



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

ALGORITHMEN ZUR EFFIZIENTEN SIMULATION GROSSER MEHRKÖRPERSYSTEME FÜR MODELICA

Christian Schubert

Geboren am: 20.10.1983 in Dresden

DISSERTATION

zur Erlangung des akademischen Grades

DOKTOR-INGENIEUR (DR.-ING.)

Erstgutachter

Prof. Dr.-Ing. habil. Günter Kunze

Zweitgutachter

Prof. Dr.-Ing. Michael Beitelschmidt

Eingereicht am: 26. Mai 2014

Verteidigt am: 05. Dezember 2014

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation mit dem Titel *Algorithmen zur effizienten Simulation großer Mehrkörpersysteme für Modelica* selbstständig und ohne unzulässige Hilfe Dritter verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Hilfsmittel und Quellen benutzt. Die wörtlichen und sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht. Es waren keine weiteren Personen an der geistigen Herstellung der vorliegenden Arbeit beteiligt. Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Dresden, den 26. Mai 2014

Christian Schubert

Zusammenfassung

In der vorliegenden Arbeit werden mithilfe von Methoden zur numerischen Behandlung schwach besetzter Matrizen $\mathcal{O}(n^3)$ - und $\mathcal{O}(n)$ -Berechnungsalgorithmen für Mehrkörpersysteme aus deren Bewegungsgleichungen abgeleitet. Durch Verwendung von Dualen Basen kann gezeigt werden, dass sich die bezüglich der Berechnungszeit effizienten Algorithmen sowohl auf Systeme mit explizit als auch implizit formulierten Bindungsgleichungen anwenden lassen. Mit diesen gewonnen Erkenntnissen wird die derzeitige Implementierung der vorgestellten Algorithmen im Sprachstandard Modelica untersucht. Es werden Ansatzmöglichkeiten aufgezeigt, mit denen ausgewählte Modelica Compiler große Mehrkörpersysteme effizienter lösen können. Zum einen wird durch eine graphentheoretische Verallgemeinerung des $\mathcal{O}(n)$ -Algorithmus dieser direkt in dem freien Modelica Werkzeug *OpenModelica* umgesetzt. Zum anderen wird die Methode der Subsysteme für den $\mathcal{O}(n)$ -Algorithmus vorgestellt. Sie ermöglicht es, beliebig komplexe Teilsysteme als eigenständige Modellelemente zu erstellen. Die Berechnung von kinematischen Schleifen kann auf diese Weise wesentlich beschleunigt werden. Ferner wird gezeigt, dass sich mit der Methode der Subsysteme Modellgleichungen eines idealen homokinetischen Gelenks ableiten lassen, die frei von Zwangsbedingungen sind. Dies führt ebenfalls zu einer schnelleren und robusteren Berechnung.

Abstract

Using methods from sparse matrix theory, $\mathcal{O}(n^3)$ - and $\mathcal{O}(n)$ -algorithms for multibody systems are derived from the equations of motion. The concept of Dual Bases reveals that efficient algorithms for explicit joint descriptions, regarding calculation time, may also be applied to systems which use implicit joint constraints. Consequently, the feasibility of implementing these results in Modelica is examined. This leads to new approaches which enable selected Modelica compilers to solve large multibody systems more efficiently. On the one hand side a graph-theoretic generalization of the $\mathcal{O}(n)$ -algorithm has been implemented into the OpenModelica compiler. On the other hand, a method of subsystems for the $\mathcal{O}(n)$ -algorithm has been devised. It allows to derive the model equations for arbitrary complex sub-systems which can be implemented as new model elements for an $\mathcal{O}(n)$ -algorithm library. This has been carried out for recurring kinematic loops of Mobile Machinery improving simulation speed considerably. Furthermore, it is shown that a fast and robust model of an ideal constant velocity joint can be derived that way.

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Professur für Baumaschinen- und Fördertechnik der Technischen Universität Dresden.

Mein besonderer Dank gilt Prof. Dr.-Ing. habil. Günter Kunze für die kontinuierliche Unterstützung und die Bereitstellung eines inspirierenden Arbeitsplatzes. An diesem *Hort der Ruhe und Gelassenheit* hatte ich die Möglichkeit, mich herausfordernden Problemstellungen intensiv und nachhaltig zu widmen. Herrn Prof. Dr.-Ing. Michael Beitelschmidt danke ich für die fachliche Orientierung auf dem komplexen Gebiet der Mehrkörperdynamik. Die zahlreichen Diskussionen und wertvollen Impulse haben wesentlich zum Inhalt der vorliegenden Arbeit beigetragen.

Den Mitarbeitern der Professur, insbesondere des Zimmers A114, danke ich für das kreative und produktive Arbeitsumfeld, das maßgeblich durch Frau Dr.-Ing. Ines Gubsch geprägt wird. Des Weiteren möchte ich Jens Frenkel für die Navigation durch die Untiefen der Welt der Modelica Compiler danken. Die Vorstellung, kinematische Schleifen durch analytische Beziehungen zu lösen, geht auf Dr.-Ing. Timo Penndorf zurück, dem ich dafür ausdrücklich danken möchte.

Dank gilt auch Thomas Beutlich, der personifizierten Unterstützung der ITI GmbH. Die Ergebnisse meiner Arbeit in einer nachhaltigen Form verwertet zu wissen, hat eine große Bedeutung für mich.

INHALTSVERZEICHNIS

1. Einleitung	13
1.1. Motivation	13
1.2. Präzisierung der Aufgabe	14
1.3. Aufbau der Arbeit	14
2. Mechanik der Mehrkörpersysteme	16
2.1. Bewegungsgleichung des starren Körpers	17
2.1.1. Kinematik des ungebundenen starren Körpers	18
2.1.2. Kinetik des ungebundenen starren Körpers	22
2.2. Beschreibung einer Bindung	26
2.2.1. Bindungsgleichung	26
2.2.2. Prinzip der virtuellen Leistung	29
2.2.3. Duale Basen einer Bindung	30
2.2.4. Globale Form der Bindungsgleichungen	32
2.2.5. Beispiele für Bindungen in einem Mehrkörpersystem	34
2.3. Bewegungsgleichung eines Mehrkörpersystems	36
2.4. Zusammenfassung zur Mechanik der Mehrkörpersysteme	38
3. Lösungsalgorithmen für Mehrkörpersysteme	39
3.1. Die Graphen eines Mehrkörpersystems	40
3.1.1. Einführung in die Graphentheorie	40
3.1.2. Graph einer quadratischen Matrix	41
3.1.3. Erweiterter Graph einer Matrix	42
3.1.4. Topologiegraph eines Mehrkörpersystems	43
3.1.5. Systemgraph eines Mehrkörpersystems	43
3.2. Lösungsalgorithmen für Systeme mit Baumstruktur	47
3.2.1. $\mathcal{O}(n^3)$ -Algorithmus für explizite Beschreibung	47
3.2.2. $\mathcal{O}(n^3)$ -Algorithmus für implizite Beschreibung	49
3.2.3. Modifizierter $\mathcal{O}(n^3)$ -Algorithmus für implizite Beschreibung	51
3.2.4. $\mathcal{O}(n)$ -Algorithmus für explizite Beschreibung	52
3.2.5. $\mathcal{O}(n)$ -Algorithmus für implizite Beschreibung	55
3.2.6. Modifizierter $\mathcal{O}(n)$ -Algorithmus für implizite Beschreibung	58
3.3. Lösungsalgorithmen am Beispiel einer ebenen Pendelkette	60
3.3.1. Allgemeine Festlegungen	60
3.3.2. $\mathcal{O}(n^3)$ -Algorithmus für implizite Beschreibungen	62

3.3.3.	$\mathcal{O}(n^3)$ -Algorithmus für explizite Beschreibung	64
3.3.4.	$\mathcal{O}(n)$ -Algorithmen	65
3.3.5.	Vergleich der Rechenzeiten von Lösungsalgorithmen	67
3.4.	Berücksichtigung kinematischer Schleifen	68
3.4.1.	Ansätze zur Berücksichtigung kinematischer Schleifen	68
3.4.2.	Vergleich der Lösungsalgorithmen	69
3.5.	Zusammenfassung der Lösungsalgorithmen eines Mehrkörpersystems	71
4.	Effiziente Berechnung von Mehrkörpersystemen	73
4.1.	Berechnung von Mehrkörpersystemen basierend auf Modelica	73
4.2.	$\mathcal{O}(n)$ -Algorithmus für Modelica Compiler	75
4.2.1.	Tearing	75
4.2.2.	Übertragung des $\mathcal{O}(n)$ -Algorithmus auf den Systemgraphen	76
4.2.3.	Übertragung des $\mathcal{O}(n)$ -Algorithmus auf Modelica Compiler	78
4.2.4.	Vergleich der Rechenzeiten	80
4.3.	$\mathcal{O}(n)$ -Algorithmus für Bibliothekselemente	81
4.3.1.	$\mathcal{O}(n)$ -MKS-Bibliothek für Modelica	81
4.3.2.	Aufbereitung eines Subsystems	83
4.3.3.	Implementierung einer kinematischen Schleife als Subsystem	84
4.3.4.	Implementierung eines Gleichlaufgelenkes als Subsystem	90
5.	Zusammenfassung und Ausblick	98
	Literaturverzeichnis	101
A.	Anhang	106
A.1.	Grundlagen der Tensorrechnung	106
A.2.	Duale Basis einer Bindung	112
A.2.1.	Einführung	112
A.2.2.	Wahl der dualen Basen	113
A.2.3.	Kinematische Differenzialgleichung	114
A.3.	Herleitung des Subsystems des Viergelenks	115
A.3.1.	Reduktion des Systems	115
A.3.2.	Numerische Lösung	117
A.3.3.	Symbolische Lösung	118
A.4.	Homokinetisches Gelenk als Subsystem	121

ABBILDUNGSVERZEICHNIS

2.1. Freier Körper	17
2.2. Statisches Kräftegleichgewicht an einem Liniensegment	22
2.3. Kräftegleichgewicht am ungebundenen Körper	24
2.4. Freigeschnittener gebundener Körper	26
2.5. Relative Lage zweier Koordinatensysteme	27
2.6. Bindung bezogen auf das Gesamtsystem	32
2.7. Drehgelenk	34
2.8. Kreuzgelenk	35
3.1. Begriffsdefinitionen an einem Graphen	41
3.2. Graphen einer Matrix \mathbf{A}	42
3.3. Topologiegraph	43
3.4. Systemgraph eines Viergelenks	44
3.5. Systemgraph eines impliziten Viergelenks	46
3.6. Mehrfachpendel	60
3.7. Pendel mit $\mathcal{O}(n^3)$ -Algorithmus	63
3.8. Verbundkörper des Pendels	65
3.9. Beispiel	66
3.10. Rechenzeitvergleich der Algorithmen	67
3.11. Modell eines Pendels mit $m = 10$ kinematischen Schleifen	69
3.12. Mehraufwand bei Berechnung mit Schleifen	70
4.1. Zuordnung von Gleichungen und Variablen im Systemgraphen eines Viergelenks	77
4.2. nummerierter Systemgraph des Viergelenks	78
4.3. Systemgraph des Viergelenks innerhalb des OpenModelica Compilers	79
4.4. Rechenzeitvergleich	81
4.5. Connection set	82
4.6. Ebenes Viergelenk	85
4.7. Skizze eines ebenen Viergelenks, nach [Mod13]	87
4.8. Normierte Rechenzeit eines Viergelenks mit verschiedenen Modelica Mechanikbibliotheken mit SimulationX 3.6	89
4.9. Mast einer Autobetonpumpe	90
4.10. Baggerarm	90
4.11. Ausführungen von Gleichlaufgelenken (aus Schmitz 1994)	91
4.12. Abstrahiertes Gleichlaufgelenk (nach [Sch94])	92
4.13. Gleichlaufgelenk, modelliert mit Modelica	93

4.14. Modell eines Gleichlaufgelenks als doppeltes Kreuzgelenk	93
4.15. Vergleich der Rechenzeiten der verschiedenen Implementierungen des homokinetischen Gelenks	97
A.1. Projektion des Vektors \vec{v} auf eine Basis $\vec{e}_{B,i}$	106

SYMBOLVERZEICHNIS

- $\vec{\mathbf{A}}_K$ Drehtensor des Körpers K . Es gilt $\vec{\mathbf{e}}_{O,i} = \vec{\mathbf{A}}_K \cdot \vec{\mathbf{e}}_{K,i}$
- \mathbf{G} Zusammenfassung aller Transformationsmatrizen eines Systems zu einer Blockmatrix deren Graph die Struktur des Topologiegraphen besitzt, siehe Gl. (3.8)
- \mathbf{G}_{PK} Transformationsmatrix zur Änderung des Bezugspunktes einer Kinemate von K nach P . Die Transponierte ändert den Bezugspunkt einer Last von P nach K . Handelt es sich bei einem Index um Kleinbuchstaben, sind die Bezugssysteme der Körper mit dem entsprechenden Index gemeint.
- \mathbf{H} Beschreibt den Zusammenhang zwischen den Geschwindigkeiten und der Lageänderung, $\mathbf{y}_K = \mathbf{H} \cdot \dot{\mathbf{z}}_K$
- \mathbf{J} Jacobimatrix, orthogonal zur Bindungsmatrix \mathbf{W}
- \mathbf{M}_S 6×6 Massenmatrix bezüglich des Punktes S
- $\hat{\mathbf{M}}$ Reduzierte Massenmatrix, wird im expliziten $\mathcal{O}(n^3)$ -Algorithmus verwendet
- \mathbf{M}^* Reduzierte Massenmatrix, wird im impliziten $\mathcal{O}(n^3)$ -Algorithmus verwendet
- $\mathbf{N}(i)$ Nachfolgerfunktion, gibt die Menge der Indizes aller Nachfolger des Körpers mit dem Index i im aufspannenden Baum des Topologiegraphen zurück
- $\mathbf{V}(i)$ Vorgängerfunktion, gibt den Index des Vorgängers des Körpers mit dem Index i im aufspannenden Baum des Topologiegraphen zurück
- \mathbf{W} Bindungsmatrix, nachgestellte Subskripte h und nh kennzeichnen den holonomen bzw. nicht holonomen Anteil
- $\vec{\mathbf{a}}_K$ Translatorische Beschleunigung des Punktes K
- \mathbf{c}_i Fasst alle Terme der expliziten Bindungsgleichung auf Beschleunigungsebene zusammen, die nicht direkt von den Beschleunigungen \mathbf{x} und $\dot{\mathbf{s}}$ abhängen
- $\vec{\mathbf{e}}_{B,i}$ i -ter Basisvektor des Koordinatensystems B
- $\vec{\mathbf{f}}_K$ Kraft angreifend am Punkt K
- \mathbf{g} Holonome implizite Bindungsgleichung auf Lageebene
- \mathbf{h}_S Vektor der geschwindigkeitsproportionalen Trägheitsterme bezüglich S

$\mathbf{l}_{A,i}$	Eingeprägte bzw. aktive Last der Bindung i
$\mathbf{l}_{C,i}$	Zwangslast der Bindung i
\mathbf{l}_K	Last bezüglich des Punktes K
$\mathbf{l}'_{K,ext}$	Fasst $\mathbf{l}_{K,ext}$ und die bekannte Last in die freien Lastrichtungen zusammen, siehe Gl. (3.6)
$\mathbf{l}_{K,ext}$	Hilfsvariable zur kompakteren Notation, die die externen Lasten und die geschwindigkeitsproportionalen Trägheitsterme zusammenfasst
$\mathbf{l}_{S,ext}^*$	Auf den Schwerpunkt S reduzierte externe Lasten, die am Körper angreifen
\mathbf{l}_i	Schnittlast der Bindung i
n_B	Anzahl der Bindungen
n_K	Anzahl der Körper
$\vec{\mathbf{p}}$	Impuls
\mathbf{p}_K	Eulerparameter zur Beschreibung der rotatorischen Lage des Körpers K
\mathbf{q}	Vektor der generalisierten Lagekoordinaten
\mathbf{q}_i	Generalisierte Lagekoordinaten der i -ten Bindung
$\vec{\mathbf{r}}_K$	Lagevektor des Punktes K
\mathbf{s}_i	Generalisierte Geschwindigkeiten der i -ten Bindung
\mathbf{t}_i	Koordinaten zur Basis der freien Bewegungsrichtungen zur Darstellung der Beschleunigung, also $\mathbf{x}_i = \Phi_i \mathbf{t}_i + \bar{\Phi}_i \bar{\mathbf{t}}_i$
$\vec{\mathbf{v}}_K$	Translatorische Geschwindigkeit des Punktes K
\mathbf{x}_K	$= \begin{bmatrix} \vec{\alpha}_K \\ \vec{\mathbf{a}}_K \end{bmatrix}$, Beschleunigung des Körpers K
\mathbf{y}_K	$= \begin{bmatrix} \vec{\omega}_K \\ \vec{\mathbf{v}}_K \end{bmatrix}$, Kinematik des Körpers K
\mathbf{z}_K	$= \begin{bmatrix} \mathbf{p}_K \\ \vec{\mathbf{r}}_K \end{bmatrix}$, Pose des Körpers K
Φ_i	Matrix der freien Bewegungsrichtungen der Bindung i , Teil der dualen Basis einer Bindung
$\bar{\Phi}_i$	Matrix der gesperrten Bewegungsrichtungen der Bindung i , Teil der dualen Basis einer Bindung
Ψ_i	Matrix der freien Lastrichtungen der Bindung i , Teil der dualen Basis einer Bindung
$\bar{\Psi}_i$	Matrix der gesperrten Lastrichtungen der Bindung i , Teil der dualen Basis einer Bindung
Θ_S	Trägheitstensor bezüglich des Punktes S
$\vec{\alpha}_K$	Rotatorische Beschleunigung des Punktes K
$\chi(\mathbf{q})$	Zwangsbedingungen der generalisierten Lagekoordinaten
$\boldsymbol{\eta}_i$	Auf die freien Bewegungsrichtungen projizierter Anteil der eingeprägten Lasten der i -ten Bindung, $\boldsymbol{\eta} = \Phi^T \mathbf{l}_{N,A}$

Symbolverzeichnis

λ_i	Koordinaten der Kraftrichtungen, also $l_N = \Psi \lambda + \overline{\Psi} \overline{\lambda}$ (auch als Lagrange Multiplikatoren bekannt)
$\vec{\omega}_K$	Drehgeschwindigkeitsvektor des Koordinatensystems K
$\vec{\pi}_S$	Drall bezüglich des Punktes S
$\sigma_{i\alpha}$	Funktion zur Abbildung der Topologie des Mehrkörpersystems, siehe Gl. (2.53)
$\vec{\tau}_K$	Moment bezüglich des Punktes K
ξ	Rheonomer Anteil der expliziten Bindungsgleichung auf Geschwindigkeitsebene
ζ	Rheonomer Anteil der impliziten Bindungsgleichung auf Geschwindigkeitsebene

1. EINLEITUNG

Simulationen schaffen die Voraussetzungen, um technische Systeme schon in frühen Phasen der Produktentwicklung untersuchen und bewerten zu können. Durch konsequente Anwendung der computergestützten Berechnung als Werkzeug in Forschung und Entwicklung kann der Bau realer Prototypen vermieden werden. Besonders für die Branche der Baumaschinenteknik bietet dies, aufgrund der hohen Stückkosten bei gleichzeitig geringen Stückzahlen, ein enormes Potenzial um Kosten, Zeit und Ressourcen einzusparen. Darüber hinaus ist es möglich, Versuche unter unveränderten Bedingungen durchzuführen, sowie das Verhalten in außergewöhnlichen oder sogar gefährlichen Situationen vorherzusagen.

1.1. MOTIVATION

Die Qualität bzw. Zuverlässigkeit der Ergebnisse hängt maßgeblich von der Güte des erstellten Modells ab und damit mittelbar von dem Wissen und der Erfahrung des Modellierers. Im Allgemeinen gilt, je detaillierter und umfangreicher ein System modelliert wird, desto genauer werden die Ergebnisse und desto mehr Effekte können berücksichtigt werden. Mit der Modelltiefe steigt der Berechnungsaufwand und somit die nötige Rechenzeit. Insofern ist vom Modellierer immer ein Kompromiss zwischen Modelltiefe und Rechenzeit zu finden.

Besonders im Rahmen von Echtzeitsimulationen werden daher nur stark vereinfachte und speziell optimierte Modelle eingesetzt. Dies wiederum schränkt die Gültigkeit des Modells und damit das Anwendungsgebiet stark ein. So kann die interaktive Echtzeitsimulation zur Ausbildung und dem Design der Bedienerschnittstellen schon heute genutzt werden, da hierfür nur ein qualitativ korrektes Verhalten erforderlich ist. Auch die Entwicklung von Steuergeräten an Hardware-in-the-Loop Prüfständen ist mittlerweile aufgrund leistungsfähigerer Rechentechnik in Grenzen möglich.

