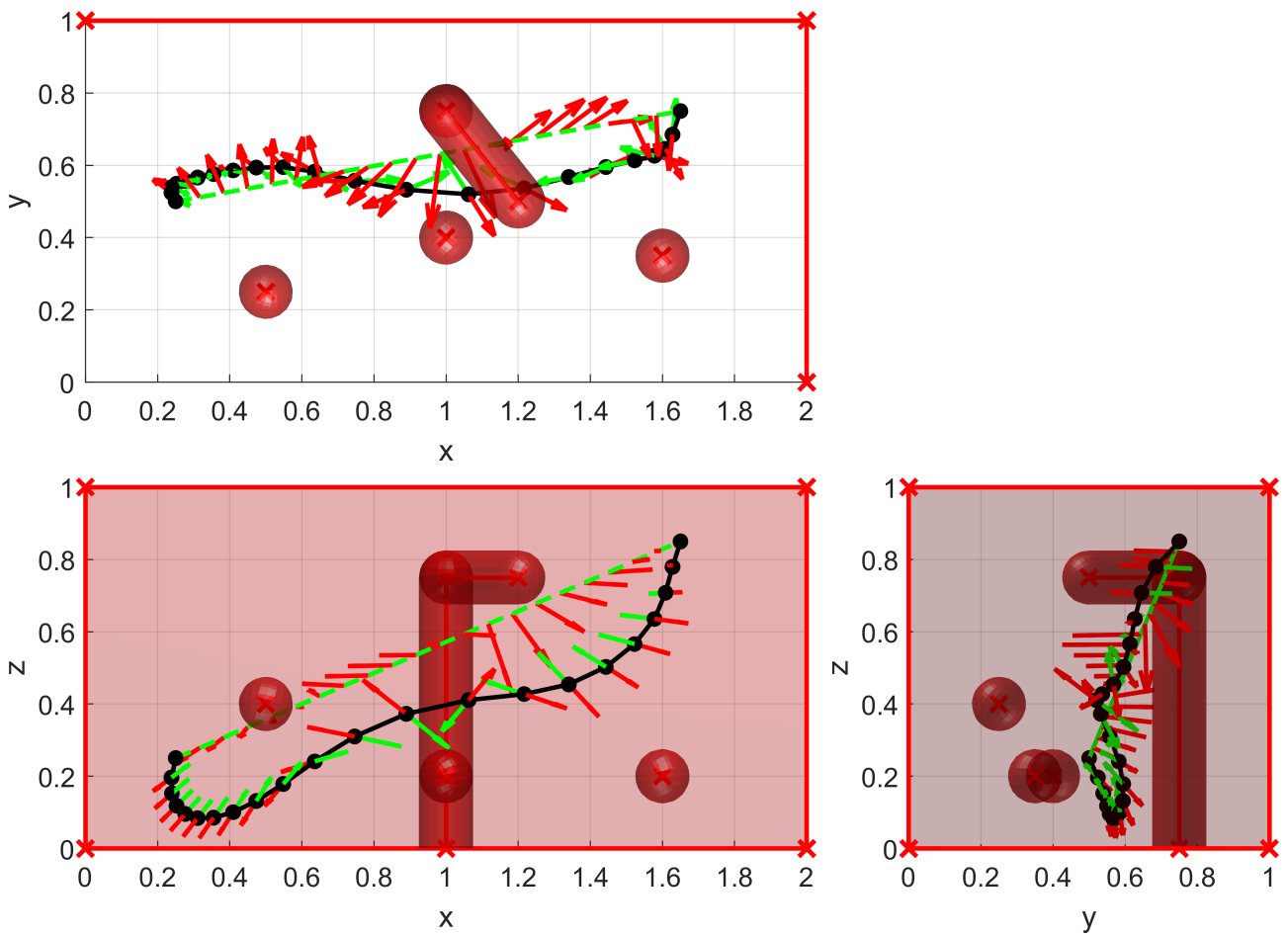


Abbildung 36: Orthogonale Ansichten für das komplexe Szenario

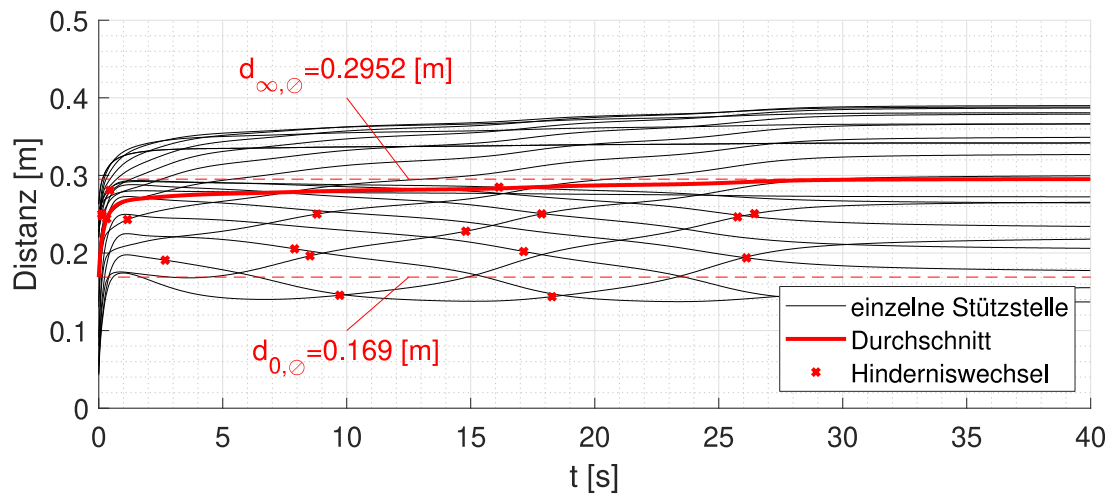


Abbildung 37: Stützstellendistanzen bei Abstoßung für das komplexe Szenario

Es wird deutlich, dass auch hier die Distanzzunahmen aufgrund der Vollbelastung initial PT_1 -ähnliche Verläufe beschreiben. Anschließend ist ein langsames „Kriechverhalten“ der Stützstellen zu beobachten. Dieses entsteht durch Wechselwirkungen zwischen den Stützstellen im zwar betragsschwachen aber durch die Überlagerungen der einzelnen Hinderniseinflüsse stark nichtlinearen Randbereich des abstoßenden Kraftfeldes. Dabei bewegen sich die Stützstellen vorwiegend tangential zur Bahn und verändern deren Verlauf nur unmerklich.

Obgleich sich dabei die Distanzen einzelner Stützstellen wieder verringern können, konvergiert die durchschnittliche Distanz stets monoton steigend in die für die gesamte Bahn energieminimale und damit entsprechend der Parametrierung optimale Lage. Ebenso ist erkennbar, dass auch bei Wechseln des einer Stützstelle nächstgelegenen Hindernisses (markiert durch rote Kreuze), deren stetige Differenzierbarkeit aufgrund der Glättung des Kraftfeldes erhalten bleibt.

Es ist offensichtlich, dass aufgrund dieser komplexen Stützstellenbewegungen im nichtlinearen Hindernisfeld eine Identifikation des Bahnverhaltens bei Abstoßung sowie die quantitative Gegenüberstellung mit dem linearen Streckenlastmodell nicht mehr möglich ist. Dass die Parametrierung der KFM auf Basis der linearen Streckenlast auch für komplexe Umgebungen Gültigkeit besitzt, soll daher später in Form von experimentellen Evaluierungsszenarien (vgl. Kap. 6.5 und Kap. 6.6) validiert werden.

5.5 Initialbahn

Dynamische Bahnplanungsmethoden, welche auf einem numerischen Lösungs- bzw. Optimierungsverfahren basieren, benötigen in der Regel eine kollisionsfreie Bahn als Initiallösung, um Stabilität und Konvergenz der Methode sicherzustellen. Die Wahl einer geeigneten Initialbahn wird typischerweise mit einer der nachfolgend aufgeführten Strategien realisiert, welche mit unterschiedlichen Nachteilen verbunden sind.

- Die Verwendung einer globalen Wegsuche zur Erzeugung einer Initialbahn ist rechenintensiv und algorithmisch komplex umzusetzen.

- Die parallele Erzeugung (und Prüfung) einer Vielzahl zufälliger Initialbahnen ist ebenfalls rechenintensiv und nicht deterministisch.
- Die Verwendung einer geraden Initialbahn zwischen Start- und Zielpunkt unter der Voraussetzung eines initial hindernisfreien Arbeitsraums ist nur für den Laborbetrieb praktikabel.

Aufgrund ihres numerisch robusten Lösungsverhaltens besitzt die KFM die Eigenschaft, dass sie für beliebige Initialbahnen stabil gelöst werden kann. Mithilfe des Durchdringungen berücksichtigenden Hindernismodells ist somit auch für eine initial kollisionsbehaftete Bahn eine monotone Konvergenz in eine (lokal) kollisionsfreie Lage sichergestellt.

Die Stützstellen werden dazu initial äquidistant zwischen dem Startpunkt \mathbf{p}_s und dem Zielpunkt \mathbf{p}_g verteilt. Somit entspricht die Initialbahn $\mathbf{p}_0 = (x_0, y_0, z_0)^T$ der stationären Endlage ($\dot{\mathbf{p}} = \mathbf{0}$) der unbelasteten KFM-DGL und kann komponentenweise durch Lösen der linearen GLS $\mathbf{A} \cdot \mathbf{x}_0 = -\mathbf{b}_x$, $\mathbf{A} \cdot \mathbf{y}_0 = -\mathbf{b}_y$ und $\mathbf{A} \cdot \mathbf{z}_0 = -\mathbf{b}_z$ bestimmt werden. Da \mathbf{A} eine Toeplitz-Matrix darstellt, erfolgt dies am effizientesten mithilfe des Thomas-Algorithmus oder die Lösung kann mit identischem Rechenaufwand auch stützstellenweise gemäß Gl. (42) direkt angegeben werden.

$$\mathbf{p}_{i,0} = \mathbf{p}_s + (\mathbf{p}_g - \mathbf{p}_s) \cdot \frac{i}{n+1} \quad (42)$$

Für die bisherigen Beispiele wurde stets eine kollisionsfreie Initialbahn gewählt. Abbildung 38 zeigt ein Szenario, bei welchem ca. 64 Prozent (rot markiert) der Initialbahn (mit $\mathbf{p}_s = (0.25, 0.85, 0.55)^T [m]$ und $\mathbf{p}_g = (1.75, 0.55, 0.15)^T [m]$) innerhalb eines Zylinderhindernisses mit $\mathbf{h}_{cyl} = (\mathbf{m}_0 = (0.4, 0.7, 0.3)^T, \mathbf{m}_1 = (1.4, 0.7, 0.3)^T, r = 0.1875) [m]$ verlaufen. Es wurde dabei dieselbe Parametrierung der KFM wie in Kap. 5.4 verwendet.

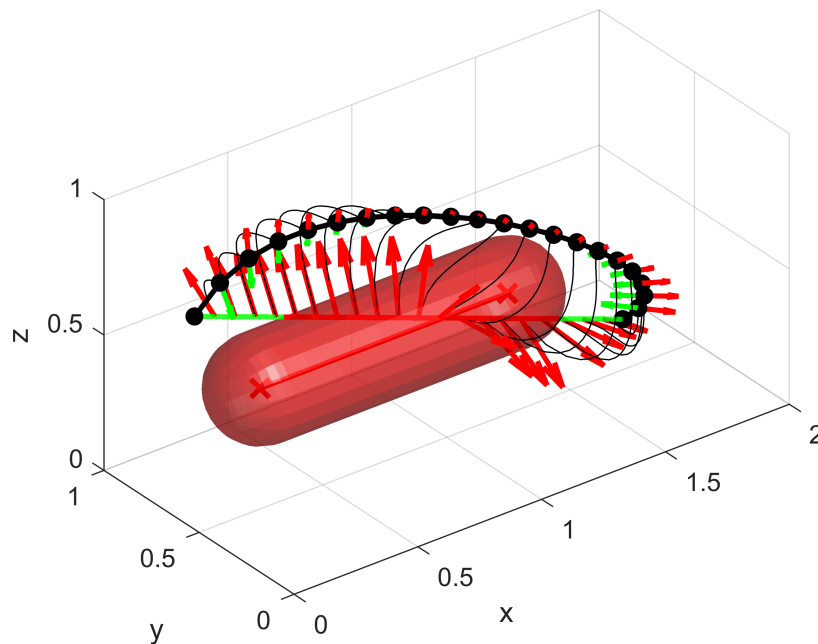


Abbildung 38: Einschwingvorgang einer kollisionsbehafteten Initialbahn

Dieses Szenario belegt, dass die KFM auch auf Basis einer kollisionsbehafteten Bahn initialisiert werden kann und somit nicht auf eine der eingangs genannten Initialisierungsstrategien angewiesen ist.

Da während dieses initialen Einschwingvorgangs die Kollisionsfreiheit der Bahn noch nicht gewährleistet werden kann, sollte mindestens die vollbelastete Stellzeit $T_{s,min}$ abgewartet werden, bevor mit der Ausführung der Bahn begonnen wird.

5.6 Interpolation auf dynamischer Bahn

Die direkte Verwendung der Stützstellen \mathbf{p} als Lagesollwerte für den Positionsregelkreis der UAVs würde aufgrund der groben Diskretisierung n zu einem unkontrollierten Bahnfolgeverhalten mit starkem Überschwingen führen.

Aus diesem Grund sollen die Stützstellen \mathbf{p} lediglich als Viapoints bzw. Kontrollpunkte für eine nachgeschaltete Bahninterpolation dienen. Mithilfe der Bahninterpolation wird ein die Dynamikgrenzen der UAVs nicht übersteigender Lagesollwert \mathbf{w} im Steuerungstakt Δt generiert. Eine besondere Herausforderung stellt hierbei die während der Ausführung variierende Form und damit ebenfalls variierende Länge der Bahn dar.

Die hier verwendete Interpolation beschreibt die Generierung des Lagesollwerts \mathbf{w} im Steuerungstakt Δt durch die nachfolgende zeitdiskrete Transitions Gleichung (43):

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \Delta t \cdot v(d_w) \cdot \left(\frac{\mathbf{p}_{next}^k - \mathbf{w}^k}{|\mathbf{p}_{next}^k - \mathbf{w}^k|} \right) \quad (43)$$

Dabei entspricht \mathbf{p}_{next}^k dem im aktuellen Steuerungstakt k jeweils als nächstes anzufliegenden Kontrollpunkt \mathbf{p}_i bzw. dem abschließend anzufliegenden Zielpunkt \mathbf{p}_g (insgesamt also $n + 1$ nacheinander anzufliegende Punkte). Der initiale Lagesollwert \mathbf{w}^0 entspricht dem Startpunkt \mathbf{p}_s und die Geschwindigkeit in Bewegungsrichtung ist v .

Damit die Stützstellen nicht mit einem Genauhalt angefliegen werden müssen, wird in Anlehnung an das Positionsüberschleifen von Point-to-Point-Steuerungen (PTP-Steuerungen) eine Überschleifkugel mit dem Radius r_{ue} sowie die zugehörige Weichschaltbedingung $|\mathbf{p}_{next}^k - \mathbf{w}^k| \leq r_{ue}$ definiert (Weber 2019).

Um am Start- und Zielpunkt ein möglichst überschwingfreies Folgeverhalten zu gewährleisten, wird der Lagesollwert mit einem beschleunigungsbegrenzten Geschwindigkeitsprofil (3-Phasenprofil) überlagert. Dieses Profil setzt sich dabei aus den folgenden 3 Bewegungsphasen zusammen:

1. Phase konstanter Beschleunigung mit $a = a_{max}$ bis maximale Geschwindigkeit $v = v_{max}$ erreicht wird
2. Phase konstanter Geschwindigkeit mit $v = v_{max}$
3. Phase konstanter Verzögerung mit $a = -a_{max}$ bis Geschwindigkeit $v = 0$ [m/s] erreicht wird

Für kurze Bahnlängen oder geringe Maximalbeschleunigungen a_{max} kann auch ein direkter Übergang von Phase 1 nach Phase 3 erfolgen, ohne dabei die Maximalgeschwindigkeit v_{max} zu erreichen.

Da die Bahn nicht in einer nach der Bogenlänge parametrisierten Beschreibung vorliegt und zusätzlich während der Ausführung deren Form und Länge variiert, ist die sonst in der sta-

tischen Bewegungsplanung übliche zeitabhängige Zuordnung der Phasen und deren Übergänge nicht ohne weiteres möglich.

Um die rechenintensive zyklische Konvertierung der Bahnparametrierung in deren Bogenlänge zu umgehen, wird nachfolgend ein vereinfachter Ansatz vorgestellt, welcher das Geschwindigkeitsprofil in Abhängigkeit der Distanz d_w zwischen aktuellem Lagesollwert \mathbf{w}^k und Start- bzw. Zielpunkt beschreibt.

Mithilfe des Weg-Zeit-Gesetzes für eine gleichmäßig beschleunigte Bewegung wird dazu zunächst die Beschleunigungs- und Verzögerungszeit $t_a = v_{max}/a_{max}$ und damit anschließend nach Gl. (44) der zugehörige Weg s_a ermittelt.

$$s_a = \frac{1}{2} \cdot \frac{v_{max}^2}{a_{max}} \quad (44)$$

Unterschreitet die euklidische Distanz d_w zwischen aktuellem Lagesollwert \mathbf{w}^k und dem Startpunkt \mathbf{p}_s oder dem Zielpunkt \mathbf{p}_g den zum Beschleunigen und Verzögern benötigten Weg s_a , wird die Geschwindigkeit v gemäß Gl. (45) und Gl. (46) proportional zu $\sqrt{d_w}$ angepasst. Die Störung ε ermöglicht dabei für $|\mathbf{p}_s - \mathbf{w}^0| = 0$ das initiale „Losinterpolieren“ am Startpunkt \mathbf{p}_s .

$$d_w = \min(|\mathbf{p}_s - \mathbf{w}^k| + \varepsilon, |\mathbf{p}_g - \mathbf{w}^k|, s_a) \quad (45)$$

$$v(d_w) = \sqrt{\frac{d_w}{s_a}} \cdot v_{max} \quad (46)$$

Da die Bahn \mathbf{p} im Allgemeinen gekrümmt verläuft, führt diese Interpolation zu einer geringfügigen Streckung der ersten bzw. dritten Phase, sodass die Phasenübergänge etwas später bzw. früher einsetzen und somit das Beschleunigungs- bzw. und Verzögerungsvermögen a_{max} nicht gänzlich ausgeschöpft wird. Abbildung 39 illustriert diesen Zusammenhang anhand des direkten (Luftlinie) und des auf der Bahn „abgerollten“ Verzögerungsweges s_a zum Zeitpunkt des Übergangs des Lagesollwerts \mathbf{w} von Phase 2 nach Phase 3. Analoges gilt auch für die Beschleunigung.

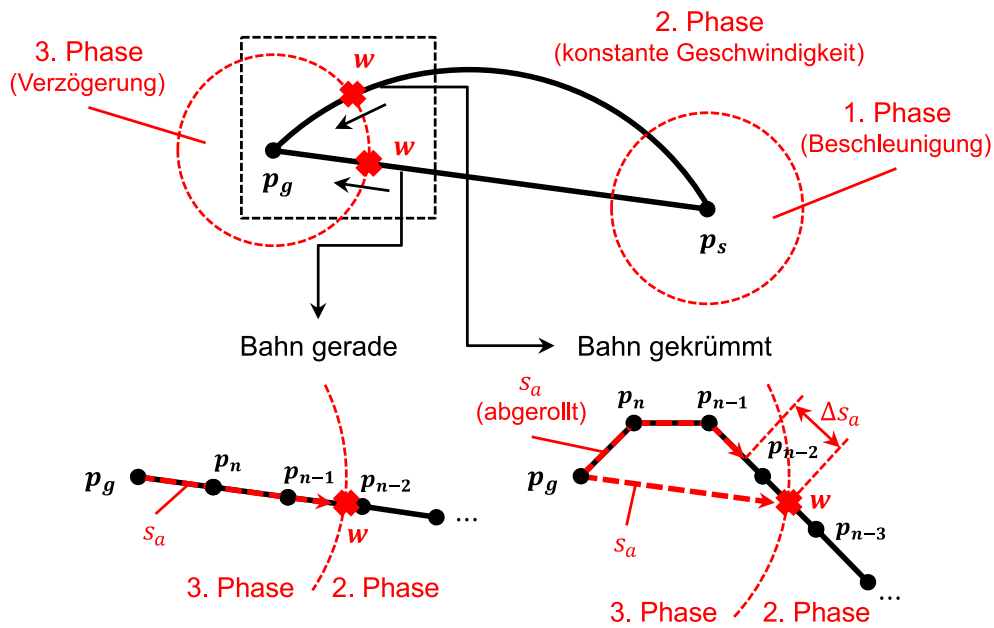


Abbildung 39: Verlängerung des Verzögerungsweges um Δs_a durch Bahnkrümmung

Für die simulative Validierung des nach Gl. (43) bis Gl. (46) generierten Lagesollwerts wurden Interpolationen für sowohl die gerade Initialbahn ($L_0 = 1.5811 \text{ [m]}$) als auch für die gekrümmte und damit längere stationäre Bahn ($L_\infty = 2.2005 \text{ [m]}$) aus Abbildung 38 durchgeführt. Die Dynamikgrenzen wurden zu $v_{max} = 0.5 \text{ [m/s]}$ und $a_{max} = 0.25 \text{ [m/s}^2\text{]}$ gewählt. Abbildung 40 zeigt die aus dem Zeitverlauf von w numerisch abgeleiteten Profile für den Weg s , die Geschwindigkeit v und die Beschleunigung a in Bewegungsrichtung.

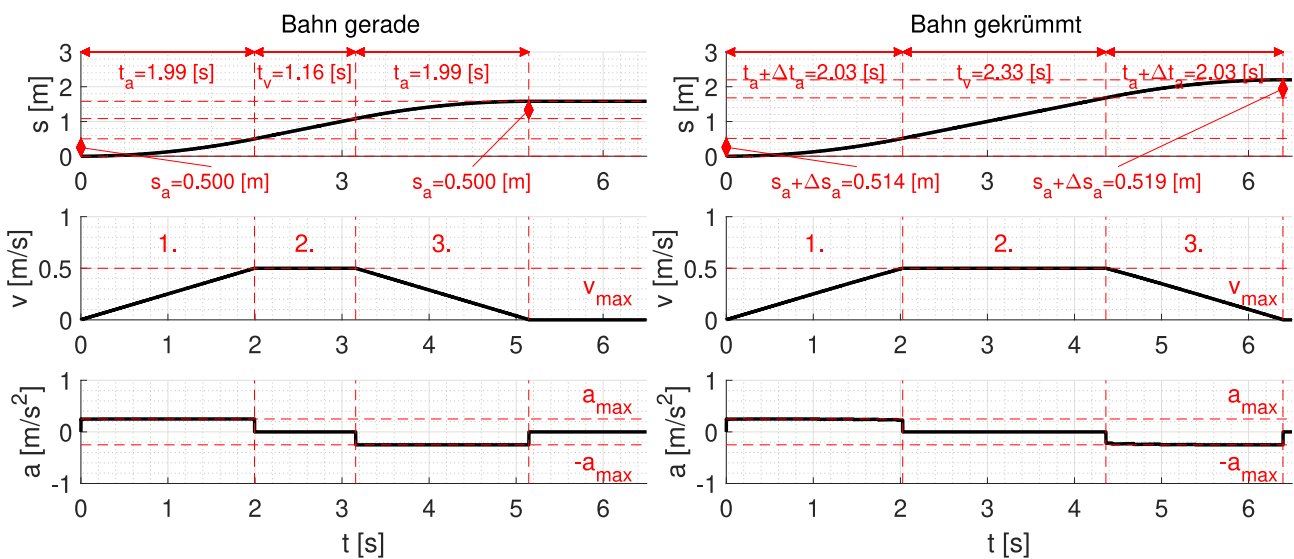


Abbildung 40: Bewegungsprofile einer geraden und einer gekrümmten Bahn

Die Verläufe belegen die Erzeugung eines die Dynamikvorgaben einhaltenden und beschleunigungsbegrenzten Sollwertprofils. Der Vergleich zwischen gerader und gekrümmter Bahn macht deutlich, dass die dabei entstehende Verlängerung der Beschleunigungs- und Verzögerungsphasen (ca. $\Delta s_a \approx 0.014[m]$ bzw. $\Delta s_a \approx 0.019[m]$ und $\Delta t_a \approx 0.04[s]$) vernachlässigbar ist, zumal dadurch das Beschleunigungs- und Verzögerungsvermögen niemals überschritten wird.

5.7 Parallele Ausführung mehrerer Instanzen der KFM

Es soll nun der parallele Betrieb der KFM für mehrere UAVs betrachtet werden. Dabei wird jedem UAV eine eigene Instanz der KFM zugeordnet, wobei alle Instanzen auf demselben Umgebungsmodell h operieren. Dieses Modell, welches bisher ausschließlich statische Hindernisse abgebildet hat, muss dazu um die UAVs als dynamische Hindernisse erweitert werden. Die UAVs werden dabei als umhüllende Kugeln mit dem Radius r_{uav} approximiert und deren Istpositionen (zunächst identisch mit den Lagesollwerten w) werden im Hindernismodell in jedem Steuerungstakt aktualisiert.

Um zu verhindern, dass die UAVs dabei jeweils ihre eigenen Bahnen verdrängen (in Kombination mit einer Interpolation auf der Bahn würde dies zudem in einer instabilen Rückkopplung resultieren), muss deren Wirkung bei der Berechnung des individuellen Kraftfeldes jeweils ignoriert werden. Es sei zusätzlich angemerkt, dass die Bahnen selbst keine physikalischen Hindernisse darstellen und sich gegenseitig nicht verdrängen. Somit sind Durchdringungen von Bahnabschnitten möglich.

Abbildung 41 zeigt ein Szenario mit drei UAVs mit jeweils einer eigenen Instanz der KFM. Um ein hohes Kollisionspotenzial zu erzwingen, wurden die Initialbahnen von UAV₁ ($p_{s,1} = (1.5, 1.7, 1.8)^T [m]$, $p_{g,1} = (0.3, 0.3, 0.3)^T [m]$, rot), UAV₂ ($p_{s,2} = (1.7, 0.2, 0.4)^T [m]$, $p_{g,2} = (0.2, 1.2, 1.7)^T [m]$, grün) und UAV₃ ($p_{s,3} = (0.2, 1.7, 0.3)^T [m]$, $p_{g,3} = (1.7, 0.4, 1.6)^T [m]$, blau) so gewählt, dass sich alle drei UAVs gegenläufig auf etwa halber Strecke treffen. Für die Interpolation aller UAVs gelten dieselben Dynamikbegrenzungen $a_{max} = 0.25 [m/s^2]$ und $v_{max} = 0.5 [m/s]$ und es wurde erneut die Parametrierung aus Kap. 5.4 für die KFM verwendet.