Die Verhaltensweise sowie die Beanspruchung einer Baumaschine hängt maßgeblich vom Bediener ab. Es bedarf daher realistischer Bedienerangaben, um in einer ganzheitlichen Simulation zu aussagekräftigen Ergebnissen zu kommen. Bemühungen, geeignete Bedienermodelle zu entwickeln, haben aufgrund der vielfältigen Wechselwirkungen und den komplexen Arbeitsaufgaben bisher nicht zum gewünschten Ergebnis geführt. Die interaktive Simulation bietet hierzu eine Lösung an, indem Sie es erlaubt, einen echten Bediener in die virtuelle Welt einzubinden. Sie ermöglicht die Bewertung stark subjektiver Kriterien, wie z.B. das Lenkverhalten, das nur schwer anhand von Diagrammen beurteilt werden kann. Jedoch sind solche Bestrebungen aufgrund der begrenzten Modelltiefe, die in Echtzeit verarbeitet werden kann, derzeit kaum denkbar.

Neben dem allgemeinen Bedürfnis, die Ergebnisse einer Simulation so schnell wie möglich zu

erhalten, gibt es im Bereich der Baumaschinen folgende Anwendungen für die die Rechenzeit kritisch ist:

- (interaktive) Echtzeitsimulation
- modellgestützte Regelung, wie z.B. Model Predictive Control
- Zustandsüberwachung, die auf einem Beobachter basieren
- numerische Parameter- und Modellstrukturoptimierung, die im Allgemeinen eine Vielzahl von Simulationen benötigt.

Die Fortschritte auf dem Gebiet der Rechentechnik führen zwar bereits zu einer Reduzierung der Rechenzeit, aber diese Einsparungen wurden meist genutzt, um die Modelle entweder zu vergrößern oder detailreicher aufzubauen. Eine weitere Möglichkeit Rechenzeiten zu verkürzen liegt darin, die verwendeten Algorithmen zur Berechnung der Modelle zu beschleunigen. Für Letzteres entdeckte der Autor während der Arbeit an der Professur für Baumaschinen- und Fördertechnik an der TU Dresden verschiedene Ansätze. Mit diesen gelang es speziell, die Simulation von Baumaschinen zu beschleunigen. Die Ergebnisse der wissenschaftlichen Tätigkeit sollen in der vorliegenden Arbeit vorgestellt werden.

1.2. PRÄZISIERUNG DER AUFGABE

Die Funktionalität einer Baumaschine entsteht durch das Zusammenspiel vieler technischer Teilsysteme aus verschiedenen Domänen, wie Mechanik, Hydraulik und Thermodynamik. Folglich muss das Modell zur Simulation des zeitlichen Verhaltens einer gesamten Maschine all diese Teilsysteme umfassen. Für eine solche Modellierungsaufgabe eignen sich Simulationswerkzeuge und -bibliotheken basierend auf dem Sprachstandard Modelica [Fri11]. Dieser wurde speziell für die elegante Beschreibung komplexer physikalischer Modelle entwickelt.

Modelica ermöglicht das Beschreiben von Verhaltensmodellen mithilfe von Differenzialgleichungen in einer werkzeugunabhängigen Form. Diese werden durch einen Modelica Compiler symbolisch aufbereitet, in eine ausführbare Form übersetzt und anschließend numerisch gelöst. Auf diese Weise wird eine vollständige Trennung der Beschreibung von der Lösung des Modells erreicht. Es finden sich eine Vielzahl an Veröffentlichungen [BO03, KSF10, KSEL10, KFSJ09, SNK11], in denen die Vorzüge der Verwendung von Modelica für Baumaschinen bereits betont wurden.

Die Rechenzeit eines Modells wird maßgeblich durch die verwendete Schrittweite des numerischen Integrators sowie der benötigten Zeit pro Rechenschritt beeinflusst. Aufgrund der meist erheblichen Abmessungen von Baumaschinen liegen die Zeitkonstanten des mechanischen Teilsystems einer Baumaschine typischerweise mehrere Zehnerpotenzen über der des Hydraulikteilmodells [KRW⁺14]. Daher wird die Rechenschrittweite meist durch die Hydraulik bestimmt, während der zeitliche Aufwand pro Schritt üblicherweise von der Mechanik dominiert wird.

Methoden zur gezielten Beeinflussung der Zeitkonstanten in der Hydraulik wurden in [Åma11] vorgestellt. Im Rahmen der vorliegenden Arbeit wird untersucht, ob und wie es möglich ist, die benötigte Zeit zur Auswertung des mechanischen Teilsystems speziell unter Berücksichtigung großer Modelle und damit die Simulationszeit des Gesamtmodells zu verringern. Auf diese Weise wird die Voraussetzung geschaffen, detailliertere Modelle beispielsweise im Rahmen einer Echtzeitsimulation nutzen zu können.

1.3. AUFBAU DER ARBEIT

Soll die Simulationszeit für das mechanische Teilsystem eines Modells verkürzt werden, muss analysiert werden, wie dessen Berechnung vonstatten geht. Die Arbeit gliedert sich dazu in die Teilaufgaben:

- Aufbauen des Grundverständnisses für die Berechnung des zeitlichen Verhaltens eines mechanischen Systems basierend auf Modelica
- Aufdeckung von Problemen, die eine effiziente Berechnung von mechanischen Teilsystemen eines Modelles verhindern
- Umsetzung von Lösungen für eine schnellere Berechnung von Modellen mechanischer Systeme basierend auf Modelica.

Der am weitesten verbreitete Ansatz zur Modellierung und Simulation dreidimensionaler mechanischer Teilmodelle ist die Methode der Mehrkörpersysteme, kurz MKS. Der Fokus der vorliegenden Arbeit richtet sich auf die Erstellung und Berechnung von Mehrkörpersystemen zur Simulation mechanischer Teilsysteme komplexer Maschinen. Ein MKS-Modell setzt sich aus einem System von starren Körpern zusammen, welche über eine Anzahl von Bindungen oder Lastelementen¹ miteinander und ihrer Umgebung in Beziehung stehen. Im Sprachgebrauch der Modellierung mechanischer Systeme werden Bindungen meist als Gelenke bezeichnet.

Algorithmen zur Berechnung von Mehrkörpersystemen stellen immer eine Verbindung zwischen der Beschreibung der Mechanik und Methoden zum Lösen eines linearen Gleichungssystems dar. Ersteres beschreibt die physikalischen Zusammenhänge des mechanischen Teilsystems und schließt mit der Herleitung der Bewegungsgleichungen eines MKS ab, die ein großes, schwach besetztes lineares Gleichungssystem bilden. Letzteres nutzt Methoden der linearen Algebra, um das lineare Gleichungssystem aufzubereiten, wobei im Ergebnis Lösungsalgorithmen für das Auswerten der Bewegungsgleichungen stehen. In der Literatur, z.B. [Pie11, SBH10], wurden diese beiden Bestandteile der MKS immer stark miteinander vermischt. Das erschwert die Suche nach Optimierungspotenzialen bezüglich einer beschleunigten Berechnung von MKS-Modellen. Um die Ursachen für eine ineffizient ausgeführte Simulation von Mehrkörpersystemen hinsichtlich der Rechenzeit zu ergründen, werden beide Teile im Rahmen dieser Arbeit strikt getrennt voneinander betrachtet.

Kapitel 2 beschäftigt sich mit der Mechanik eines Mehrkörpersystems. Es wird erläutert wie durch die Berücksichtigung der Kinematik und Kinetik des Systems die Bewegungsgleichungen unter zur Hilfenahme einer kompakten Tensorschreibweise in allgemeiner Form aufgestellt werden können. Es wird ferner gezeigt, dass sich das resultierende Gleichungssystem zur Berechnung der Systembeschleunigungen ungeachtet dessen, ob diese explizit oder implizit beschrieben werden, immer aus drei Grundgleichungen und deren erster und zweiter Ableitung nach der Zeit zusammensetzt.

Kapitel 3 befasst sich mit den Methoden der linearen Algebra und leitet mithilfe der Graphentheorie die bekannten Lösungsalgorithmen für Mehrkörpersysteme her. Diese unterscheiden sich zum einen in der verwendeten Formulierung der Bindungsgleichungen und zum anderen in der Skalierung der Rechenzeit bei einer Vergrößerung des Mehrkörpersystems. Es wird gezeigt, dass sie alle auf das in Kapitel 2 erarbeitete Gleichungssystem zurückzuführen sind, auch wenn der Bezug zu diesem sich oftmals nicht auf den ersten Blick offenbart.

Bezug nehmend auf das Ziel der vorliegenden Arbeit werden die Vor- und Nachteile der hergeleiteten Lösungsalgorithmen in Kapitel 4 diskutiert. Darüber hinaus wird erläutert, wie die verbreitetsten Modelica Werkzeuge die Lösungsalgorithmen für Mehrkörpersysteme implementiert haben. In Verbindung mit den Erkenntnissen aus den Kapiteln 2 und 3 werden damit die Ursachen für lange Berechnungszeiten von mechanischen Teilsystemen eines Baumaschinenmodells ergründet. Dies ist die Grundlage für die im Kapitel 4 vorgestellten erarbeiteten Lösungen, die wie an praktischen Beispielen gezeigt werden kann, eine Reduktion der Rechenzeit von MKS-Modellen zur Folge haben.

¹Im Rahmen dieser Arbeit wird anstelle von *Kraftelement* der Begriff *Lastelement* benutzt, da diese auch Momente in das System einbringen können.

2. MECHANIK DER MEHRKÖRPERSYSTEME

Ziel dieses Kapitels ist es, die Bewegungsgleichungen eines Mehrkörpersystems herzuleiten. Die Bewegungsgleichungen werden durch ein differenzial-algebraisches Gleichungssystem (DAE) beschrieben, dessen Lösung den zeitlichen Verlauf aller kinematischen und kinetischen Größen wiedergibt. Unter einem Mehrkörpersystem wird in dieser Arbeit ein System aus starren Körpern, idealen zweiseitigen Bindungen und masselosen Lastelementen, die die wirkenden Kräfte und Momente im System beschreiben, verstanden.

Um zu den Bewegungsgleichungen eines Mehrkörpersystems zu gelangen, soll zunächst in Abschnitt 2.1 die Bewegungsgleichung eines einzelnen starren Körpers hergeleitet werden. Es wird aus der Kinematik und Kinetik des starren Körpers die Newton-Euler-Gleichung des ungebundenen Körpers abgeleitet. Diese wird anschließend um die Schnittlasten der angrenzenden Bindungen erweitert. Im Ergebnis steht die Bewegungsgleichung des gebundenen starren Körpers.

In Abschnitt 2.2 schließt sich die Betrachtung der zweiseitigen Bindungen an, die die Bewegungsfreiheit zweier Körper zueinander einschränkt. Diese kinematischen Beziehungen werden gemeinhin als Bindungsgleichungen bezeichnet. Sie können entweder in impliziter (Beschreibung der verbotenen Bewegungen) oder in expliziter Form (Beschreibung der zulässigen Bewegungen) vorliegen. Abhängig von der konkreten Bindung lässt sich meist eine Form leichter beschreiben als die andere. Vereinzelt existiert sogar nur eine der beiden Beschreibungsformen.

Aus dem Prinzip der virtuellen Leistung lassen sich bindungsspezifische Bedingungen an die Schnittlasten ableiten, die für die Aufstellung der Bewegungsgleichung notwendig sind. Unter der Verwendung von dualen Basen (auch *Modes of Motion* [RS88] genannt) lassen sich das Prinzip der virtuellen Leistung und die Bindungsgleichungen auf Lageebene so umformulieren, dass nicht mehr zwischen expliziter und impliziter Beschreibung unterschieden werden muss. Neben Vorteilen in der Notation werden sich die speziellen Eigenschaften einer dualen Basis auch in Kapitel 3 als nützlich erweisen. Anschließend werden die Bindungsgleichungen so umgeschrieben, dass sie sich in die Bewegungsgleichung des Mehrkörpersystems einbetten lassen. Der Abschnitt wird durch die Betrachtung eines Dreh- und eines Schubgelenks abgeschlossen, für welche die Bindungsgleichungen in globaler Form unter Nutzung der dualen Basen hergeleitet werden.

Abschnitt 2.3 führt die Ergebnisse der beiden vorhergehenden Abschnitte zusammen, um zu den Bewegungsgleichungen eines Mehrkörpersystems als DAE mit dem Index 3 zu gelangen. Es enthält die Bindungsgleichungen auf Lageebene. Zur numerischen Berechnung der Lösung des Systems wird die DAE meist auf den Index 1 reduziert, was einer Ersetzung der Bindungsgleichungen auf Lageebene mit denen auf Beschleunigungsebene entspricht. Anhand des dadurch

indexreduzierten Systems lässt sich zeigen, dass der wesentliche Teil der Bewegungsgleichungen letztlich nur aus drei verschiedenen Gleichungen besteht, die ein großes lineares schwach besetztes Gleichungssystem bilden. Dessen effiziente Lösung wird mit den bis dahin erworbenen Erkenntnissen in Kapitel 3 diskutiert.

2.1. BEWEGUNGSGLEICHUNG DES STARREN KÖRPERS

Für die Herleitung der Bewegungsgleichung eines starren Körpers werden an dieser Stelle Vereinbarungen zu den in der Arbeit verwendeten Notationen getroffen. Ein Symbol mit aufgesetztem Vektorpfeil, z.B. \vec{r} , beschreibt einen Tensor erster Stufe und damit das geometrische Objekt, dargestellt durch eine gerichtete Strecke einer bestimmten Länge. Ist dem Symbol ein tiefgestellter Index vorangesetzt, z.B. ${}_K\vec{r}$, so ist die Darstellung des Tensors als 3×1 Matrix bezüglich der Basis bzw. des Koordinatensystems K gemeint. Bei der Ableitung von Vektoren nach der Zeit ist immer darauf zu achten, bezüglich welcher Basis diese ausgeführt bzw. beobachtet wird. Der Term $\frac{d^B}{dt}\vec{r}$ zeigt an, dass die Basis B bei der Differentiation als ruhend angenommen wird. Ein aufgesetzter Punkt wird ausschließlich für Differentiation nach der Zeit bezüglich des Inertialsystems verwendet. Für eine ausführlichere Diskussion sei auf A.1 verwiesen.

Im Folgenden soll die Bewegung eines ungebundenen starren Körpers im Raum beschrieben werden, um daraus dessen Bewegungsgleichung ableiten zu können. Es wird ein orthonormales Koordinatensystem K eingeführt, das fest mit dem Körper verbunden ist. Entsprechend Abb. 2.1, lässt sich die Lage \vec{r}_{OP} eines beliebigen körperfesten Punktes P bezüglich eines Inertialsystems O bestimmen als [Sha10]

$$\vec{r}_{OP} = \vec{r}_{OK} + \vec{r}_{KP}. \quad (2.1)$$

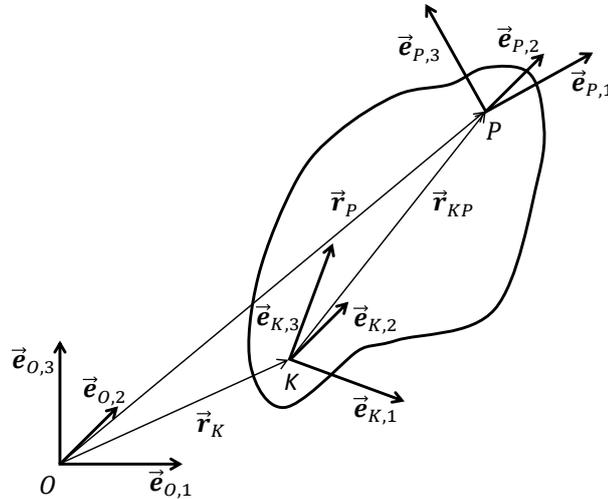


Abbildung 2.1.: Freier Körper

Sofern keine Verwechslungsgefahr besteht, wird im Folgenden der Index O weggelassen um zu einer übersichtlicheren Schreibweise zu gelangen. Beispielsweise wird an Stelle \vec{r}_{OP} kurz \vec{r}_P geschrieben. Vektoren, die relativ zu O gemessen werden, besitzen auf diese Weise nur einen Index und sind somit als absolute Größen zu erkennen.

Sofern die Geometrie eines starren Körpers bekannt ist, ist auch die Lage ${}_K\vec{r}_{KP}$ eines jeden Punktes P auf selbigem bezüglich des körperfesten Koordinatensystems K bekannt. Um diese Information in eine tensorielle Beschreibung zu überführen wird, die Lage der Basisvektoren $\vec{e}_{K,i}$ des körperfesten Koordinatensystems i benötigt. Diese Information ist im Drehtensor $\vec{\vec{A}}_K$

gespeichert, für den gilt

$$\vec{e}_{O,i} = \vec{\mathbf{A}}_K \cdot \vec{e}_{K,i}.$$

Zusätzlich ist es möglich, dass die Basisvektoren von P gedreht zu denen von K liegen.

$$\vec{e}_{K,i} = \vec{\mathbf{A}}_{KP} \cdot \vec{e}_{P,i}$$

Da es sich um einen Starrkörper handelt, muss der Drehtensor $\vec{\mathbf{A}}_{KP}$ bezüglich der Zeit konstant sein.

2.1.1. KINEMATIK DES UNGEBUNDENEN STARREN KÖRPERS

Um zu den Bewegungsgleichungen des ungebundenen starren Körpers zu gelangen, müssen zuerst dessen kinematische Zusammenhänge beschrieben werden.

LAGEEBENE

Die translatorischen und rotatorischen Lageinformationen des Körpers K werden in der *Pose* \mathbf{z}_K zusammengefasst. Anstelle jedoch den Drehtensor $\vec{\mathbf{A}}_K$ zu verwenden, hat es sich in der Praxis bewährt lediglich mit den zugehörigen Eulerparametern \mathbf{p}_K , siehe Gl. (A.13), zu arbeiten. Daher wird im Rahmen dieser Arbeit die Pose definiert als

$$\mathbf{z}_K = \begin{bmatrix} \mathbf{p}_K \\ \vec{\mathbf{r}}_K \end{bmatrix}. \quad (2.2)$$

Somit ist es mithilfe der Pose \mathbf{z}_K und Gl. (2.1) immer möglich, die Lage eines beliebigen Punktes P auf einem starren Körper K eindeutig zu bestimmen. Zusätzlich muss die Zwangsbedingung

$$\mathbf{p}_K^T \cdot \mathbf{p}_K = 1 \quad (2.3)$$

beachtet werden. Das Koordinatensystem P besitzt eine relative Pose bezüglich des Körperbezugssystems K .

$$\mathbf{z}_{PK} = \begin{bmatrix} \mathbf{p}_{PK} \\ \vec{\mathbf{r}}_{PK} \end{bmatrix}$$

Es muss beachtet werden, dass \mathbf{z}_K als Spaltenvektor definiert wurde, da es sich bei \mathbf{p}_K selbst um eine Matrix handelt. Um \mathbf{z}_K numerisch auswerten zu können, muss jeder Tensor in eine Komponentendarstellung überführt werden. Es ist üblich, für die translatorischen und rotatorischen Größen verschiedene Basen zu verwenden. Obwohl die Einträge in den Matrizen Tensoren sind, ist die Notation so gewählt, dass sie sich direkt auf die Komponentendarstellung mit reellen Zahlen übertragen lässt.

Über der Menge aller Posen kann eine nicht-abelsche Gruppe definiert werden. Deren Additionsoperator \oplus wird definiert als

$$\mathbf{z}_1 \oplus \mathbf{z}_2 := \begin{bmatrix} \mathbf{p}_1 \star \mathbf{p}_2 \\ \vec{\mathbf{r}}_1 + \vec{\mathbf{r}}_2 \end{bmatrix}.$$

Der Operator \star bezeichnet die Multiplikation von Quaternionen, entsprechend Gl. (A.18). Das

Inverse Element ist definiert als

$$\mathbf{z}_1^{-1} = \begin{bmatrix} \bar{\mathbf{p}}_1 \\ -\bar{\mathbf{r}}_1 \end{bmatrix}$$

entsprechend (A.17). Das Nullelement lautet

$$\mathbf{z}_0 = \begin{bmatrix} \mathbf{p}_e \\ \bar{\mathbf{0}} \end{bmatrix}, \mathbf{p}_e = \begin{bmatrix} 1 \\ \bar{\mathbf{0}} \end{bmatrix}.$$

Der Einfachheit halber sei zusätzlich die Subtraktion als

$$\mathbf{z}_1 \ominus \mathbf{z}_2 = \mathbf{z}_1 \oplus \mathbf{z}_2^{-1}$$

definiert. Auf diese Weise kann Gl. (2.1) zusammen mit der Lage der Basisvektoren als

$$\mathbf{z}_P = \mathbf{z}_K \oplus \mathbf{z}_{KP}$$

und damit sehr kompakt formuliert werden.

GESCHWINDIGKEITS- UND BESCHLEUNIGUNGSEBENE

Die Geschwindigkeit $\vec{\mathbf{v}}_P$ des Punktes P auf dem Körper K bezüglich des Inertialsystems O ergibt sich durch Differenziation von Gl. (2.1) bezüglich der Zeit:

$$\begin{aligned} \vec{\mathbf{v}}_P &= \frac{d^O}{dt} \vec{\mathbf{r}}_P \\ &= \frac{d^O}{dt} \vec{\mathbf{r}}_K + \frac{d^O}{dt} \vec{\mathbf{r}}_{KP} \\ &= \vec{\mathbf{v}}_K + \frac{d^O}{dt} \vec{\mathbf{r}}_{KP} \end{aligned}$$

Für jeden Starrkörper gilt, dass alle Punkte einen bezüglich der Zeit konstanten Abstand zueinander besitzen. Deshalb muss

$$\frac{d^K}{dt} \vec{\mathbf{r}}_{KP} = \bar{\mathbf{0}}$$

gelten. Mit dieser Information kann der Term $\frac{d^O}{dt} \vec{\mathbf{r}}_{KP}$ durch Anwendung von Gl. (A.11) vereinfacht werden und es folgt die sogenannte Starrkörperformel

$$\vec{\mathbf{v}}_P = \vec{\mathbf{v}}_K + \vec{\boldsymbol{\omega}}_K \times \vec{\mathbf{r}}_{KP}. \quad (2.4)$$

Damit ist ersichtlich, dass der Geschwindigkeitszustand jedes Punktes eines starren Körpers vollständig über die translatorische Geschwindigkeit $\vec{\mathbf{v}}_K$ und die rotatorische Geschwindigkeit $\vec{\boldsymbol{\omega}}_K$ des Punktes K definiert ist. Diese Größen werden zusammen als *Kinemate* \mathbf{y}_K des Körpers K zusammengefasst.

$$\mathbf{y}_K = \begin{bmatrix} \vec{\boldsymbol{\omega}}_K \\ \vec{\mathbf{v}}_K \end{bmatrix} \quad (2.5)$$

Für eine ausführliche Diskussion des Konzepts der Kinemate und des dahinterliegenden theoretischen Grundgerüsts wird auf die *Bewegungswinder* aus [MMS05], bzw. die *Spatial Vector Algebra* aus [Fea07] verwiesen.

Der differenzielle Zusammenhang der Kinemate und der Pose kann mithilfe der Gl. (A.15) ge-

finden werden.

$$\begin{aligned}
 \mathbf{y}_K &= \begin{bmatrix} 2\mathbf{P} \frac{d}{dt} \mathbf{p}_K \\ \frac{d}{dt} \vec{\mathbf{r}}_K \end{bmatrix} \\
 &= \begin{bmatrix} 2\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \vec{\mathbf{E}} \end{bmatrix} \cdot \dot{\mathbf{z}}_K \\
 &= \mathbf{H} \cdot \dot{\mathbf{z}}_K
 \end{aligned} \tag{2.6}$$

Die Gleichungen (2.6) und (2.3) bilden zusammen folgendes differenzial-algebraisches Gleichungssystem (DAE) mit dem Index 1, das den Zusammenhang zwischen Lageänderungen und Geschwindigkeiten beschreibt.

$$\mathbf{y}_K = \mathbf{H} \cdot \dot{\mathbf{z}}_K \tag{2.7a}$$

$$\mathbf{p}_K^T \cdot \mathbf{p}_K = 1 \tag{2.7b}$$

Dieses muss als Teil der Bewegungsgleichung für jeden Körper zur Ermittlung der Pose gelöst werden. Unter Verwendung von Gl. (A.16) kann es direkt nach $\dot{\mathbf{z}}_K$ umgestellt werden.

$$\begin{aligned}
 \dot{\mathbf{z}}_K &= \begin{bmatrix} \frac{1}{2}\mathbf{P}^T & \mathbf{0} \\ \mathbf{0} & \vec{\mathbf{E}} \end{bmatrix} \cdot \mathbf{y}_K \\
 &= \mathbf{H}^{-1} \cdot \mathbf{y}_K
 \end{aligned} \tag{2.8}$$

Diese Gleichung berücksichtigt jedoch nur die Zeitableitung der Nebenbedingung (2.3). Daher sollte diese Form nicht ohne eine Maßnahme zur Stabilisierung der Zwangsbedingung für die numerische Integration der Pose $\dot{\mathbf{z}}_K$ benutzt werden. Andernfalls würde sich ein numerischer Drift einstellen.

Die Kinemate \mathbf{y}_K gilt ausschließlich für den Punkt K , also den Referenzpunkt des körperfesten Koordinatensystems K , da sich im Allgemeinen die translatorischen Geschwindigkeiten verschiedener Punkte auf einem Körper unterscheiden. Mithilfe von Gleichung (2.4) kann die Kinemate auf einen anderen Punkt P transformiert werden

$$\begin{aligned}
 \mathbf{y}_P &= \begin{bmatrix} \vec{\omega}_P \\ \vec{\mathbf{v}}_P \end{bmatrix} = \begin{bmatrix} \vec{\omega}_K \\ \vec{\mathbf{v}}_K + \vec{\omega}_K \times \vec{\mathbf{r}}_{KP} \end{bmatrix} \\
 &= \begin{bmatrix} \vec{\mathbf{E}} & \vec{\mathbf{0}} \\ \vec{\mathbf{r}}_{PK} & \vec{\mathbf{E}} \end{bmatrix} \cdot \mathbf{y}_K \\
 &= \mathbf{G}_{PK} \cdot \mathbf{y}_K
 \end{aligned} \tag{2.9}$$

wobei $\vec{\mathbf{r}}_{PK}$ einen Tensor zweiter Stufe mit denselben Abbildungseigenschaften wie $\vec{\mathbf{r}}_{PK} \times$ bezeichnet, siehe Gleichung (A.4). Die Matrix

$$\mathbf{G}_{PK} = \begin{bmatrix} \vec{\mathbf{E}} & \vec{\mathbf{0}} \\ \vec{\mathbf{r}}_{PK} & \vec{\mathbf{E}} \end{bmatrix}, \tag{2.10}$$

deren Einträge kartesische Tensoren sind, wird im Rahmen dieser Arbeit Transformationsmatrix

von K nach P genannt. Es ist zu beachten, dass bei Überführung der Elemente von \mathbf{G}_{PK} in eine Matrixdarstellung, die Einheitstensoren auf der Hauptdiagonalen, abhängig von der Wahl der Basisvektoren, die Gestalt von Rotationsmatrizen annehmen. Es lässt sich zeigen, dass für Transformationsmatrizen Rechenregeln, ähnlich derer von Rotationsmatrizen, wie folgt gelten:

$$\mathbf{G}_{AB} \cdot \mathbf{G}_{BC} = \mathbf{G}_{AC}$$

und

$$\mathbf{G}_{PK}^{-1} = \begin{bmatrix} \vec{\mathbf{E}} & \vec{\mathbf{0}} \\ \vec{\mathbf{r}}_{PK}^T & \vec{\mathbf{E}} \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{E}} & \vec{\mathbf{0}} \\ \vec{\mathbf{r}}_{KP} & \vec{\mathbf{E}} \end{bmatrix} = \mathbf{G}_{KP}.$$

Ein erneutes Ableiten der Gleichung (2.4) führt auf den Zusammenhang der Beschleunigungen.

$$\begin{aligned} \vec{\mathbf{a}}_P &= \frac{d}{dt} \vec{\mathbf{v}}_P \\ &= \frac{d}{dt} \vec{\mathbf{v}}_K + \left(\frac{d}{dt} \vec{\boldsymbol{\omega}}_K \right) \times \vec{\mathbf{r}}_{KP} + \vec{\boldsymbol{\omega}}_K \times \left(\frac{d}{dt} \vec{\mathbf{r}}_{KP} \right) \\ &= \vec{\mathbf{a}}_K + \vec{\boldsymbol{\alpha}}_K \times \vec{\mathbf{r}}_{KP} + \vec{\boldsymbol{\omega}}_K \times (\vec{\boldsymbol{\omega}}_K \times \vec{\mathbf{r}}_{KP}) \end{aligned} \quad (2.11)$$

Analog zur Kinemate \mathbf{y}_K soll nun die Beschleunigung \mathbf{x}_K des Körpers K eingeführt werden.

$$\mathbf{x}_K = \begin{bmatrix} \vec{\boldsymbol{\alpha}}_K \\ \vec{\mathbf{a}}_K \end{bmatrix} \quad (2.12)$$

Die Beschleunigung \mathbf{x}_K geht direkt durch Ableitung aus Gl. (2.5) hervor.

$$\dot{\mathbf{y}}_K = \mathbf{x}_K \quad (2.13)$$

Analog zur Kinemate kann Gl. (2.11) als Transformation des Bezugspunktes der Beschleunigung verstanden werden, welche sich zu

$$\mathbf{x}_P = \mathbf{G}_{PK} \cdot \mathbf{x}_K + \boldsymbol{\xi}_{PK} \quad (2.14)$$

ergibt, wobei gilt:

$$\boldsymbol{\xi}_{PK} = \begin{bmatrix} \vec{\mathbf{0}} \\ \vec{\boldsymbol{\omega}}_K \times (\vec{\boldsymbol{\omega}}_K \times \vec{\mathbf{r}}_{KP}) \end{bmatrix}.$$

Mit den Bewegungsgrößen Pose \mathbf{z}_K , Kinemate \mathbf{y}_K und Beschleunigung \mathbf{x}_K kann die Kinematik eines starren Körpers vollständig beschrieben werden.

2.1.2. KINETIK DES UNGEBUNDENEN STARREN KÖRPERS

Im Folgenden soll nun auf die Kinetik eines starren Körpers eingegangen werden. Sie setzt sich mit der Ursache der Änderung der Bewegungsgrößen, hervorgerufen durch die Einwirkung von Kräften und Momenten, auseinander.

LAST

Abb. 2.2 zeigt ein Linienelement, an dessen Ende K eine Kraft \vec{f}_K und ein Moment $\vec{\tau}_K$ angreifen. Ähnlich der Kinemate und der Beschleunigung, werden diese beiden Größen als *Last* bezüglich des Punktes K

$$\mathbf{l}_K = \begin{bmatrix} \vec{\tau}_K \\ \vec{f}_K \end{bmatrix} \quad (2.15)$$

zusammengefasst. Mithilfe des statischen Kräfte- und Momentengleichgewichts

$$\vec{f}_P = \vec{f}_K \quad (2.16a)$$

$$\vec{\tau}_P = \vec{\tau}_K + \vec{r}_{PK} \times \vec{f}_K \quad (2.16b)$$

lässt sich eine äquivalente Last \mathbf{l}_P mit derselben Kraftwirkung ableiten. Gl. (2.16) lässt sich kurz schreiben als

$$\mathbf{l}_P = \mathbf{G}_{PK}^{-T} \cdot \mathbf{l}_K. \quad (2.17)$$

Die Ähnlichkeit zur Transformation der Kinemate, siehe Gl. (2.9), ist eine Folge aus der Forderung, dass die virtuelle Leistung δP_P bzw. δP_K gleich sein muss, da die beiden Lasten äquivalent sind.

$$\delta P_P = \delta \mathbf{y}_P^T \cdot \mathbf{l}_P \quad (2.18a)$$

$$= (\mathbf{G}_{PK} \delta \mathbf{y}_K)^T \cdot (\mathbf{G}_{PK}^{-T} \mathbf{l}_K) \quad (2.18b)$$

$$= \delta \mathbf{y}_K^T (\mathbf{G}_{PK}^T \mathbf{G}_{PK}^{-T}) \mathbf{l}_K \quad (2.18c)$$

$$= \delta \mathbf{y}_K^T \cdot \mathbf{l}_K \quad (2.18d)$$

$$= \delta P_K \quad (2.18e)$$

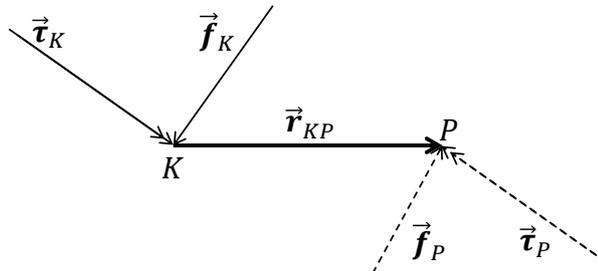


Abbildung 2.2.: Statisches Kräftegleichgewicht an einem Liniensegment

IMPULS UND DRALL

Der Impuls $d\vec{p}_P$ eines differenziell kleinen Massenelements, das sich am Punkt P befindet ist definiert als das Produkt aus seiner Geschwindigkeit \vec{v}_P und seiner Masse dm . Wird ein Starrkörper als eine Menge von differenziell kleinen Massenelementen betrachtet, so ergibt sich dessen Gesamtimpuls \vec{p} zu

$$\begin{aligned}\vec{p} &= \int_{P \in K} d\vec{p}_P \\ &= \int_{P \in K} \vec{v}_P dm \\ &= \int_{P \in K} (\vec{v}_S + \vec{\omega}_K \times \vec{r}_{SP}) dm \\ &= \vec{v}_S \int_{P \in K} dm + \vec{\omega}_K \times \int_{P \in K} \vec{r}_{SP} dm.\end{aligned}$$

Wird der Schwerpunkt S des Körpers als Bezugspunkt gewählt, verschwindet der Ausdruck $\int_{P \in K} \vec{r}_{SP} dm$ und es folgt

$$\vec{p} = m\vec{v}_S. \quad (2.19)$$

Analog zum Impuls \vec{p} kann der Drall $\vec{\pi}_S$ bezüglich eines Punktes S als Integral über alle differenziell kleinen Teilchen der Masse dm und des Dralls $d\vec{\pi}_{S,P}$ berechnet werden.

$$\begin{aligned}\vec{\pi}_S &= \int_{P \in K} d\vec{\pi}_{S,P} \\ &= \int_{P \in K} \vec{r}_{SP} \times d\vec{p}_P \\ &= \int_{P \in K} \vec{r}_{SP} \times (\vec{v}_S + \vec{\omega}_K \times \vec{r}_{SP}) dm \\ &= -\vec{v}_S \times \underbrace{\int_{P \in K} \vec{r}_{SP} dm}_{=\vec{0}} + \int_{P \in K} \vec{r}_{SP} \times (\vec{\omega}_K \times \vec{r}_{SP}) dm \\ &= \int_{P \in K} \overset{\sim}{\vec{r}}_{SP} \overset{\sim}{\vec{r}}_{SP}^T dm \cdot \vec{\omega}_K \\ &= \overset{\Rightarrow}{\Theta}_S \cdot \vec{\omega}_K\end{aligned} \quad (2.20)$$

Dabei wird die Größe $\overset{\Rightarrow}{\Theta}_S$ als Trägheitstensor bezeichnet.

BEWEGUNGSGLEICHUNG DES UNGEBUNDENEN STARREN KÖRPERS

Impuls- und Drallsatz besagen, dass die Änderung des Dralls bzw. des Impulses gerade dem äußeren Moment $\vec{\tau}_{S,ext}$ respektive der äußeren Kraft \vec{f}_{ext}

$$\begin{aligned}\frac{d}{dt}\vec{\pi}_S &= \vec{\tau}_{S,ext} \\ \frac{d}{dt}\vec{p} &= \vec{f}_{ext}\end{aligned}$$

entsprechen, siehe Abb. 2.3.

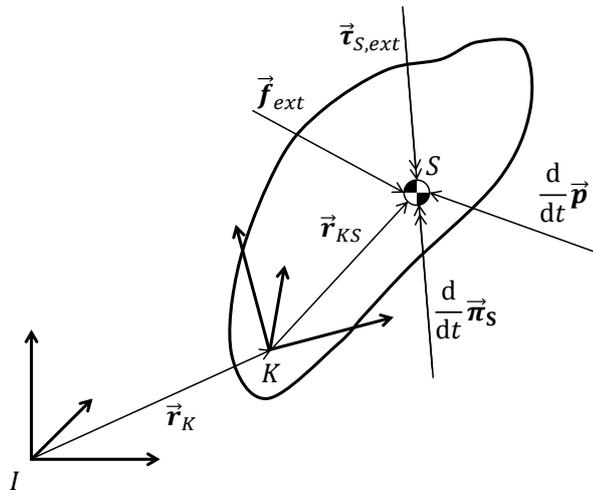


Abbildung 2.3.: Kräftegleichgewicht am ungebundenen Körper

Werden die Beziehungen für den Impuls (2.19) und den Drall (2.20) eingesetzt und gilt, dass die Masse m konstant ist und sich der Trägheitstensor $\vec{\Theta}_S$ bezüglich des körperfesten Systems K nicht ändert, also $\frac{d^K}{dt}\vec{\Theta}_S = \vec{0}$, folgt die Bewegungsgleichung eines ungebundenen starren Körpers bezüglich seines Schwerpunkts S . Diese sind als *Newton-Euler-Gleichungen* bekannt.

$$\begin{aligned}\vec{\Theta}_S \cdot \vec{\alpha}_K + \vec{\omega}_K \times \left(\vec{\Theta}_S \cdot \vec{\omega}_K \right) &= \vec{\tau}_{S,ext} \\ m\vec{a}_S &= \vec{f}_{ext}.\end{aligned}$$

Durch Einführen von Beschleunigung und Last nach Gl. (2.12) und (2.15) ergibt sich eine kompaktere Schreibweise

$$\mathbf{M}_S \cdot \mathbf{x}_S + \mathbf{h}_S = \mathbf{l}_{S,ext}^* \quad (2.21)$$

mit der 6×6 Massenmatrix \mathbf{M}_S bezüglich S

$$\mathbf{M}_S = \begin{bmatrix} \vec{\Theta}_S & \vec{0} \\ \vec{0} & m\mathbf{E} \end{bmatrix}$$

und dem Vektor der geschwindigkeitsproportionalen Trägheitsterme \mathbf{h}_S bezüglich S

$$\mathbf{h}_S = \begin{bmatrix} \vec{\omega}_K \times \left(\overset{\rightrightarrows}{\Theta}_S \cdot \vec{\omega}_K \right) \\ \vec{0} \end{bmatrix}.$$

Unter Nutzung der Transformationsbeziehungen (2.14) und (2.17) für die Beschleunigung und die Lasten, können die Bewegungsgleichungen für einen beliebigen anderen körperfesten Bezugspunkt K formuliert werden. Dazu werden beide Seiten der Newton-Euler-Gleichung (2.21), entsprechend der Transformation der Last aus Gleichung (2.17)

$$\mathbf{l}_K = \mathbf{G}_{KS}^{-T} \cdot \mathbf{l}_S$$

von links mit \mathbf{G}_{KS}^{-T} multipliziert. Anschließend wird die Transformationsbeziehung (2.14)

$$\mathbf{x}_K = \mathbf{G}_{KS} \cdot \mathbf{x}_S + \boldsymbol{\xi}_{KS}$$

zum Umrechnen der Beschleunigung vom Punkt K auf den Schwerpunkt S eingesetzt

$$\mathbf{G}_{KS}^{-T} \cdot \left(\mathbf{M}_S \cdot \underbrace{\left(\mathbf{G}_{KS}^{-1} (\mathbf{x}_K - \boldsymbol{\xi}_{KS}) \right)}_{=\mathbf{x}_S} + \mathbf{h}_S \right) = \underbrace{\mathbf{G}_{KS}^{-T} \cdot \mathbf{l}_{S,ext}^*}_{=\mathbf{l}_{K,ext}^*}$$

und es folgt die Bewegungsgleichung des ungebundenen starren Körpers für einen beliebigen Bezugspunkt K

$$\mathbf{M}_K \cdot \mathbf{x}_K + \mathbf{h}_K = \mathbf{l}_{K,ext}^* \quad (2.22)$$

mit

$$\begin{aligned} \mathbf{M}_K &= \begin{bmatrix} \overset{\rightrightarrows}{\Theta}_K & m \vec{r}_{SK} \\ m \vec{r}_{SK}^T & m \mathbf{E} \end{bmatrix} \\ \mathbf{h}_K &= \begin{bmatrix} \vec{\omega}_K \times \left(\overset{\rightrightarrows}{\Theta}_K \cdot \vec{\omega}_K \right) \\ \vec{\omega}_K \times \left(m \vec{r}_{SK} \cdot \vec{\omega}_K \right) \end{bmatrix} \\ \overset{\rightrightarrows}{\Theta}_K &= \overset{\rightrightarrows}{\Theta}_S + m \vec{r}_{SK}^T \vec{r}_{SK}. \end{aligned} \quad (2.23)$$

Die dabei entstandene Gleichung (2.23) ist unter dem Namen *Satz von Steiner* bekannt und dient zur Umrechnung des Trägheitstensors vom Schwerpunkt auf einen beliebigen Bezugspunkt. Im Folgenden wird für die Formulierung der Bewegungsgleichung immer ein vom Schwerpunkt verschiedener Bezugspunkt verwendet, siehe Gl. (2.24). Dies erlaubt eine deutlich kompaktere und effizientere Auswertung der gesamten Bewegungsgleichungen eines Mehrkörpersystems. Eine Diskussion der Wahl des Bezugspunktes folgt in Abschnitt 3.1.5.

Da für den weiteren Verlauf der Arbeit die Unterscheidung zwischen \mathbf{h}_K und $\mathbf{l}_{K,ext}^*$ nicht von Bedeutung ist, werden diese beiden Größen, zugunsten einer kompakteren Notation, in dem Vektor $\mathbf{l}_{K,ext} = \mathbf{l}_{K,ext}^* - \mathbf{h}_K$ vereinigt und Gl. (2.22) vereinfacht sich zu

$$\mathbf{M}_K \cdot \mathbf{x}_K = \mathbf{l}_{K,ext}. \quad (2.24)$$

Diese soll im nächsten Abschnitt für einen gebundenen Körper erweitert werden.