In Abbildung 41 sind links die sich durch die Wechselwirkungen der drei KFM-Instanzen ergebenden Sollwerttrajektorien (Kreuzmarkierungen im halben Sekundenabstand) sowie rechts die zugehörigen Verläufe der Abstände zwischen den UAVs dargestellt. Zusätzlich sind die Abstände ohne eine Anpassung der Bahnen durch die KFM, welche zu Kollisionen zwischen den UAVs führen, gestrichelt eingezeichnet. Da alle Hindernisse denselben Radius besitzen, ist die Durchdringung auf $2 \cdot r_{uav} = 0.3 [m]$ begrenzt.

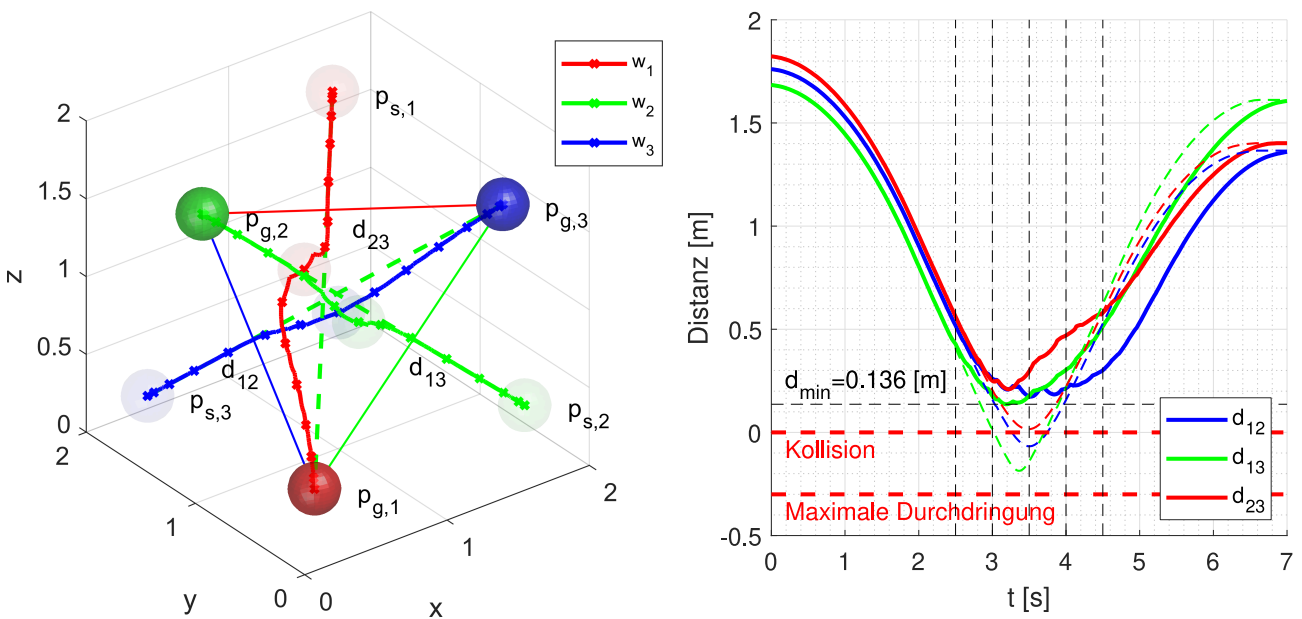


Abbildung 41: Sollwerttrajektorien und Abstände für drei UAVs

Die UAVs weichen einander in diesem Szenario mit einem $d_{min} = 0.136 [m]$ nicht unterschreitenden Abstand aus. Der kollisionskritische Bereich zwischen $t = 2.5 [s]$ und $t = 4.5 [s]$ ist in Abbildung 42 nochmals detailliert mit den jeweils aktuellen Bahnen dargestellt. Die in die drei Koordinatenebenen projizierten orthogonalen Ansichten verdeutlichen dabei nochmals, dass zu jedem Zeitpunkt Kollisionsfreiheit zwischen allen drei UAVs besteht. Kollisionsfreiheit zwischen zwei UAVs ist jeweils dann gegeben, wenn sie sich in mindestens einer Projektion nicht überlappen.

5 Funktionen für die KFM in komplexen Umgebungen

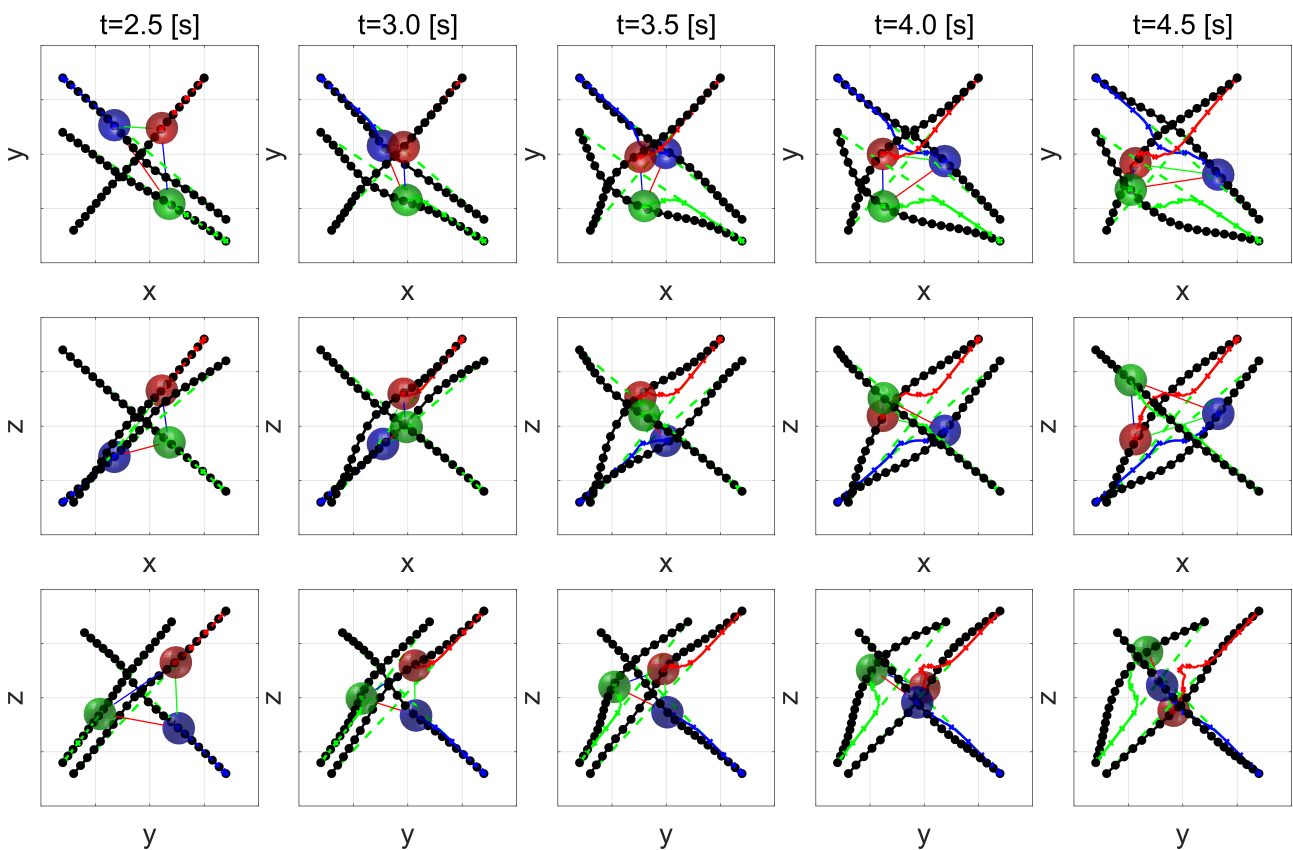


Abbildung 42: Orthogonale Ansichten der Ausweichbewegung

In einem weiteren Szenario, welches neben den UAVs selbst als dynamische Hindernisse zusätzliche statische Hindernisse enthält, wurden die in Abbildung 43 dargestellten Gegebenheiten in einem Maschinenbaulabor der Hochschule Esslingen nachgebildet.

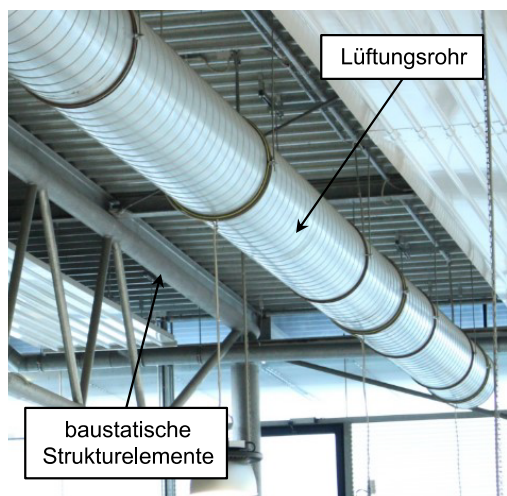


Abbildung 43: Luftraum innerhalb einer Maschinenhalle

Dabei wurde ein Ausschnitt des Luftraums als durch (statische) Ebenenhindernisse begrenzter Arbeitsraum der Größe $5 [m] \times 4 [m] \times 3 [m]$ für die UAVs definiert. Mithilfe weiterer (statischer) Zylinderhindernisse wurden die den Arbeitsraum durchziehenden baustatischen Strukturelemente sowie das Lüftungsrohr modelliert.

Es wurden erneut die sich ergebenden Sollwerttrajektorien von drei UAVs mit diesmal kollisionsbehafteten Initialbahnen ($\mathbf{p}_{s,1} = (3.0, 0.5, 2.6)^T [m]$, $\mathbf{p}_{g,1} = (2.0, 3.5, 0.5)^T [m]$, $\mathbf{p}_{s,2} = (1.2, 0.6, 2.5)^T [m]$, $\mathbf{p}_{g,2} = (4.5, 3.0, 0.4)^T [m]$, $\mathbf{p}_{s,3} = (1.0, 3.0, 1.0)^T [m]$, $\mathbf{p}_{g,3} = (4.0, 0.5, 1.5)^T [m]$) aufgezeichnet. Abgesehen von einer geringfügigen Erhöhung der Hinderniswirkweite ($r_0 = 0.8 [m]$ und $r_u = 0.9 [m]$) blieb die Parametrierung der KFM unverändert. Die Abstände zwischen den UAVs und dem jeweils nächstgelegenen Hindernis (andere UAVs oder statisches Hindernis) sind in Abbildung 44 dargestellt.

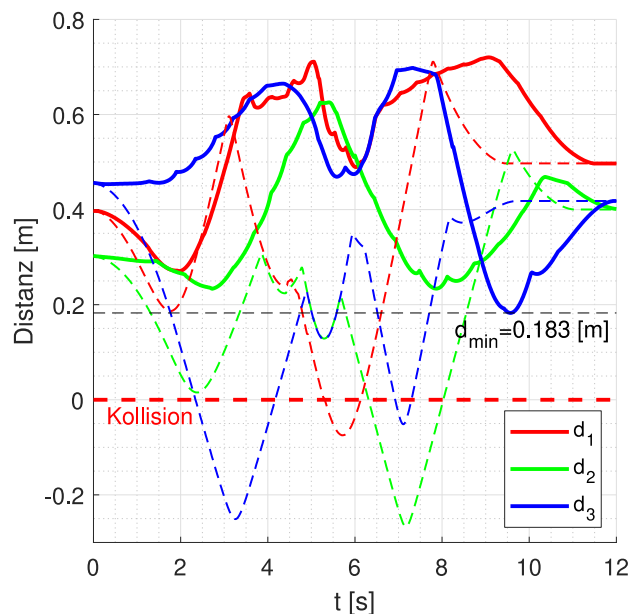


Abbildung 44: Abstände für drei UAVs mit statischen Hindernissen

Gegenüber der Interpolation auf den unbelasteten Initialbahnen (gestrichelt dargestellt) mit deutlich sichtbaren Kollisionen (Durchdringungen bis ca. $0.25 [m]$) wird auch hier ein zum ersten Szenario ähnlicher Mindestabstand zu den Hindernissen von $d_{min} = 0.183 [m]$ nicht unterschritten. Abbildung 45 zeigt die zugehörigen Sollwerttrajektorien sowie die sich bei Simulationensende $t = 12 [s]$ ergebenden Bahnen.

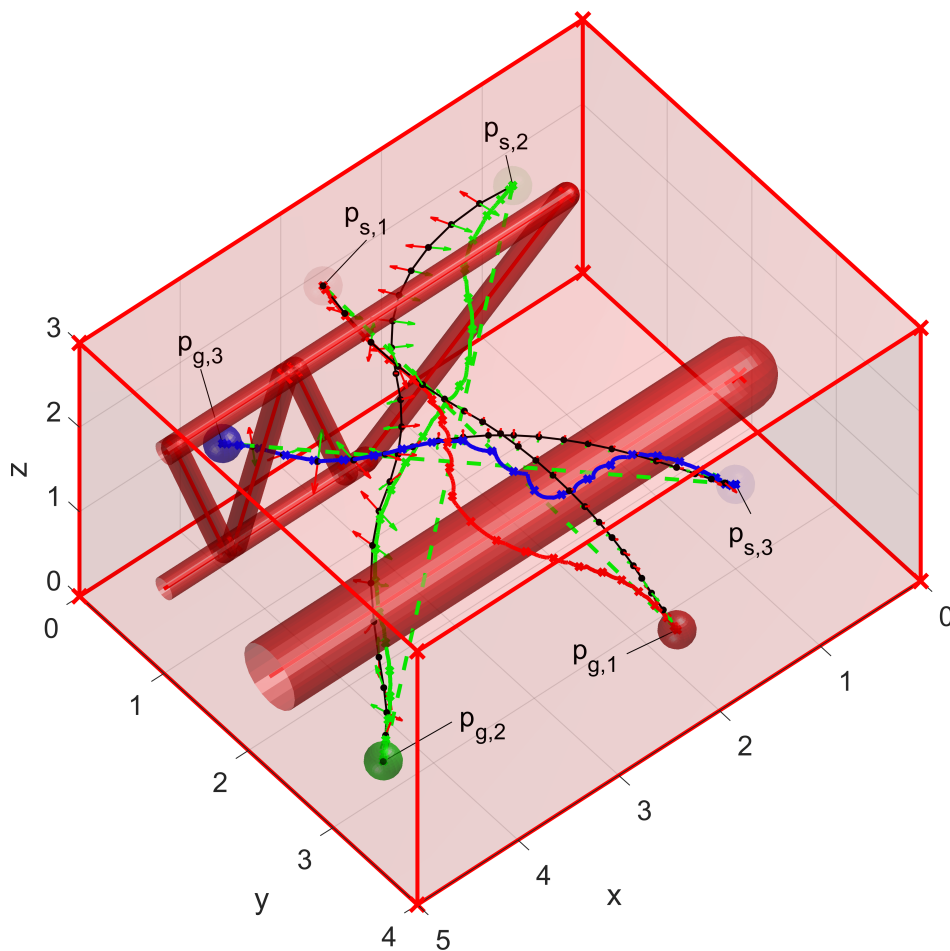


Abbildung 45: Sollwerttrajektorien für drei UAVs mit statischen Hindernissen

Diese Beispiele bestätigen die Eignung der KFM, als instanziiierbare und für eine beliebige Anzahl von UAVs parallel ausgeführte Bahnplanungsmethode eingesetzt zu werden.

5.8 Start- und Landevorgänge

Um eine Interaktion (z.B. Be- und Entladevorgänge beim Kleinteiletransport) mit den sich unterhalb des Arbeitsraums befindlichen Bearbeitungsmaschinen zu ermöglichen, werden (typischerweise vertikale) Start- und Landevorgänge zwischen definierten Übergabepositionen an den Maschinen außerhalb der Arbeitsraumbegrenzung sowie Vorpositionen innerhalb der Arbeitsraumbegrenzung benötigt. Diese sollen als Linearinterpolationen innerhalb eines geschützten Flugkorridors und frei von äußeren Hinderniseinflüssen ausgeführt werden. Dabei muss sichergestellt sein, dass die Sichtverbindung zwischen Vor- und Übergabeposition frei von statischen Hindernissen ist.

Für die Start- und Landevorgänge wird die jeweilige Instanz der KFM in der folgenden geringfügig modifizierten Form mit \mathbf{p}_s und \mathbf{p}_g als Übergabe- und Vorposition ausgeführt. Dabei gilt $K = 0$, sodass die unbelastete Bahn eine gerade Verbindung (Linearinterpolation) zwischen Übergabe- und Vorposition beschreibt. Zusätzlich wird das dem UAV zugeordnete Hindernis zu einem die gesamte Bahn umhüllenden Zylinder (ebenfalls mit dem Radius r_{uav}) erweitert. Dieser gesicherte Flugkorridor erlaubt anschließend einen von den übrigen UAVs bzw. Hindernissen unbeeinflussten Start bzw. Landung.

Abbildung 46 illustriert dies exemplarisch für einen Start von UAV₁ (rot) und eine gleichzeitige Landung von UAV₃ (blau). Deren als Zylinderhindernisse modellierte gesicherte Flugkorridore werden dabei von UAV₂ umflogen.

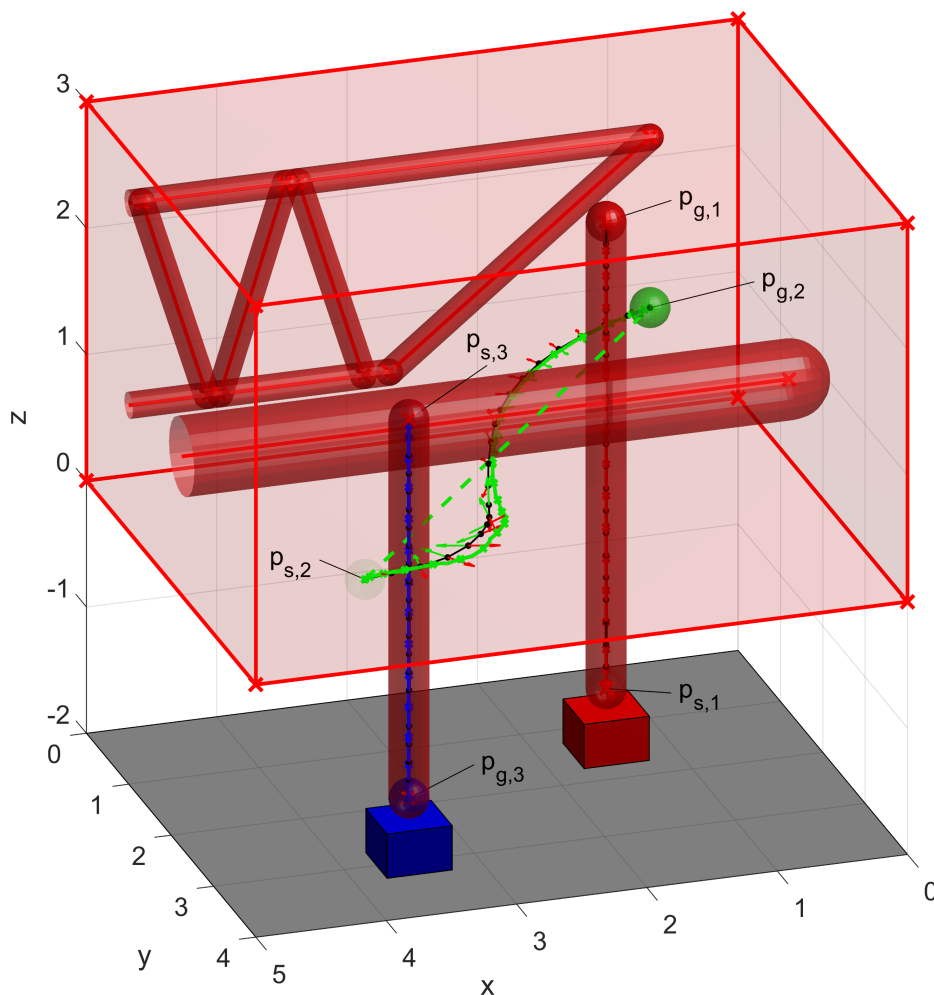


Abbildung 46: Vertikaler Start- und Landevorgang

6 Implementierung und Realisierung einer Testumgebung für die KFM

In diesem Kapitel wird die KFM in industrielle Steuerungstechnik integriert. Es wird zunächst eine abstrahierte Steuerungsarchitektur vorgestellt, welche den parallelen Betrieb einer beliebigen Anzahl von Instanzen der KFM zulässt. Anschließend werden die für die Realisierung dieser Architektur verwendeten Komponenten vorgestellt. Die KFM wird als eigene über Ablaufprogramme aufrufbare Interpolationsart unter harter Echtzeit in eine SPS implementiert und in drei parallelen Instanzen ausgeführt. Die Evaluierung der KFM erfolgt im Rahmen einer virtuellen Inbetriebnahme zunächst unter Berücksichtigung von Dynamikmodellen der UAVs und abschließend mit realen UAVs.

6.1 Architektur

Entsprechend der Problemstellung wird die KFM zur Gewährleistung der Skalierbarkeit als dezentrales (verteiltes) System auf mehreren vernetzten Industriesteuerungen implementiert, wobei jeder Steuerung genau ein UAV zugeordnet wird. Die Testumgebung wird hier für drei Instanzen der KFM aufgebaut. Die Komponenten des Gesamtsystems sowie die zwischen ihnen zu realisierenden Informationsflüsse sind in Abbildung 47 dargestellt.

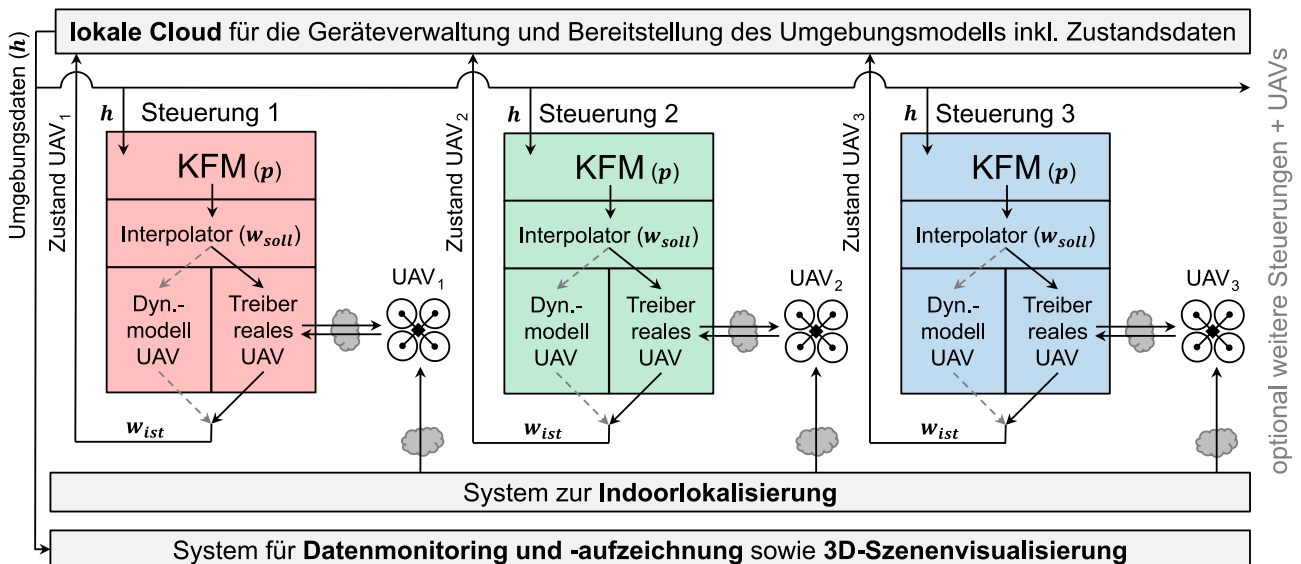


Abbildung 47: Architektur des Gesamtsystems zur Realisierung der KFM

Auf Basis der aktuellen Umgebungsdaten (\mathbf{h}) wird die KFM-Bahn (\mathbf{p}) aktualisiert und dem Interpolator zugeführt. Der resultierende Lagesollwert (\mathbf{w}_{soll}) wird anschließend wahlweise einem Dynamikmodell oder einem Treiber für ein reales UAV zugeführt. Das Dynamikmodell wird entkoppelt vom eigentlichen Steuerungscode implementiert. Beim Betrieb mit realem UAV leitet der entsprechende Treiber die Lagesollwerte über eine Drahtlosschnittstelle an das UAV weiter. Auf dessen Mikrocontroller findet die Lageregelung auf Basis einer absoluten Lagereferenz von einem separaten Indoorlokalisierungssystem (GPS ist zu ungenau und nur im Freien verfügbar) statt. Der aktuelle Lageistwert (\mathbf{w}_{ist}) wird wieder zurück an den Steuerungsrechner gesendet. Aus Sicht der Steuerung macht es keinen Unterschied, ob das Dynamikmodell oder das reale UAV betrieben wird, da beide Varianten dieselben Schnittstellen bedienen.

Alle Steuerungen sind mit einer lokalen Cloud vernetzt, welche kontinuierlich die Lageistwerte und weitere relevante Zustandsdaten aller aktiven UAVs ausliest und zusammen mit den statischen Hindernisinformationen zu einem globalen Umgebungsmodell (\mathbf{h}) bündelt. Dieses Modell wird schließlich wieder allen Instanzen der KFM als Eingang zugeführt, wodurch der Signalkreislauf geschlossen wird.

Zusätzlich wird das Umgebungsmodell für das Monitoring und die Datenaufzeichnung sowie für die 3D-Szenenvisualisierung abgegriffen. Um den Steuerungsablauf nicht zu beeinflussen, wird hierfür ein separates Rechensystem verwendet.

6.2 Verwendete Soft- und Hardware

Als industrielle Steuerungen wurden drei Exemplare des bereits in Kap. 4.12 vorgestellten Industrie-PCs C6930-0050 der Firma Beckhoff mit deren Automatisierungssoftware TwinCAT (Version 3.1.4020.14) verwendet. TwinCAT läuft dabei als Echtzeiterweiterung innerhalb von Windows 7 und ermöglicht dadurch eine einfach zu realisierende parallele Ausführung und Kopplung von nach IEC 61131-3 implementierten SPS-Programmen in harter Echtzeit und weiteren Nichtezeit-Softwarekomponenten innerhalb desselben Rechnersystems. Dazu gehören beispielsweise auf Hersteller-APIs basierende Schnittstellenbrücken (Gateways) zu Hardware, welche nicht direkt unter TwinCAT über industrielle

Feldbusse unterstützt wird. Typischerweise zählen hierzu auch die meisten UAVs. Für die interne Kommunikation nutzt TwinCAT das TCP/IP-basierte Automation Device Specification Protokoll (ADS) mit zugehörigem Nachrichtenrouter, welches über eine dynamische Laufzeitbibliothek (DLL) auch externen Nichtechtzeitprozessen für den Zugriff auf die SPS-Variablen zur Verfügung steht.