BEWEGUNGSGLEICHUNG DES GEBUNDENEN STARREN KÖRPERS

Wird ein Mehrkörpersystem betrachtet, so sind die Körper meist über Bindungen bzw. Gelenke miteinander verbunden. Sie schränken die Bewegungsmöglichkeiten ein und erzeugen Bindungslasten. Diese müssen zusätzlich in Gl. (2.24) berücksichtigt werden, um auf die Bewegungsgleichung des gebundenen starren Körpers zu gelangen.

Zur Bestimmung der Bindungslasten wird der Körper, wie in Abb. 2.4, freigeschnitten. Für jede angrenzende Bindung i entsteht dabei eine Bindungs- oder auch Schnittlast \mathbf{l}_i , die mithilfe der Transformationsmatrix \mathbf{G}_{iK}^T auf den Körperbezugspunkt K transformiert werden kann.

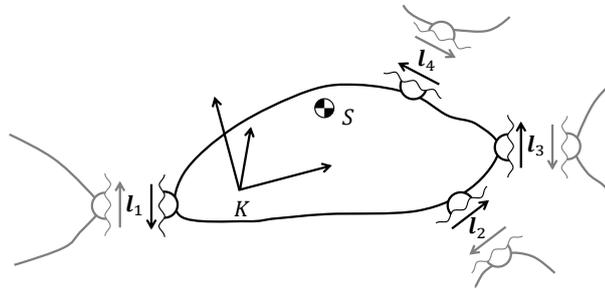


Abbildung 2.4.: Freigeschnittener gebundener Körper

Die Bewegungsgleichung des gebundenen Körpers lautet somit

$$\mathbf{M}_K \cdot \mathbf{x}_K = \mathbf{l}_{K,ext} + \sum_i \sigma_{iK} \mathbf{G}_{iK}^T \cdot \mathbf{l}_i. \quad (2.25)$$

Die σ_{iK} bestimmen das Vorzeichen, mit der die Schnittlast \mathbf{l}_i in die Bewegungsgleichung eingeht. Dies hängt davon ab, ob das positive oder das negative Schnittufer an den Körper angrenzt.

Somit wurde die erste der drei Gleichungen bestimmt, aus denen sich die Bewegungsgleichung eines Mehrkörpersystems zusammensetzt.

2.2. BESCHREIBUNG EINER BINDUNG

Wie bereits im vorangegangenen Abschnitt erwähnt, müssen zur Abbildung komplexer mechanischer Strukturen mithilfe eines Mehrkörpersystems zwischen Körpern bzw. mit ihnen fest verbundenen Koordinatensystemen Bindungen eingeführt werden. Die durch sie verursachten Einschränkungen der Bewegungsfreiheit zweier verbundener Körper wird mit der sogenannten Bindungsgleichung beschrieben, welche in diesem Kapitel hergeleitet werden soll.

2.2.1. BINDUNGSGLEICHUNG

Abb. 2.5 zeigt zwei Koordinatensysteme V und N . Das System V ist fest mit dem unteren Körper verbunden, der in diesem Kontext als Vorgänger verstanden wird. Auf dem nachfolgenden Körper sei das Koordinatensystem N fixiert.

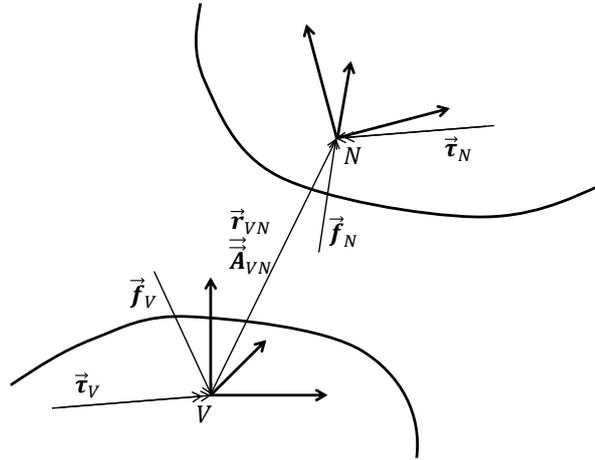


Abbildung 2.5.: Relative Lage zweier Koordinatensysteme

Der Abstand ihrer Ursprünge beträgt

$$\vec{r}_{VN} = \vec{r}_N - \vec{r}_V.$$

Die Orientierungen der Koordinatensysteme seien durch ihre zwei Basen $\vec{e}_{V,i}$ und $\vec{e}_{N,i}$ gegeben, die über den Drehtensor \vec{A}_{VN} miteinander in Beziehung stehen.

$$\vec{e}_{V,i} = \vec{A}_{VN} \cdot \vec{e}_{N,i}$$

Analog Gl. (2.2) wird für die Bindung die relative Pose z_{VN} der Koordinatensysteme V und N definiert als

$$z_{VN} = z_N \ominus z_V = \begin{bmatrix} \mathbf{p}_{VN} \\ \vec{r}_{VN} \end{bmatrix},$$

wobei die Eulerparameter \mathbf{p}_{VN} den Drehtensor \vec{A}_{VN} parametrisieren.

Eine implizit formulierte zweiseitige Bindung ist eine Funktion der Dimension n_{ch}

$$\mathbf{g}(z_{VN}, t) = \mathbf{0} \in \mathbb{R}^{n_{ch}}, \quad (2.26)$$

die den Drehtensor \vec{A}_{VN} bzw. die Eulerparameter \mathbf{p}_{VN} zusammen mit dem Abstandsvektor \vec{r}_{VN} und der Zeit t auf eine Menge von Bindungsverletzungen abbildet. Die Forderung nach dem Verschwinden der Bindungsverletzungen schränkt die relative Pose ein und es werden auf diese Weise bestimmte Relativbewegungen untersagt. Einseitige Bindungen, die durch eine Ungleichung beschrieben werden, sollen in dieser Arbeit nicht behandelt werden. Für eine weiterführende Diskussion dieses Sachverhalts sei auf [PG96] verwiesen. Oft hängt \mathbf{g} nicht direkt von t ab, so dass $\frac{\partial \mathbf{g}}{\partial t} = \mathbf{0}$ gilt. In diesem Fall wird von einer *skleronomen* Bindung gesprochen. Andernfalls wird sie als *rheonome* Bindung bezeichnet. Die Wahl von \mathbf{g} ist jedoch nicht eindeutig. So kann es verschiedene implizite Formulierungen für dasselbe Gelenk geben.

Meist ist es möglich, die komplette Lösungsvielfalt von Gl. (2.26) explizit anzugeben. In diesem Fall wird die relative Pose

$$z_{VN} = \bar{\mathbf{g}}(\mathbf{q}, t) \quad (2.27)$$

durch eine Funktion $\bar{\mathbf{g}}$ der Zeit t und eines Parametervektors $\mathbf{q} \in \mathbb{R}^{n_q}$ beschrieben. Die Elemente von \mathbf{q} werden als *generalisierte Lagekoordinaten* der Bindung bezeichnet. Sie müssen jedoch nicht

zwingend unabhängig voneinander sein. Im Allgemeinen muss davon ausgegangen werden, dass sie gewissen Zwangsbedingungen

$$\boldsymbol{\chi}(\mathbf{q}) = \mathbf{0} \quad (2.28)$$

unterliegen, wie es der Fall wäre, wenn \mathbf{q} z.B. Eulerparameter zur Beschreibung einer Drehung enthalten würde. Demnach ergibt sich die Anzahl der Lagefreiheitsgrade zu $n_{fh} = \dim(\mathbf{q}) - \dim(\boldsymbol{\chi}) = 6 - n_{ch}$. Durch Einsetzen von Gleichung (2.27) in (2.26) vereinfacht sich letztere zu trivialen Gleichungen, die für jede mit der Zwangsbedingung (2.28) kompatible Wahl von \mathbf{q} erfüllt werden.

Gleichung (2.26) kann bezüglich der Zeit t abgeleitet werden. Die Ableitung wird vorerst bezüglich der Basis V vorgenommen und durch den hochgestellten Index V gekennzeichnet, also

$$\mathbf{y}_{VN}^V = \mathbf{H} \cdot \frac{d^V \mathbf{z}_{VN}}{dt}$$

Die Anwendung der Kettenregel und der Gl. (2.8) liefert

$$\begin{aligned} \mathbf{0} = \frac{d^V}{dt} \mathbf{g} &= \frac{\partial \mathbf{g}}{\partial \mathbf{z}_{VN}} \cdot \frac{d^V \mathbf{z}_{VN}}{dt} + \frac{\partial \mathbf{g}}{\partial t} \\ &= \frac{\partial \mathbf{g}}{\partial \mathbf{z}_{VN}} \cdot \mathbf{H}^{-1} \cdot \mathbf{y}_{VN}^V + \frac{\partial \mathbf{g}}{\partial t}. \end{aligned}$$

Werden die folgenden Größen

$$\begin{aligned} \mathbf{W}_h^T &= \frac{\partial \mathbf{g}}{\partial \mathbf{z}_{VN}} \cdot \mathbf{H}^{-1} \\ \boldsymbol{\zeta}_h &= -\frac{\partial \mathbf{g}}{\partial t} \end{aligned}$$

zusammengefasst, folgt eine verkürzte Schreibweise mit

$$\mathbf{W}_h^T \cdot \mathbf{y}_{VN}^V = \boldsymbol{\zeta}_h. \quad (2.29)$$

Zwischen den Kinemat \mathbf{y}_N und \mathbf{y}_V , beobachtet aus dem Inertialsystem, und der relativen Kinemate \mathbf{y}_{VN}^V , beobachtet aus dem Vorgängerkoordinatensystem V , gilt folgender Zusammenhang nach Gl. (A.11)

$$\begin{aligned} \mathbf{y}_N &= \mathbf{y}_V + \mathbf{y}_{VN}^V \\ \mathbf{y}_N &= \mathbf{y}_V + \mathbf{y}_{VN}^V + \begin{bmatrix} \vec{\mathbf{0}} \\ \vec{\boldsymbol{\omega}}_V \times \vec{\mathbf{r}}_{VN} \end{bmatrix} \\ \mathbf{y}_N &= \mathbf{G}_{NV}(\mathbf{z}_{VN}) \cdot \mathbf{y}_V + \mathbf{y}_{VN}^V. \end{aligned} \quad (2.30)$$

Die Einbettung einer Bindung in das Mehrkörpersystem wird zu einem späteren Zeitpunkt in Abschnitt 2.2.4 behandelt.

Gleichung (2.29) beschreibt die Menge aller *holonomen* Bindungen. Dies sind Bindungen, die die relative Lage und Geschwindigkeiten gleichermaßen einschränken. Hinzu kommen die *nichtholonomen* Bindungen, die lediglich Auswirkungen auf die relative Geschwindigkeit besitzen, da sie nicht integrierbar sind. Als Beispiel für nichtholonome Bindungen seien das ideale Rollen eines starren Rades und das ideale Gleiten eines Schlittschuhs genannt. Laut [RS88] können alle mechanischen nichtholonomen Bindungen angeschrieben werden als

$$\mathbf{W}_{nh}^T \cdot \mathbf{y}_{VN}^V = \boldsymbol{\zeta}_{nh} \quad (2.31)$$

und somit können Gl. (2.29) und (2.31) zusammengefasst werden zu

$$\mathbf{W}^T \cdot \mathbf{y}_{VN}^V = \boldsymbol{\zeta}' \quad (2.32)$$

unter der Verwendung von

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} \mathbf{W}_h & \mathbf{W}_{nh} \end{bmatrix} \\ \boldsymbol{\zeta}' &= \begin{bmatrix} \boldsymbol{\zeta}_h \\ \boldsymbol{\zeta}_{nh} \end{bmatrix}. \end{aligned}$$

An dieser Stelle sei zu beachten, dass \mathbf{W} und $\boldsymbol{\zeta}$ selbst nichtlineare Funktionen von \mathbf{z}_{VN} sein können. Im Folgenden wird angenommen, dass \mathbf{W} keine linear abhängigen Spalten, bzw. die Bindung keine redundanten Bindungsgleichungen, besitzt. Für eine ausführlichere Diskussion der Behandlung eines Rangabfalls sei auf [Rei97] verwiesen.

Da Gl. (2.32) linear bezüglich der relativen Kinemate \mathbf{y}_{VN}^V ist, lassen sich alle Lösungen des linearen Gleichungssystem als

$$\mathbf{y}_{VN}^V = \mathbf{J} \mathbf{s}' + \boldsymbol{\xi}' \quad (2.33)$$

für einen beliebigen Parametervektor $\mathbf{s}' \in \mathbb{R}^{n_f}$ schreiben. Daher wird \mathbf{s}' als Vektor der *generalisierten Geschwindigkeiten* bezeichnet und es gilt $n_f = 6 - n_c$. Die Jacobimatrix \mathbf{J} kann als Zusammenstellung der Vektoren der freien Richtungen verstanden werden. Da es sich bei Gl. (2.33) um eine Lösung von Gl. (2.32) handelt, muss sie auch für $\mathbf{s} = \mathbf{0}$ gelten. Dadurch entsteht eine Beziehung für $\boldsymbol{\xi}$ mit

$$\mathbf{W}^T \boldsymbol{\xi}' = \boldsymbol{\zeta}'.$$

$\boldsymbol{\xi}$ stellt die inhomogene Lösung der Gl. (2.32). Für die homogene Lösung muss

$$\mathbf{W}^T \mathbf{J} = \mathbf{0} \quad (2.34)$$

gelten, da sie für alle \mathbf{s}' erfüllt sein soll.

Die impliziten und expliziten Bindungsgleichungen (2.26), (2.32) und (2.27), (2.33) auf Lage- und Geschwindigkeitsebene bilden die Grundlage für die zweite Klasse der Gleichungen aus denen sich die Bewegungsgleichung eines Mehrkörpersystems zusammensetzt. Sie werden im Abschnitt 2.2.3 wieder aufgegriffen. Mithilfe der dualen Basen werden sie umgeschrieben und anschließend im Abschnitt 2.2.4 so umformuliert, dass sie sich direkt auf die Bezugspunkte der angrenzenden Körper beziehen.

2.2.2. PRINZIP DER VIRTUELLEN LEISTUNG

Um die Integrität einer Bindung zwischen den angeschlossenen Körpern zu gewährleisten, können in den Bindungen Zwangslasten entstehen. Die in Kapitel 2.1.2 eingeführten Schnittlasten \mathbf{l} einer Bindung setzen sich aus diesen Zwangslasten \mathbf{l}_C und ggf. einer eingepprägten Last \mathbf{l}_A zusammen, die z.B. durch Effekte wie Reibung oder einen Antrieb verursacht werden kann.

$$\mathbf{l} = \mathbf{l}_C + \mathbf{l}_A$$

Das *Prinzip der virtuellen Leistung* besagt, dass die die virtuelle Leistung der Zwangslasten ver-

schwindet [Bud90, SE11]. Unter Benutzung der Symbole $\mathbf{l}_{V,C} = \begin{bmatrix} \vec{\tau}_{V,C} \\ \vec{\mathbf{f}}_{V,C} \end{bmatrix}$ und $\mathbf{l}_{N,C} = \begin{bmatrix} \vec{\tau}_{N,C} \\ \vec{\mathbf{f}}_{N,C} \end{bmatrix}$ für die wirkenden Zwangslasten an dem Vorgänger V und dem Nachfolger N einer Bindung, muss demnach

$$\delta \mathbf{y}_V^T \cdot \mathbf{l}_{V,C} + \delta \mathbf{y}_N^T \cdot \mathbf{l}_{N,C} = 0 \quad (2.35)$$

gelten. Die virtuelle Geschwindigkeit des Nachfolgers $\delta \mathbf{y}_N$ hängt jedoch mit der des Vorgängers zusammen. Unter Ausnutzung der Gleichungen (2.30) und (2.33) lässt sich folgender Zusammenhang finden

$$\delta \mathbf{y}_N = \mathbf{G}_{NV} \cdot \delta \mathbf{y}_V + \mathbf{J} \cdot \delta \mathbf{s}'.$$

Wird dies in Gl. (2.35) eingesetzt, resultiert

$$\delta \mathbf{y}_V^T \cdot \left(\mathbf{l}_{V,C} + \mathbf{G}_{NV}^T \cdot \mathbf{l}_{N,C} \right) + \delta \mathbf{s}'^T \cdot \mathbf{J}^T \cdot \mathbf{l}_{N,C} = 0.$$

Die Zwangslasten $\mathbf{l}_{V,C}$ und $\mathbf{l}_{N,C}$ hängen entsprechend den Gesetzen der Statik, siehe Gl. (2.17), miteinander zusammen. Daher muss der Ausdruck $\left(\mathbf{l}_{V,C} + \mathbf{G}_{NV}^T \cdot \mathbf{l}_{N,C} \right)$ und mit ihm der erste Summand verschwinden. Alle Komponenten der $\delta \mathbf{s}'$ sind voneinander unabhängig. Daher muss zusätzlich der Koeffizient $\mathbf{J}^T \mathbf{l}_{N,C}$ verschwinden, um die Gleichung für alle $\delta \mathbf{s}'$ zu erfüllen. Es folgt damit die zentrale Aussage

$$\mathbf{J}^T \mathbf{l}_{N,C} = 0 \quad (2.36)$$

$$\mathbf{J}^T (\mathbf{l}_N - \mathbf{l}_{N,A}) = 0$$

$$\mathbf{J}^T \mathbf{l}_N = \boldsymbol{\eta}', \quad (2.37)$$

wobei

$$\boldsymbol{\eta}' = \mathbf{J}^T \cdot \mathbf{l}_{N,A}$$

die Projektion der von außen wirkenden aktiven Lasten $\mathbf{l}_{N,A}$ auf die freien Bewegungsrichtungen darstellt. Anschaulich besagt Gl. (2.37), dass ein ideales Gelenk entlang der freien Richtungen mit Ausnahme der eingepprägten Lasten keine Kräfte oder Momente überträgt.

Gleichung (2.37) ist Bestandteil der Bewegungsgleichungen des Mehrkörpersystems und soll im folgenden Abschnitt durch Nutzung der dualen Basen leicht umformuliert werden.

2.2.3. DUALE BASEN EINER BINDUNG

Für das spätere Arbeiten mit den Bewegungsgleichungen eines Mehrkörpersystems ist es hilfreich, analog zu [RS88] für die Kinemate \mathbf{y}_{VN}^V und die Gelenklast \mathbf{l}_N zwei spezielle Basen bestehend aus den Basisvektoren $\boldsymbol{\phi}_i$ bzw. $\boldsymbol{\psi}_i$, $i = 1 \dots 6$ einzuführen. Um eine kompaktere Darstellung zu erlangen, werden die ersten n_f (Anzahl der Freiheitsgrade des Gelenks) und die verbleibenden Basisvektoren zu den Matrizen

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_i], \quad i = 1 \dots n_f$$

$$\overline{\boldsymbol{\Phi}} = [\boldsymbol{\phi}_i], \quad i = (n_f + 1) \dots 6$$

$$\boldsymbol{\Psi} = [\boldsymbol{\psi}_i], \quad i = 1 \dots n_f$$

$$\overline{\boldsymbol{\Psi}} = [\boldsymbol{\psi}_i], \quad i = (n_f + 1) \dots 6$$

zusammengefasst. Sie sollen die besondere Eigenschaft aufweisen, dass die Basis $[\Psi, \bar{\Psi}]$ gerade die duale Basis [LCE10] zu $[\Phi, \bar{\Phi}]$ darstellt. Dies bedeutet, dass

$$[\Psi, \bar{\Psi}]^T \cdot [\Phi, \bar{\Phi}] = \begin{bmatrix} E & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix} \quad (2.38)$$

ist. Bilden die $[\Phi, \bar{\Phi}]$ selbst eine orthonormale Basis, so stellen sie zugleich die duale Basis zu sich selbst dar.

Jeder Vektor des Vektorraums und damit auch die Kinemate \mathbf{y}_{VN}^V und die Gelenklast \mathbf{l}_N können bezüglich den neu eingeführten Basen angeschrieben werden.

$$\mathbf{y}_{VN}^V = \Phi \mathbf{s} + \bar{\Phi} \bar{\mathbf{s}} \quad (2.39)$$

$$\mathbf{l}_N = \Psi \boldsymbol{\lambda} + \bar{\Psi} \bar{\boldsymbol{\lambda}} \quad (2.40)$$

\mathbf{s} , $\bar{\mathbf{s}}$, $\boldsymbol{\lambda}$ und $\bar{\boldsymbol{\lambda}}$ stellen die entsprechenden Koordinatenvektoren dar. Die in Φ und $\bar{\Psi}$ enthaltenen Basisvektoren können frei unter Beachtung von Gl. (2.38) gewählt werden. Zusätzlich soll

$$\text{span}(\Phi) = \text{span}(\mathbf{J}) \quad (2.41)$$

$$\text{span}(\bar{\Psi}) = \text{span}(\mathbf{W}) \quad (2.42)$$

gelten. Sind alle Vektoren in \mathbf{J} und \mathbf{W} orthogonal zueinander, kann direkt $\Phi = \mathbf{J}$ und $\bar{\Psi} = \mathbf{W}$ gewählt werden. Das Auffinden der dualen Basen ist für jede Bindung möglich.

Wird nun die Darstellung der Kinemate bzgl. der neuen Basis (2.39) in die implizite Bindungsgleichung auf Geschwindigkeitsebene (2.32) eingebunden, so folgt aufgrund der Eigenschaften (2.42) und (2.38)

$$(\mathbf{W}^T \bar{\Phi}) \cdot \bar{\mathbf{s}} = \zeta'$$

und damit eine Bestimmungsgleichung für die Koordinaten $\bar{\mathbf{s}}$. Die Koordinaten \mathbf{s} , zugehörig zu den Basisvektoren Φ , werden von der impliziten Bindungsgleichung nicht eingeschränkt und bleiben frei. Daher wird im Falle der Vektoren in Φ von den *freien Bewegungsrichtungen* gesprochen, während $\bar{\Phi}$ die *gesperrten Bewegungsrichtungen* beschreibt. Analog kann die neue Darstellung der Zwangslasten (2.40) in das Prinzip der virtuellen Leistung (2.37) eingesetzt werden. Mit

$$(\mathbf{J}^T \Psi) \cdot \boldsymbol{\lambda} = \eta' \quad (2.43)$$

ergibt sich eine Bestimmungsgleichung für die Koordinaten $\boldsymbol{\lambda}$, während die $\bar{\boldsymbol{\lambda}}$ frei bleiben. Folglich stellen die Ψ die *freien Lastrichtungen* und die $\bar{\Psi}$ die *gesperrten Lastrichtungen* dar. Ist die Basis $[\Phi, \bar{\Phi}]$ orthonormal gewählt, was für viele Gelenke möglich ist, fallen jeweils die gesperrten und freien Bewegungs- und Lastrichtungen zusammen, also $\Phi = \Psi$ und $\bar{\Phi} = \bar{\Psi}$.

Da aufgrund von Gl. (2.42) die Matrizen $\bar{\Psi}$ und \mathbf{W} den gleichen Raum aufspannen, kann die implizite Bindungsgleichung (2.32) auf die neue Basis umgeschrieben werden zu

$$\bar{\Psi}^T \cdot \mathbf{y}_{VN}^V = \zeta. \quad (2.44)$$

Analog kann für die explizite Bindungsgleichung vorgegangen werden. Es folgt

$$\mathbf{y}_{VN}^V = \Phi \mathbf{s} + \boldsymbol{\xi}, \quad (2.45)$$

wobei die neuen generalisierten Geschwindigkeiten \mathbf{s} eine Linearkombination der alten \mathbf{s}' darstellen. Der Vorteil der dualen Basen kann anhand der neuen expliziten Bindungsgleichung demonstriert werden. Wird diese auf die gesperrten Lastrichtungen $\bar{\Psi}$ projiziert, so ergibt sich die implizite Bindungsgleichung. Wird sie hingegen auf die freien Lastrichtungen Ψ projiziert, so resultiert dies in einer Vorschrift zur Berechnung der generalisierten Geschwindigkeiten \mathbf{s} aus der relativen Kinemate.

Zuletzt kann auch das Prinzip der virtuellen Leistung (2.37) mit der neuen Basis dargestellt werden als

$$\Phi^T \mathbf{l}_N = \boldsymbol{\eta}. \quad (2.46)$$

Diese Gleichung besagt anschaulich, dass Bindungen entlang der freien Bewegungsrichtungen keine Kräfte oder Momente übertragen, sofern sie nicht angetrieben sind ($\boldsymbol{\eta} \neq \mathbf{0}$). Dies stellt die dritte der drei Gleichungen dar, aus denen sich die Bewegungsgleichung eines Mehrkörpersystems zusammensetzt.

2.2.4. GLOBALE FORM DER BINDUNGSGLEICHUNGEN

Für die Beschreibung der Bewegungsgleichungen eines Mehrkörpersystems ist es von Vorteil, wenn die Bindungsgleichungen nicht die relativen kinematischen Größen zwischen dem Vorgängerkoordinatensystem V und dem Nachfolgerkoordinatensystem N , sondern die absoluten Größen der Bezugspunkte der beteiligten Starrkörper beschreiben, siehe Abb. 2.6.

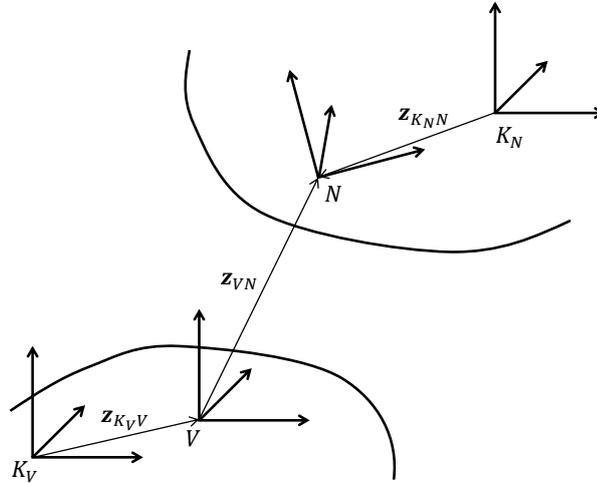


Abbildung 2.6.: Bindung bezogen auf das Gesamtsystem

Auf Lageebene ergibt sich die relative Pose \mathbf{z}_{VN} zu

$$\begin{aligned} \mathbf{z}_{VN} &= \mathbf{z}_N \ominus \mathbf{z}_V \\ &= (\mathbf{z}_{K_N} \oplus \mathbf{z}_{K_N,N}) \ominus (\mathbf{z}_{K_V} \oplus \mathbf{z}_{K_V,V}). \end{aligned}$$

Alle holonomen Bindungen können damit wie folgt beschrieben werden

$$\begin{cases} \mathbf{0} = \mathbf{g}(\mathbf{z}_{K_V}, \mathbf{z}_{K_N}, t) & \text{implizit formulierte Bindung} \\ \mathbf{z}_{K_N} = \bar{\mathbf{g}}(\mathbf{z}_{K_V}, \mathbf{q}, t) & \text{explizit formulierte Bindung.} \end{cases} \quad (2.47)$$

2. Mechanik der Mehrkörpersysteme

Die somit global formulierten Bindungsgleichungen liefern den zweiten Gleichungstyp, der in den Bewegungsgleichungen des Mehrkörpersystems vorkommt.

Da die Bewegungsgleichungen mit den Bindungsgleichungen auf Lageebene schwierig auszuwerten sind, werden an deren Stelle oft die Bindungsgleichungen auf Geschwindigkeits- oder Beschleunigungsebene verwendet. Diese sollen im Folgenden zusammengetragen werden.

Die relative Kinemate \mathbf{y}_{VN} kann unter Anwendung von Gl. (2.9) wie folgt berechnet werden

$$\mathbf{y}_{VN} = \mathbf{G}_{N,K_N} \cdot \mathbf{y}_{K_N} - \mathbf{G}_{V,K_V} \cdot \mathbf{y}_{K_V}. \quad (2.48)$$

Da in dem vorangegangenen Abschnitt lediglich die Ableitung $\frac{d^V}{dt}$, beobachtet aus dem Koordinatensystem V , berechnet wurde, muss diese noch auf eine Ableitung bezüglich des ruhenden Systems O umgerechnet werden. Nach einigen Umformungen folgt die Vorschrift

$$\mathbf{G}_{N,K_N} \cdot \mathbf{y}_{K_N} = \mathbf{G}_{N,K_V} \cdot \mathbf{y}_{K_V} + \mathbf{y}_{VN}^V. \quad (2.49)$$

Die Bewegung des Nachfolgekörpers mit der Kinemate \mathbf{y}_{K_N} setzt sich demnach aus zwei Bewegungen zusammen. Der erste Anteil beschreibt die Bewegung, die der Nachfolgekörper ausführen würde, wenn er fest mit dem Vorgänger verbunden wäre. Der zweite Anteil enthält die Bewegung des Nachfolgekörpers, die durch das Gelenk bei einem ruhenden Vorgängerkörper verursacht wird. Durch Einsetzen der expliziten Bindungsgleichung (2.45) in Gleichung (2.49) folgt mit

$$\mathbf{G}_{N,K_N} \cdot \mathbf{y}_{K_N} = \mathbf{G}_{N,K_V} \cdot \mathbf{y}_{K_V} + \Phi \mathbf{s} + \xi \quad (2.50)$$

die Geschwindigkeitsformulierung der Bindungsgleichungen, die auch die nichtholonomen Gelenke enthalten kann. Auf Geschwindigkeitsebene ist es nicht mehr nötig, zwischen expliziten und impliziten Bindungsgleichungen zu unterscheiden, da die implizite Bindungsgleichung enthalten ist. Sie kann durch Multiplikation mit $\bar{\Psi}^T$ von links sofort gewonnen werden. Darin liegt der Vorteil in der Verwendung der dualen Basen.

Durch Ableiten von Gl. (2.50) nach der Zeit folgt die Bindungsgleichung auf Beschleunigungsebene mit

$$\mathbf{G}_{N,K_N} \cdot \mathbf{x}_{K_N} = \mathbf{G}_{N,K_V} \cdot \mathbf{x}_{K_V} + \Phi \dot{\mathbf{s}} + \mathbf{c}, \quad (2.51)$$

wobei der Vektor \mathbf{c} alle Terme zusammenfasst, die nicht direkt von den Beschleunigungen \mathbf{x} und $\dot{\mathbf{s}}$ abhängen. Der Vektor \mathbf{c} setzt sich zusammen aus

$$\mathbf{c} = \mathbf{c}_G + \mathbf{c}_\Phi + \dot{\xi}. \quad (2.52)$$

Der Anteil $\mathbf{c}_\Phi = \left(\frac{d^O}{dt} \Phi\right) \mathbf{s}$ geht direkt aus dem Gelenk hervor, wohingegen der Anteil \mathbf{c}_G aus der Ableitung der Transformationsmatrix der Beschleunigung stammt. Letzterer berechnet sich zu

$$\begin{aligned} \mathbf{c}_G &= \left(\frac{d^O}{dt} \mathbf{G}_{N,K_V}\right) \mathbf{y}_{K_V} - \left(\frac{d^O}{dt} \mathbf{G}_{N,K_N}\right) \cdot \mathbf{y}_{K_N} \\ &= \left[\begin{array}{c} \vec{\mathbf{0}} \\ \vec{\omega}_V \times \left(\tilde{\vec{\mathbf{r}}}_{N,K_V} \cdot \vec{\omega}_V\right) - \vec{\omega}_N \times \left(\tilde{\vec{\mathbf{r}}}_{N,K_N} \cdot \vec{\omega}_N\right) \end{array} \right]. \end{aligned}$$

Mit den hier gewonnenen Bindungsgleichungen auf Lage-, Geschwindigkeits und Beschleunigungsebene, die die kinematischen Größen der angrenzenden Körperbezugspunkte enthalten, wurden

alle relevanten Gleichungen für den zweiten Bestandteil der Bewegungsgleichung eines Mehrkörpersystems vorgestellt. Letzteres soll im Anschluss an den folgenden Abschnitt in 2.3 aufgebaut werden, in dem die Bindungsgleichungen für zwei Beispiele erarbeitet werden.

2.2.5. BEISPIELE FÜR BINDUNGEN IN EINEM MEHRKÖRPERSYSTEM

An dieser Stelle sollen die Bindungsgleichungen und speziell die dualen Basen anhand von zwei praktischen Beispielen, dem Dreh- und dem Kreuzgelenk, vorgestellt werden.

Drehgelenk Es wird angenommen, dass die körperfesten Koordinatensysteme V des Vorgängers und N des Nachfolgers durch die orthonormalen Basisvektoren \vec{a}_i bzw. \vec{b}_i , $i = 1 \dots 3$ beschrieben werden. Ohne Beschränkung der Allgemeinheit soll für die Drehachse \vec{n}

$$\vec{n} = \vec{a}_3 = \vec{b}_3$$

gelten. Die implizite Zwangsbedingung \mathbf{g} des einachsigen Drehgelenks, siehe Abb. 2.7, lautet

$$\mathbf{g}(z_{VN}) = \begin{bmatrix} 0, \vec{b}_1 \cdot \mathbf{p}_{VN} \\ 0, \vec{b}_2 \cdot \mathbf{p}_{VN} \\ \vec{r}_{VN} \end{bmatrix} = \mathbf{0}.$$

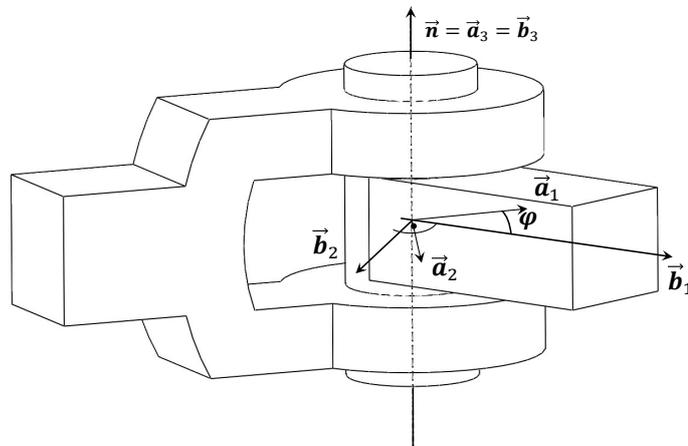


Abbildung 2.7.: Drehgelenk

Ebenso ist es möglich, eine explizite Formulierung zu finden. Dazu wird der Drehwinkel $q = \varphi$ zwischen den Basisvektoren \vec{a}_1 und \vec{b}_1 eingeführt.

$$z_{VN} = \begin{bmatrix} \cos \frac{q}{2} \\ \sin \frac{q}{2} \cdot \vec{n} \\ \mathbf{0} \end{bmatrix}$$

Auf Geschwindigkeitsebene lautet die implizite Zwangsbedingung

$$\underbrace{\begin{bmatrix} \vec{0} & \vec{b}_1 & \vec{b}_2 \\ \vec{E} & \vec{0} & \vec{0} \end{bmatrix}^T}_{\mathbf{W}^T} \mathbf{y}_{VN}^V = \mathbf{0}$$

bzw. in der expliziten Formulierung mit der Winkelgeschwindigkeit $s = \omega = \dot{\varphi}$

$$\mathbf{y}_{VN}^V = \underbrace{\begin{bmatrix} \vec{n} \\ \vec{0} \end{bmatrix}}_{\mathbf{J}} s.$$

Die duale Basis entspricht nun den Spaltenvektoren in \mathbf{J} und \mathbf{W} .

$$\begin{aligned} \Phi &= \Psi = \mathbf{J} \\ \bar{\Phi} &= \bar{\Psi} = \mathbf{W} \end{aligned}$$

Da das Gelenk skleronom ist, sind die Größen ζ und ξ identisch Null. Der von der Bindung abhängige Teil \mathbf{c}_Φ des Vektors \mathbf{c} , aus Gl. (2.52), kann wie folgt berechnet werden:

$$\begin{aligned} \mathbf{c}_\Phi &= \left(\frac{d^O}{dt} \Phi \right) s \\ &= \begin{bmatrix} \vec{\omega}_V \times \vec{n} \\ \vec{0} \end{bmatrix} s. \end{aligned}$$

Damit wurden alle nötigen Informationen zur Berechnung des Drehgelenkes hergeleitet.

Kreuzgelenk Ein Kreuzgelenk, siehe Abb. 2.8, besitzt zwei Drehfreiheitsgrade und kann als Reihenschaltung von zwei Drehgelenken mit den Drehachsen \vec{n}_1 und \vec{n}_2 beschrieben werden.

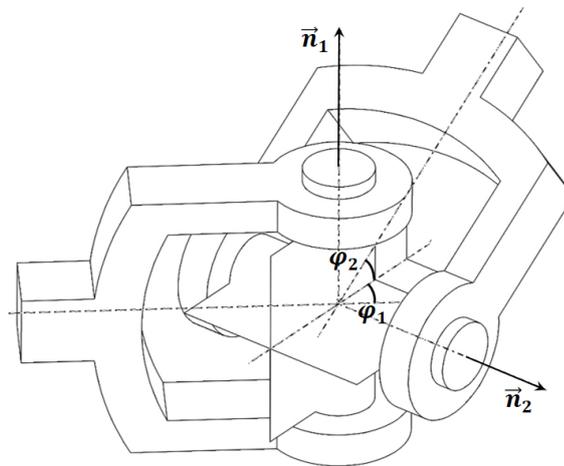


Abbildung 2.8.: Kreuzgelenk

Dabei stehen \vec{n}_1 und \vec{n}_2 senkrecht aufeinander. Letzteres definiert die implizite Bindungsgleichung

$$\mathbf{g}(z_{VN}) = \begin{bmatrix} \vec{n}_1 \cdot \vec{A}(p_{VN}) \cdot \vec{n}_2 \\ \vec{r}_{VN} \end{bmatrix} = \mathbf{0}.$$

Die explizite Bindungsgleichung ergibt sich aus der Aneinanderreihung der beiden Drehgelenke mit den Gelenkkoordinaten $\mathbf{q}^T = [\varphi_1, \varphi_2]^T$ zu

$$\begin{aligned} z_{VN} &= z_{VN,1} \oplus z_{VN,2} \\ &= \begin{bmatrix} \begin{bmatrix} \cos \frac{\varphi_1}{2} \\ \sin \frac{\varphi_1}{2} \vec{n}_1 \end{bmatrix} \star \begin{bmatrix} \cos \frac{\varphi_2}{2} \\ \sin \frac{\varphi_2}{2} \vec{n}_2 \end{bmatrix} \\ \vec{0} \end{bmatrix}. \end{aligned}$$

Mithilfe der Gelenkgeschwindigkeiten $\mathbf{s}^T = [\dot{\varphi}_1, \dot{\varphi}_2]^T$ lässt sich durch Addition der Drehgeschwindigkeiten beider Drehgelenke die relative Kinematik leicht bestimmen.

$$\mathbf{y}_{VN}^V = \underbrace{\begin{bmatrix} \vec{n}_1 & \vec{n}_2 \\ \vec{0} & \vec{0} \end{bmatrix}}_{\mathbf{J}} \mathbf{s}$$

Da die Spaltenvektoren in \mathbf{J} selbst orthonormal zueinander sind, können diese sofort als freie Bewegungsrichtungen Φ übernommen werden. Durch die Wahl der gesperrten Bewegungsrichtungen zu

$$\overline{\Phi} = \begin{bmatrix} \vec{0} & \vec{n}_1 \times \vec{n}_2 \\ \vec{E} & \vec{0} \end{bmatrix}$$

bildet $[\Phi, \overline{\Phi}]$ eine vollständige orthonormale Basis. Damit fallen die Kraft- und Bewegungsrichtungen zusammen und es gilt $\Psi = \Phi$ und $\overline{\Psi} = \overline{\Phi}$.

Der nichtlineare Anteil bei der Transformation der Beschleunigung \mathbf{c}_Φ lautet

$$\begin{aligned} \mathbf{c}_\Phi &= \left(\frac{d^O}{dt} \Phi \right) \mathbf{s} \\ &= \begin{bmatrix} \vec{\omega}_V \times \vec{n}_1 s_1 + (\vec{\omega}_V + \vec{n}_1 s_1) \times \vec{n}_2 s_2 \\ \vec{0} \end{bmatrix}. \end{aligned}$$

Das Kreuzgelenk ist damit vollständig bestimmt.

2.3. BEWEGUNGSGLEICHUNG EINES MEHRKÖRPERSYSTEMS

Unter einem Mehrkörpersystem wird ein System aus n_K Körpern verstanden, die über n_B Bindungen miteinander in Beziehung stehen. Jeder der Körper und jede der Bindungen wird mit einem eindeutigen Index aus dem Intervall $[1, \dots, n_K]$ bzw. $[1, \dots, n_B]$ versehen. Zur besseren Unterscheidung werden die Indizes der Körper mit griechischen und die Indizes der Bindungen mit lateinischen Buchstaben angegeben. Der Aufbau des Mehrkörpersystems wird über drei Funktionen dargestellt. Die Vorgängerfunktion $\alpha = V(i)$ und die Nachfolgerfunktion $\beta = N(i)$ geben jeweils den Index des Vorgänger- bzw. Nachfolgekörpers einer Bindung i wieder. Diese Information wird benötigt, um die Vorzeichen der Gelenklasten bestimmen zu können. Diese

werden über die dritte Funktion

$$\sigma_{i\alpha} = \begin{cases} 1 & \alpha = N(i) \\ -1 & \alpha = V(i) \\ 0 & \text{sonst} \end{cases} \quad (2.53)$$

festgelegt.

An Stelle des Index K_α wird nur noch α geschrieben, welcher das Bezugssystem des Körpers α beschreibt. Analog wird für das Nachfolgesystem N_i der tiefgestellte Index i angehängt, d.h.

$$M_\alpha := M_{K_\alpha}$$

$$G_{i\alpha} := G_{N_i K_\alpha}$$

$$l_i := l_{N_i}.$$

Um die Bewegung eines Mehrkörpersystems berechnen zu können, müssen folgende Gleichungen der Körper und Bindungen gemeinsam gelöst werden.