Die Implementierung des Dynamikmodells erfolgt in der Echtzeitsimulationssoftware Virtuos (Version 2.4.23) der Industriellen Steuerungstechnik GmbH Stuttgart (ISG). Virtuos wird hauptsächlich für die Hardware-In-The-Loop-Simulation (HiL) im Rahmen der virtuellen Inbetriebnahme verwendet. Dabei wird das Verhalten einer realen Maschine bis zu einem gewissen Abstraktionsgrad in Form eines mathematischen Modells nachgebildet und synchron zur Realzeit berechnet. Da das Modell idealerweise über identische Ein- und Ausgangsschnittstellen zur Steuerung verfügt, kann es gänzlich ohne eine Anpassung an der Steuerung anstelle der realen Maschine für realistische und gleichzeitig risikofreie Vorabtests bei der (virtuellen) Inbetriebnahme genutzt werden. Die Modellierung erfolgt dabei in der Entwicklungsumgebung Virtuos M (Modeller) unter Nichtechtzeit und in Form von blockorientierten Signalflussdiagrammen, während der Echtzeitrechenkern Virtuos S (Solver) als eigene Task in TwinCAT eingebunden ist.

Als reale UAVs werden drei Exemplare des Crazyflie (Version 2.0), einem etwa handflächengroßen Quadrocopter, der Firma Bitcraze verwendet. Der Crazyflie ist vollständig quelloffen und adressiert primär Anwendungen aus Forschung, Entwicklung und Lehre. Mit einem maximalen Abfluggewicht von ca. 42 Gramm und einer Flugdauer von bis zu 7 Minuten ist dieser noch nicht für den industriellen Kleinteiletransport geeignet und soll im Rahmen dieser Arbeit lediglich prototypisch die Anwendbarkeit der KFM auf reale UAVs belegen.

Für die Kommunikation zwischen Steuerungsrechner und dem Crazyflie 2.0 wird über den Crazyradio PA USB-Dongle von Bitcraze basierend auf einem nRF24L01-Modul eine Funkstrecke im 2.4 GHz ISM-Band mit einer Reichweite von bis zu einem Kilometer realisiert. Darüber können Soll- und Istwerte in Datenpaketen zu jeweils 32 Byte und einer Datenrate von bis zu 2 Mbit/s ausgetauscht werden. Auf dem Crazyflie 2.0 ist die

Kommunikationsschnittstelle bereits standardmäßig in die Firmware integriert. Steuerungsseitig wird das Crazyradio PA über eine in Python verfasste und damit innerhalb der Python Laufzeitumgebung und nicht unter Echtzeit ausgeführte API angesprochen. Die Schnittstelle zu TwinCAT wird hier über PyADS, eine Wrapperbibliothek, welche die ADS-DLL kapselt, realisiert. Abbildung 48 zeigt den Crazyflie und den Crazyradio PA USB Dongle.

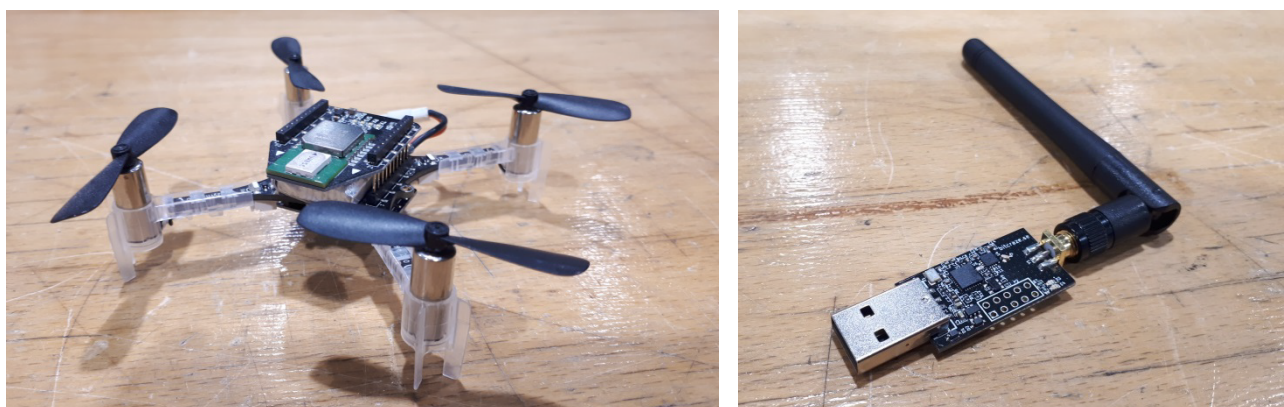


Abbildung 48: Crazyflie mit LPS-Deck und Crazyradio PA USB Dongle

Für Flugversuche mit realen UAVs steht in den Maschinenbaulaboren der Hochschule Esslingen eine Flugzelle zur Verfügung (vgl. Abbildung 49). Dieser mit Netzen abgesicherte Arbeitsraum besitzt eine Abmessung von ca. $4 [m] \times 5 [m] \times 4.5 [m]$ und entspricht damit der Größenordnung der in Kap. 5 dargestellten Szenarien.

Für die Indoorlokalisierung der UAVs wird das Loco Positioning System (LPS) der Firma Bitcraze verwendet. Es besteht aus einem Erweiterungsdeck für den Crazyflie sowie aus mindestens vier Ankern im Arbeitsraum, deren Positionen bekannt sein müssen. Mithilfe von Funksignalen im Ultrabreitband (UWB) werden kontinuierlich Laufzeitmessungen zwischen Ankern und Deck durchgeführt und daraus die absolute Position des Crazyflie bestimmt. Das LPS stellt dafür die Betriebsmodi Two Way Ranging (TWR), bei welchem die Anker zur Bestimmung der absoluten Laufzeit vom Deck aktiv angetriggert werden, und Time Difference of Arrival (TDoA), bei welchem das passive Deck zur Bestimmung der relativen Laufzeit ein kontinuierliches Signal von den Ankern empfängt, zur Verfügung. Die Funktionsweise von TDoA entspricht damit einem lokalen GPS im kleinen Maßstab und lediglich dieses Verfahren lässt sich mit mehreren simultanen Decks betreiben. Um eine

möglichst hohe Genauigkeit zu erreichen, wird das LPS mit insgesamt 8 jeweils in den Eckbereichen der Flugzelle angebrachten Ankern im TDoA-Modus verwendet. In Abbildung 49 ist die Flugzelle sowie ein einzelner LPS-Anker dargestellt.



Abbildung 49: Flugzelle und ein einzelner LPS-Anker

Um die zwangsläufigen Messschwankungen des LPS zu kompensieren, wird der Crazyflie zur Stabilisierung zusätzlich mit dem Flowdeck der Firma Bitcraze betrieben. Mithilfe einer Kombination aus optischem Flusssensor und einem Sensor zur elektrooptischen Entfernungsmessung ermöglicht dieses Erweiterungsdeck die Detektion der relativen Lage mit einer höheren Präzision als das LPS. Alle positionsrelevanten Messwerte (absolute Lage aus LPS und relative Lage aus Inertialsensoren und Flowdeck) werden in der Firmware des Crazyflie mit einem Kalman-Filter für dessen Positionsschätzung fusioniert.

Als lokale Cloud für die Bereitstellung des Umgebungsmodells wird die am Virtual Automation Lab (VAL) der Hochschule Esslingen entwickelte und auf einer separaten Serverhardware (Intel Nuc Mini) laufende VAL Edge Cloud verwendet. Diese stellt einen unter Node.js (einer auf Javascript basierenden Laufzeitumgebung zur Entwicklung von Netzwerk- und Serveranwendungen) entwickelten OPC-UA Client dar, mit welchem alle OPC-UA-fähigen Geräte innerhalb eines Netzwerks verbunden und deren Maschinendaten weiterhin auf dem Server implementierten Mehrwertdiensten zur Verfügung gestellt werden

können. OPC-UA (Open Platform Communications Unified Automation) ist ein plattform-unabhängiges Netzwerkprotokoll für die Maschine-zu-Maschine-Kommunikation. TwinCAT wird dazu als OPC-UA-Server konfiguriert und stellt die gewünschten Daten mit einer Aktualisierungsrate von 50 Millisekunden bereit.

Innerhalb der VAL Edge Cloud wurde mit dem UAV Control Network (UCN) ein eigener Mehrwertdienst implementiert, welcher das Netzwerk nach verfügbaren UAVs durchsucht und deren Zustandsdaten kontinuierlich mit den aus einer Konfigurationsdatei ausgelesenen statischen Hindernisdaten zu einem globalen Umgebungsmodell bündelt und dieses wiederum auf die verfügbaren Steuerungen zurückschreibt.

Die Übermittlung der Daten zwischen Server und Steuerungsrechner mittels TCP/IP erlaubt keine harte Echtzeitkommunikation. Aufgrund direkter Verbindungen ohne Switches und ohne Datenpaketkollisionen kann diese dennoch als echtzeitnah (weiche Echtzeit) betrachtet werden. Erweiterungen durch das Time-Sensitive Networking (TSN) eröffnen zukünftig auch hier die Möglichkeit einer Verbindung in harter Echtzeit.

Die 3D-Szenenvisualisierung findet im ebenfalls am VAL entwickelten VAL 3D-Webstudio statt (Schnierle et al. 2018; Schnierle et al. 2019). Dieses stellt einen weiteren innerhalb der VAL Edge-Cloud laufenden Mehrwertdienst dar und ermöglicht die webbrowserbasierte Visualisierung von kinematisierten und realdatengetriebenen CAD-Modellen. Für den Zugriff auf das VAL 3D-Webstudio wird ein zusätzlicher Standard-PC als Client verwendet.

Derselbe Rechner wird ebenfalls für das Monitoring und die Aufzeichnung der Daten in einem TwinCAT Measurement Projekt verwendet. Dieses Projekt läuft direkt in der ins Microsoft Visual Studio integrierten Entwicklungsumgebung von TwinCAT und ermöglicht per ADS-Remotezugriff das Lesen, die Visualisierung sowie die Aufzeichnung (jeweils lückenlos) beliebiger Steuerungsdaten im selben Netzwerk.

Abbildung 50 fasst die in diesem Kapitel vorgestellten Soft- und Hardwarekomponenten sowie deren Kommunikationsschnittstellen und die jeweils verwendeten bzw. erreichbaren Zykluszeiten und Datenraten nochmals zusammen. Abbildung 51 zeigt ein Foto des Gesamtaufbaus. Zudem wurde eine weitere Beschreibung des Systems veröffentlicht (Bergen et al. 2018).

6 Implementierung und Realisierung einer Testumgebung für die KFM

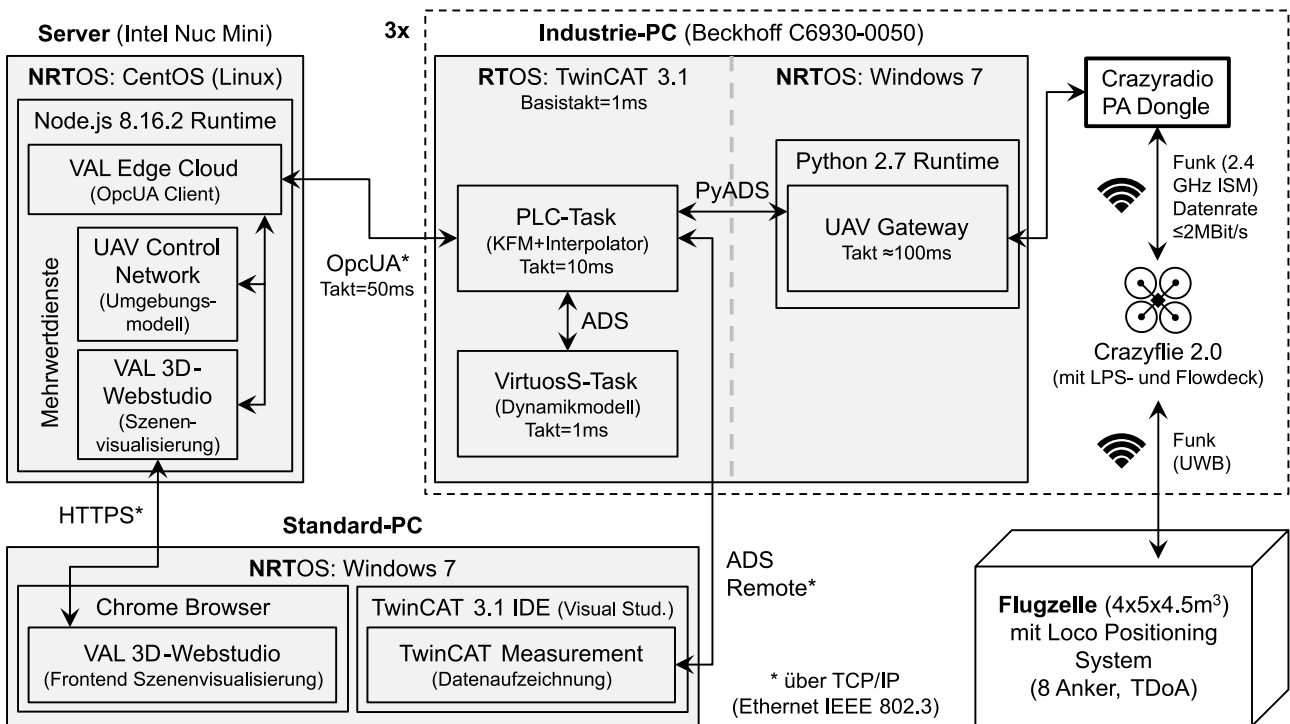


Abbildung 50: Überblick über die verwendeten Soft- und Hardwarekomponenten



Abbildung 51: Gesamtaufbau

6.3 Implementierung der KFM in einer SPS

Aus Vorarbeiten am VAL stand bereits eine vollständig eigenentwickelte Steuerung auf Basis der TwinCAT SPS für einen KUKA youBot 5-Achsknickarmroboter zur Verfügung. Die bereits vorhandenen Steuerungsfunktionalitäten umfassten dabei unter anderem verschiedene Betriebsmodi (manuelles Verfahren oder Automatikbetrieb auf Basis eines Steuerungsprogramms), verschiedene kinematische Transformationen (Verfahren in einzelnen Achswinkeln oder in kartesischen Endeffektorkoordinaten), verschiedene Interpolationsarten (synchrone und asynchrone Point-to-Point-Interpolation) mit verschiedenen Sollwertprofilen (ruckbegrenztetes Spline- und 7-Phasen-Profil), verschiedene Sicherheitsfunktionen (Arbeitsraum- und Bahnfehlerüberwachung) sowie eine HMI für die Benutzerinteraktion. Der in Structured Text (ST) verfasste Steuerungscode wurde im Rahmen dieser Arbeit für den Betrieb eines UAVs mit der KFM adaptiert. Dazu wurden unter anderem die folgenden Anpassungen und Erweiterungen durchgeführt:

- Anpassung der Achskonfiguration von den 5 rotatorischen Freiheitsgraden des Roboters zu 3 translatorischen (Positionen x , y und z) und einem rotatorischen (Orientierung ψ) Freiheitsgrad des UAVs
- Implementierung einer Bibliothek für Vektor- und Matrixarithmetik
- Erweiterung der Interpolationsarten um die KFM als eigene (kollisionsfreie) Point-to-Point Interpolationsart mit Sollwertprofilgenerator in Form des Funktionsbausteins „FB_KFM“
- Erweiterung der Programmbefehle für den Automatikbetrieb um die Befehle „PTP_KFM“ (kollisionsfreie Interpolation mit der KFM) sowie „PTP_KFM_START“ und „PTP_KFM_LAND“ (gesicherter Start- und Landevorgang), welche jeweils den Funktionsbaustein „FB_KFM“ in unterschiedlichen Modi ausführen
- Implementierung der Schnittstelle zur VAL Edge Cloud für die Bereitstellung der eigenen Zustandsdaten sowie zum Empfang des Umgebungsmodells

Um die KFM möglichst intuitiv und flexibel in die bestehende Steuerungsarchitektur integrieren zu können, wurde sie ebenso wie die bereits vorhandenen Interpolationsarten in Anlehnung an die PLC-Open Spezifikation (Erweiterung der IEC 61131 zur standardisier-

ten Realisierung von Motion-Applikationen) in Form des instanzitierbaren Funktionsbausteins „FB_KFM“ mit den in Abbildung 52 dargestellten Ein- und Ausgangsvariablen implementiert.

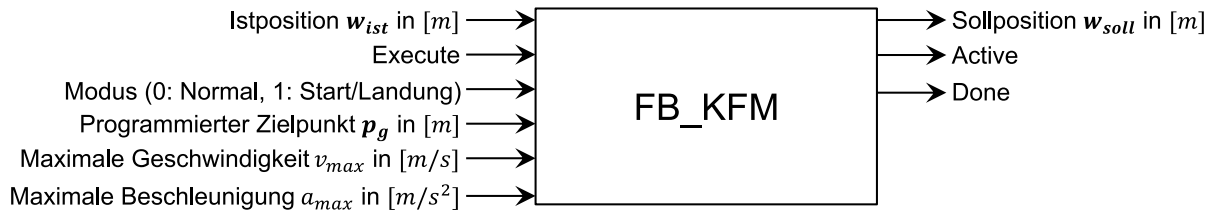


Abbildung 52: Ein- und Ausgangsvariablen des Funktionsbausteins „FB_KFM“

Auf der obersten Ebene kann der Funktionsbaustein „FB_KFM“ durch den idealisierten Pseudocode in Code 1 veranschaulicht werden.

001	IF (Execute=TRUE AND Active=FALSE) THEN
002	$p_s := w_{ist};$
003	$w_{soll} := w_{ist};$
004	$p := \text{KFM_Initialize}(p_s, p_g);$
005	Done := FALSE;
006	Active := TRUE;
007	ELSIF (ABS($w_{soll} - p_g$) > eps) THEN
008	IF (Modus=0) THEN
009	$f := \text{KFM_Force}(p);$
010	$p := \text{KFM_Integrate}(p, p_s, p_g, f);$
011	END IF
012	$w_{soll} := \text{KFM_Interpolate}(p, p_s, p_g, w_{soll});$
013	ELSE
014	Done := TRUE;
015	Active := FALSE;
016	END IF

Code 1: Oberste Ebene des Funktionsbausteins „FB_KFM“

Die KFM wird auf dieser Ebene in die vier Hauptfunktionen „KFM_Initialize“, „KFM_Force“, „KFM_Integrate“ und „KFM_Interpolate“ aufgeteilt.

„KFM_Initialize“ wird nur einmalig im ersten Takt des Bewegungssatzes (steigende Flanke des Eingangs „Execute“) durchlaufen und initialisiert die Bahn p entsprechend der als globale Variable vorgegebenen Diskretisierung n als gerade Verbindung zwischen dem programmierten Zielpunkt p_g und der aktuellen Istposition des UAVs w_{ist} als Startpunkt p_s .

Außerdem werden hier die Matrizen L und U der LU-Zerlegung berechnet und global gespeichert. Gleichzeitig werden die restlichen Eingänge „Modus“, v_{max} und a_{max} eingelesen und sind bis zum Ende des Satzes gültig.

„KFM_Force“ wertet in einer Reihe verschachtelter Unterfunktionen (Iteration über Stützstellen, Feldnormierung, Distanzrechnungen) die abstoßende Kraft f aus. Der Zugriff auf das von der VAL Edge Cloud bereitgestellte Umgebungsmodell h wird über einen globalen Speicher realisiert. Gleiches gilt für die weiteren zur Berechnung von f benötigten Parameter r_0 , r_u und p .

„KFM_Integrate“ aktualisiert die Bahn p durch einen Integrationsschritt des impliziten Eulerverfahrens mit den Eingangsvektoren p_s , p_g und f . Auf Basis des Thomas-Algorithmus muss dafür lediglich die Rücksubstitution mit den bei der Initialisierung bestimmten globalen Matrizen L und U durchgeführt werden. Die Parameter K und T sowie die mit der Zykluszeit der Steuerung identische Integrationsschrittweite Δt stehen als globale Variablen zur Verfügung.

Sowohl „KFM_Force“ als auch „KFM_Integrate“ werden für Start oder Landung (Eingang „Modus“ hat den Wert 1) nicht durchlaufen, da hier ausschließlich auf der geraden und unbelasteten Initialbahn interpoliert wird.

„KFM_Interpolate“ aktualisiert den Lagesollwert w_{soll} auf der durch p , p_s und p_g vorgegebenen Bahn unter Berücksichtigung der Dynamikgrenzen v_{max} und a_{max} . Der initiale Lagesollwert wird im ersten Takt des Bewegungssatzes einmalig mit dem Lageistwert w_{ist} abgeglichen.

Im Rahmen dieser Arbeit wird der Funktionsbaustein „FB_KFM“ durch ein individuelles und auf jeder Steuerung manuell zu startendes Ablaufprogramm, welches die nacheinander anzufliegenden Zielpunkte p_g enthält, aufgerufen. Später ist überdies auch die direkte Beauftragung durch eine übergeordnete Leitsteuerung, welche die Aufträge innerhalb des Produktionssystems verwaltet, denkbar.

In Code 2 ist ein exemplarisches Ablaufprogramm zum sukzessiven Abfliegen mehrerer Punkte mit der KFM dargestellt. Die Achsen A1 bis A4 entsprechen dabei der Position in

x -, y - und z -Richtung sowie der Orientierung ψ des UAV. Dabei hat ψ keinen Einfluss auf die generierte Bahn.

001	// Einstiegspunkt
002	PRG_START();
003	// Start
004	PTP_KFM_START(A1:=1.000, A2:=1.000, A3:=2.000, A4:=0.000);
005	// Nacheinander mehrere Punkte anfliegen
006	PTP_KFM(A1:=3.000, A2:=1.000, A3:=2.000, A4:=0.000);
007	PTP_KFM(A1:=3.000, A2:=3.000, A3:=2.000, A4:=0.000);
008	PTP_KFM(A1:=1.000, A2:=3.000, A3:=2.000, A4:=0.000);
009	// Punkt oberhalb Landeposition anfliegen (Vorposition)
010	PTP_KFM(A1:=1.000, A2:=1.000, A3:=2.000, A4:=0.000);
011	// Wartezeit 2 Sekunden für Exakthalt
012	SLEEP(T:=2.000);
013	// Landung
014	PTP_KFM_LAND(A1:=1.000, A2:=1.000, A3:=0.000, A4:=0.000);

Code 2: Beispielhaftes Ablaufprogramm zum Abfliegen mehrerer Punkte mit der KFM

6.4 Dynamikmodell der UAVs

Für die virtuelle Inbetriebnahme der KFM mithilfe eines Dynamikmodells der UAVs, welches aus einem Modell der Regelstrecke und dem Regler besteht, wurden zunächst die nichtlinearen Bewegungsgleichungen eines verallgemeinerten und als Starrkörper modellierten UAVs angewandt (Kurak et al. 2018). Dieses Streckenmodell wird mit dem Zustandsvektor $\mathbf{X} = (X_1, \dots, X_{12})^T = (\varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}, z, \dot{z}, x, \dot{x}, y, \dot{y})^T$ sowie dem Eingangsvektor $\mathbf{U} = (U_1, \dots, U_4)^T$ gemäß Gl. (47) im Zustandsraum beschrieben. Dabei sind m die Masse des UAVs, I_{xx} , I_{yy} und I_{zz} dessen körpereigene Hauptträgheitsmomente und l der Abstand zwischen Schwerpunkt und Rotorachsen sowie g die Gravitationskonstante. Verzögernde bzw. dämpfende Einflüsse wie beispielsweise der Luftwiderstand oder das Motorverhalten wurden vernachlässigt, sodass jeder Freiheitsgrad näherungsweise durch eine zweifache Integration beschrieben wird.

$$\begin{pmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \\ \dot{X}_4 \\ \dot{X}_5 \\ \dot{X}_6 \\ \dot{X}_7 \\ \dot{X}_8 \\ \dot{X}_9 \\ \dot{X}_{10} \\ \dot{X}_{11} \\ \dot{X}_{12} \end{pmatrix} = \begin{pmatrix} X_2 \\ X_4 \cdot X_6 \cdot (I_{yy} - I_{zz})/I_{xx} + l/I_{xx} \cdot U_2 \\ X_4 \\ X_2 \cdot X_6 \cdot (I_{zz} - I_{xx})/I_{yy} + l/I_{yy} \cdot U_3 \\ X_6 \\ X_2 \cdot X_4 \cdot (I_{xx} - I_{yy})/I_{zz} + l/I_{zz} \cdot U_4 \\ X_8 \\ -g + (\cos(X_1) \cdot \cos(X_3))/m \cdot U_1 \\ X_{10} \\ (\cos(X_1) \cdot \sin(X_3) \cdot \cos(X_5) + \sin(X_1) \cdot \sin(X_5))/m \cdot U_1 \\ X_{12} \\ (\cos(X_1) \cdot \sin(X_3) \cdot \sin(X_5) - \sin(X_1) \cdot \cos(X_5))/m \cdot U_1 \end{pmatrix} \quad (47)$$

Das Modell beschreibt die Zustände X und Eingänge U in drei unterschiedlichen Koordinatensystemen (KS):

- Im raumfesten KS_1 werden die translatorischen Zustände (X_7 bis X_{12}) beschrieben.
- KS_2 besitzt raumfeste Achsen (identisch mit KS_1) und einen körperfesten Ursprung (identisch mit KS_3). In KS_2 werden die rotatorischen Zustände (X_1 bis X_6) beschrieben.
- Im körperfesten KS_3 werden die Eingänge (U_1 bis U_4) beschrieben.

Abbildung 53 illustriert X und U in den verschiedenen Koordinatensystemen.