Jeder Körper α trägt seine Bewegungsgleichung (2.25) sowie die kinematischen Differenzialgleichungen zur Bestimmung der Pose (2.7) und der Kinemate (2.13) bei. Jede Bindung i wird entweder durch ihre implizite holonome oder durch ihre (semi-)explizite¹ Bindungsgleichung auf Lageebene charakterisiert (siehe Gl. (2.26) bzw. Gl. (2.27)). Ggf. müssen nichtholonome Bindungsgleichungen nach Gl. (2.31) berücksichtigt werden. Zusätzlich muss für jede Bindung das Prinzip der virtuellen Leistung nach Gl. (2.46) beachtet werden. Es ergibt sich somit folgendes Gleichungssystem:

$$\mathbf{H} \cdot \dot{\mathbf{z}}_\alpha = \mathbf{y}_\alpha \quad (2.54a)$$

$$\mathbf{p}_\alpha^T \cdot \mathbf{p}_\alpha = 0 \quad (2.54b)$$

$$\dot{\mathbf{y}}_\alpha = \mathbf{x}_\alpha \quad (2.54c)$$

$$M_\alpha \cdot \mathbf{x}_\alpha = l_{\alpha,ext} + \sum_i \sigma_{i\alpha} G_{i\alpha}^T \cdot l_i \quad (2.54d)$$

$$\mathbf{0} = \mathbf{g}(z_V, z_N, t) \quad (2.54e)$$

$$z_N = \bar{\mathbf{g}}(z_V, \mathbf{q}, t) \quad (2.54f)$$

$$\chi_i(\mathbf{q}_i) = \mathbf{0} \quad (2.54g)$$

$$\mathbf{W}_{nh,i}^T \cdot \left(\sum_\alpha \sigma_{i\alpha} G_{i\alpha}^T \mathbf{y}_\alpha \right) = \zeta_{nh,i} \quad (2.54h)$$

$$\Phi_i^T \cdot l_i = \eta_i \quad (2.54i)$$

Dabei handelt es sich um ein differenzial-algebraisches Gleichungssystem (DAE) mit dem Index 3. Es ist bekannt [ESF98], dass die Berechnung der Lösung solch einer DAE mit großen Schwierigkeiten verbunden ist. Obwohl es spezielle Integrationsverfahren, wie z.B. RADAU5 [HW10] gibt, die mit solchen Systemen umgehen können, werden in den meisten Anwendungen aus Gründen der Geschwindigkeit und Robustheit andere Techniken zum Lösen von DAEs eingesetzt. Zu den bekanntesten Verfahren gehören Constraint-Stabilization [Bau72] und die Projektion [LNPE92, Bur10]. Das gebräuchlichste Vorgehen in der Mechanik besteht in der

¹Da die \mathbf{q} ggf. impliziten Zwangsbedingungen genügen.

Umformulierung des Systems auf Basis von Minimalkoordinaten. Im Falle von offenen kinematischen Anordnungen stellen meist die Lage- und Geschwindigkeitsgrößen der Gelenke eine günstige Wahl der Minimalkoordinaten dar. Auch im Falle von geschlossenen kinematischen Schleifen lassen sich meist Minimalkoordinaten und immer Minimalgeschwindigkeiten mithilfe des Coordinate-Partitioning [WHB81, SBK10] bzw. der Dummy Derivative Methode [MS93] finden. Es ist jedoch möglich, dass die gewählten Minimalkoordinaten lokal zu Singularitäten führen, weshalb ggf. eine Umschaltung zwischen verschiedenen Koordinatensätzen vorgesehen sein sollte. Diese Technik ist als Dynamic State Selection bekannt und in [MOE00] beschrieben. Allen hier genannten Verfahren ist gemein, dass eine Indexreduktion des Systems (2.54) auf den Index 1 ausgeführt wird. Dabei werden alle Bindungsgleichungen durch ihre Formulierungen auf Beschleunigungsebene (2.51) ersetzt [ESF98].

$$\mathbf{M}_\alpha \cdot \mathbf{x}_\alpha = \mathbf{l}_{\alpha,ext} + \sum_i \sigma_{i\alpha} \mathbf{G}_{i\alpha}^T \cdot \mathbf{l}_i \quad (2.55a)$$

$$\mathbf{0} = \sum_\alpha \sigma_{i\alpha} \mathbf{G}_{i\alpha} \mathbf{x}_\alpha + \Phi_i \cdot \dot{\mathbf{s}}_i + \mathbf{c}_i \quad (2.55b)$$

$$\Phi_i^T \cdot \mathbf{l}_i = \boldsymbol{\eta}_i \quad (2.55c)$$

2.4. ZUSAMMENFASSUNG ZUR MECHANIK DER MEHRKÖRPERSYSTEME

Das Gleichungssystem (2.55) fasst die Bewegungsgleichungen eines allgemeinen starren Mehrkörpersystems zusammen. Da die Bindungsgleichungen nun auf Beschleunigungsebene vorliegen, muss nicht mehr zwischen holonomen und nichtholonomen Bindungen unterschieden werden. Aufgrund der Verwendung der dualen Basen muss außerdem nicht mehr zwischen impliziten und expliziten Bindungen unterschieden werden, da die implizite Form einfach durch Projektion auf die gesperrten Lastrichtungen $\bar{\Psi}$ gewonnen werden kann, siehe Abschnitt 3.1.5.

Das Gleichungssystem (2.55) ist linear in den Unbekannten

- Beschleunigungen der Körper \mathbf{x}_α ,
- Gelenkbeschleunigungen $\dot{\mathbf{s}}_i$ und
- Gelenklasten \mathbf{l}_i .

Es besitzt eine eindeutige Lösung, sofern die Matrix der Koeffizienten Vollrang besitzt. Fälle in denen dies nicht zutrifft, werden in [Rei97] diskutiert. Die Zusammenhänge der Ableitungen der kinematischen Größen sind in (2.55) vernachlässigt worden. Dies hängt stark von der gewählten Methode der Indexreduktion ab. Im einfachsten Fall können die mit den Bindungen verträglichen, \mathbf{x}_α und $\dot{\mathbf{s}}_i$ zu den Kinematoren \mathbf{y}_α , Posen \mathbf{z}_α , bzw. Gelenkgeschwindigkeiten \mathbf{s}_i und Gelenkkoordinaten \mathbf{q}_i aufintegriert werden. Dabei müsste jedoch mit einem Drift, also einem Fehler, in den Bindungsgleichungen auf Lage- und Geschwindigkeitsebene gerechnet werden [ESF98].

3. LÖSUNGsalgorithmen für MEHRKÖRPERSYSTEME

Es lassen sich eine Vielzahl von Algorithmen [ESF98, Fea07, Nik05, RS10] zur Lösung der Bewegungsgleichungen eines Mehrkörpersystems finden, die letztlich alle Lösungen des linearen Gleichungssystems (2.55) aus dem vorherigen Kapitel darstellen. Die Lösungsalgorithmen für Mehrkörpersysteme lassen sich in zweierlei Hinsicht unterteilen. Zum einen kann eine Unterscheidung zwischen $\mathcal{O}(n)$ -, oder auch rekursiven Algorithmen genannt, und $\mathcal{O}(n^3)$ -Algorithmen, also nicht-rekursive Algorithmen, erfolgen. Merkmal ist die Skalierung des Rechenaufwands für ein Mehrkörpersystem mit steigender Anzahl n der Körper bzw. Gelenke. Zum anderen kann unterschieden werden, ob die Bindungsgleichungen in expliziter Form oder in impliziter Form, siehe Abschnitt 3.1.5, vorliegen. Auf der zweiten Einteilung gründend haben sich zwei Herangehensweisen zur Lösung von Mehrkörpersystemen herausgebildet. Die erste [BJO86], welche vorwiegend im Bereich der Robotik angewendet wird, befasst sich ausschließlich mit der expliziten Formulierung (2.27), aus der hoch effiziente Algorithmen entwickelt wurden. In [SH95] wird gezeigt, dass ihre Benutzung der relativen Kinematik am effizientesten ist und sich mit dem speziell entwickelten $\mathcal{O}(n)$ -Algorithmus auch große Systeme berechnen lassen. Die zweite Herangehensweise [Erl04], welche aus der Computergraphik stammt, bevorzugt die implizite Formulierung (2.26), da die Herleitung der Bewegungsgleichungen mit der absoluten Kinematik deutlich einfacher sei. Des Weiteren müssen kinematische Schleifen nicht gesondert behandelt werden, da die linearen Gleichungssysteme mit generischen Algorithmen gelöst werden können. Interessanterweise vertritt auch diese Seite die Auffassung, eine sehr effiziente Lösung zu erhalten [JB93].

Ziel dieses Kapitels ist es, abweichend von der in der Literatur üblichen Darstellung, die Algorithmen beider Ideologien mithilfe der Gauß-Elimination für lineare Gleichungssysteme herzuleiten. Diese abstrakte Sicht auf die Algorithmen bildet die Grundlage für die Lösungen zur effizienteren Berechnung von mechanischen Teilsystemen, die in den folgenden Kapiteln vorgestellt werden.

In Abschnitt 3.1 wird zuerst die Struktur des Gleichungssystems (2.55) genauer analysiert. Wie in der Behandlung von schwach besetzten Gleichungssystemen üblich, wird dazu die Graphentheorie eingesetzt. Mit deren Hilfe ist es möglich, die regelmäßige Struktur in den Gleichungen zu erkennen. Es wird gezeigt, dass im Vergleich zur Verwendung expliziter Bindungsbeschreibungen sich die Struktur bei impliziten Bindungsbeschreibungen nur marginal verändert. Dies führt zu der These, dass sich die Berechnungen mit Relativkoordinaten (explizite Bindungsgleichungen) und Absolutkoordinaten (implizite Bindungsgleichungen) doch nicht so stark unterscheiden, wie in der Literatur meist behauptet wird [Bar96, Nik05].

In Abschnitt 3.2.1 wird für Mehrkörpersysteme ohne kinematische Schleifen gezeigt, dass sich

alle bekannten Mehrkörperalgorithmen aus dem linearen Gleichungssystem (2.55) durch Gauß-Elimination herleiten lassen, sofern die Gleichungen und Variablen vorher entsprechend sortiert wurden. Durch die Verwendung der dualen Basen treten interessante Gemeinsamkeiten zwischen den Algorithmen für die expliziten und die impliziten Bindungsgleichungen zu Tage. Weiterhin wird gezeigt, dass es möglich ist, mit einer rein impliziten Bindungsformulierung (Absolutkoordinaten) durch entsprechende Projektion der Gleichungen Algorithmen zu konstruieren, die fast identisch zu denen für die explizite Bindungsformulierung sind. Sofern die beschreibenden Koordinatensysteme zur Darstellung günstig gewählt werden, entspricht die benötigte Projektion lediglich einer geeigneten Umsortierung der Gleichungen und Variablen. Somit kann gezeigt werden, dass es aus Sicht der Berechnung praktisch keinen Unterschied zwischen der impliziten und expliziten Formulierung der Bindungsgleichungen gibt.

Abschnitt 3.3 soll durch die Anwendung der vorgestellten Algorithmen auf das Mehrkörpersystem eines ebenen Pendels die gefundenen Zusammenhänge nochmals praktisch darstellen.

Die Einbeziehung von kinematischen Schleifen in die vorgestellten Algorithmen wird in Abschnitt 3.4 diskutiert und anhand eines Beispiels der Mehraufwand für die jeweiligen Berechnungszeiten präsentiert.

3.1. DIE GRAPHEN EINES MEHRKÖRPERSYSTEMS

In diesem Abschnitt soll ein kurzer Überblick über das Gebiet der Graphentheorie gegeben werden. Für eine detailliertere Behandlung sei auf Standardwerke, wie z.B. [Die10, Bap10] verwiesen.

3.1.1. EINFÜHRUNG IN DIE GRAPHENTHEORIE

Ein Graph $G = (V, E)$, wie in Abb. 3.1 dargestellt, beschreibt ein Paar einer endlichen nicht-leeren Menge von Knoten V (engl. *vertices*) zusammen mit einer Menge an Kanten E (engl. *edges*). Die Kantenmenge E enthält Paare von Knoten, die durch die jeweilige Kante miteinander verbunden werden. Sind die Paare in E geordnet, wird von einem gerichteten bzw. Digraphen (engl. *directed Graph*) gesprochen andernfalls von einem ungerichteten Graphen. Zwei Knoten sind *adjazent*, wenn sie durch eine Kante miteinander verbunden sind. Eine Knotensequenz v_0, v_1, \dots, v_n in dem jeder Knoten adjazent zu seinem Vorgängerknoten ist, wird *Pfad* genannt. Die *Länge des Pfades* entspricht der Anzahl der Kanten die ein Pfad besitzt. Sind der erste und der letzte Knoten eines Pfades gleich und alle anderen voneinander verschieden, so wird von einer *Schleife* gesprochen. Ein Graph ist *schwach zusammenhängend*, sofern es möglich ist, einen Pfad von einem beliebigen Knoten zu jedem anderen Knoten zu finden, ohne die Kantenrichtung zu berücksichtigen. Ein Graph mit *Baumstruktur* ist ein schwach zusammenhängender Graph, der keine Schleifen enthält. Der *aufspannende Baum* G_t (engl. *spanning tree*) eines Graphen G enthält alle Knoten und eine Untermenge der Kanten aus G . Die Menge der Kanten, die in G aber nicht in G_t enthalten sind, werden *Cut-Set* genannt. Die Untermenge der Kanten von G_t besitzt die definierende Eigenschaft, dass der Graph eine Baumstruktur besitzt und dass diese Eigenschaft durch das Hinzufügen einer beliebigen Kante aus dem Cut-Set verloren gehen würde. Die Wahl des aufspannenden Baums ist nicht eindeutig.

Durch Wahl eines Referenzknotens, auch *Wurzel* genannt, innerhalb eines Graphen mit Baumstruktur lassen sich eindeutige Vorgänger-Nachfolger-Beziehungen aufstellen. Der *Vorgänger* $V(i)$ eines Knotens i ist der nächste Knoten entlang des Pfades zur Wurzel des Graphen. Die *Nachfolger* $N(i)$ eines Knotens i sind alle die Knoten, die i als Vorgänger besitzen. Die Menge der erreichbaren Knoten $E(i)$ enthält alle Knoten, die von i aus über einen beliebigen Pfad erreicht werden können. Ein hochgestelltes $+$ soll darauf hinweisen, dass der Knoten i selbst in der Menge enthalten ist. So gilt z.B. $N^+(i) = N(i) \cup \{i\}$.

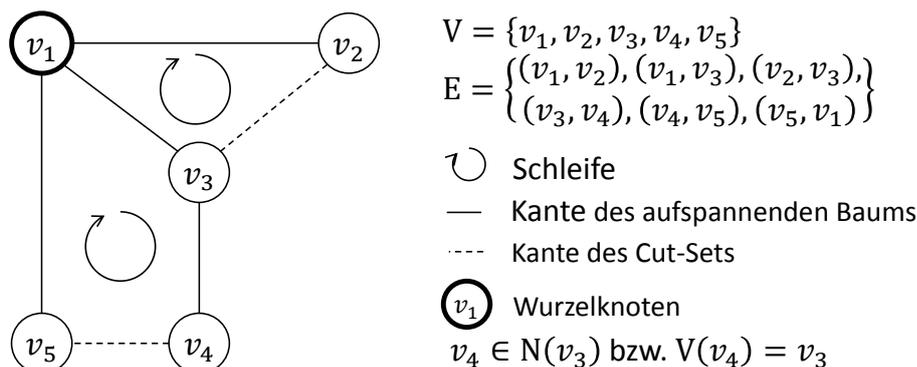


Abbildung 3.1.: Begriffsdefinitionen an einem Graphen

Der *Erreichbarkeitsgraph* zu einem Graphen G enthält die selben Knoten wie G . Kanten zwischen den Knoten existieren genau dann, wenn diese in G unter Beachtung der Kantenrichtung durch einen Pfad verbunden sind.

3.1.2. GRAPH EINER QUADRATISCHEN MATRIX

Graphen können zur Visualisierung der Struktur von Matrizen verwendet werden. Speziell für große schwach besetzte Matrizen ist die Darstellung als Graph gegenüber der Darstellung in klassischer Matrixschreibweise vorteilhaft, da nur die von Null verschiedenen Elemente sichtbar sind. Darüber hinaus ist das Aussehen des Graphen unabhängig von der Permutation der zugrundeliegenden Matrix, d.h. von der Anordnung der Zeilen und Spalten. Daher bilden Graphen die Grundlage fast aller effizienten Algorithmen zur Behandlung schwach besetzter Matrizen [DER86, Dav06].

Der Digraph $G(\mathbf{A})$ einer quadratischen Matrix $\mathbf{A} = [a_{ij}]_{n \times n}$ besitzt n Knoten. Die Elemente a_{ij} können entweder Skalare oder selbst Matrizen bzw. Tensoren sein. Eine gerichtete Kante von Knoten i zu Knoten j existiert, sofern $a_{ij} \neq 0$. Abb. 3.2a und 3.2b zeigen eine exemplarische Matrix \mathbf{A} und den zugehörigen Graphen. Falls \mathbf{A} symmetrisch ist, werden die entgegengesetzten Kanten zusammengefasst und es resultiert in einem ungerichteten Graphen. Das Aussehen des Graphen ist invariant gegenüber symmetrischen Zeilen- und Spaltenpermutationen. Es werden lediglich die Knotennummern verändert.

Einfache Matrixoperationen lassen sich effizient im Graphen durchführen. So lässt es sich leicht verifizieren, dass die Kantenmenge $E_{\mathbf{A} \pm \mathbf{B}}$ des Graphen $G(\mathbf{A} \pm \mathbf{B})$ der Menge $E_{\mathbf{A}} \cup E_{\mathbf{B}}$ der Graphen $G(\mathbf{A})$ und $G(\mathbf{B})$ entspricht. Summanden, die sich zu Null ergeben werden in dieser Sichtweise nicht berücksichtigt. Der Graph $G(\mathbf{A}^T)$ kann direkt aus dem Graphen $G(\mathbf{A})$ durch eine Richtungsumkehr aller Kanten gewonnen werden. Das Produkt zweier Matrizen $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ ist elementweise definiert als

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$

Daraus folgt, dass der Graph $G(\mathbf{A} \cdot \mathbf{B})$ eine Kante von j nach i enthält ($c_{ij} \neq 0$), wenn die Knoten i und j durch jeweils eine Kante aus $G(\mathbf{A})$ und $G(\mathbf{B})$ über einen beliebigen Knoten k verbunden sind. Im Falle von $G(\mathbf{A}^2)$ entspricht dies den Pfaden der Länge 2. Verallgemeinert auf $G(\mathbf{A}^n)$ enthält dieser Graph eine Kante zwischen den Knoten j und i sofern zwischen ihnen ein Pfad der Länge n existiert. Besitzt $G(\mathbf{A})$ eine Baumstruktur, so kann es jedoch keinen Pfad geben, der mehr als n Knoten umfasst. Andernfalls müsste ein Knoten mehrmals im Pfad enthalten sein. Dies würde aber bedeuten, dass $G(\mathbf{A})$ mindestens eine Schleife besitzt und damit keine Baumstruktur hätte. Somit sind Matrizen deren Graph eine Baumstruktur besitzt nilpotent, d.h. $\mathbf{A}^{n+1} = \mathbf{0}$. Durch Ausmultiplizieren lässt sich zeigen, dass für eine Diagonalmatrix

D und eine nilpotente Matrix A gilt

$$(D - A)^{-1} = D^{-1} \left(E + \sum_{i=1}^n (D^{-1}A)^i \right). \quad (3.1)$$

Ist $D = E$ so vereinfacht sich die Beziehung zu

$$(E - A)^{-1} = E + \sum_{i=1}^n A^i. \quad (3.2)$$

Dieser Zusammenhang wird für die Herleitung der $\mathcal{O}(n^3)$ -Algorithmen benötigt und in Abschnitt 3.2.1 aufgegriffen.

3.1.3. ERWEITERTER GRAPH EINER MATRIX

Der erweiterte Graph $G(\text{aug}(\mathbf{B}))$ einer nicht notwendigerweise quadratischen Matrix $\mathbf{B} = [b_{ij}]_{m \times n}$ ist definiert als Graph der erweiterten Matrix

$$\text{aug}(\mathbf{B}) = \begin{bmatrix} \mathbf{E} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}.$$

Der Graph besitzt somit m Knoten für jede Zeile und n Knoten für jede Spalte der Matrix \mathbf{B} . Ist das Element $b_{ij} \neq 0$, so gibt es eine ungerichtete Kante von dem Zeilenknoten i zu dem Spaltenknoten j . Da sich die Knotenmenge in zwei disjunkte Teilmengen zerlegen lässt und keine Kante zwischen Knoten der gleichen Teilmenge existiert, wird der Graph als *bipartit* bezeichnet. Der erweiterte Graph ist invariant gegenüber Zeilen- und Spaltenpermutationen. Es ändert sich lediglich die Nummerierung der Zeilen- oder Spaltenknoten.