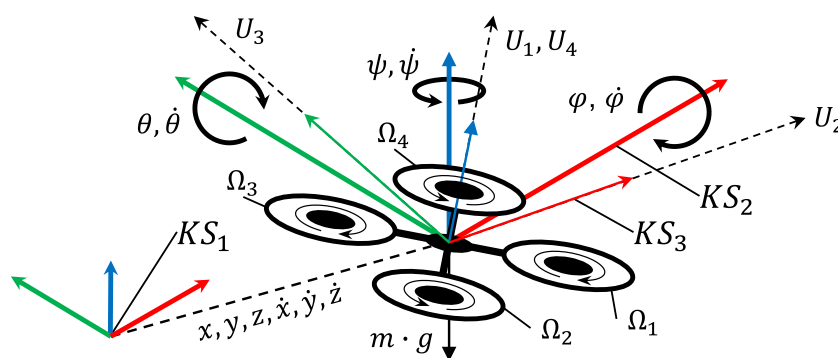


Abbildung 53: Koordinatensysteme zur Modellierung eines UAVs in „H“-Konfiguration

Die hier verwendeten abstrakten Eingangsgrößen stellen somit die auf das UAV aufgeprägten Momente um die körperfesten Rotationsachsen (U_2 , U_3 und U_4 in $[N \cdot m]$) sowie die Schubkraft in Richtung der körperfesten Hochachse (U_1 in $[N]$) dar. Entgegen der di-

rekten Vorgabe der Rotordrehzahlen $\Omega = (\Omega_1, \Omega_2, \Omega_3, \Omega_4)^T$ in $[rad/s]$, welche die realen Stellgrößen des UAVs darstellen und zur Ausführung einzelner Flugbewegungen stets gekoppelt variiert werden müssen, kann die Beeinflussung und Regelung von Höhe, Position und Orientierung über U weitestgehend entkoppelt erfolgen.

Für ein UAV wie den Crazyflie in „H-“ bzw. „X“-Konfiguration (Rotoren um 45° zur Flugrichtung versetzt) kann der Zusammenhang zwischen U und den Rotordrehzahlen Ω über die Motormatrix und die Rotorkonstanten K_f und K_m gemäß Gl. (48) hergestellt werden. K_f und K_m fassen dabei die u. a. von Rotorfläche und Luftdichte beeinflusste Abhängigkeit zwischen Drehzahl und resultierender Auftriebskraft (K_f) und resultierendem Moment (K_m) an einem einzelnen Rotor zusammen.

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} K_f & K_f & K_f & K_f \\ -l/\sqrt{2} \cdot K_f & -l/\sqrt{2} \cdot K_f & l/\sqrt{2} \cdot K_f & l/\sqrt{2} \cdot K_f \\ -l/\sqrt{2} \cdot K_f & l/\sqrt{2} \cdot K_f & l/\sqrt{2} \cdot K_f & -l/\sqrt{2} \cdot K_f \\ -K_m & K_m & -K_m & K_m \end{pmatrix} \cdot \begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} \quad (48)$$

Für den Eingang $U_h = (m \cdot g, 0, 0, 0)^T$ bzw. die Rotordrehzahlen $\Omega_h = \sqrt{m \cdot g / (4 \cdot K_f)} \cdot (1, 1, 1, 1)^T$ befindet sich das UAV in einem stabilen Schwebезustand. Ausgehend davon illustriert Abbildung 54 für exemplarische Flugmanöver die erforderlichen Eingangsgrößen sowie die für deren Ausführung erforderliche (gekoppelte) Variation der Rotordrehzahlen.

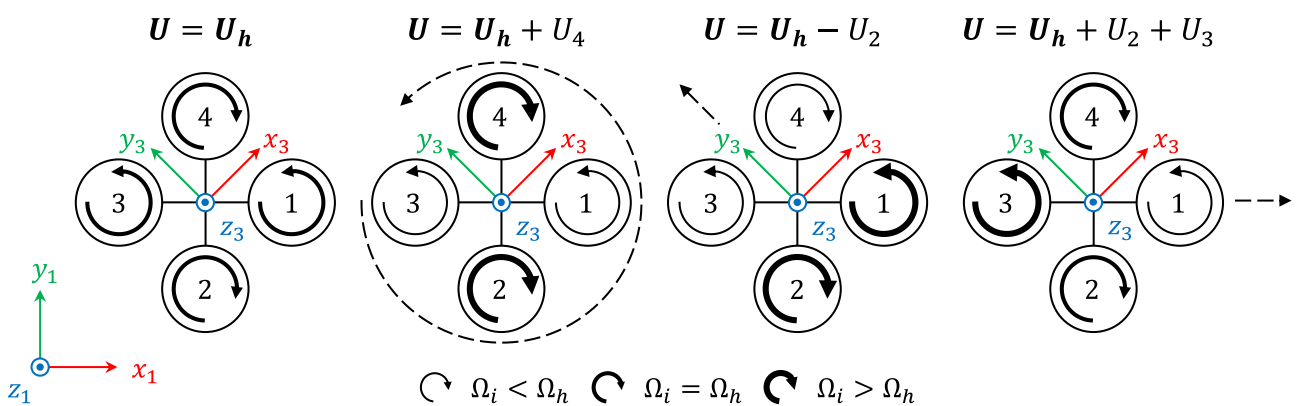


Abbildung 54: Modelleingänge und Drehzahlen für exemplarische Flugmanöver

Da das Dynamikmodell ausschließlich dazu dienen soll, bei der Inbetriebnahme ein aus Sicht der Steuerung „plausibles“ Flugverhalten abzubilden, wurde darauf verzichtet die

komplexe Regelungsstruktur des realen Crazyflie mit kaskadierten PID-Reglern möglichst exakt nachzubilden. Stattdessen wurde mithilfe von MATLAB ein automatisierter Entwurf eines Zustandsreglers durchgeführt.

Da der Entwurf eines Zustandsregler auf einem linearen Modell der Regelstrecke basiert, wurden zunächst die nichtlinearen Bewegungsgleichungen (47) um den sich für den Eingang U_h ergebenden stabilen Schwebestand linearisiert ($U = U_h + \Delta U$). Mit den Regelgrößen $Y = (x, y, z, \psi)^T$ als Systemausgang ergibt sich das steuer- und beobachtbare lineare Zustandsraummodell (49) und (50).

$$\dot{X} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot X + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & l/I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & l/I_{yy} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l/I_{zz} \\ 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \Delta U \quad (49)$$

$$Y = \begin{pmatrix} x \\ y \\ z \\ \psi \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot X + \vec{0} \cdot \Delta U \quad (50)$$

Bei der Zustandsregelung ergibt sich die Stellgröße ΔU durch subtraktive Aufschaltung der über eine Rückführmatrix K gewichteten Zustände X auf die Führungsgröße w . Da gegenüber einem Standardregelkreis somit keine Rückkopplung der Regelgrößen Y erfolgt, muss der dadurch bedingte stationäre Fehler durch einen Vorfilter V , welcher aus K und dem Streckenmodell zu ermitteln ist, eliminiert werden. Für die Berechnung der Stellgröße gilt somit $\Delta U = V \cdot w - K \cdot X$.

Die Berechnung der Rückführmatrix K erfolgte auf Basis der LQ-Optimierung (Lunze 2014). Gegenüber der Polzuweisung, bei welcher ausschließlich die Gesamtdynamik des geschlossenen Regelkreises gezielt vorgegeben wird, erlaubt dieses Entwurfsverfahren

eine Begrenzung des Stellaufwandes durch relative Gewichtung der Zustands- und Stell- bzw. Eingangsgrößen.

Die für den Reglerentwurf benötigten Streckenparameter des Crazyflie wurden bereits bestehenden Arbeiten (Hustiu et al. 2018; Luis 2016; Förster 2015) entnommen und sind in Tabelle 11 nochmals wiedergegeben.

Masse m	0.0299 [kg]
Gravitationskonstante g	9.81 [m/s^2]
Hauptträgheitsmoment I_{xx}	$1.395 \cdot 10^{-5}$ [$kg \cdot m^2$]
Hauptträgheitsmoment I_{yy}	$1.436 \cdot 10^{-5}$ [$kg \cdot m^2$]
Hauptträgheitsmoment I_{zz}	$2.173 \cdot 10^{-5}$ [$kg \cdot m^2$]
aerodynamische Konstante (Auftriebskraft) K_f	$2.22 \cdot 10^{-8}$ [$N/(rad/s)^2$]
aerodynamische Konstante (Drehmoment) K_m	$1 \cdot 10^{-9}$ [$N \cdot m/(rad/s)^2$]
Abstand zwischen Schwerpunkt und Rotoren l	0.046 [m]

Tabelle 11: Für den Reglerentwurf verwendete Parameter für den Crazyflie

Die sich nach Gl. (48) aus K_f ergebende Schwebedrehzahl von $\Omega_h = 1909$ [rad/s] zeigt, dass der Crazyflie bei einer maximalen Drehzahl von $\Omega_{max} \approx 2513$ [rad/s] (Ottoßon 2017) lediglich eine geringe Leistungsreserve der Stellgrößen besitzt.

Daher wurde das Optimierungskriterium so eingestellt, dass die Regeldynamik nicht dominierend aus den Stellgrößen, sondern aus den Zuständen resultiert, wodurch außerdem extreme Fluglagen außerhalb der Gültigkeit der Linearisierung (Kleinwinkelnäherung) vermieden werden. Zusätzlich wurde bei der horizontalen Positionierung (x und y) gegenüber der Höhen- und Orientierungsregelung (z und ψ) eine etwas höhere Dynamik und ein Überschwingen von ca. 5 Prozent zugelassen.

Durch experimentelle Feinjustierung anhand simulierter Sprungantworten konnte die dafür geeignete Gewichtungsvorgabe von $\mathbf{Q} = \mathbf{I} \cdot (50, 5, 50, 5, 20, 5, 50, 5, 100, 1, 100, 1)$ (Zustände) und $\mathbf{R} = \mathbf{I} \cdot (10, 60000, 60000, 60000)$ (Eingangs- bzw. Stellgrößen) identifiziert werden. Die daraus resultierende Rückführmatrix \mathbf{K} und Vorfiltermatrix \mathbf{V} sind im Anhang in Kap. 9.5 aufgeführt. Abbildung 55 zeigt für diese Reglereinstellung die resultierenden Ant-

worten der Regelgrößen Y auf einen Einheitssprung der Führungsgrößen w für das linearisierte als auch für das davon nur minimal abweichende nichtlineare Modell.

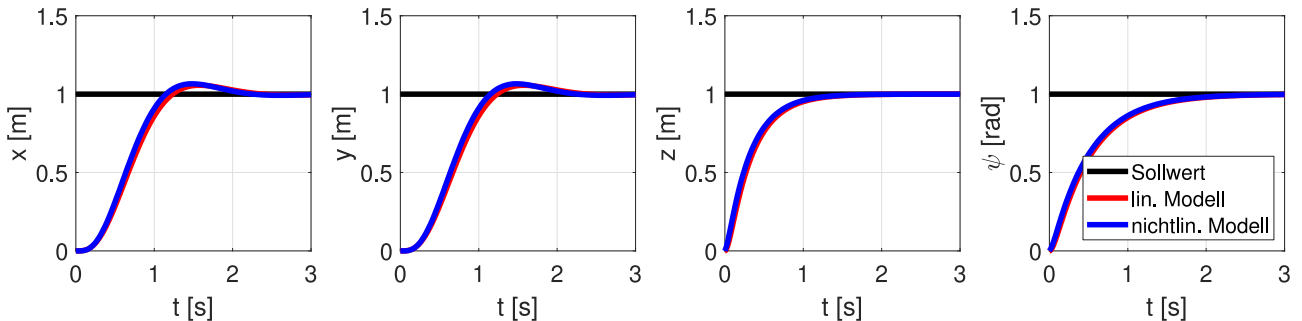


Abbildung 55: Sprungantworten der Regelgrößen des geregelten Dynamikmodells

Abbildung 56 zeigt zusätzlich den damit verbundenen Stellaufwand und verdeutlicht nochmals die entkoppelte Wirkung von U auf die Regelgrößen Y sowie den gekoppelten und nichtlinearen Zusammenhang zu den Rotordrehzahlen Ω .

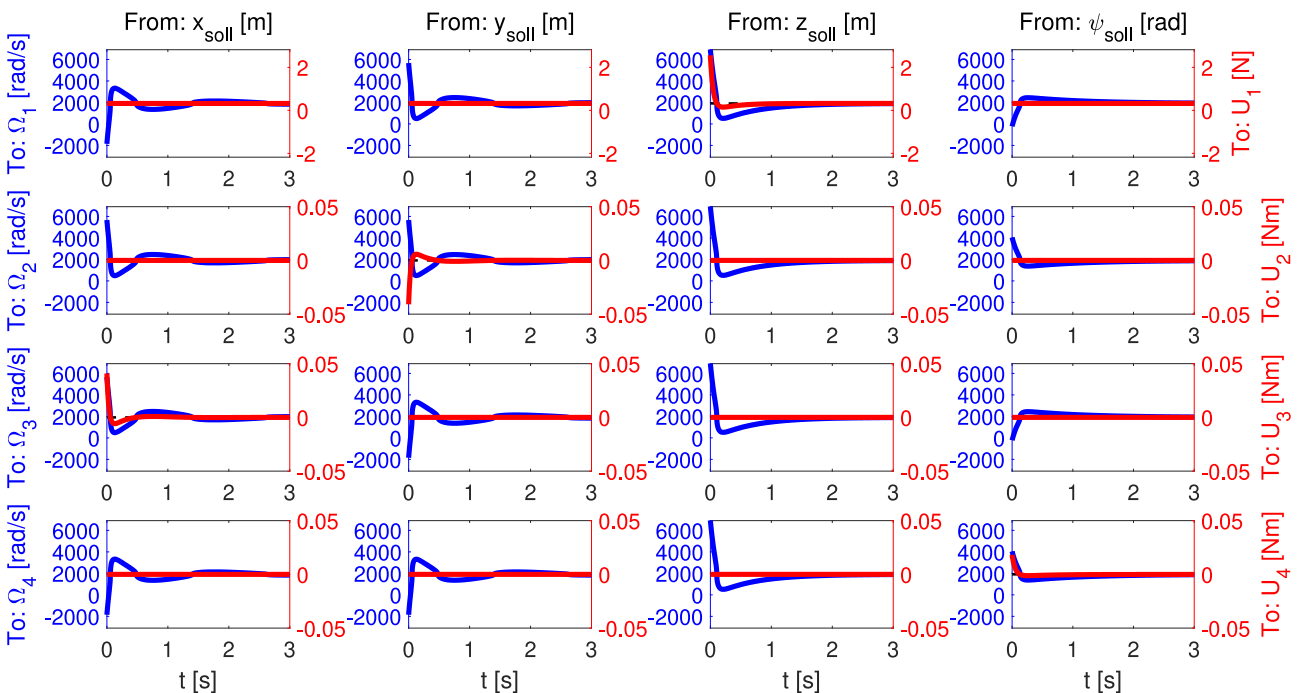


Abbildung 56: Sprungantworten der Stellgrößen des geregelten Dynamikmodells

Das hier beschriebene Modell eines positionsgeregelten UAVs wurde in der Echtzeitsimulationssoftware ISG Virtuos in Form eines Signalflussplans implementiert. Die oberste Ebene des Modells mit sichtbarer Reglerstruktur ist in Abbildung 57 dargestellt.

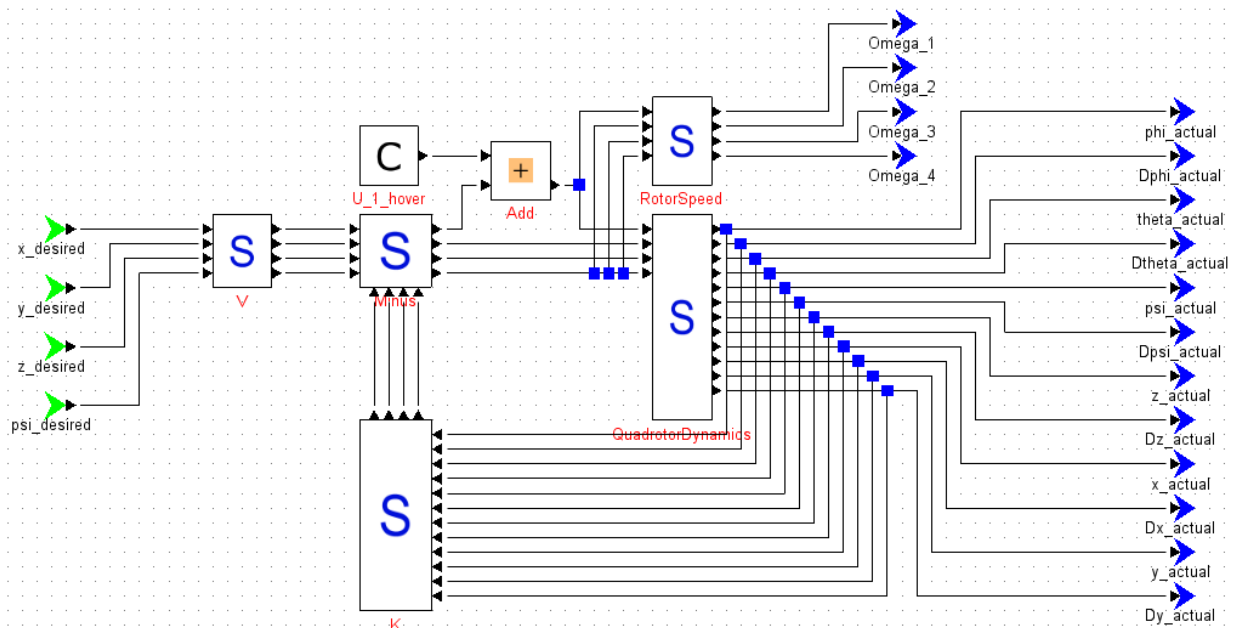


Abbildung 57: Zustandsgeregeltes Dynamikmodell als Signalflussplan in Virtuos

6.5 Evaluierung mit Dynamikmodell

Für die Validierung unter Realbedingungen wurden für die in den vorangegangenen Kapiteln beschriebene Implementierung für drei Instanzen der KFM verschiedene Evaluierungsszenarien entworfen.

Da reale Flugversuche aus Sicherheitsgründen ausschließlich innerhalb der Flugzelle durchgeführt werden können, wurde der zur Verfügung stehende Flugraum bei allen Szenarien auf deren Abmessungen von ca. 4 [m] × 5 [m] × 4.5 [m] (Breite in x -Richtung, Tiefe in y -Richtung und Höhe in z -Richtung) begrenzt. In z -Richtung wurden ausschließlich die obersten drei Meter als „Arbeitsraum“ für Flugbewegungen mit der KFM genutzt. Die unteren 1.5 Meter wurden als „Start-/Landraum“ reserviert.

Als statische Hindernisse innerhalb des Arbeitsraums wurden erneut die in die Flugzelle projizierten digitalen Abbilder von Dachträgern und einem Lüftungsrohr aus den Maschinenbaulaboren der Hochschule Esslingen verwendet. Da Arbeitsraum und Hindernisse somit nahezu identisch mit den in Kap. 5.7 und Kap. 5.8 vorgestellten Simulationen sind, wurde hier ebenfalls dieselbe Parametrierung der KFM verwendet ($n = 20$, $K = 0.1139$, $T = 0.0454$, $r_0 = 0.8$ [m] und $r_u = 0.9$ [m]).

Abbildung 58 zeigt die Online-Visualisierung der Testumgebung im VAL 3D-Webstudio. Zusätzlich sind die Initialpositionen $(x_0, y_0, z_0)^T$ der drei UAVs eingezeichnet. Es sei angemerkt, dass dabei die UAVs (Crazyflie) zur besseren Sichtbarkeit im doppelten Maßstab dargestellt sind. Alle Flugbewegungen wurden im durch das Loco Positioning System definierte Inertialsystem mit Ursprung im Anker #4 beschrieben.

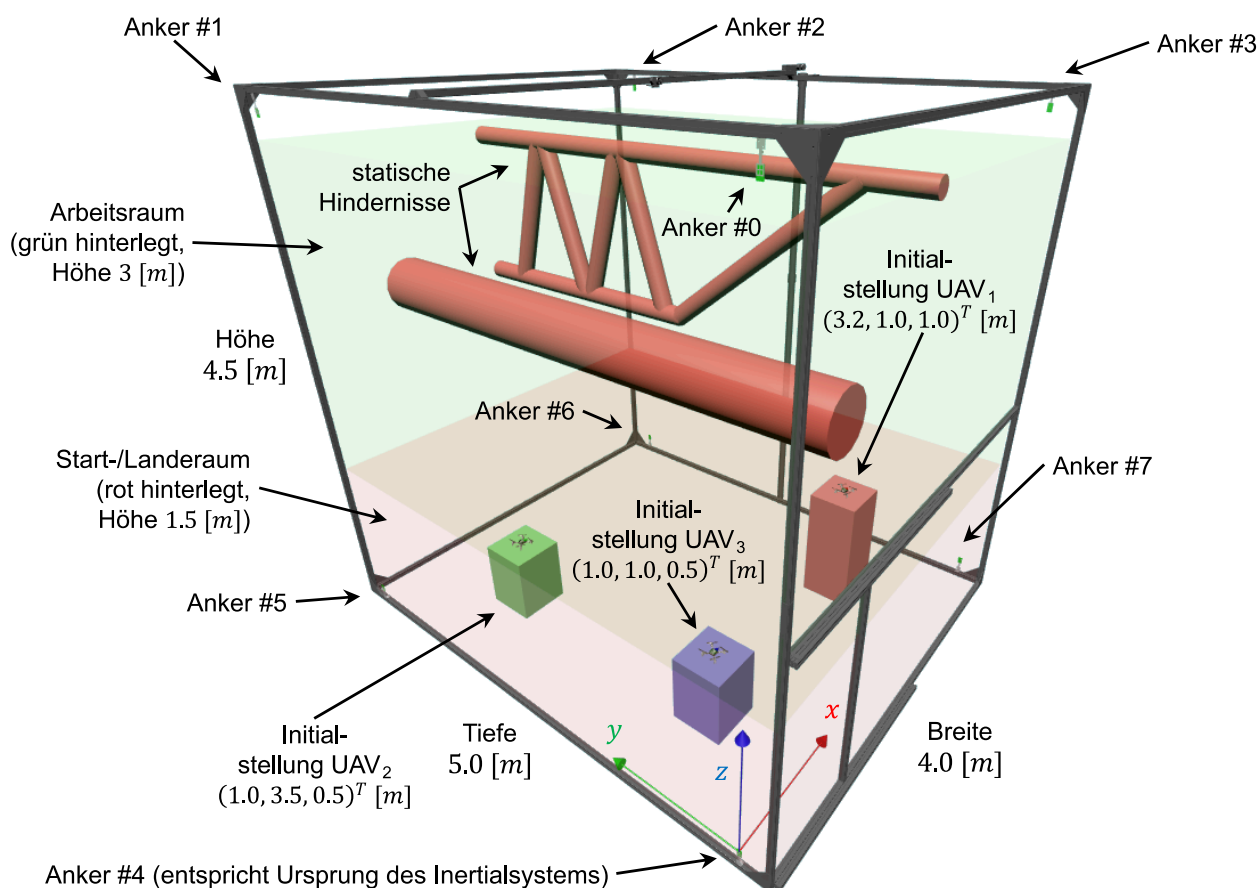


Abbildung 58: Online-Visualisierung der Testumgebung im VAL 3D-Webstudio

Zunächst wurden die Steuerungen im Sinne einer virtuellen Inbetriebnahme an eine rein simulierte Umgebung mit den Dynamikmodellen der UAVs angebunden (Hardware-in-the-Loop-Simulation). In einem ersten Evaluierungsszenario wurden von den UAVs jeweils mehrere vorgegebene Positionen in drei parallel ausgeführten Programmen angefliegen. Tabelle 12 zeigt die anzufliegenden Positionen für die drei UAVs in tabellarischer Form. Nach dem Start und vor der Landung wurde jeweils ein zusätzlicher Genauhalt mit 2 Sekunden Wartezeit („Sleep“) ausgeführt.

-	UAV ₁			UAV ₂			UAV ₃		
	x	y	z	x	y	z	x	y	z
Initialstellung	3.2	1	1	1	3.5	0.5	1	1	0.5
Start	3.2	1	2	1	3.5	2	1	1	2
Position 1	3.5	3	4.1	3.4	1.4	4	3.5	4	3
Position 2	0.5	2	2	2	1	3	1	1	2
Position 3	3.2	1	2	1	3.5	2	-	-	-
Landung	3.2	1	1	1	3.5	0.5	1	1	0.5

Tabelle 12: Anzufliegende Positionen des ersten Szenarios

Die drei Programme wurden auf den einzelnen Steuerungen manuell und nacheinander gestartet. Deren Ausführung erfolgte somit zeitlich unabhängig voneinander. Um dabei Kollisionen, welche durch einen hohen Bahnfehler bedingt sind, zu vermeiden, wurde die Interpolationsdynamik mit $a_{max} = 0.1 [m/s^2]$ und $v_{max} = 0.5 [m/s]$ zunächst niedrig gehalten.

Abbildung 59 zeigt die sich für eine exemplarische Ausführung ergebenden Verläufe der Positionen in x -, y - und z -Richtung der UAVs sowie deren Abstand d zum jeweils nächstgelegenen Hindernis. Bei den Positionsdaten sind jeweils Ist- (dick), Soll- (dünn) und Zielwert (gestrichelt) dargestellt.

Abschnitte, in welchen sich die UAVs in Initial- bzw. Wartestellung befinden oder einen Start bzw. eine Landung ausführen, sind für die Abstands- bzw. Kollisionsbewertung irrelevant. Da sich die UAVs in ihrer Initialstellung außerhalb des Arbeitsraums befinden, ist hier der Abstand d negativ (Hindernisdurchdringung) und wird erst während des Starts (Transition vom Start-/Landraum in den Arbeitsraum) positiv.

Zusätzlich sind für dieses Experiment in Abbildung 60 die räumlichen Trajektorien der Soll- (dünn) und Istpositionen (dick mit Markierungen im Abstand von einer Sekunde) sowie in Abbildung 61 die Online-Visualisierung mit den online geplanten KFM-Bahnen in fünfsekündigen Zeitschritten abgebildet. Die Positionen der UAVs sind in allen Bildern farblich markiert, wobei die UAVs selbst aufgrund der Perspektive vereinzelt verdeckt sind.