Ähnlich dem Graphen einer Matrix können auch im erweiterten Graphen Matrixoperationen komfortabel dargestellt werden. Von besonderem Interesse im Kontext dieser Arbeit ist der Zusammenhang zwischen $G(\mathbf{B}^T \mathbf{B})$ und $G(\text{aug}(\mathbf{B}))$. Der Graph $G(\mathbf{B}^T \mathbf{B})$ besitzt genauso viele Knoten, wie \mathbf{B} Spalten besitzt. Eine Kante von i nach j existiert, wenn es in $G(\text{aug}(\mathbf{B}))$ jeweils eine Kante von einem beliebigen Zeilenknoten k zu den Spaltenknoten i und j gibt.

Abb. 3.2 zeigt die beiden Graphen für eine frei gewählte Matrix.

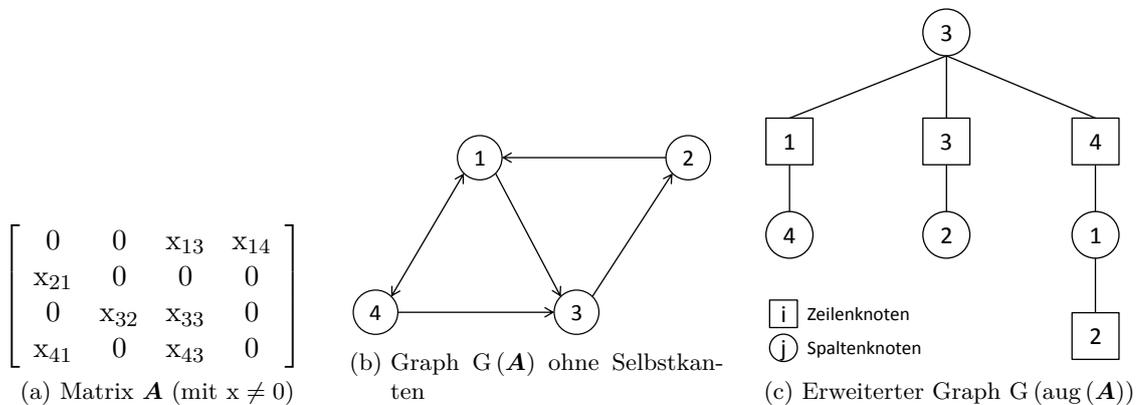


Abbildung 3.2.: Graphen einer Matrix \mathbf{A}

3.1.4. TOPOLOGIEGRAPH EINES MEHRKÖRPERSYSTEMS

Im Topologiegraphen wird jedem Körper $\alpha \in [1, \dots, n_K]$ des Mehrkörpersystems ein Knoten zugewiesen. Zusätzlich wird das Inertialsystem als Knoten 0 hinzugefügt, der die Wurzel des Topologiegraphen bzw. seines aufspannenden Baumes bildet. Jede Bindung wird als Kante zwischen den beteiligten Körpern abgebildet. Sofern das Mehrkörpersystem Körper ohne Verbindung zu dem Inertialsystem besitzt, wird an einem geeigneten Körper eine gedachte Bindung mit 6 Freiheitsgraden eingefügt. Besitzt der Topologiegraph eine Baumstruktur, so wird auch das zugehörige Mehrkörpersystem als baumstrukturiert bezeichnet. Andernfalls besitzt es *kinematische Schleifen*. Abb. 3.3 zeigt den Topologiegraphen für ein ebenes Viergelenk.

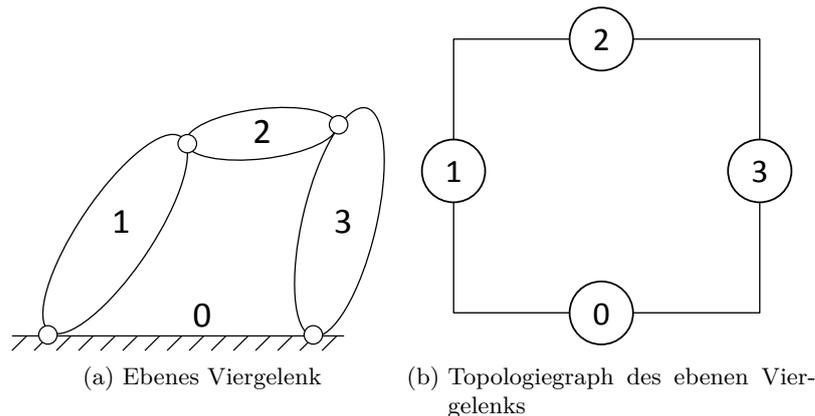


Abbildung 3.3.: Topologiegraph

Innerhalb des Topologiegraphen eines aufspannenden Baumes lassen sich die Gelenke in zwei Gruppen unterteilen. Gelenke, deren Kanten Teil des aufspannenden Baumes sind, werden *primäre* Gelenke genannt. Gelenke, die Teil des Cut Sets sind, werden als *sekundäre Gelenke* bezeichnet. Die Wahl des Cut Sets ist dabei nicht eindeutig. Abhängig von der konkreten Modellstruktur, kann es sein, dass ein Cut Set eine effizientere Berechnung zulässt als ein anderes. Für eine weitergehende Diskussion wird auf [MOE00] verwiesen.

3.1.5. SYSTEMGRAPH EINES MEHRKÖRPERSYSTEMS

Für ein beliebiges nichtlineares System mit den Gleichungen $\mathbf{h}(\mathbf{z}) = \mathbf{0}$ und Variablen \mathbf{z} ist der zugehörige Systemgraph als erweiterter Graph der Jacobimatrix $\mathbf{J} = \frac{\partial \mathbf{h}}{\partial \mathbf{z}}$ definiert. Jeder Zeilenknoten repräsentiert eine Gleichung und jeder Spaltenknoten eine Variable des Systems. Eine Kante zwischen dem Gleichungsknoten i und dem Variablenknoten j existiert, wenn $\frac{\partial h_i}{\partial z_j} \neq 0$ bzw. Variable j in der Gleichung i vorkommt. Die Nummerierung der Gleichungen und Variablen beeinflusst das Aussehen des Graphen nicht.

DARSTELLUNG MIT EXPLIZITEN BINDUNGSGLEICHUNGEN

Mithilfe des aufspannenden Baums kann jedem Körper α genau ein primäres Gelenk i zugeordnet werden. Wird das Körperbezugssystem K_α nun genau in den Punkt gelegt, an dem das Gelenk i mit dem Körper verbunden ist, so vereinfacht sich die Transformationsmatrix $\mathbf{G}_{i\alpha}$ zu einer Einheitsmatrix. Dadurch ist eine effektivere Auswertung des Mehrkörpersystems möglich. Das System (2.55) lässt sich unter Nutzung der expliziten Bindungsbeschreibung darstellen als

$$\mathbf{M}_i \cdot \mathbf{x}_i = \mathbf{l}_{i,ext} + \mathbf{l}_i + \sum_k \sigma_{ki} \mathbf{G}_{ki}^T \cdot \mathbf{l}_k \quad (3.3a)$$

$$\mathbf{x}_i = \mathbf{G}_{iV(i)} \mathbf{x}_{V(i)} + \mathbf{\Phi}_i \cdot \dot{\mathbf{s}}_i + \mathbf{c}_i \quad (3.3b)$$

$$\mathbf{\Phi}_i^T \cdot \mathbf{l}_i = \boldsymbol{\eta}_i. \quad (3.3c)$$

Im Unterschied zum Ausgangssystem (2.55) tauchen die Schnittlasten \mathbf{l}_i und die Beschleunigungen \mathbf{x}_i nun ohne Vorfaktoren im System auf.

Der zugehörige Systemgraph besitzt eine leiterähnliche Struktur. Er lässt zwei *Holme* mit *Griffen* erkennen, die über *Sprossen* miteinander verbunden sind. Der *kinematische Holm* wird durch die Variablenknoten der Beschleunigungen \mathbf{x}_i und durch die Gleichungsknoten A_i (3.3b) gebildet. Die Struktur entspricht dem des topologischen Graphen. Die Variablenknoten der Gelenkbeschleunigungen $\dot{\mathbf{s}}_i$ stellen die *Griffe* des kinematischen Holms dar. Der *kinetische Holm* wird durch die Variablenknoten der Schnittlasten \mathbf{l}_i und die Gleichungsknoten I_i (3.3a) gebildet. Auch die Struktur des kinetischen Holms entspricht dem des topologischen Graphen. Die Griffe des kinetischen Holms werden durch die Gleichungsknoten D_i (3.3c) repräsentiert. Die *Sprossen* entstehen durch die Abhängigkeit der Gleichung I_i von den \mathbf{x}_i . Sie stellt die einzige Verbindung zwischen dem kinematischen und dem kinetischen Holm dar. Der Systemgraph für das ebene Viergelenk ist in Abb. 3.4 dargestellt.

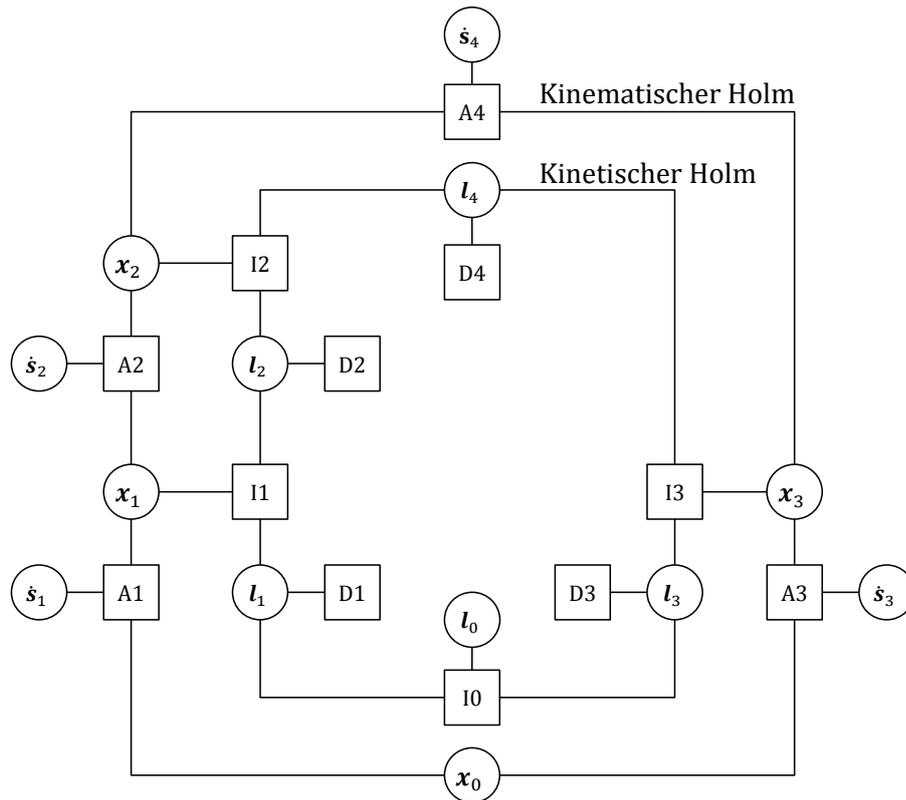


Abbildung 3.4.: Systemgraph eines Viergelenks

DARSTELLUNG MIT IMPLIZITEN BINDUNGSGLEICHUNGEN

Viele Algorithmen arbeiten nicht direkt mit dem System (3.3), sondern nutzen die implizite Formulierung der Bindungsgleichungen. Um das System in die implizite Formulierung zu überführen, wird Gl. (3.3b) einmal mit Ψ_i^T und einmal mit $\bar{\Psi}_i^T$ multipliziert und es folgt

$$\Psi_i^T (\mathbf{x}_i - \mathbf{G}_{iV(i)} \mathbf{x}_{V(i)} - \mathbf{c}_i) = \dot{\mathbf{s}}_i \quad (3.4a)$$

$$\bar{\Psi}_i^T (\mathbf{x}_i - \mathbf{G}_{iV(i)} \mathbf{x}_{V(i)}) = \bar{\Psi}_i^T \mathbf{c}_i. \quad (3.4b)$$

Auf diese Weise können die $\dot{\mathbf{s}}_i$ aus Gleichung (3.3b) separiert werden und es bleibt eine Gleichung zur Berechnung der Gelenkbeschleunigung sowie eine Gleichung mit der impliziten Zwangsbedingung übrig.

Im nächsten Schritt wird \mathbf{l}_i durch die Darstellung (2.40) bezüglich der Basis Ψ_i und $\bar{\Psi}_i^T$ ersetzt. Die Koordinaten λ_i lassen sich analog zu Gl. (2.43) berechnen und Gl. (3.3a) kann somit umgeschrieben werden zu

$$\mathbf{M}_i \cdot \mathbf{x}_i = \mathbf{l}'_{i,ext} + \bar{\Psi}_i \bar{\lambda}_i + \sum_k \sigma_{ki} \mathbf{G}_{ki}^T \cdot \bar{\Psi}_k \bar{\lambda}_k \quad (3.5)$$

mit

$$\mathbf{l}'_{i,ext} = \mathbf{l}_{i,ext} + \Psi_i \lambda_i + \sum_k \sigma_{ki} \mathbf{G}_{ki}^T \cdot \Psi_k \lambda_k. \quad (3.6)$$

Letzteres enthält zusätzlich die Anteile der angeschlossenen angetriebenen Gelenke. Das lineare Gleichungssystem der impliziten Bindungsgleichungen lautet

$$\mathbf{M}_i \cdot \mathbf{x}_i = \mathbf{l}'_{i,ext} + \bar{\Psi}_i \bar{\lambda}_i + \sum_k \sigma_{ki} \mathbf{G}_{ki}^T \cdot \bar{\Psi}_k \bar{\lambda}_k \quad (3.7a)$$

$$\bar{\Psi}_i^T (\mathbf{x}_i - \mathbf{G}_{iV(i)} \mathbf{x}_{V(i)}) = \bar{\Psi}_i^T \mathbf{c}_i. \quad (3.7b)$$

Es ist gegenüber dem Gleichungssystem (3.3) in der Anzahl der Gleichungen und Variablen reduziert, da nur noch die Unbekannten \mathbf{x}_i und $\bar{\lambda}_i$ mit den Gleichungen (3.7a), abgekürzt als I_i , und (3.7b), abgekürzt als X_i , vorkommen. Der Systemgraph ist in Abb. 3.5 dargestellt.

Der Systemgraph unterscheidet sich von dem der expliziten Formulierung durch die fehlenden seitlichen Griffe, da die Gelenkbeschleunigungen aus dem System entfernt und das Prinzip der virtuellen Leistung genutzt wurde, um die Koordinaten λ der freien Lastrichtung der Schnittkräfte zu eliminieren. Die implizite Version kann somit als kompaktierte Darstellung der expliziten Formulierung verstanden werden.

3. Lösungsverfahren für Mehrkörpersysteme

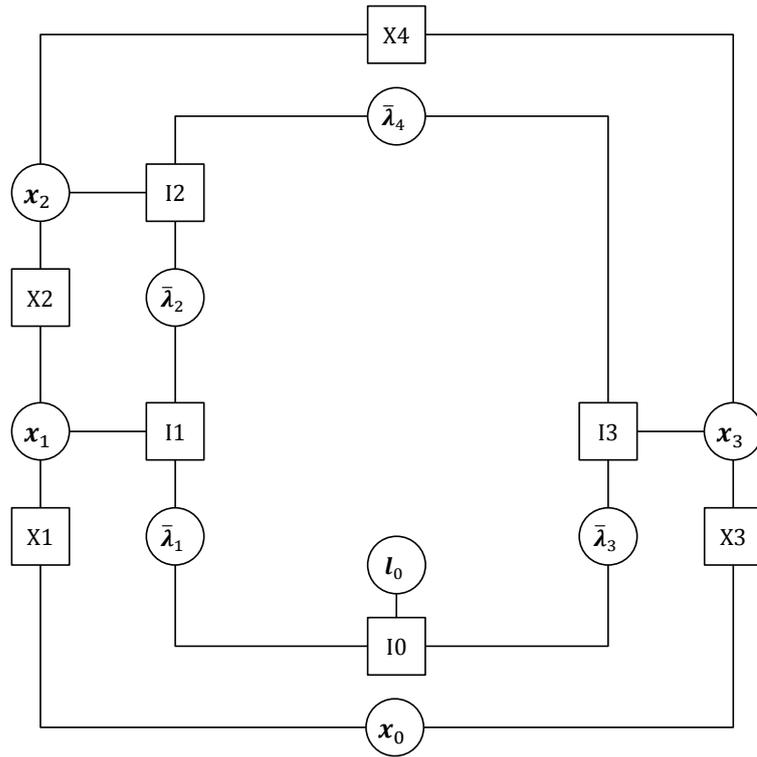


Abbildung 3.5.: Systemgraph eines impliziten Vieregelks

3.2. LÖSUNGsalgorithmen FÜR SYSTEMe MIT BAUMSTRUKTUR

Vorerst sollen Algorithmen zur Berechnung der Bewegungsgleichungen für den aufspannenden Baum eines Mehrkörpersystems präsentiert werden. Die Einbeziehung der sekundären Gelenke folgt in Abschnitt (3.4).

3.2.1. $\mathcal{O}(n^3)$ -ALGORITHMUS FÜR EXPLIZITE BESCHREIBUNG

Die Gleichungen des Systems (3.3) lassen sich zusammenfassen und können wie folgt als Matrixgleichungen angeschrieben werden:

$$\begin{bmatrix} \mathbf{G}^T & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} & \mathbf{\Phi} \\ \mathbf{\Phi}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{l} \\ \mathbf{x} \\ \dot{\mathbf{s}} \end{bmatrix} = \begin{bmatrix} \mathbf{l}_{ext} \\ \mathbf{c} \\ \boldsymbol{\eta} \end{bmatrix} \quad (3.8)$$

mit

$$\begin{aligned} \mathbf{M} &= \text{blkdiag}(\mathbf{M}_1, \dots, \mathbf{M}_{n_K}) \\ \mathbf{G} &= [\sigma_{ij} \mathbf{G}_{ij}]_{n_K \times n_K} \\ \mathbf{x} &= [\mathbf{x}_1^T, \dots, \mathbf{x}_{n_K}^T]^T \\ \mathbf{l}_{ext} &= [\mathbf{l}_{1,ext}^T, \dots, \mathbf{l}_{n_K,ext}^T]^T \\ \mathbf{\Phi} &= \text{blkdiag}(\mathbf{\Phi}_1, \dots, \mathbf{\Phi}_{n_K}) \\ \mathbf{l} &= [\mathbf{l}_1^T, \dots, \mathbf{l}_{n_K}^T]^T \\ \mathbf{c} &= [\mathbf{c}_1^T, \dots, \mathbf{c}_{n_K}^T]^T \\ \dot{\mathbf{s}} &= [\dot{\mathbf{s}}_1^T, \dots, \dot{\mathbf{s}}_{n_K}^T]^T \\ \boldsymbol{\eta} &= [\boldsymbol{\eta}_1^T, \dots, \boldsymbol{\eta}_{n_K}^T]^T. \end{aligned}$$

Das Gleichungssystem (3.8) kann nun über eine Gauß-Elimination in eine obere Dreiecksmatrix überführt werden und es folgt

$$\begin{bmatrix} \mathbf{G}^T & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} & \mathbf{\Phi} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \mathbf{l} \\ \mathbf{x} \\ \dot{\mathbf{s}} \end{bmatrix} = \begin{bmatrix} \mathbf{l}_{ext} \\ \mathbf{c} \\ \hat{\boldsymbol{\eta}} \end{bmatrix} \quad (3.9)$$

mit

$$\begin{aligned} \hat{\mathbf{M}} &= (\mathbf{G}^{-1} \mathbf{\Phi})^T \mathbf{M} (\mathbf{G}^{-1} \mathbf{\Phi}) \\ \hat{\boldsymbol{\eta}} &= \boldsymbol{\eta} + (\mathbf{G}^{-1} \mathbf{\Phi})^T (\mathbf{M} \mathbf{G}^{-1} \mathbf{c} - \mathbf{l}_{ext}). \end{aligned} \quad (3.10)$$

Die unterste Zeile von Gleichung (3.9) beschreibt nun die am weitesten verbreitete Schreibweise für die Bewegungsgleichung eines Mehrkörpersystems

$$\hat{\mathbf{M}} \dot{\mathbf{s}} = \hat{\boldsymbol{\eta}}.$$

3. Lösungsalgorithmen für Mehrkörpersysteme

Die Gelenkbeschleunigungen $\dot{\mathbf{s}}$ können durch Lösen des Gleichungssystems berechnet werden. Die Dimension der reduzierten Massenmatrix $\hat{\mathbf{M}}$ entspricht $\dim(\mathbf{s})$. Je nachdem wie Gleichung (3.10) zur Berechnung der reduzierten Massenmatrix $\hat{\mathbf{M}}$ ausgewertet wird, ergeben sich verschiedene Algorithmen. Das Befolgen der Klammern aus Gl. (3.10) zu diesem Zweck wird als *Velocity Transformation* Algorithmus [Nik05] bezeichnet. Die Lösung

$$\hat{\mathbf{M}} = \Phi^T (\mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1}) \Phi \quad (3.11)$$

wird als *Composite-Rigid-Body-Algorithmus* [Fea07] bezeichnet.