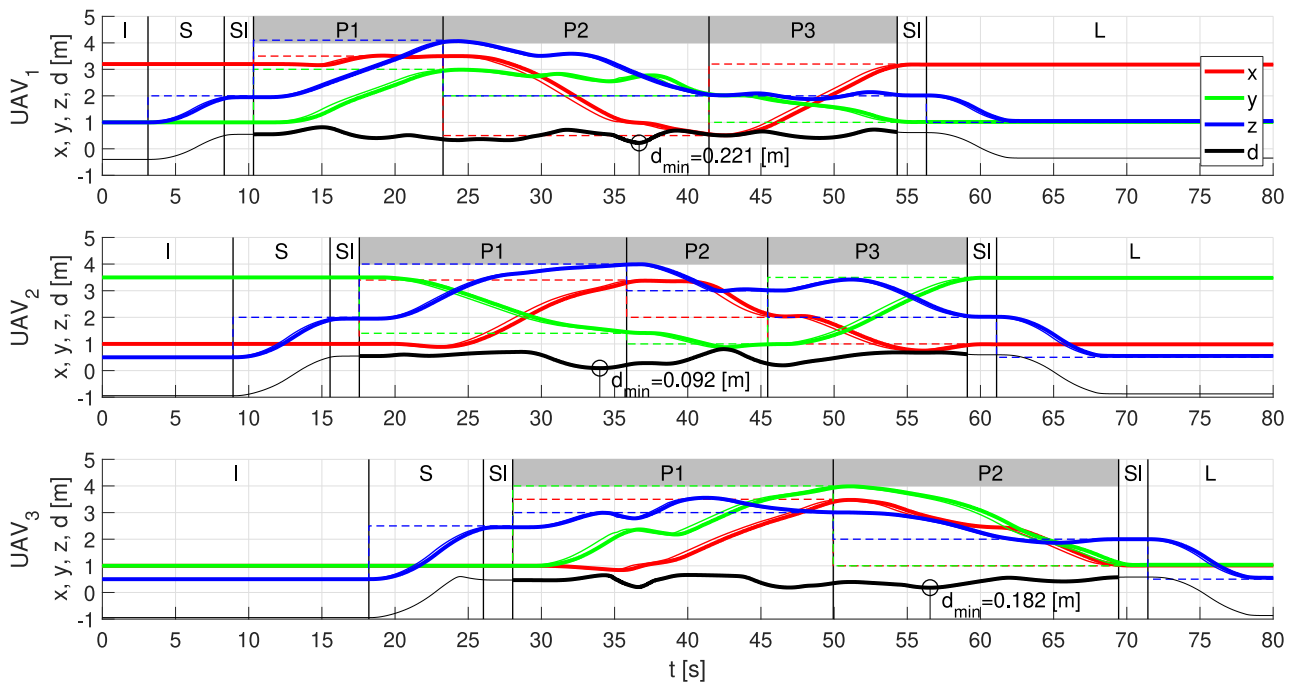


Abbildung 59: Positions- und Distanzverläufe des ersten Szenarios

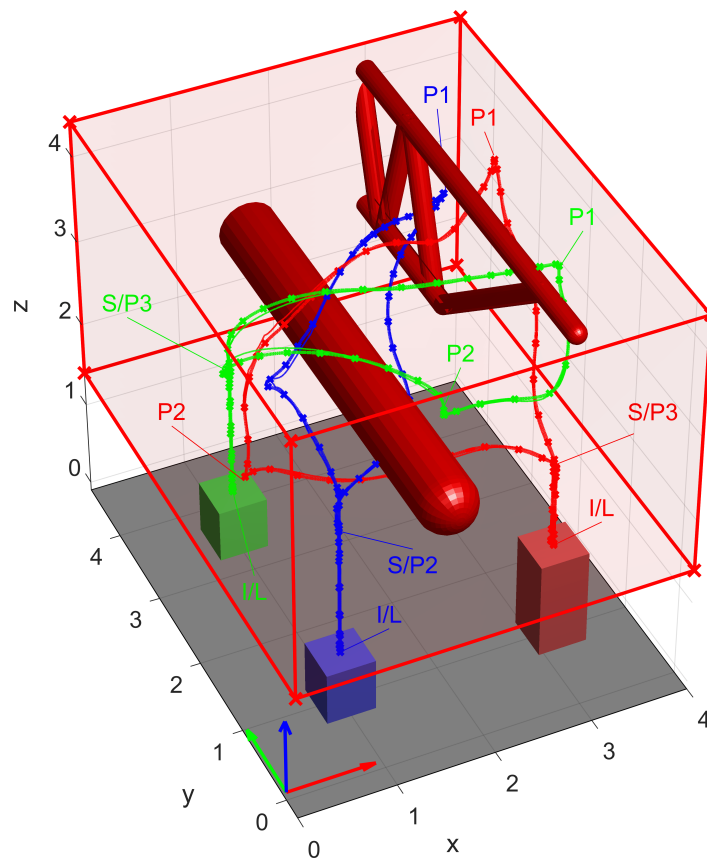


Abbildung 60: Trajektorien des ersten Szenarios

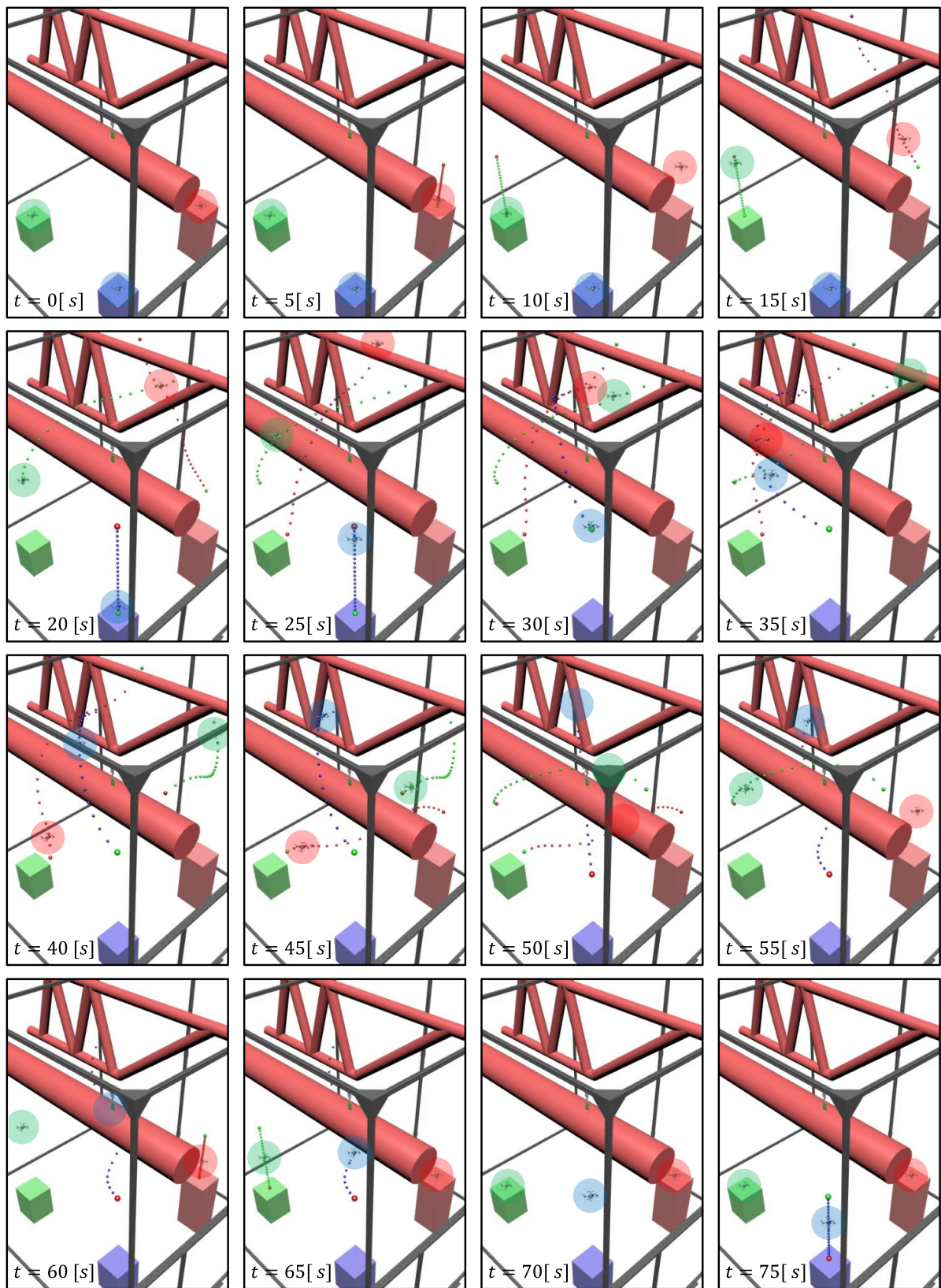


Abbildung 61: Online-Visualisierung des ersten Szenarios

Dieses Szenario veranschaulicht simulativ die Fähigkeit der KFM für unabhängig ausgelöste Flugbewegungen autonome und sich kontinuierlich an den Umgebungszustand anpassende Solltrajektorien zu generieren. Die Distanzverläufe belegen dabei die Kollisionsfreiheit der von den UAVs geflogenen Isttrajektorien.

Das zweite Evaluierungsszenario bestand aus mehreren Dauertests, bei welchen mindestens 15 Minuten lang zufällige Zielpositionen innerhalb derselben Testumgebung angefliegen wurden. Jeweils nach zehn angeflogenen Positionen wurde eine Landung durchgeführt. Code 3 zeigt das Ablaufprogramm, welches dafür auf allen drei Steuerungen im Dauerlauf ausgeführt wurde.

001	PRG START ();
002	PTP KFM START (A1:=x ₀ , A2:=y ₀ , A3:=2.000, A4:=0.000);
003	SLEEP (T:=2.000);
004	FOR i:=1 TO 10 DO
005	PTP KFM(A1:=RAND(), A2:=RAND(), A3:=RAND(), A4:=0.000);
006	END FOR
007	PTP KFM(A1:=x ₀ , A2:=y ₀ , A3:=2.000, A4:=0.000);
008	SLEEP (T:=2.000);
009	PTP KFM LAND (A1:=x ₀ , A2:=y ₀ , A3:=z ₀ , A4:=0.000);
010	SLEEP (T:=2.000);

Code 3: Programm für das zweite Szenario (Dauertests)

Um sicherzustellen, dass der Zielpunkt nicht mit einem Hindernis oder den (statischen) Start- und Zielpunkten der anderen UAVs kollidiert, wurde dessen zufällige Auswahl gegebenenfalls wiederholt, bis der Abstand zum nächsten potentiellen Kollisionspartner mindestens 0.5 Meter betrug. Später ist die Sicherstellung, dass bei der Beauftragung der UAVs lediglich valide Zielpunkte vorgegeben werden, Aufgabe einer übergeordneten Leitsteuerung.

Die Parametrierung der KFM wurde in diesem Szenario erneut beibehalten. Lediglich die Interpolationsdynamik wurde in mehreren Stufen gesteigert, um die Robustheit der von der KFM generierten Bahnen anhand des dabei zwangsläufig wachsenden Bahnfehlers zu untersuchen. Abbildung 62 zeigt die Verläufe der Abstände während der Dauertests für die einzelnen UAVs. Die nicht kollisionsrelevanten Abstände bei Starts und Landungen sind

dabei dünn dargestellt. Die Durchdringungen der Nulllinie entsprechen dabei den Übergängen zwischen Arbeits- und Start-/Landraum.

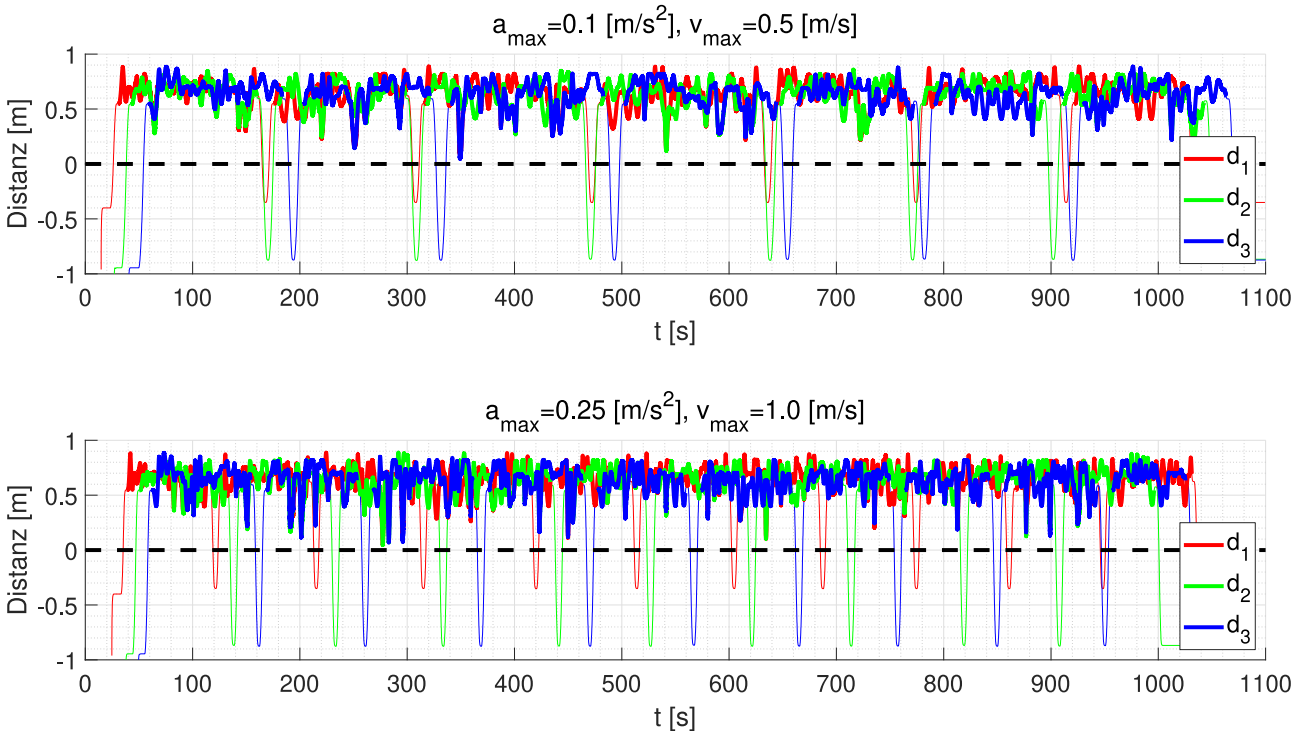


Abbildung 62: Distanzverläufe der Dauertests mit steigender Interpolationsdynamik

Tabelle 13 fasst die wichtigsten Kennzahlen der Dauertests für die einzelnen UAVs nochmals zusammen.

		-	UAV ₁	UAV ₂	UAV ₃
Interpolationsdynamik	$a_{max} = 0.1 [m/s^2]$ $v_{max} = 0.5 [m/s]$	Dauer in [s]	1011.5	1020.1	1025.0
		Angeflogene Positionen*	77	77	77
		Ø-Bahnfehler* in [m]	0.102	0.098	0.096
		Anzahl Kollisionen	0	0	0
	$a_{max} = 0.25 [m/s^2]$ $v_{max} = 1.0 [m/s]$	Dauer in [s]	1006.1	960.8	978.9
		Angeflogene Positionen*	121	110	110
		Ø-Bahnfehler* in [m]	0.156	0.148	0.143
		Anzahl Kollisionen	0	0	0
* ohne Starts und Landungen					

Tabelle 13: Zusammenfassung der Ergebnisse aus den Dauertests

Die Dauertests belegen die Robustheit der von der KFM generierten Solltrajektorien gegenüber Unsicherheiten bei den tatsächlichen Istpositionen der UAVs. Für die hier verwendete Parametrierung von KFM und Dynamikmodell resultierten selbst sehr große durchschnittliche Bahnfehler von ca. 14 bis 16 Zentimetern nicht in einer Kollision. Erst bei einer weiteren Steigerung der Interpolationsdynamik auf $a_{max} = 0.5 [m/s^2]$ und $v_{max} = 2.0 [m/s]$ und einem daraus resultierenden durchschnittlichen Bahnfehler von ca. 19 bis 21 Zentimetern konnten vereinzelte Kollisionen provoziert werden.

Bei allen Versuchen wurde die SPS mit einer Zykluszeit von $\Delta t = 10 [ms]$ und das Dynamikmodell mit $\Delta t = 1 [ms]$ betrieben. Der gesamten TwinCAT Laufzeitumgebung wurde eine maximale Systemauslastung von 90 Prozent gewährt. Die Echtzeitauslastung lag dabei stets zwischen 21 bis 23 Prozent. Zusätzlich wurden separate Zeitmessungen der vier Hauptfunktionen der KFM (vgl. Kap. 6.3) durchgeführt. Die Summe derer durchschnittlicher Rechenzeiten und Jitter (repräsentiert durch die Standardabweichung σ) ist in Tabelle 14 wiedergegeben.

Rechenzeit pro Zyklus in [ms]					
UAV ₁		UAV ₂		UAV ₃	
\emptyset	σ	\emptyset	σ	\emptyset	σ
1.9281	±0.0482	1.9322	±0.0473	1.9339	±0.0443

Tabelle 14: Rechenzeiten der KFM bei den Dauertests

Die durchschnittlichen Rechenzeiten für die Ausführung der KFM betragen auf allen Steuerungen weniger als 2 Millisekunden und die durchschnittliche Abweichung betrug weniger als 2.5 Prozent.

6.6 Evaluierung mit realen UAVs

Das dritte Evaluierungsszenario wurde mit realen UAVs (Crazyflie) innerhalb der Flugzelle durchgeführt. Somit waren in diesem Szenario neben den UAVs selbst ebenfalls fünf der sechs Arbeitsraumbegrenzungen physisch vorhanden. Die verbleibenden Hindernisse (Dachträger, Lüftungsrohr sowie die untere horizontale Arbeitsraumbegrenzung) wurden nur im Umgebungsmodell als virtuelle Hindernisse abgebildet.

6 Implementierung und Realisierung einer Testumgebung für die KFM

Es wurden erneut fest vorgegebene Positionen angefliegen, welche in Tabelle 15 aufgeführt sind. Da die Sensorik zur Lageerfassung des Crazyflie nicht in Wandnähe eingesetzt werden kann, war es notwendig die anzufliegenden Positionen gegenüber dem ersten Evaluierungsszenario abzuändern. Die Interpolationsdynamik wurde hier zu $a_{max} = 0.25 [m/s^2]$ und $v_{max} = 1.0 [m/s]$ gewählt. Dies entspricht der maximalen Einstellung, welche in den Dauertests noch als vollständig kollisionsfrei evaluiert wurde.

-	UAV ₁			UAV ₂			UAV ₃		
	x	y	z	x	y	z	x	y	z
Initialstellung	3.2	1	1	1	3.5	0.5	1	1	0.5
Start	3.2	1	1.75	1	3.5	1.75	1	1	1.75
Position 1	1.2	4	2	2.5	1.4	2.8	2.6	3.75	3.2
Position 2	1.2	2	2	1	3.5	1.75	1	1	1.75
Position 3	3.2	1	1.75	-	-	-	-	-	-
Landung	3.2	1	1	1	3.5	0.5	1	1	0.5

Tabelle 15: Anzufliegende Positionen des dritten Szenarios

Abbildung 63 zeigt die Positions- und Distanzverläufe für eine exemplarische Ausführung.

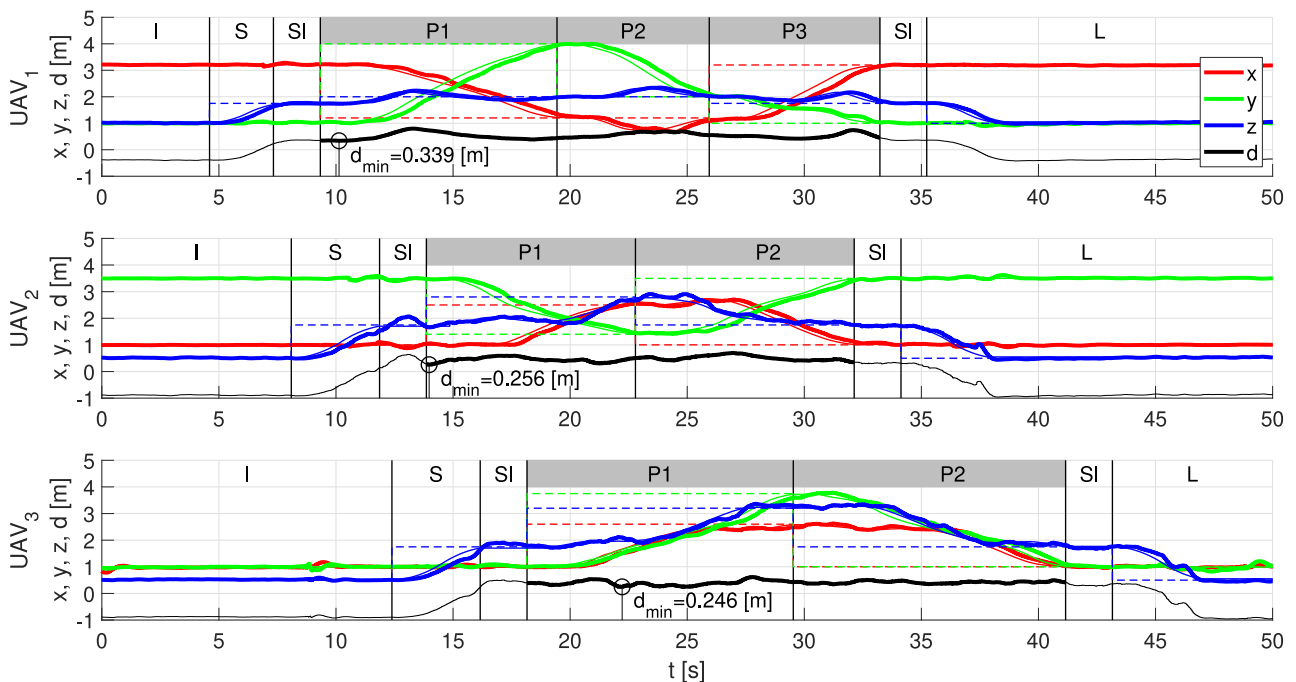


Abbildung 63: Positions- und Distanzverläufe des dritten Szenarios

In Abbildung 64 sind die zugehörigen räumlichen Trajektorien der Soll- und Istpositionen dargestellt.

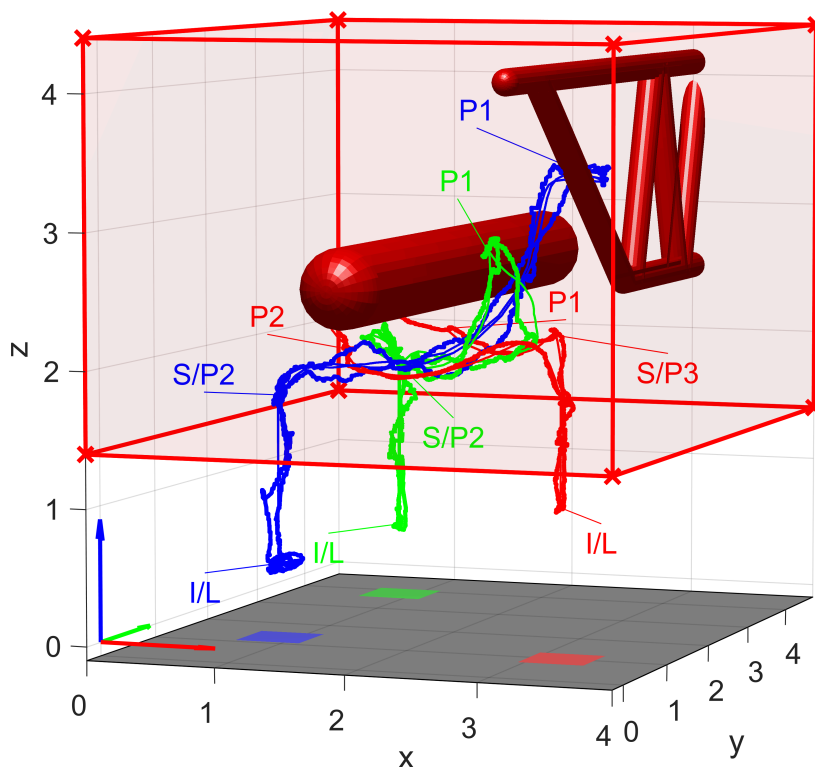


Abbildung 64: Trajektorien des dritten Szenarios

Der durchschnittliche Bahnfehler (ohne Starts und Landungen) für die einzelnen UAVs in diesem Szenario ist in Tabelle 16 aufgeführt.

-	UAV ₁	UAV ₂	UAV ₃
Ø-Bahnfehler in [m]	0.185	0.187	0.177

Tabelle 16: Bahnfehler des dritten Szenarios

Abbildung 65 zeigt in einer Bewegungsstudie die aus der Videoaufzeichnung des Szenarios extrahierten Positionen der UAVs. Diese sind zeitsynchron zu Abbildung 63 und mit den entsprechenden Zeitstempeln markiert. Die Dreiecke auf dem Boden der Flugzelle dienen lediglich der Erhöhung des Kontrastes für die Detektion durch das Flowdeck.

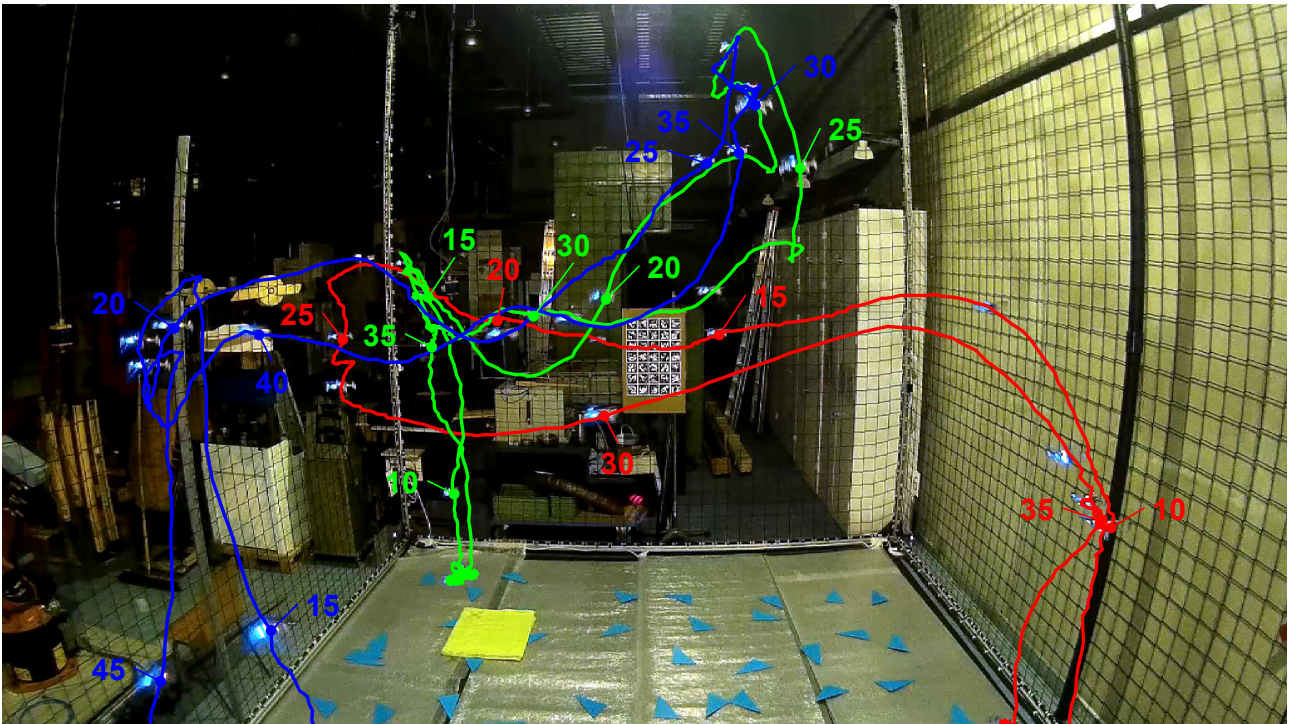


Abbildung 65: Bewegungsstudie des dritten Szenarios

In Abbildung 66 ist die Online-Visualisierung mit den jeweils aktuellen KFM-Bahnen ebenfalls in 2.5-sekündigen Zeitschritten abgebildet. Die UAVs sind dabei auf Basis ihrer realen Zustandsdaten abgebildet worden.

Die Ergebnisse bestätigen auch in diesem abschließenden Szenario die Kollisionsfreiheit der mithilfe der KFM generierten Solltrajektorien und damit die Anwendbarkeit der KFM auf reale UAVs.

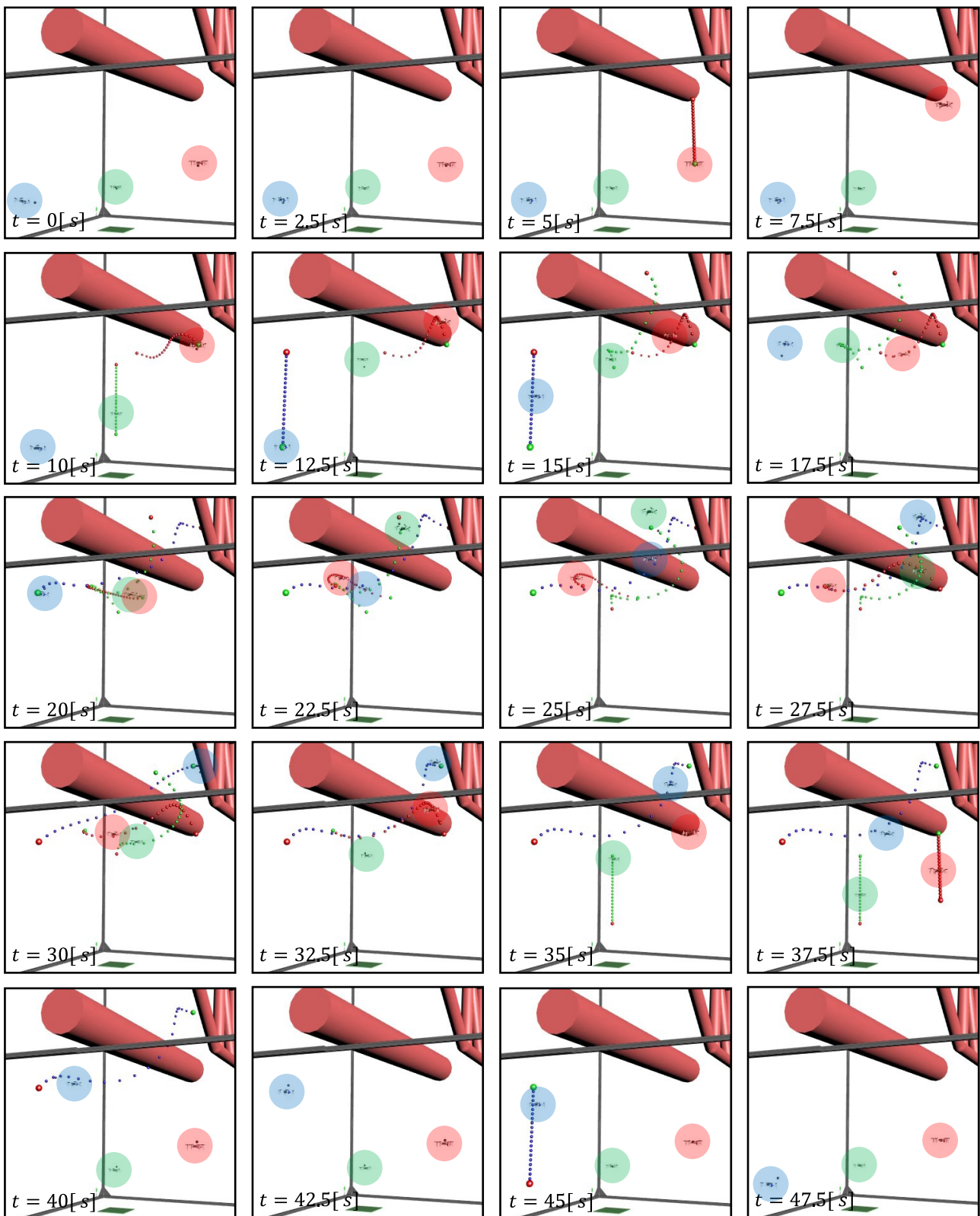


Abbildung 66: Online-Visualisierung des dritten Szenarios

7 Zusammenfassung

Aktuelle Entwicklungen in der Produktionstechnik sind vom Wunsch nach einem hohen Individualisierungsgrad der Produkte geprägt. Die damit verbundene Varianz bei der Abfolge einzelner Produktionsschritte sowie die bedarfsabhängige Auslastung der zur Verfügung stehenden Bearbeitungsmaschinen erfordert gleichzeitig eine ebenso flexible Materialflusseinrichtung. Für den Transport von Kleinteilen ist hierbei der Einsatz unbemannter Flugfahrzeuge (UAVs) Gegenstand aktueller Forschung. Gegenüber bodengebundenen oder starren Materialflusseinrichtungen (FTS, Förderband, Fördermatrix) ermöglichen UAVs die Erschließung des meist ungenutzten Luftraums innerhalb der Produktionsstätten. Parallele und voneinander unabhängige Beauftragungen der UAVs erzeugen dabei eine Umgebung mit hohem Kollisionspotenzial, in welcher die Flugbewegungen der einzelnen UAVs nicht mehr a priori geplant werden können.

Aus dieser in Kap. 1 formulierten Problemstellung wurde das Ziel der Arbeit abgeleitet. Dieses bestand in der Entwicklung und prototypischen Realisierung einer Online-Bahnplanungsmethode, welche in der Lage ist, Bahnen für mehrere UAVs zur Laufzeit an den aktuellen Umgebungszustand anzupassen. Dabei konnten die folgenden Anforderungen an die Methode identifiziert werden: Generierung kollisionsfreier Bahnen in dynamischer Umgebung, Echtzeitfähigkeit, Recheneffizienz, numerische Robustheit, Optimalität, 3D-Fähigkeit und skalierbare Implementierbarkeit.

In Kap. 2 wurde zunächst der aktuelle Stand der Forschung und Technik mit Fokus auf Bahnplanungsmethoden in dynamischer Umgebung beleuchtet und gemäß den genannten Anforderungen bewertet. Von den untersuchten Methoden erfüllte keine alle Anforderungen. Dennoch konnte die Elastic-Bands-Methode (EBM) trotz Defiziten hinsichtlich ihrer numerischen Robustheit in ihrer Kernfunktionalität als den Anforderungen am nächsten kommend identifiziert werden.

Die EBM beschreibt die Bahn als Abfolge einzelner Stützstellen, welche durch Aufprägung virtueller Abstoßungs- und Rückstellkräfte kontinuierlich in einen paretooptimalen Zustand aus einer Maximierung des Abstandes zu den Hindernissen bei gleichzeitiger Minimierung der Bahnlänge konvergieren.

In Kap. 3 wurde das Bahnverhalten der EBM im Rahmen einer Voruntersuchung analysiert. Dabei konnte empirisch gezeigt werden, dass die numerische Stabilität aufgrund der Aktualisierung der Bahn mittels einer expliziten Transitions Gleichung durch die Wahl von Parametrierung und Zeitschrittweite (sowie der Diskretisierung und dem aktuellen Betriebspunkt) begrenzt wird. Aufgrund der Verwendung von nichtlinearen Kraftgesetzen kann weder die Stabilitätsgrenze der EBM analytisch bestimmt werden noch ist es möglich, das Bahnverhalten aufgrund der Parametrierung vorherzusagen, sodass auch diese empirisch ermittelt werden muss.

Mit der Kurvenflussmethode (KFM) wurde in Kap. 4 eine neue Bahnplanungsmethode entwickelt, welche das Funktionsprinzip der EBM aufgreift und gleichzeitig deren beschriebene Schwächen behebt. Dies konnte einerseits durch die Entwicklung einer Methodik, welche die Parametrierung des Bahnverhaltens auf Basis physikalischer Größen zulässt und zum anderen durch die Verwendung eines impliziten Lösungsverfahrens für die Aktualisierung der Bahn erreicht werden.

Basierend auf geometrischen Kurvenflüssen wurde dazu zunächst ein lineares Ersatzmodell für das Rückstellverhalten der Bahn hergeleitet. Dieses beschreibt die Stützstellen einer ortsdiskretisierten Bahn als gekoppelte Verzögerungsglieder 1. Ordnung und kann mit systemtheoretischen Methoden für LZI-Systeme analysiert werden. Durch die spezielle Toeplitz-Struktur der zugrundeliegenden Systemmatrix können deren Eigenwerte und damit ebenfalls die sich ergebende Bahndynamik beim Rückstellen in Abhängigkeit der Diskretisierung explizit angegeben werden.

Das System wurde anschließend um die Abstoßungskraft sowie um Parameter zur Gewichtung beider Kräfte, durch welche das Bahnverhalten (Bahndynamik und Sicherheitsabstand) beeinflusst werden kann, ergänzt. Um die Linearität des Systems für die Analyse zu erhalten, wurde zunächst eine lineare und eindimensional wirkende Streckenlast als Hindernismodell verwendet. Diese stellt die maximal mögliche Belastung der Bahn dar und wurde im Folgenden als Referenz für die Parametrierung der KFM verwendet. Es konnte gezeigt werden, dass die Bahn bei vollbelasteter Abstoßung die höchste und bei unbelasteter Rückstellung die geringste Dynamik aufweist. Mit diesen beiden analytisch bestimm-

baren Grenzwerten war es möglich, die zu erwartende Bahndynamik für eine gegebene Diskretisierung und Parametrierung für alle Belastungsfälle einzugrenzen.

Die Bewertung des Sicherheitsabstandes der Bahn erfolgte auf Basis ihrer stationären Endlage, welche ebenfalls analytisch berechnet werden kann. Mithilfe mathematischer Normen konnte der Sicherheitsabstand der gesamten Bahn in einer einzelnen repräsentativen Kenngröße bemessen werden.

Um die abstrakten Parameter der KFM nicht empirisch ermitteln zu müssen, wurde deren inverser Zusammenhang zum Bahnverhalten hergeleitet, womit die Parametrierung der KFM durch Vorgabe physikalischer Kenngrößen (Stellzeit und stationäre Endlage) erfolgen konnte.

Die bisher zeitkontinuierlich betrachtete Bahn wurde anschließend mit dem impliziten Eulerverfahren zeitdiskretisiert. Für die recheneffiziente (semiimplizite) Anwendung des Verfahrens wurde die im Allgemeinen nichtlineare Abstoßungskraft explizit ausgewertet und floss als Eingangsgröße in die nunmehr als linear betrachtete Differentialgleichung ein. Dadurch konnte der semiimplizite Eulerschritt auf die Lösung eines linearen Gleichungssystems reduziert werden, welches durch Anwendung des Thomas-Algorithmus mit linearer Rechenkomplexität gelöst werden konnte.

Anhand einer Wiederholung der Experimente aus der Voruntersuchung wurde ein Vergleich zwischen KFM und EBM durchgeführt, wobei die KFM unter Vorgabe des bei der EBM identifizierten Bahnverhaltens parametrierung wurde. Damit konnten folgende signifikanten Vorteile der KFM gegenüber der EBM bestätigt werden: Numerisch stabile Lösbarkeit, analytische Parametrierbarkeit, Skalierungsunabhängigkeit bezüglich der Lagegrößen sowie eine etwas höhere Recheneffizienz.

Unabhängig von der EBM wurde die KFM in Kap. 5 um weitere Steuerungsfunktionen erweitert. Dazu gehörte zunächst die Entwicklung eines räumlichen Hindernismodells, welches einerseits in der Lage war komplexe Umgebungen, wie sie im Luftraum von Maschinenhallen vorzufinden sind, abzubilden und gleichzeitig recheneffiziente Abstandsrechnungen zulässt. Dieses Modell wurde auf Basis umhüllender Körper mit geometrischen Primitiven realisiert. Das resultierende skalare Distanzfeld wurde zugunsten einer

numerisch robusten Auswertung geglättet, sodass die Abstoßungskräfte direkt aus dem Betrag und dem Gradienten dieses Feldes abgeleitet werden konnten.

Zusätzlich erlaubte die Berücksichtigung von Durchdringungen einen Einschwingvorgang der Bahn aus einem kollisionsbehafteten Zustand heraus, wodurch die bei der EBM notwendige initiale Wegsuche entfällt.

Die Interpolation auf der dynamischen Bahn erfolgte mit einem beschleunigungsbegrenzten 3-Phasenprofil und Start- und Landemanöver der UAVs von bzw. zu Bearbeitungsstationen unterhalb des Luftraums wurden in Form von gesicherten Flugkorridoren realisiert.

In Kap. 6 wurden mehrere jeweils auf einer eigenen Industriesteuerung und unter Echtzeit ausgeführte Instanzen der KFM in Betrieb genommen. Jede Instanz steuerte dabei ein UAV und konnte mithilfe von Ablaufprogrammen beauftragt werden. Die Bereitstellung des Umgebungsmodells sowie der Austausch der Zustandsdaten der einzelnen UAVs wurde über eine lokale Cloud realisiert.

Es wurden verschiedene Evaluierungsszenarien mit Dynamikmodellen der UAVs (Hardware-in-the-Loop-Simulation) sowie mit realen UAVs durchgeführt. Im Rahmen dieser Szenarien konnten die zu Beginn formulierten Anforderungen an die zu entwickelnde Bahnplanungsmethode für die KFM überprüft und bestätigt werden. Im Folgenden werden die Ergebnisse abschließend nochmals den eingangs formulierten Anforderungen gegenübergestellt:

Kollisionsfreiheit in dynamischer Umgebung:

Es wurden Dauertests durchgeführt, bei welchen es selbst bei Bahnfehlern der UAVs von 15 Zentimetern zu keinerlei Kollisionen kam. Dies unterstreicht die Robustheit der durch die KFM generierten Lagesollwerttrajektorien gegenüber einem schlechten Folgeverhalten und belegt die Kollisionsfreiheit der KFM in dynamischer Umgebung.

Recheneffizienz:

Dauermessungen ergaben einen Rechenzeitbedarf von unter zwei Millisekunden pro Aktualisierung der Bahn. Dies liegt innerhalb der geforderten Aktualisierungsrate von wenigen Millisekunden und belegt die Recheneffizienz der KFM.

Echtzeitfähigkeit:

Die KFM wurde unter einem Echtzeitbetriebssystem implementiert und bei fester Zykluszeit ausgeführt. Dauermessungen ergaben eine Standardabweichung vom Mittelwert der Rechenzeit von lediglich 2.5 Prozent. Dies bestätigt die deterministische Ausführbarkeit und belegt damit die Echtzeitfähigkeit der KFM.

Optimalität:

Die der KFM zugrunde liegende Differentialgleichung beschreibt ein Evolutionsproblem, welches stets in einen energieminimalen Zustand konvergiert. Dieser entspricht dem über die Parametrierung der KFM definierten Paretooptimum und belegt damit die (lokale) Optimalität der KFM.

Numerische Robustheit:

Durch die Verwendung des impliziten Eulerverfahrens zur Aktualisierung der Bahn wurde ein (für LZI-Systeme) A-stabiles Integrationsverfahren gewählt. Die hohe numerische Dämpfung des Verfahrens kompensiert Unsicherheiten, welche durch dessen semiimplizite Anwendung aufgrund der nichtlinearen Abstoßungskraft entstehen können. Obgleich die Stabilität nichtlinearer Systeme analytisch nicht nachgewiesen werden kann, untermauert die stets monotone Konvergenz der Bahnen während der Dauertests die numerische Robustheit der KFM.

3D-Fähigkeit:

Als generalisierte Methode kann die KFM unter Voraussetzung eines entsprechenden Hindernismodells für Bahnplanungsprobleme von beliebiger Dimension eingesetzt werden. Im Rahmen dieser Arbeit wurden räumliche Bahnen geplant, was die 3D-Fähigkeit der KFM belegt.

Skalierbarkeit:

Im Rahmen der Evaluierungsszenarien wurden drei vernetzte Instanzen der KFM parallel betrieben. Die zugrundeliegende Steuerungsarchitektur erlaubt grundsätzlich die Vernetzung einer beliebigen Anzahl von Instanzen und belegt damit die Skalierbarkeit der KFM.

8 Literaturverzeichnis

- Altschuler 1991** Altschuler, Steven J., 1991.
Singularities of the curve shrinking flow for space curves.
Journal of Differential Geometry **34** (2), S. 491–514.
DOI: 10.4310/jdg/1214447218
- Barraquand et al. 1991** Barraquand, Jerome; Latombe, Jean-Claude, 1991.
Robot motion planning: A distributed representation approach.
The International Journal of Robotics Research **10** (6), S. 628–649.
DOI: 10.1177/027836499101000604
- Barraquand et al. 1992** Barraquand, Jerome; Langlois, Bruno; Latombe, Jean-Claude, 1992.
Numerical potential field techniques for robot path planning.
IEEE Transactions on Systems, Man, and Cybernetics **22** (2), S. 1012–1017.
DOI: 10.1109/21.148426
- Batterman 2003** Batterman, Robert W., 2003.
Falling cats, parallel parking, and polarized light.
Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics **34** (4), S. 527–557.
DOI: 10.1016/S1355-2198(03)00062-5

- Benenson et al. 2006** Benenson, Rodrigo; Petti, Stephane; Fraichard, Thierry; Parent, Michel, 2006.
Integrating Perception and Planning for Autonomous Navigation of Urban Vehicles.
In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 98–104.
ISBN 1-4244-0258-1
DOI: 10.1109/IROS.2006.281806
- Bergen et al. 2018** Bergen, Timm von; Huptych, Marcel; Röck, Sascha, 2018.
Autonome Flugrobotik für die wandlungsfähige Fabrik – Steuerung, Infrastruktur und Simulation.
wt Werkstattstechnik online **108** (11-12), S. 834–839.
DOI: 10.37544/1436-4980-2018-11-12-94
- Berti et al. 2008** Berti, Hugo; Sappa, Angel Domingo; Agamennoni, Osvaldo Enrique, 2008.
Improved Dynamic Window Approach by using Lyapunov Stability Criteria.
Latin American applied research (38), S. 289–298.
- Bettenhausen et al. 2013** Bettenhausen, Kurt D.; Kowalewski, Stefan, 2013.
Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation.
Report, VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik
- Borenstein et al. 1989** Borenstein, Johann; Koren, Yoram, 1989.
Real-time obstacle avoidance for fast mobile robots.
IEEE Transactions on Systems, Man, and Cybernetics **19** (5), S. 1179–1187.
DOI: 10.1109/21.44033

- Borenstein et al. 1991** Borenstein, Johann; Koren, Yoram, 1991.
The vector field histogram-fast obstacle avoidance for mobile robots.
IEEE Transactions on Robotics and Automation **7** (3), S. 278–288.
DOI: 10.1109/70.88137
- Bortoff 2000** Bortoff, Scott A., 2000.
Path planning for UAVs.
In: *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, S. 364–368.
ISBN 0-7803-5519-9
DOI: 10.1109/ACC.2000.878915
- Brandt et al. 2005** Brandt, Thorsten; Sattel, Thomas, 2005.
Path planning for automotive collision avoidance based on elastic bands.
In: *Proceedings of the 16th IFAC World Congress*, S. 210–215.
ISBN 978-3-902661-75-3
DOI: 10.3182/20050703-6-CZ-1902.01245
- Brock 2000** Brock, Oliver, 2000.
Generating robot motion: The integration of planning and execution.
Standford, Univ., Diss., 2000

Brock et al. 1999

Brock, Oliver; Khatib, Oussama, 1999.

High-speed navigation using the global dynamic window approach.

In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, S. 341–346.

ISBN 0-7803-5180-0

DOI: 10.1109/ROBOT.1999.770002

Brock et al. 2002a

Brock, Oliver; Khatib, Oussama; Viji, Sriram, 2002.

Task-consistent obstacle avoidance and motion behavior for mobile manipulation.

In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, S. 388–393.

ISBN 0-7803-7272-7

DOI: 10.1109/ROBOT.2002.1013391

Brock et al. 2002b

Brock, Oliver; Khatib, Oussama, 2002.

Elastic strips: A framework for motion generation in human environments.

The International Journal of Robotics Research **21** (12), S. 1031–1052.

DOI: 10.1177/0278364902021012002

- Bubeck et al. 2014** Bubeck, Alexander; Gruhler, Matthias; Reiser, Ulrich; Weißhardt, Florian, 2014.
Vom fahrerlosen Transportsystem zur intelligenten mobilen Automatisierungsplattform.
In: Bauernhansl, Thomas; Hompel, Michael ten; Vogel-Heuser, Birgit (Hrsg.): *Industrie 4.0 in Produktion, Automatisierung und Logistik*.
Wiesbaden: Springer Vieweg, S. 221–233.
DOI: 10.1007/978-3-658-04682-8_11
- Calisi 2007** Calisi, Daniele, 2007.
Motion Planning, Reactive Methods, and Learning Techniques for Mobile Robot Navigation.
Report, Sapienza University of Rome
- Castro et al. 2002** Castro, Daniel; Nunes, Urbano; Ruano, Antonio, 2002.
Reactive local navigation.
In: *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, S. 2427–2432.
ISBN 0-7803-7474-6
DOI: 10.1109/IECON.2002.1185353
- Chen et al. 2013** Chen, Xia; Zhang, Jing, 2013.
The Three-Dimension Path Planning of UAV Based on Improved Artificial Potential Field in Dynamic Environment.
In: *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, S. 144–147.
ISBN 978-0-7695-5011-4
DOI: 10.1109/IHMSC.2013.181

- Chou et al. 2001** Chou, Kai-Seng; Zhu, Xi-Ping, 2001.
The curve shortening problem.
New York: Chapman and Hall/CRC.
ISBN 978-1-4200-3570-4
DOI: 10.1201/9781420035704
- Chow et al. 2007** Chow, Bennett; Glickenstein, David, 2007.
Semidiscrete Geometric Flows of Polygons.
The American Mathematical Monthly **114** (4), S. 316–328.
DOI: 10.1080/00029890.2007.11920419
- Connolly et al. 1990** Connolly, Christopher I.; Burns, John Brian; Weiss, Rich,
1990.
Path planning using Laplace’s equation.
In: *Proceedings., IEEE International Conference on Robotics and Automation*, S. 2102–2106.
ISBN 0-8186-9061-5
DOI: 10.1109/ROBOT.1990.126315
- Da Fonseca 2007** Da Fonseca, Carlos M., 2007.
On the eigenvalues of some tridiagonal matrices.
Journal of Computational and Applied Mathematics **200**
(1), S. 283–286.
DOI: 10.1016/j.cam.2005.08.047
- Dechter et al. 1985** Dechter, Rina; Pearl, Judea, 1985.
Generalized best-first search strategies and the optimality
of A*.
Journal of the ACM (JACM) **32** (3), S. 505–536.
DOI: 10.1145/3828.3830

- Demeester et al. 2005** Demeester, Eric; Nuttin, Marnix; Vanhooydonck, Dirk; Vanacker, Gerolf; van Brussel, Hendrik, 2005.
Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section.
In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 2357–2362.
ISBN 0-7803-8912-3
DOI: 10.1109/IROS.2005.1545383
- Devadoss et al. 2011** Devadoss, Satyan L.; O'Rourke, Joseph, 2011.
Discrete and computational geometry.
Princeton, Oxford: Princeton University Press.
ISBN 978-0-6911-4553-2
- Dijkstra 1959** Dijkstra, Edsger W., 1959.
A note on two problems in connexion with graphs.
Numerische Mathematik **1** (1), S. 269–271.
DOI: 10.1007/BF01386390
- Dorst 2015** Dorst, Wolfgang, 2015.
Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0.
Report, BITKOM e.V.
- Dubins 1957** Dubins, Lester E., 1957.
On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents.
American Journal of mathematics **79** (3), S. 497–516.
DOI: 10.2307/2372560

- Elbanhawi et al. 2014** Elbanhawi, Mohamed; Simic, Milan, 2014.
Sampling-Based Robot Motion Planning: A Review.
Ieee access (2), S. 56–77.
DOI: 10.1109/ACCESS.2014.2302442
- Elfes 1989** Elfes, Alberto, 1989.
Using occupancy grids for mobile robot perception and navigation.
Computer **22** (6), S. 46–57.
DOI: 10.1109/2.30720
- Erdmann et al. 1987** Erdmann, Michael; Lozano-Perez, Tomas, 1987.
On multiple moving objects.
Algorithmica **2**, S. 477–521.
DOI: 10.1007/BF01840371
- Ericson 2004** Ericson, Christer, 2004.
Real-Time Collision Detection.
Boca Raton: CRC Press.
ISBN 978-1-5586-0732-3
- Felix 2017** Felix, Rudolf, 2017.
Vom Fließband zur gesteuerten Flexibilität: Kennzahlen-optimiert produziert.
Verfügbar unter: <https://www.it-production.com/allgemein/vom-fliessband-zur-gesteuerten-flexibilitaet/>
Zugriff am: 27 05 2021
- Filippis et al. 2011** Filippis, Luca de; Guglier, Giorgio; Quagliotti, Fulvia, 2011.
Path Planning Strategies for UAVS in 3D Environments.
Journal of Intelligent & Robotic Systems **65**, S. 247–264.
DOI: 10.1007/s10846-011-9568-2

Fiorini et al. 1998

Fiorini, Paolo; Shiller, Zvi, 1998.

Motion Planning in Dynamic Environments Using Velocity Obstacles.

The International Journal of Robotics Research **17** (7), S. 760–772.

DOI: 10.1177/027836499801700706

Förster 2015

Förster, Julian, 2015.

System Identification of the Crazyflie 2.0 Nano Quad-roboter.

Zürich, Univ., Bachelorarbeit, 2015

DOI: 10.3929/ethz-b-000214143

Fox et al. 1997

Fox, Dieter; Burgard, Wolfram; Thrun, Sebastian, 1997.

The dynamic window approach to collision avoidance.

IEEE Robotics & Automation Magazine **4** (1), S. 23–33.

DOI: 10.1109/100.580977

Fraichard 1993

Fraichard, Thierry, 1993.

Dynamic trajectory planning with dynamic constraints: A 'state-time space' approach.

In: *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, S. 1393–1400.

ISBN 0-7803-0823-9

DOI: 10.1109/IROS.1993.583794

Frazzoli et al. 2002

Frazzoli, Emilio; Dahleh, Munther A.; Feron, Eric, 2002.

Real-Time Motion Planning for Agile Autonomous Vehicles.

Journal of Guidance, Control, and Dynamics **25** (1), S. 116–129.

DOI: 10.2514/2.4856

- Fujimura 2012** Fujimura, Kikuo, 2012.
Motion Planning in Dynamic Environments.
Tokyo: Springer.
Computer Science Workbench.
ISBN 978-4-431-68165-6
DOI: 10.1007/978-4-431-68165-6
- Gammell et al. 2014** Gammell, Jonathan D.; Srinivasa, Siddhartha S; Barfoot, Timothy D., 2014.
Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic.
In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 2997–3004.
ISBN 978-1-4799-6934-0
DOI: 10.1109/IROS.2014.6942976
- Gershgorin 1931** Gershgorin, Semyon Aranovich, 1931.
Über die Abgrenzung der Eigenwerte einer Matrix.
Известия Российской академии наук. Серия математическая (6), S. 749–754.
- Gipser 1999** Gipser, Michael, 1999.
Systemdynamik und Simulation.
Wiesbaden: Springer Vieweg.
ISBN 978-3-519-02743-0
DOI: 10.1007/978-3-663-11581-6
- Goerzen et al. 2010** Goerzen, Chad; Kong, Zhaodan; Mettler, Bernard, 2010.
A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance.
Journal of Intelligent and Robotic Systems **57**, S. 65–100.
DOI: 10.1007/s10846-009-9383-1

- Golub et al. 2013** Golub, Gene H.; van Loan, Charles F., 2013.
Matrix Computations.
4. Auflage.
Baltimore: Johns Hopkins University Press.
ISBN 978-1-4214-0794-4
- Graf et al. 2004** Graf, Birgit; Hans, Matthias; Schraft, Rolf D., 2004.
Care-O-bot II—Development of a Next Generation Robotic
Home Assistant.
Autonomous Robots **16** (2), S. 193–205.
DOI: 10.1023/B:AURO.0000016865.35796.e9
- Graf et al. 2009** Graf, Birgit; Parlitz, Christopher; Hägele, Martin, 2009.
Robotic Home Assistant Care-O-bot® 3 Product Vision
and Innovation Platform.
In: Julie A. Jacko (Hrsg.): *International Conference on
Human-Computer Interaction*, S. 312–320.
ISBN 978-3-642-02576-1
DOI: 10.1007/978-3-642-02577-8_34
- Günter 2014** Günter, Ullrich, 2014.
*Fahrerlose Transportsysteme: Eine Fibel-mit Praxisan-
wendungen-zur Technik-für die Planung*.
2. Auflage.
Wiesbaden: Springer Vieweg.
ISBN 978-3-8348-2591-9
DOI: 10.1007/978-3-8348-2592-6

- Günthner et al. 2006** Günthner, Willibald; Wilke, Michael; Zäh, Michael; Aull, Florian; Rudolf, Henning, 2006.
Produktion individualisierter Produkte.
In: Lindemann, Udo; Reichwald, Ralf; Zäh, Michael F. (Hrsg.): *Individualisierte Produkte—Komplexität beherrschen in Entwicklung und Produktion*.
Berlin, Heidelberg: Springer, S. 63–87.
DOI: 10.1007/3-540-34274-5_5
- Hart et al. 1968** Hart, Peter E.; Nilsson, Nils J.; Raphael, Bertram, 1968.
A Formal Basis for the Heuristic Determination of Minimum Cost Paths.
IEEE Transactions on Systems Science and Cybernetics 4 (2), S. 100–107.
DOI: 10.1109/TSSC.1968.300136
- Hrabar 2008** Hrabar, Stefan, 2008.
3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs.
In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 807–814.
ISBN 978-1-4244-2057-5
DOI: 10.1109/IROS.2008.4650775
- Hrabar 2011** Hrabar, Stefan, 2011.
Reactive obstacle avoidance for Rotorcraft UAVs.
In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 4967–4974.
ISBN 978-1-61284-454-1
DOI: 10.1109/IROS.2011.6094629

Hsu et al. 1997

Hsu, David; Latombe, Jean-Claude; Motwani, Rajeev, 1997.

Path planning in expansive configuration spaces.

In: *Proceedings of International Conference on Robotics and Automation*, S. 2719–2726.

ISBN 0-7803-3612-7

DOI: 10.1109/ROBOT.1997.619371

Huiliang et al. 2003

Huiliang, Zhang; Ying, Huang Shell, 2003.

Dynamic map for obstacle avoidance.

In: *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, S. 1152–1157.

ISBN 0-7803-8125-4

DOI: 10.1109/ITSC.2003.1252665

Huptych et al. 2011a

Huptych, Marcel; Groh, Konrad; Röck, Sascha, 2011.

Online Path Planning for Industrial Robots in Varying Environments Using the Curve Shortening Flow Method.

In: Jeschke, Sabina; Liu, Honghai; Schilberg, Daniel (Hrsg.): *Intelligent Robotics and Applications. ICIRA 2011.*, S. 73–82.

ISBN 978-3-642-25485-7

DOI: 10.1007/978-3-642-25486-4_8

- Huptych et al. 2011b** Huptych, Marcel; Röck, Sascha; Verl, Alexander, 2011.
Online-Roboterbahnplanung mit Hilfe der Kurvenflussmethode.
In: Brecher, Christian; Krüger, Jörg; Uhlmann, Eckart; Verl, Alexander (Hrsg.): *Sichere Mensch-Maschine-Interaktion*.
Düsseldorf: VDI-Verlag, S. 64–72.
ISBN 978-3-18-368102-0
- Huptych et al. 2012** Huptych, Marcel; Röck, Sascha, 2012.
Ein Beitrag zur sicheren Automatisierung von Manipulatoren in veränderlichen Umgebungen.
In: *Automation 2012, 13. Branchentreff der Mess- und Automatisierungstechnik*, S. 267–271.
ISBN 978-3-18-092171-6
- Huss et al. 1987** Huss, Janice E.; Pennline, James A., 1987.
A Comparison of Five Benchmarks.
Report, Lewis Research Center, National Aeronautics and Space Administration
- Hustiu et al. 2018** Hustiu, Sofia; Lupascu, Marian; Popescu, Stefan; Burlacu, Adrian; Kloetzer, Marius, 2018.
Stable hovering architecture for nanoquadcopter applications in indoor environments.
In: *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, S. 659–663.
ISBN 978-1-5386-4444-7
DOI: 10.1109/ICSTCC.2018.8540759

Hwang et al. 1992

Hwang, Yong K.; Ahuja, Narendra, 1992.
Gross motion planning-a survey.
ACM Computing Surveys (CSUR) **24** (3), S. 219–291.
DOI: 10.1145/136035.136037

Jarvis 1993

Jarvis, Ray, 1993.
Distance transform based path planning for robot navigation.
In: Zheng, Yuan F. (Hrsg.): *Recent Trends in Mobile Robots*.
Singapore: World Scientific Publishing Co. Pte. Ltd., S. 3–31.
ISBN 981-02-1511-8

Kallman et al. 2004

Kallman, Marcelo; Mataric, Maja, 2004.
Motion planning using dynamic roadmaps.
In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, S. 4399–4404.
ISBN 0-7803-8232-3
DOI: 10.1109/ROBOT.2004.1302410

Kant et al. 1988

Kant, Kamal; Zucker, Steven W., 1988.
Planning collision-free trajectories in time-varying environments: a two-level hierarchy.
The Visual Computer **3** (5), S. 304–313.
DOI: 10.1007/BF01914866

- Kanzow 2005** Kanzow, Christian, 2005.
Numerik linearer Gleichungssysteme: Direkte und iterative Verfahren.
Berlin, Heidelberg: Springer-Verlag.
Springer-Lehrbuch.
ISBN 978-3-540-26743-0
DOI: 10.1007/b138019
- Karaman et al. 2011** Karaman, Sertac; Frazzoli, Emilio, 2011.
Sampling-based algorithms for optimal motion planning.
The International Journal of Robotics Research **30** (7), S.
846–894.
DOI: 10.1177/0278364911406761
- Kavraki et al. 1996** Kavraki, Lydia E.; Svestka, Petr; Latombe, Jean-Claude;
Overmars, Mark H., 1996.
Probabilistic Roadmaps for Path Planning in High-
Dimensional Configuration Spaces.
IEEE Transactions on Robotics and Automation **12** (4), S.
566–580.
DOI: 10.1109/70.508439
- Kavraki et al. 2008** Kavraki, Lydia E.; LaValle, Steven M., 2008.
Motion planning.
In: Siciliano, Bruno; Khatib, Oussama (Hrsg.): *Springer Handbook of Robotics.*
Berlin, Heidelberg: Springer, S. 109–131.
DOI: 10.1007/978-3-540-30301-5_6

Khatib 1985

Khatib, Oussama, 1985.
Real-time obstacle avoidance for manipulators and mobile robots.
In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, S. 500–505.
ISBN 0-8186-0615-0
DOI: 10.1109/ROBOT.1985.1087247

Khatib et al. 1997

Khatib, Maher; Jaouni, Hazem; Chatila, Raja; Laumond, Jean-Paul, 1997.
Dynamic path modification for car-like nonholonomic mobile robots.
In: *Proceedings of International Conference on Robotics and Automation*, S. 2920–2925.
ISBN 0-7803-3612-7
DOI: 10.1109/ROBOT.1997.606730

Koditschek 1987

Koditschek, Daniel, 1987.
Exact robot navigation by means of potential functions: Some topological considerations.
In: *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, S. 1–6.
ISBN 0-8186-0787-4
DOI: 10.1109/ROBOT.1987.1088038

Koenig et al. 2002

Koenig, Sven; Likhachev, Maxim, 2002.
D* Lite.
In: Dechter, Rina; Kearns, Michael; Sutton, Rich (Hrsg.): *Eighteenth National Conference on Artificial Intelligence*, S. 476–483.
ISBN 978-0-262-51129-2

- Koren et al. 1991** Koren, Yoram; Borenstein, Johann, 1991.
Potential field methods and their inherent limitations for mobile robot navigation.
In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, S. 1398–1404.
ISBN 0-8186-2163-X
DOI: 10.1109/ROBOT.1991.131810
- Köther 2018** Köther, Reinhard, 2018.
Taschenbuch der Logistik.
5. Auflage.
München: Carl Hanser Verlag GmbH & Co. KG.
ISBN 978-3-446-45414-9
DOI: 10.3139/9783446457676
- Kühnel 2013** Kühnel, Wolfgang, 2013.
Differentialgeometrie: Kurven-Flächen-Mannigfaltigkeiten.
6. Auflage.
Wiesbaden: Springer Spektrum.
Aufbaukurs Mathematik.
ISBN 978-3-658-00615-0
DOI: 10.1007/978-3-658-00615-0
- Kunz et al. 2010** Kunz, Tobias; Reiser, Ulrich; Stilman, Mike; Verl, Alexander, 2010.
Real-time path planning for a robot arm in changing environments.
In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 5906–5911.
ISBN 978-1-4244-6674-0
DOI: 10.1109/IROS.2010.5653275

- Kurak et al. 2018** Kurak, Sevkuthan; Hodzic, Migdat, 2018.
Control and estimation of a quadcopter dynamical model.
Periodicals of Engineering and Natural Sciences **6** (1), S.
63–75.
DOI: 10.21533/pen.v6i1.164
- Kuwata et al. 2009** Kuwata, Yoshiaki; Teo, Justin; Fiore, Gaston; Karaman,
Sertac; Frazzoli, Emilio; How, Jonathan P., 2009.
Real-Time Motion Planning With Applications to Autono-
mous Urban Driving.
IEEE Transactions on Control Systems Technology **17** (5),
S. 1105–1118.
DOI: 10.1109/TCST.2008.2012116
- Landrock et al. 2018** Landrock, Holm; Baumgärtel, Anne, 2018.
*Die Industriedrohne-der fliegende Roboter: Professionelle
Drohnen und ihre Anwendung in der Industrie 4.0.*
Wiesbaden: Springer Vieweg.
ISBN 978-3-658-21354-1
DOI: 10.1007/978-3-658-21355-8
- Latombe 1991** Latombe, Jean-Claude, 1991.
Robot Motion Planning.
Boston: Springer.
The Springer International Series in Engineering and
Computer Science 124.
ISBN 978-0-7923-9206-4
DOI: 10.1007/978-1-4615-4022-9

- Laumond et al. 1998** Laumond, Jean-Paul; Sekhavat, Sepanta; Lamiroux, Florent, 1998.
Guidelines in nonholonomic motion planning for mobile robots.
In: Laumond, Jean-Paul (Hrsg.): *Robot motion planning and control*.
Berlin, Heidelberg: Springer, S. 1–53.
ISBN 978-3-540-76219-5
DOI: 10.1007/BFb0036070
- LaValle 2006** LaValle, Steven M., 2006.
Planning Algorithms.
New York: Cambridge university press.
ISBN 978-0-5218-6205-9
- LaValle et al. 2001** LaValle, Steven M.; Kuffner Jr., James J., 2001.
Randomized Kinodynamic Planning.
The International Journal of Robotics Research **20** (5), S. 378–400.
DOI: 10.1177/02783640122067453
- Leven et al. 2002** Leven, Peter; Hutchinson, Seth, 2002.
A Framework for Real-time Path Planning in Changing Environments.
The International Journal of Robotics Research **21** (12), S. 999–1030.
DOI: 10.1177/0278364902021012001
- Likhachev et al. 2003** Likhachev, Maxim; Gordon, Geoffrey J.; Thrun, Sebastian, 2003.
ARA: Formal Analysis*.
Report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Lingner 2016

Lingner, Margrit, 2016.

Wenn der Lieferroboter drei Mal klingelt: Lieferroboter und Paketdrohnen.

Verfügbar unter: <https://www.it-business.de/wenn-der-lieferroboter-drei-mal-klingelt-a-544855/>

Zugriff am: 20.05.2021

Lozano-Perez 1990

Lozano-Perez, Tomas, 1990.

Spatial planning: A configuration space approach.

In: Cox, Ingemar J; Wilfong, Gordon T. (Hrsg.): *Autonomous Robot Vehicles.*

New York: Springer, S. 259–271.

ISBN 978-0-387-97240-4

DOI: 10.1007/978-1-4613-8997-2_20

Luis 2016

Luis, Carlos, 2016.

Design of a Trajectory Tracking Controller for a Nanoquadcopter.

Report, Electrical Engineering Department, École Polytechnique de Montréal

Verfügbar unter: <https://arxiv.org/pdf/1608.05786>

Lunze 2014

Lunze, Jan, 2014.

Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung.

8. Auflage.

Berlin, Heidelberg: Springer Vieweg.

Springer-Lehrbuch.

ISBN 978-3-642-53943-5

DOI: 10.1007/978-3-642-53944-2

- Mantegazza et al. 2018** Mantegazza, Carlo; Novaga, Matteo; Pluda, Alessandra; Schulze, Felix, 2018.
Evolution of networks with multiple junctions
Verfügbar unter: <https://arxiv.org/abs/1611.08254>
- Manz et al. 1993** Manz, Allan; Liscano, Ramiro; Green, David, 1993.
A comparison of realtime obstacle avoidance methods for mobile robots.
In: Chatila, Raja; Hirzinger, Gerd (Hrsg.): *Experimental Robotics II*, S. 299–316.
ISBN 978-3-540-19851-2
DOI: 10.1007/BFb0036147
- McCauley 2016** McCauley, Dana, 2016.
Domino's unveils drone delivery trial in New Zealand.
Verfügbar unter:
<https://www.news.com.au/finance/business/retail/dominos-unveils-drone-delivery-trial-in-new-zealand/news-story/ea2d5e92d90fc904e0ceb6e73f186cab>
Zugriff am: 20.05.2021
- McMahon 1986** McMahon, Frank H., 1986.
The Livermore Fortran Kernels: A computer test of the numerical performance range.
Report, Lawrence Livermore National Lab., CA, USA
- Minguez et al. 2004** Minguez, Javier; Montano, Luis, 2004.
Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios.
IEEE Transactions on Robotics and Automation **20** (1), S. 45–59.
DOI: 10.1109/TRA.2003.820849

- Mitrinovic et al. 1970** Mitrinovic, Dragoslav S.; Vasic, Petar M., 1970.
Analytic inequalities.
Berlin, Heidelberg: Springer-Verlag.
Grundlehren der mathematischen Wissenschaften 165.
ISBN 978-3-642-99972-7
DOI: 10.1007/978-3-642-99970-3
- Mokhtarian et al. 1992** Mokhtarian, Farzin; Mackworth, Alan K., 1992.
A theory of multiscale, curvature-based shape representation for planar curves.
IEEE Transactions on Pattern Analysis & Machine Intelligence (8), S. 789–805.
DOI: 10.1109/34.149591
- Mullins 1956** Mullins, William W., 1956.
Two-Dimensional Motion of Idealized Grain Boundaries.
Journal of Applied Physics **27** (8), S. 900–904.
DOI: 10.1063/1.1722511
- Naderi et al. 2015** Naderi, Kouros; Rajamäki, Joose; Hämäläinen, Perttu, 2015.
RT-RRT*: a real-time path planning algorithm based on RRT*.
In: *MIG '15: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, S. 113–118.
ISBN 978-1-4503-3991-9
DOI: 10.1145/2822013.2822036

- Noschese et al. 2013** Noschese, Silvia; Pasquini, Lionello; Reichel, Lothar, 2013.
Tridiagonal Toeplitz matrices: properties and novel applications.
Numerical Linear Algebra with Applications **20** (2), S. 302–326.
DOI: 10.1002/nla.1811
- O’Dunlaing et al. 1985** O’Dunlaing, Colm; Yap, Chee K., 1985.
A “retraction” method for planning the motion of a disc.
Journal of Algorithms **6** (1), S. 104–111.
DOI: 10.1016/0196-6774(85)90021-5
- Otte et al. 2015** Otte, Michael; Frazzoli, Emilio, 2015.
RRT X: Real-Time Motion Planning/Replanning for Environments with Unpredictable Obstacles.
In: Akin, H. L; Amato, Nancy M; Isler, Volkan; van der Stappen, A. F. (Hrsg.): *Algorithmic Foundations of Robotics XI*.
Basel: Springer International Publishing, S. 461–478.
ISBN 978-3-319-16594-3
- Otte et al. 2016** Otte, Michael; Frazzoli, Emilio, 2016.
RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning.
The International Journal of Robotics Research **35** (7), S. 797–822.
DOI: 10.1177/0278364915594679

- Ottosson 2017** Ottosson, Erik, 2017.
Identification and Estimation of DC-Motor Dynamics for the Crazyflie Quadcopter.
Lund, Univ., Masterarbeit, 2017
Verfügbar unter: <http://lup.lub.lu.se/student-papers/record/8928602>
- Petti et al. 2005** Petti, Stephane; Fraichard, Thierry, 2005.
Safe motion planning in dynamic environments.
In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 2210–2215.
ISBN 0-7803-8912-3
DOI: 10.1109/IROS.2005.1545549
- Philippsen 2004** Philippsen, Roland, 2004.
Motion planning and obstacle avoidance for mobile robots in highly cluttered dynamic environments.
Lausanne, Univ., Diss., 2004
DOI: 10.5075/epfl-thesis-3146
- Phillips et al. 2011** Phillips, Mike; Likhachev, Maxim, 2011.
Sipp: Safe interval path planning for dynamic environments.
In: *2011 IEEE International Conference on Robotics and Automation*, S. 5628–5635.
ISBN 978-1-61284-386-5
DOI: 10.1109/ICRA.2011.5980306

- Pin et al. 1994** Pin, Francois G.; Killough, Stephen M., 1994.
A new family of omnidirectional and holonomic wheeled platforms for mobile robots.
IEEE Transactions on Robotics and Automation **10** (4), S. 480–489.
DOI: 10.1109/70.313098
- Plato 2010** Plato, Robert, 2010.
Numerische Mathematik kompakt.
4. Auflage.
Wiesbaden: Vieweg+Teubner Verlag.
ISBN 978-3-8348-1018-2
DOI: 10.1007/978-3-8348-9644-5
- Pomarlan et al. 2013** Pomarlan, Mihai; Sucan, Ioan A., 2013.
Motion planning for manipulators in dynamically changing environments using real-time mapping of free workspace.
In: *2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, S. 483–487.
ISBN 978-1-4799-0194-4
DOI: 10.1109/CINTI.2013.6705245
- Portugal 2010** Portugal, David Bina Siassipour, 2010.
A study on local planning techniques for mobile robot navigation.
Report, Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Coimbra

- Pottmann et al. 2007** Pottmann, Helmut; Asperl, Andreas; Hofer, Michael; Kilian, Axel, 2007.
Architectural Geometry.
Exton: Bentley Institute Press.
ISBN 978-1-9344-9304-5
- Quinlan 1994** Quinlan, Sean, 1994.
Real-time modification of collision-free paths.
Standord, Univ., Diss., 1994
- Quinlan et al. 1993** Quinlan, Sean; Khatib, Oussama, 1993.
Elastic bands: Connecting path planning and control.
In: [1993] *Proceedings IEEE International Conference on Robotics and Automation*, S. 802–807.
ISBN 0-8186-3450-2
DOI: 10.1109/ROBOT.1993.291936
- Radmanesh et al. 2018** Radmanesh, Mohammadreza; Kumar, Manish; Guentert, Paul H.; Sarim, Mohammad, 2018.
Overview of Path-Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study.
Unmanned Systems **6** (2), S. 95–118.
DOI: 10.1142/S2301385018400022
- Rastgoo et al. 2014** Rastgoo, Mohammad Naim; Nakisa, Bahareh; Nasrudin, Mohammad Faidzul; Nazri, Mohd Zakree Ahmed, 2014.
A Critical Evaluation of Literature on Robot Path Planning in Dynamic Environment.
Journal of Theoretical and Applied Information Technology **70** (1), S. 177–185.

- Reeds et al. 1990** Reeds, James; Shepp, Lawrence, 1990.
Optimal paths for a car that goes both forwards and backwards.
Pacific Journal of Mathematics **145** (2), S. 367–393.
- Röck et al. 2010** Röck, Sascha; Hoher, Simon; Huptych, Marcel, 2010.
Simulationsbasierte Methoden für die Automatisierungstechnik in der Produktion.
In: *Stuttgarter Impulse - Fertigungstechnik für die Zukunft; Schriftliche Fassung der Vorträge zum Fertigungstechnischen Kolloquium am 29 und 30. September 2010 in Stuttgart; FTK 2010*, S. 159–188.
ISBN 978-3-925589-51-5
- Rösmann et al. 2015** Rösmann, Christoph; Hoffmann, Frank; Bertram, Torsten, 2015.
Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control.
In: *2015 European Control Conference (ECC)*, S. 3352–3357.
ISBN 978-3-9524-2693-7
DOI: 10.1109/ECC.2015.7331052
- Safadi 2007** Safadi, Hani, 2007.
Local Path Planning Using Virtual Potential Field.
Report, School of Computer Science, McGill University
- Saravanakumar et al. 2013** Saravanakumar, Subramanian; Asokan, Thondiyath, 2013.
Multipoint potential field method for path planning of autonomous underwater vehicles in 3D space.
Intelligent Service Robotics **6** (4), S. 211–224.
DOI: 10.1007/s11370-013-0138-2

- Schnierle et al. 2018** Schnierle, Marc; Röck, Sascha, 2018.
Plattform für die Mixed-Reality-in-the-Loop-Simulation –
Ein Beitrag zur Mixed-Reality-in-the-Loop-Simulation als
Erweiterung der X-in-the-Loop-Methodenreihe.
wt Werkstattstechnik online **108** (9), S. 623–630.
DOI: 10.37544/1436-4980-2018-09-59
- Schnierle et al. 2019** Schnierle, Marc; Polak, Christopher; Röck, Sascha, 2019.
Mensch-Roboter-Interaktion mit Mixed Reality auf Basis
einer Digital-Twin-as-a-Service-Plattform.
atp magazin **61** (5), S. 84–91.
DOI: 10.17560/atp.v61i5.2415
- Shiller 2015** Shiller, Zvi, 2015.
Off-line and on-line trajectory planning.
In: Carbone, Giuseppe; Gomez-Bravo, Fernando (Hrsg.):
Motion and Operation Planning of Robotic Systems.
Cham: Springer, S. 29–62.
ISBN 978-3-319-14704-8
DOI: 10.1007/978-3-319-14705-5_2
- Shkolnik et al. 2009** Shkolnik, Alexander; Walter, Matthew; Tedrake, Russ,
2009.
Reachability-guided sampling for planning under differen-
tial constraints.
In: *2009 IEEE International Conference on Robotics and
Automation*, S. 2859–2865.
ISBN 978-1-4244-2788-8
DOI: 10.1109/ROBOT.2009.5152874

- Short et al. 2016** Short, Andrew; Pan, Zengxi; Larkin, Nathan; van Duin, Stephen, 2016.
Recent progress on sampling based dynamic motion planning algorithms.
In: *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, S. 1305–1311.
ISBN 978-1-5090-2065-2
DOI: 10.1109/AIM.2016.7576950
- Siciliano et al. 2009** Siciliano, Bruno; Sciavicco, Lorenzo; Villani, Luigi; Oriolo, Giuseppe, 2009.
Robotics: Modelling, Planning and Control.
London: Springer-Verlag.
Advanced Textbooks in Control and Signal Processing.
ISBN 978-1-84628-641-4
DOI: 10.1007/978-1-84628-642-1
- Smith et al. 2007** Smith, Stephen L.; Broucke, Mireille E.; Francis, Bruce A., 2007.
Curve shortening and the rendezvous problem for mobile autonomous robots.
IEEE Transactions on Automatic Control **52** (6), S. 1154–1159.
DOI: 10.1109/TAC.2007.899024
- Stichweh 2015** Stichweh, Heiko, 2015.
Aktorik für Industrie 4.0.
In: Tompel, Michael ten; Vogel-Heuser, Birgit; Bauernhansl, Thomas (Hrsg.): *Handbuch Industrie 4.0*.
Berlin, Heidelberg: Springer Vieweg, S. 1–21.
ISBN 978-3-662-45537-1
DOI: 10.1007/978-3-662-45537-1_7-1

Stoller 2014

Stoller, Detlef, 2014.

DHL fliegt Medikamente mit Drohne auf die Nordseeinsel Juist: Weltpremiere für Paketkopter.

Verfügbar unter:

<https://www.ingenieur.de/technik/fachbereiche/verkehr/dhl-fliegt-medikamente-drohne-nordseeinsel-juist/>

Zugriff am: 27.05.2021

Thesseling et al. 2015

Thesseling, Matthias; Kimmerle, David; Boberg, Christopher; Stichweh, Heiko, 2015.

A novel concept for distributed low-power-drives in multidirectional conveyor matrix.

In: *Proceedings of PCIM Europe 2015; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, S. 1–8.

ISBN 978-3-8007-3924-0

Thomas 1949

Thomas, Llewellyn Hilleth, 1949.

Elliptic Problems in Linear Difference Equations over a Network.

Watson Sci. Comput. Lab. Rept., Columbia University, New York 1

Tsourdos et al. 2011

Tsourdos, Antonios; White, Brian; Shanmugavel, Madhavan, 2011.

Cooperative Path Planning of Unmanned Aerial Vehicles.

West Sussex: John Wiley & Sons, Ltd.

ISBN 978-0-470-74129-0

DOI: 10.1002/9780470974636

- Tusseyeva et al. 2013** Tusseyeva, Inara; Kim, Seong-Gon; Kim, Yong-Gi, 2013. 3D Global Dynamic Window Approach for Navigation of Autonomous Underwater Vehicles. *International Journal of Fuzzy Logic and Intelligent Systems* **13** (2), S. 91–99.
DOI: 10.5391/IJFIS.2013.13.2.91
- Ulrich et al. 1998** Ulrich, Iwan; Borenstein, Johann, 1998. VFH+: Reliable obstacle avoidance for fast mobile robots. In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, S. 1572–1577.
ISBN 0-7803-4300-X
DOI: 10.1109/ROBOT.1998.677362
- Ulrich et al. 2000** Ulrich, Iwan; Borenstein, Johann, 2000. VFH*: Local obstacle avoidance with look-ahead verification. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, S. 2505–2511.
ISBN 0-7803-5886-4
DOI: 10.1109/ROBOT.2000.846405
- van den Berg 2007** van den Berg, Jur Pieter, 2007. *Path Planning in Dynamic Environments*. Utrecht, Univ., Diss., 2007.
ISBN 978-90-393-4480-4

Varadhan et al. 2004

Varadhan, Gokul; Manocha, Dinesh, 2004.
Accurate Minkowski sum approximation of polyhedral models.
In: *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, S. 392–401.
ISBN 0-7695-2234-3
DOI: 10.1109/PCCGA.2004.1348370

Wang et al. 2002

Wang, Lim Chee; Yong, Lim Ser; Ang, Marcelo H., 2002.
Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment.
In: *Proceedings of the IEEE International Symposium on Intelligent Control*, S. 821–826.
ISBN 0-7803-7620-X
DOI: 10.1109/ISIC.2002.1157868

Weber 2019

Weber, Wolfgang, 2019.
Industrieroboter: Methoden der Steuerung und Regelung.
4. Auflage.
München: Carl Hanser Verlag GmbH Co. KG.
ISBN 978-3-446-45952-6
DOI: 10.3139/9783446460607.fm

Werner 2011

Werner, Dirk, 2011.
Funktionalanalysis.
7. Auflage.
Berlin, Heidelberg: Springer-Verlag.
Springer-Lehrbuch.
ISBN 978-3-642-21017-4
DOI: 10.1007/978-3-642-21017-4

- Westkämper et al. 2008** Westkämper, Engelbert; Zahn, Erich, 2008.
Wandlungsfähige Produktionsunternehmen: Das Stuttgarter Unternehmensmodell.
Berlin, Heidelberg: Springer.
ISBN 978-3-540-21889-0
DOI: 10.1007/978-3-540-68890-7
- Wise et al. 2000** Wise, Kevin D.; Bowyer, Adrian, 2000.
A survey of global configuration-space mapping techniques for a single robot in a static environment.
The International Journal of Robotics Research **19** (8), S. 762–779.
DOI: 10.1177/02783640022067157
- Yang et al. 2010** Yang, Yuandong; Brock, Oliver, 2010.
Elastic roadmaps—motion generation for autonomous mobile manipulation.
Autonomous Robots **28** (1), S. 113–130.
DOI: 10.1007/s10514-009-9151-x
- Yang et al. 2015** Yang, Liang; Qi, Juntong; Xiao, Jizhong; Yong, Xia, 2015.
A literature review of UAV 3D path planning.
In: *Proceeding of the 11th World Congress on Intelligent Control and Automation*, S. 2376–2381.
ISBN 978-1-4799-5825-2
DOI: 10.1109/WCICA.2014.7053093
- Zohaib et al. 2013** Zohaib, Muhammad; Pasha, Mustafa; Riaz, Raja Ali; Javaid, Nadeem; Ilahi, Manzoor; Khan, Rahim Dad, 2013.
Control strategies for mobile robot with obstacle avoidance.
Journal of Basic Sciences and Applied Research **3** (4), S. 1027–1036.

9 Anhang

9.1 Pseudocode für die Berechnung des Parameters K als inverses Problem

Code 4 zeigt die Lösung des in Kap. 4.9 beschriebenen nichtlinearen inversen Problems zur Bestimmung des Sicherheitsparameters K mit dem Newton-Raphson-Verfahren (Plato 2010) in an MATLAB angelehntem Pseudocode. Eine Newtoniteration wird dabei durch das LGS $J(\mathbf{y}_F) \cdot \Delta \mathbf{y}_F = -\mathbf{F}(\mathbf{y}_F)$ und einer anschließenden Aktualisierung der Näherungslösung $\mathbf{y}_F = \mathbf{y}_F + \Delta \mathbf{y}_F$ beschrieben. Dabei ist \mathbf{y}_F der Vektor der Unbekannten, \mathbf{F} das zu lösende Nullstellenproblem und $J = \partial \mathbf{F} / \partial \mathbf{y}_F$ die mehrdimensionale Ableitung des Nullstellenproblems nach den Unbekannten (Jacobimatrix).

Die in Code 4 verwendeten Funktionen `ones()`, `norm()`, `length()`, `zeros()`, `size()`, `sum()`, `min()`, `max()` und `sqrt()` geben dabei eine Einismatrix, die 2-Norm, die Vektorenlänge, eine Nullmatrix, die Matrixdimension, die Vektorsumme, den Minimal- und Maximalwert eines Vektors sowie die Quadratwurzel zurück.

001	<code>n=10; %Stützstellenanzahl</code>
002	<code>y0=0.5*ones(n,1); %Startwerte setzen</code>
003	<code>K0=1.0;</code>
004	<code>yF=[y0;K0];</code>
005	<code>while norm(F(yF))>1e-5 %Newtoniteration bis Residuum err.</code>
006	<code> Delta yF=J(yF)\-F(yF); %Lineares GLS lösen</code>
007	<code> yF=yF+Delta yF; %Näherungslösung iterieren</code>
008	<code>end</code>
009	<code>K=yF(n+1); %K ausgeben</code>
010	<code>%FUNKTION J (Numerische Jacobimatrix)</code>
011	<code>function rJ=J(yF)</code>
012	<code> epsilon=1e-10; %Störschrittweite</code>
013	<code> for i=1:length(yF) %Spaltenweise differenzieren</code>
014	<code> yF(i)=yF(i)+epsilon; %Positiver Störschritt</code>
015	<code> Fp=F(yF);</code>
016	<code> yF(i)=yF(i)-2*epsilon; %Negativer Störschritt</code>
017	<code> Fm=F(yF);</code>
018	<code> yF(i)=yF(i)+epsilon; %Störung zurücksetzen</code>
019	<code> rJ(:,i)=(Fp-Fm)/(2*epsilon); %Zentraler DQ</code>
020	<code> end</code>

021	end
022	%FUNKTION F (Formulierung Nullstellenproblem)
023	function rF= F (y_F)
024	r ₀ =1.0; %Wirkweite
025	rF=zeros(size(y_F));
026	n=length(y_F)-1;
027	K= y_F (n+1);
028	for i=1:n
029	if i>1
030	rF(i)=rF(i)+ y_F (i-1);
031	end
032	rF(i)=rF(i)+ y_F (i)*(-2-K)+K*r ₀ ;
033	if i<n
034	rF(i)=rF(i)+ y_F (i+1);
035	end
036	end
037	R=1; %Randbedingung
038	switch R
039	case 1 %Durchschnitt
040	y _{0,soll} =r ₀ *0.5;
041	rF(n+1)=sum(y_F (1:n))/n-y _{0,soll} ;
042	case 2 %Minimum
043	y _{min,soll} =r ₀ *0.3;
044	rF(n+1)=min(y_F (1:n))-y _{min,soll} ;
045	case 3 %Maximum
046	y _{max,soll} =r ₀ *0.95;
047	rF(n+1)=max(y_F (1:n))-y _{max,soll} ;
048	case 4 %Länge
049	L _{soll} =L ₀ *2;
050	dL=L ₀ /(n+1);
051	rF(n+1)=sqrt(dL^2+ y_F (1)^2);
052	for i=2:n
053	rF(n+1)=rF(n+1)+...
054	sqrt(dL^2+(y_F (i)- y_F (i-1))^2);
055	end
056	rF(n+1)=rF(n+1)+sqrt(dL^2+ y_F (n)^2);
057	rF(n+1)=rF(n+1)-L _{soll} ;
058	end
059	end

Code 4: Berechnung von K als inverses Problem mit dem Newton-Verfahren

9.2 Pseudocode KFM

Nachfolgend ist in Code 5 die KFM für ein ebenes Problem ($D = 2$) in an MATLAB angelehntem Pseudocode dargestellt. Die Parameter n , T , K und Δt der KFM (Zeilen 003 bis 006), die Start- und Zielkoordinaten x_s , y_s , x_g und y_g (Zeilen 008 bis 011), die Initialbelegung der Zustandsvektoren bzw. Stützstellen x und y (Zeilen 041 und 042) sowie die Berechnung der Abstoßungskräfte f_x und f_y auf Basis eines Umgebungsmodells (Zeilen 046 und 047) sind vom Anwender problemspezifisch festzulegen bzw. zu implementieren.

Für eine Erweiterung der KFM auf ein räumliches oder gar höherdimensionales Problem ($D \geq 3$) müssen die Anweisungen nach den Zeilen 011, 025, 042, 047, 050, 053, 056, 060 und 063 äquivalent für die weiteren Dimensionen fortgeführt werden. Die in Code 5 zusätzlich verwendete Funktion `eye()` gibt eine Einheitsmatrix zurück.

001	%INITIALISIERUNG
002	%Parameter
003	$n = \dots;$ %Anzahl Stützstellen
004	$T = \dots;$ %Dynamikparameter
005	$K = \dots;$ %Sicherheitsparameter
006	$\Delta t = \dots;$ %Zeitschrittweite
007	%Randwerte
008	$x_s = \dots;$
009	$y_s = \dots;$
010	$x_g = \dots;$
011	$y_g = \dots;$
012	%Einheitsmatrix
013	$I = \text{eye}(n, n);$
014	%Systemmatrix
015	$A = \text{zeros}(n, n);$
016	$A(1, 1:2) = [-2, 1];$
017	for $i = 2:n-1$
018	$A(i, i) = -2;$
019	$A(i, i-1) = 1;$
020	$A(i, i+1) = 1;$
021	end
022	$A(n, n-1:n) = [1, -2];$
023	%Randvektoren
024	$b_x = [x_s; \text{zeros}(n-2, 1); x_g];$

025	$\mathbf{b}_y = [y_s; \text{zeros}(n-2, 1); y_g];$
026	<code>%Koeffizientenmatrix</code>
027	$\mathbf{C} = \mathbf{I} - \Delta t / T * \mathbf{A};$
028	<code>%LU-Zerlegung</code>
029	$\mathbf{L} = \text{eye}(n, n);$
030	$\mathbf{U} = \text{zeros}(n, n);$
031	$\mathbf{U}(1, 1) = \mathbf{C}(1, 1);$
032	$\mathbf{U}(1, 2) = \mathbf{C}(2, 1);$
033	<code>for i=2:n-1</code>
034	$\mathbf{L}(i, i-1) = \mathbf{C}(i, i-1) / \mathbf{U}(i-1, i-1);$
035	$\mathbf{U}(i, i) = \mathbf{C}(i, i) - \mathbf{L}(i, i-1) * \mathbf{U}(i-1, i);$
036	$\mathbf{U}(i, i+1) = \mathbf{C}(i+1, i);$
037	<code>end</code>
038	$\mathbf{L}(n, n-1) = \mathbf{C}(n, n-1) / \mathbf{U}(n-1, n-1);$
039	$\mathbf{U}(n, n) = \mathbf{C}(n, n) - \mathbf{L}(n, n-1) * \mathbf{U}(n-1, n);$
040	<code>%Stützstellen</code>
041	$\mathbf{x} = \dots;$
042	$\mathbf{y} = \dots;$
043	<code>%SCHLEIFE</code>
044	<code>while (1)</code>
045	<code> %Kraftvektoren</code>
046	$\mathbf{f}_x = \dots;$
047	$\mathbf{f}_y = \dots;$
048	<code> %Eingangsvektoren</code>
049	$\mathbf{r}_x = \mathbf{x} + \Delta t / T * (\mathbf{b}_x + \mathbf{K} * \mathbf{f}_x);$
050	$\mathbf{r}_y = \mathbf{y} + \Delta t / T * (\mathbf{b}_y + \mathbf{K} * \mathbf{f}_y);$
051	<code> %Vorwärtssubstitution</code>
052	$\hat{\mathbf{x}}(1) = \mathbf{r}_x(1);$
053	$\hat{\mathbf{y}}(1) = \mathbf{r}_y(1);$
054	<code> for i=2:n</code>
055	$\hat{\mathbf{x}}(i) = \mathbf{r}_x(i) - \mathbf{L}(i, i-1) * \hat{\mathbf{x}}(i-1);$
056	$\hat{\mathbf{y}}(i) = \mathbf{r}_y(i) - \mathbf{L}(i, i-1) * \hat{\mathbf{y}}(i-1);$
057	<code> end</code>
058	<code> %Rückwärtssubstitution</code>
059	$\mathbf{x}(n) = \hat{\mathbf{x}}(n) / \mathbf{U}(n, n);$
060	$\mathbf{y}(n) = \hat{\mathbf{y}}(n) / \mathbf{U}(n, n);$
061	<code> for i=n-1:-1:1</code>
062	$\mathbf{x}(i) = (\hat{\mathbf{x}}(i) - \mathbf{U}(i, i+1) * \mathbf{x}(i+1)) / \mathbf{U}(i, i);$
063	$\mathbf{y}(i) = (\hat{\mathbf{y}}(i) - \mathbf{U}(i, i+1) * \mathbf{y}(i+1)) / \mathbf{U}(i, i);$
064	<code> end</code>
065	<code>end</code>

Code 5: KFM

Es sei angemerkt, dass zur Optimierung des Speicherbedarfs sowie der Zugriffszeiten bei Anwendung des Thomas-Algorithmus (Thomas 1949) typischerweise die lediglich dünn besetzten Matrizen L und U diaogonalenweise in Vektoren bzw. eindimensionalen Arrays gespeichert werden. Für eine bessere Lesbarkeit wurden sie in Code 5 dennoch als Matrizen dargestellt.

9.3 Pseudocode Abstoßungskraft

Code 6 zeigt die in dieser Arbeit verwendete Berechnung der räumlichen Abstoßungskraft f in an MATLAB angelehntem Pseudocode. Die hier zusätzlich verwendeten Funktionen `abs()`, `dot()`, `sign()`, und `size()` liefern dabei den Betrag, das Skalarprodukt, das Vorzeichen sowie die Dimension des jeweiligen Arguments bzw. der jeweiligen Argumente zurück.

001	%FUNKTION f
002	<code>function rf=f(p, h, ε, r₀, r_u, p)</code>
003	<code>rf=zeros(size(p));</code>
004	<code>n=size(p, 2);</code>
005	<code>for i=1:n</code>
006	<code> p_i=p(:, i);</code>
007	<code> %Gradientenberechnung mit Vorwärts-DQ</code>
008	<code> rho0=ρ(p_i, h, r₀, r_u, p);</code>
009	<code> rhox=ρ(p_i+ [ε; 0; 0], h, r₀, r_u, p);</code>
010	<code> rhoy=ρ(p_i+ [0; ε; 0], h, r₀, r_u, p);</code>
011	<code> rhoz=ρ(p_i+ [0; 0; ε], h, r₀, r_u, p);</code>
012	<code> grd=(1/ε)*[(rho0-rhox); (rho0-rhoy); (rho0-rhoz)];</code>
013	<code> ngrd=norm(grd);</code>
014	<code> if ngrd>0</code>
015	<code> grd=grd/ngrd;</code>
016	<code> else</code>
017	<code> grd=[0; 0; 0];</code>
018	<code> end</code>
019	<code> rf(:, i)=rho0*grd;</code>
020	<code>end</code>
021	<code>end</code>
022	%FUNKTION ρ
023	<code>function rrho=ρ(p_i, h, r₀, r_u, p)</code>
024	<code>s=(r_u-r₀)/(π/2-1);</code>

025	$r_l = r_0 - s;$
026	$rH = H(\mathbf{p}_i, \mathbf{h}, p);$
027	if $rH \leq r_l$ %Phase 1
028	$rrho = r_0 - rH;$
029	elseif $rH \geq r_u$ %Phase 3
030	$rrho = 0;$
031	else %Phase 2
032	$rrho = s * (1 - \sin((rH - r_l) / s));$
033	end
034	end
035	%FUNKTION H
036	function $rH = H(\mathbf{p}_i, \mathbf{h}, p)$
037	$rH = 0;$
038	$m = \text{size}(\mathbf{h}, 2);$ %Anzahl Hindernisse
039	for $j = 1:m$
040	$\mathbf{h}_j = \mathbf{h}(:, j);$
041	$d_j = d(\mathbf{p}_i, \mathbf{h}_j);$
042	$rH = rH + \text{sign}(d_j) * \text{abs}(d_j)^p;$
043	end
044	$rH = \text{sign}(rH) * \text{abs}(rH)^{1/p};$
045	end
046	%FUNKTION d
047	function $rd = d(\mathbf{p}_i, \mathbf{h}_j)$
048	$\text{type} = \mathbf{h}_j(1);$ %Hindernistyp
049	switch type
050	case 1 %Kugel
051	$\mathbf{m}_0 = \mathbf{h}_j(2:4);$
052	$r = \mathbf{h}_j(5);$
053	$rd = d_{sph}(\mathbf{p}_i, \mathbf{m}_0, r);$
054	case 2 %Zylinder
055	$\mathbf{m}_0 = \mathbf{h}_j(2:4);$
056	$\mathbf{m}_1 = \mathbf{h}_j(5:7);$
057	$r = \mathbf{h}_j(8);$
058	$rd = d_{cyl}(\mathbf{p}_i, \mathbf{m}_0, \mathbf{m}_1, r);$
059	case 3 %Ebene
060	$\mathbf{m}_0 = \mathbf{h}_j(2:4);$
061	$\mathbf{n} = \mathbf{h}_j(5:7);$
062	$rd = d_{pln}(\mathbf{p}_i, \mathbf{m}_0, \mathbf{n});$
063	end
064	end
065	%FUNKTION d_{sph}

066	<code>function rdsph=$d_{sph}(\mathbf{p}_i, \mathbf{m}_0, r)$</code>
067	<code> rdsph=norm($\mathbf{p}_i - \mathbf{m}_0$) - r;</code>
068	<code>end</code>
069	<code>%FUNKTION d_{cyl}</code>
070	<code>function rdcyl=$d_{cyl}(\mathbf{p}_i, \mathbf{m}_0, \mathbf{m}_1, r)$</code>
071	<code> v=$\mathbf{m}_1 - \mathbf{m}_0$;</code>
072	<code> w=$\mathbf{p}_i - \mathbf{m}_0$;</code>
073	<code> c1=dot(w, v);</code>
074	<code> if (c1<=0)</code>
075	<code> rdcyl=norm($\mathbf{p}_i - \mathbf{m}_0$) - r;</code>
076	<code> return</code>
077	<code> end</code>
078	<code> c2=dot(v, v);</code>
079	<code> if (c2<=c1)</code>
080	<code> rdcyl=norm($\mathbf{p}_i - \mathbf{m}_1$) - r;</code>
081	<code> return</code>
082	<code> end</code>
083	<code> b=c1/c2;</code>
084	<code> pb=$\mathbf{m}_0 + b * \mathbf{v}$;</code>
085	<code> rdcyl=norm($\mathbf{p}_i - \text{pb}$) - r;</code>
086	<code>end</code>
087	<code>%FUNKTION d_{pln}</code>
088	<code>function rdpln=$d_{pln}(\mathbf{p}_i, \mathbf{m}_0, \mathbf{n})$</code>
089	<code> sn=-dot($\mathbf{n}, \mathbf{p}_i - \mathbf{m}_0$);</code>
090	<code> sd=dot(\mathbf{n}, \mathbf{n});</code>
091	<code> sb=sn/sd;</code>
092	<code> b=$\mathbf{p}_i + sb * \mathbf{n}$;</code>
093	<code> rdpln=norm($\mathbf{p}_i - \text{b}$);</code>
094	<code> if norm($\mathbf{p}_i - \mathbf{m}_0 + \mathbf{n}$) < norm($\mathbf{p}_i - \mathbf{m}_0 - \mathbf{n}$)</code>
095	<code> rdpln=-rdpln;</code>
096	<code> end</code>
097	<code>end</code>

Code 6: Berechnung der Abstoßungskraft \mathbf{f} der KFM

9.4 Hindernisdaten

Nachfolgend sind in Gl. (51) die Daten zum in Kap. 5.2 verwendeten Hindernis aufgeführt.

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}_1 = \mathbf{h}_{sph} = (\mathbf{m}_0 = (1, 0.4, 0.2)^T, r = 0.075) \\ \mathbf{h}_2 = \mathbf{h}_{sph} = (\mathbf{m}_0 = (1.6, 0.35, 0.2)^T, r = 0.075) \\ \mathbf{h}_3 = \mathbf{h}_{sph} = (\mathbf{m}_0 = (0.5, 0.25, 0.4)^T, r = 0.075) \\ \mathbf{h}_4 = \mathbf{h}_{cyl} = (\mathbf{m}_0 = (1, 0.75, 0)^T, \mathbf{m}_1 = (1, 0.75, 0.75)^T, r = 0.075) \\ \mathbf{h}_5 = \mathbf{h}_{cyl} = (\mathbf{m}_0 = (1, 0.75, 0.75)^T, \mathbf{m}_1 = (1.2, 0.5, 0.75)^T, r = 0.075) \\ \mathbf{h}_6 = \mathbf{h}_{pln} = (\mathbf{m}_0 = (2, 0.5, 0.75)^T, \mathbf{n} = (-1, 0, 0)^T) \\ \mathbf{h}_7 = \mathbf{h}_{pln} = (\mathbf{m}_0 = (1, 1, 0.5)^T, \mathbf{n} = (0, -1, 0)^T) \end{pmatrix} [m] \quad (51)$$

9.5 Reglereinstellung für das Dynamikmodell

Die Parameter der Rückführmatrix \mathbf{K} und der Vorfiltermatrix \mathbf{V} des in Kap. 6.4 entworfenen Zustandsreglers sind in Gl. (52) und Gl. (53) dargestellt. Aus Platzgründen ist \mathbf{K} transponiert dargestellt.

$$\mathbf{K} = \begin{pmatrix} \sim 0 & 0.0807 & \sim 0 & \sim 0 \\ \sim 0 & 0.0115 & \sim 0 & \sim 0 \\ \sim 0 & \sim 0 & 0.0809 & \sim 0 \\ \sim 0 & \sim 0 & 0.0116 & \sim 0 \\ \sim 0 & \sim 0 & \sim 0 & 0.0183 \\ \sim 0 & \sim 0 & \sim 0 & 0.0100 \\ 2.2361 & \sim 0 & \sim 0 & \sim 0 \\ 0.8047 & \sim 0 & \sim 0 & \sim 0 \\ \sim 0 & \sim 0 & 0.0408 & \sim 0 \\ \sim 0 & \sim 0 & 0.0263 & \sim 0 \\ \sim 0 & -0.0408 & \sim 0 & \sim 0 \\ \sim 0 & -0.0262 & \sim 0 & \sim 0 \end{pmatrix}^T \quad (52)$$

$$\mathbf{V} = \begin{pmatrix} \sim 0 & \sim 0 & 2.2361 & \sim 0 \\ \sim 0 & -0.0408 & \sim 0 & \sim 0 \\ 0.0408 & \sim 0 & \sim 0 & \sim 0 \\ \sim 0 & \sim 0 & \sim 0 & 0.0183 \end{pmatrix} \quad (53)$$



Aktuelle Trends in der Produktionstechnik sind stark von der Forderung nach einem hohen Individualisierungsgrad der Produkte geprägt. Zur Realisierung der damit einhergehenden Flexibilität bei der Abfolge einzelner Produktionsschritte bedarf es eines ebenso flexiblen Materialflusssysteme. Derzeit wird in diesem Zusammenhang der Einsatz unbemannter Flugfahrzeuge (UAVs) für den Transport von Kleinteilen erforscht. Entgegen starren oder bodengebundenen Fördereinrichtungen (Fließband, Fördermatrix, FTS) sind UAVs in der Lage, den ungenutzten Luftraum innerhalb der Produktionsstätten auszunutzen. Parallele und voneinander unabhängige Beauftragungen der UAVs erzeugen dabei allerdings eine Umgebung mit hohem Kollisionspotenzial, in welcher die einzelnen Flugbewegungen nicht mehr im Voraus geplant werden können.

Vor dem Hintergrund dieser Problematik beschreibt das vorliegende Buch die Entwicklung und Implementierung einer neuen Online-Bahnplanungsmethode auf Basis ortsdiskreter Kurvenflüsse, welche in der Lage ist, kollisionsfreie Flugbahnen für mehrere UAVs durch Aufprägung virtueller Kräfte kontinuierlich an einen sich verändernden Umgebungszustand anzupassen.