Beide Algorithmen können graphentheoretisch ausgewertet werden, um zu einer geschlossenen Gleichung für $\hat{\mathbf{M}}$ im Falle von baumstrukturierten Systemen zu gelangen.

Gemäß Gl. (3.8) kann \mathbf{G} geschrieben werden als

$$\mathbf{G} = -(\mathbf{E} - \mathbf{S}),$$

da aufgrund der speziellen Wahl der Körperbezugssysteme K_α die $\mathbf{G}_{ii} = -\mathbf{E}$ sind, siehe Abschnitt 3.1.5. Der Graph der Matrix \mathbf{S} repräsentiert den topologischen Graphen des Systems und besitzt somit Baumstruktur. Gleichung (3.2) zur Berechnung von \mathbf{G}^{-1} kann angewandt werden und es gilt

$$[\mathbf{G}^{-1}]_{ij} = \prod_{(k_1, k_2) \in \text{Pfad}(i, j)} \mathbf{G}_{k_1, k_2} = \begin{cases} \mathbf{G}_{ij} & i \in N^+(j) \\ \mathbf{0} & \text{sonst.} \end{cases}$$

Das Produkt $\mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1}$ ergibt sich somit zu

$$[\mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1}]_{ij} = \sum_k \mathbf{G}_{ki}^T \mathbf{M}_k \mathbf{G}_{kj}, \text{ mit } k = N^+(i) \cap N^+(j)$$

und es können die Elemente der reduzierten Massenmatrix wie folgt berechnet werden:

$$[\hat{\mathbf{M}}]_{ij} = \sum_k \Phi_i^T \mathbf{G}_{ki}^T \mathbf{M}_k \mathbf{G}_{kj} \Phi_j^T, \text{ mit } k = N^+(i) \cap N^+(j).$$

Für den Composite-Rigid-Body-Algorithmus lässt sich diese Summe effektiver auswerten. Dazu werden die Variablen

$$\bar{\mathbf{M}}_i = \mathbf{M}_i + \sum_{l \in N(i)} \mathbf{G}_{li}^T \bar{\mathbf{M}}_l \mathbf{G}_{li} \quad (3.12)$$

definiert. Entsprechend den Transformationsbeziehungen (2.14) und (2.17) beschreiben die $\bar{\mathbf{M}}_i$ die Trägheitseigenschaften des Körpers i , der mit all seinen Nachfolgern fest verbunden ist.

Die Elemente der reduzierten Massenmatrix können nun direkt über die Beziehung

$$[\hat{\mathbf{M}}]_{ij} = \begin{cases} \mathbf{0} & N^+(i) \cap N^+(j) = \emptyset \\ \Phi_i^T \mathbf{G}_{ki}^T \bar{\mathbf{M}}_k \mathbf{G}_{kj} \Phi_j^T & k = \max(i, j) \end{cases}$$

berechnet werden. Ein analoges Vorgehen für die rechte Seite führt auf die Berechnungsvorschrift

$$[\hat{\boldsymbol{\eta}}]_i = \boldsymbol{\eta}_i - \Phi_i^T \hat{\mathbf{l}}_{i, \text{ext}}$$

mit

$$\begin{aligned}\hat{\mathbf{l}}_{i,ext} &= \bar{\mathbf{l}}_{i,ext} + \sum_{l \in N(i)} \mathbf{G}_{li}^T [\bar{\mathbf{l}}_{ext}]_i \\ [\bar{\mathbf{l}}_{ext}]_i &= \mathbf{l}_{i,ext} - \mathbf{M}_i \bar{\mathbf{c}}_i \\ \bar{\mathbf{c}}_i &= \mathbf{c}_i + \mathbf{G}_{iV} \bar{\mathbf{c}}_V.\end{aligned}$$

Der Algorithmus lässt sich dann wie folgt zusammenfassen:

Algorithmus 3.1 $\mathcal{O}(n^3)$ -Algorithmus für die explizite Beschreibung

Vorwärtsrekursion:

```
for i=1:n
     $\bar{\mathbf{c}}_i = -\mathbf{c}_i + \mathbf{G}_{iV} \bar{\mathbf{c}}_V$ 
     $\bar{\mathbf{l}}_{i,ext} = \mathbf{l}_{i,ext} - \mathbf{M}_i \bar{\mathbf{c}}_i$ 
end
```

Rückwärtsrekursion:

```
for i=n:-1:1
     $\bar{\mathbf{M}}_i = \mathbf{M}_i + \sum_{k \in N(i)} \mathbf{G}_{ki}^T \bar{\mathbf{M}}_k \mathbf{G}_{ki}$ 
     $\hat{\mathbf{l}}_{i,ext} = -\bar{\mathbf{l}}_{i,ext} + \sum_{l \in N(i)} \mathbf{G}_{li}^T \bar{\mathbf{l}}_{l,ext}$ 
end
```

Erstellen der Matrizen

$$\begin{aligned}\hat{\mathbf{M}}_{ij} &= \begin{cases} \mathbf{0} & N^+(i) \cap N^+(j) = \emptyset \\ \mathbf{\Phi}_i^T \mathbf{G}_{ki}^T \bar{\mathbf{M}}_k \mathbf{G}_{kj} \mathbf{\Phi}_j^T & k = \max(i, j) \end{cases} \\ \hat{\boldsymbol{\eta}}_i &= \boldsymbol{\eta}_i - \mathbf{\Phi}_i^T \hat{\mathbf{l}}_{i,ext}\end{aligned}$$

Lösen von

$$\hat{\mathbf{s}} = \hat{\mathbf{M}}^{-1} \hat{\boldsymbol{\eta}}$$

3.2.2. $\mathcal{O}(n^3)$ -ALGORITHMUS FÜR IMPLIZITE BESCHREIBUNG

Der Algorithmus für eine implizite Beschreibung der Bindungsgleichung basiert auf dem Gleichungssystem (3.7), dessen Gleichungen sich ebenfalls als Matrixgleichungen zu

$$\begin{bmatrix} \mathbf{M} & \mathbf{G}^T \bar{\boldsymbol{\Psi}} \\ \bar{\boldsymbol{\Psi}}^T \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \bar{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{l}'_{ext} \\ \boldsymbol{\gamma} \end{bmatrix} \quad (3.13)$$

zusammenfassen lassen, wenn die Definitionen aus Gl. (3.8) und zusätzlich

$$\begin{aligned}\bar{\boldsymbol{\Psi}} &= \text{blkdiag}(\bar{\boldsymbol{\Psi}}_1, \dots, \bar{\boldsymbol{\Psi}}_{n_K}) \\ \mathbf{l}'_{ext} &= [\mathbf{l}'_{1,ext}, \dots, \mathbf{l}'_{n_K,ext}]^T \\ \bar{\boldsymbol{\lambda}} &= [\bar{\boldsymbol{\lambda}}_1, \dots, \bar{\boldsymbol{\lambda}}_{n_B}]^T \\ \boldsymbol{\gamma} &= [\mathbf{c}_1^T \bar{\boldsymbol{\Psi}}_1, \dots, \mathbf{c}_{n_B}^T \bar{\boldsymbol{\Psi}}_{n_B}]^T\end{aligned}$$

3. Lösungsalgorithmen für Mehrkörpersysteme

angewendet wird. Wird das Gleichungssystem (3.13) nun einer Block-Gaußelimination

$$\begin{bmatrix} \mathbf{M} & \mathbf{G}^T \bar{\Psi} \\ \mathbf{0} & \mathbf{M}^* \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{l}'_{ext} \\ \bar{\Psi}^T \mathbf{G} \mathbf{M}^{-1} \mathbf{l}'_{ext} - \gamma \end{bmatrix}$$

mit

$$\mathbf{M}^* = \bar{\Psi}^T (\mathbf{G} \mathbf{M}^{-1} \mathbf{G}^T) \bar{\Psi} \quad (3.14)$$

unterzogen, können die Koordinaten $\bar{\lambda}$ der Zwangslasten mithilfe der letzten Zeile berechnet werden. Dazu muss jedoch die Matrix \mathbf{M}^* invertiert werden.

Durch Auswerten von Gleichung (3.14) mithilfe von Graphen lässt sich ein geschlossener Ausdruck für \mathbf{M}^* finden.

\mathbf{M} ist eine Blockdiagonalmatrix, deren Inverse blockweise berechnet werden kann.

$$\mathbf{M}^{-1} = \text{blkdiag}(\mathbf{M}_1^{-1}, \dots, \mathbf{M}_{n_K}^{-1})$$

Demnach besitzt $\mathbf{G} \mathbf{M}^{-1}$ denselben erweiterten Graphen wie \mathbf{G} , da sich die Positionen der von Null verschiedenen Einträge nicht verändern.

$$[\mathbf{G} \mathbf{M}^{-1}]_{ij} = \mathbf{G}_{ij} \mathbf{M}_j^{-1}$$

Der erweiterte Graph G (aug(\mathbf{G})) besitzt je einen Knoten für jede Bindung und je einen Knoten für jeden Körper. Ausgehend von den Bindungsknoten verläuft jeweils eine Kante zu dem Knoten des Vorgänger- und Nachfolgekörpers.

$$[\mathbf{G}]_{ij} = \sigma_{ij} \mathbf{G}_{ij} = \begin{cases} -\mathbf{E} & j = i \\ \mathbf{G}_{ij} & j = V(i) \\ \mathbf{0} & \text{sonst} \end{cases}$$

Wird nun der Graph $G(\mathbf{G} \mathbf{M}^{-1} \mathbf{G}^T)$ betrachtet, wie in Abschnitt 3.1.3 beschrieben wurde, repräsentieren dessen Knoten die Körper des Systems. Eine Kante zwischen zwei Knoten existiert, wenn die beiden Körper über eine Bindung miteinander in Beziehung stehen, oder einen gemeinsamen Vorgänger besitzen.

$$\begin{aligned} [\mathbf{G} \mathbf{M}^{-1} \mathbf{G}^T]_{ij} &= \sum_k \sigma_{ik} \sigma_{jk} \mathbf{G}_{ik} \mathbf{M}_k^{-1} \mathbf{G}_{jk}^T \\ &= \begin{cases} \mathbf{M}_i^{-1} + \mathbf{G}_{iV} \mathbf{M}_V^{-1} \mathbf{G}_{iV}^T & i = j, V = V(i) \\ \mathbf{G}_{ik} \mathbf{M}_k^{-1} \mathbf{G}_{jk}^T & k = V^+(i) \cap V^+(j) \end{cases} \end{aligned}$$

Somit berechnen sich die Elemente von \mathbf{M}^* zu

$$\mathbf{M}_{ij}^* = \begin{cases} \bar{\Psi}_i^T (\mathbf{M}_i^{-1} + \mathbf{G}_{iV} \mathbf{M}_V^{-1} \mathbf{G}_{iV}^T) \bar{\Psi}_i & i = j, V = V(i) \\ \bar{\Psi}_i^T (\mathbf{G}_{ik} \mathbf{M}_k^{-1} \mathbf{G}_{jk}^T) \bar{\Psi}_j & k = V^+(i) \cap V^+(j). \end{cases}$$

Der zugehörige Algorithmus zur Berechnung der Gelenkbeschleunigungen lautet:

Algorithmus 3.2 Impliziter $\mathcal{O}(n^3)$ -Algorithmus

```

parfor i=1:n
     $\mathbf{M}_{ii}^* = \bar{\Psi}_i^T \left( \mathbf{M}_i^{-1} + \mathbf{G}_{iV} \mathbf{M}_V^{-1} \mathbf{G}_{iV}^T \right) \bar{\Psi}_i$ 
     $\mathbf{M}_{kl}^* = \bar{\Psi}_k^T \left( \mathbf{G}_{ki} \mathbf{M}_i^{-1} \mathbf{G}_{lk}^T \right) \bar{\Psi}_l, (k, l) \in (N(i) \cup i) \times (N(i) \cup i)$ 
     $\gamma_i^* = \bar{\Psi}_i^T \left( \mathbf{M}_i^{-1} \mathbf{l}'_{i,ext} + \mathbf{G}_{iV} \mathbf{M}_V^{-1} \mathbf{l}'_{V,ext} \right) - \gamma_i$ 
end

Lösen von
 $\mathbf{M}^* \bar{\lambda} = \gamma^*$ 

parfor i=1:n
     $\mathbf{x}_i = \mathbf{M}_i^{-1} \left( \mathbf{l}'_{i,ext} + \bar{\Psi}_i \bar{\lambda}_i + \sum_{k \in N(i)} \mathbf{G}_{ki}^T \bar{\Psi}_k \bar{\lambda}_k \right)$ 
end

und optional
for i=1:n
     $\dot{\mathbf{s}}_i = \Psi_i^T \left( \mathbf{x}_i - \mathbf{G}_{iV(i)} \mathbf{x}_{V(i)} - \mathbf{c}_i \right)$ 
end

```

Weil der Algorithmus für die implizite Bindungsbeschreibung keine quadratische Matrix \mathbf{G} benötigt, kann er ebenso auf Systeme mit kinematischen Schleifen übertragen werden.

3.2.3. MODIFIZIERTER $\mathcal{O}(n^3)$ -ALGORITHMUS FÜR IMPLIZITE BESCHREIBUNG

Bei einem Vergleich der beiden vorgestellten Algorithmen fällt auf, dass sich Anzahl und Position der Nicht-Null-Elemente der Matrizen $\hat{\mathbf{M}}$ und \mathbf{M}^* stark unterscheiden. Der Algorithmus für die implizite Beschreibung scheint ungeeignet für breit gefächerte Topologien zu sein, da Einträge zwischen Körpern existieren, die einen gemeinsamen Vorgänger besitzen. Der explizite Algorithmus ist ungünstig für lange Ketten, da die Nicht-Null-Elemente zwischen den Körpern auftreten, die auf einem gemeinsamen Pfad liegen.

Zu berücksichtigen sind außerdem die Dimensionen der Matrizen. Haben die einzelnen Gelenke nur eine geringe Anzahl an Freiheitsgraden, also $\dim(\mathbf{s}) < \dim(\bar{\lambda})$, so ist die Faktorisierung der Matrix $\hat{\mathbf{M}}$ im Allgemeinen weniger aufwendig als die von \mathbf{M}^* .

Interessanterweise sind die Berechnungsvorschriften der Massenmatrizen (3.10) und (3.14) sehr ähnlich. Im Falle von $\hat{\mathbf{M}}$ werden jedoch die zusammengefassten Massen $\mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1}$ auf die freien Richtungen Φ projiziert, während für \mathbf{M}^* die Inverse der zusammengefassten Massen $\mathbf{G} \mathbf{M}^{-1} \mathbf{G}^T$ auf die gesperrten Lastrichtungen $\bar{\Psi}$ projiziert werden.

Verursacht wird dies durch die unterschiedliche Permutation des linearen Gleichungssystems. Insbesondere wird die Bewegungsgleichung im expliziten Fall nach den Schnittlasten \mathbf{l} und im impliziten Fall nach den Beschleunigungen \mathbf{x} umgestellt. Die Berechnung der Schnittlasten mithilfe der Bewegungsgleichungen besitzt jedoch mehrere Vorteile. Zum einen muss die Massenmatrix nicht invertiert werden und muss deshalb nicht zwingend regulär sein.¹ Zum anderen ist die Hauptdiagonale der Matrix \mathbf{G} aufgrund der günstigen Lage der Körperbezugssysteme

¹Dies ist von Vorteil, wenn Modellierer die Trägheitseigenschaften bezüglich einer gesperrten Richtung zu Null setzen.

mit Einheitstensoren besetzt. Somit kann die Bewegungsgleichung des Körpers i ohne Inversion einer Matrix nach der zugehörigen Schnittlast umgestellt werden. Es sei darauf hingewiesen, dass dies jedoch nur erreicht werden kann, wenn die Symmetrie der Koeffizientenmatrix wie in Gleichung (3.8) aufgegeben wird. Andernfalls muss immer die Massenmatrix \mathbf{M} auf der Hauptdiagonalen stehen. Die meisten verfügbaren numerischen Lösungsverfahren für schwach besetzte Matrizen leiten die Permutation jedoch basierend auf symmetrischen Matrizen her. Daher ist nicht davon auszugehen, dass bei deren Verwendung, wie bei der Formulierung mit impliziten Bindungsgleichungen üblich, diese effizientere Sortierung genutzt wird.

Aufgrund der unterschiedlichen Dimensionen kann in der impliziten Formulierung die Bewegungsgleichung nicht ohne weiteres nach den Koordinaten $\bar{\lambda}$ der Schnittlasten umgestellt werden. Mithilfe der speziellen Eigenschaften (2.38) des $\bar{\Psi}$ können die Gleichungen aber geschickt projiziert werden. Dazu wird die Bewegungsgleichung auf die freien Richtungen Φ und auf die gesperrten Richtungen $\bar{\Phi}$ projiziert. Zusätzlich werden die Beschleunigungen \mathbf{x} der Körper mithilfe der Koordinaten t_i bezüglich der Basis $\mathbf{G}^{-1} \cdot [\Phi, \bar{\Phi}]$ dargestellt.

$$\mathbf{x} = \mathbf{G}^{-1}\Phi\mathbf{t} + \mathbf{G}^{-1}\bar{\Phi}\bar{\mathbf{t}}$$

Da die Matrix \mathbf{G}^{-1} den Erreichbarkeitsgraphen repräsentiert, bedeutet diese Wahl der Basis, dass eine rekursive Kinematik verwendet wird, in der die Koordinaten \mathbf{t} für die Beschleunigung eine ähnliche Funktion wie die Gelenkgeschwindigkeiten \mathbf{s} für die Kinematik übernehmen.

Somit ergibt sich

$$\begin{bmatrix} \mathbf{E} & (\bar{\Phi}^T \mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1} \bar{\Phi}) & (\bar{\Phi}^T \mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1} \Phi) \\ \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & (\Phi^T \mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1} \bar{\Phi}) & (\Phi^T \mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1} \Phi) \end{bmatrix} \begin{bmatrix} \bar{\lambda} \\ \bar{\mathbf{t}} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \bar{\Phi}^T \mathbf{G}^{-T} \mathbf{l}'_{ext} \\ \gamma \\ \Phi^T \mathbf{G}^{-T} \mathbf{l}'_{ext} \end{bmatrix}.$$

Wird nun wieder die Gauß-Elimination ausgeführt, folgt

$$\begin{bmatrix} \mathbf{E} & (\bar{\Phi}^T \mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1} \bar{\Phi}) & (\bar{\Phi}^T \mathbf{G}^{-T} \mathbf{M} \mathbf{G}^{-1} \Phi) \\ \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \bar{\lambda} \\ \bar{\mathbf{t}} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \bar{\Phi}^T \mathbf{G}^{-T} \mathbf{l}'_{ext} \\ \gamma \\ \hat{\boldsymbol{\eta}}' \end{bmatrix}$$

mit

$$\hat{\boldsymbol{\eta}}' = \Phi^T \mathbf{G}^{-T} (\mathbf{l}'_{ext} - \mathbf{M} \mathbf{G}^{-1} \bar{\Phi} \gamma).$$

Auf diese Weise wurde durch eine geeignete Transformation und Umsortierung der Gleichungen und Variablen der Algorithmus für die explizite Beschreibung

$$\hat{\mathbf{M}}\mathbf{t} = \hat{\boldsymbol{\eta}}'$$

mithilfe der impliziten Formulierung hergeleitet. Voraussetzung dafür ist jedoch die Kenntnis der dualen Basen der Bindung, speziell der freien Bewegungsrichtungen Φ und der gesperrten Lastrichtungen $\bar{\Psi}$.

3.2.4. $\mathcal{O}(n)$ -ALGORITHMUS FÜR EXPLIZITE BESCHREIBUNG

Zur Berechnung großer Mehrkörpersysteme sind die Algorithmen der $\mathcal{O}(n^3)$ Klasse ungeeignet, da der Aufwand zum Lösen des linearen Gleichungssystems mit der Anzahl der Körper

zu stark ansteigt. Daher wurden in den 80er Jahren verschiedene $\mathcal{O}(n)$ -Algorithmen entwickelt. Es begann mit [Arm79] und wurde dann in [Fea83] verbessert. Den Durchbruch brachten [BJO86, BJO87] im deutsch- und [Fea87] im englischsprachigen Raum. Letztere enthielten erstmals Ansätze zur Berücksichtigung kinematischer Schleifen. Kurz danach wurde in [Weh88] gezeigt, dass sich die $\mathcal{O}(n)$ -Algorithmen als eine Matrixfaktorisierung darstellen lassen. Statt die Gleichungen blockweise wie in den $\mathcal{O}(n^3)$ -Algorithmen üblich im Gleichungssystem anzuordnen, werden sie nun körperweise und entsprechend des Topologiegraphen angeordnet. Dazu werden die Gleichungen (3.3a), (3.3b) und (3.3c) eines jeden Körper-Gelenkpaars zusammengefasst. Anschließend wird beginnend bei der Wurzel eine *Tiefensuche* oder eine *Breitensuche* ausgeführt und die Körper-Gelenk-Paare in umgekehrter Reihenfolge in das Gleichungssystem eingefügt. Somit wird sichergestellt, dass die Gleichungsblöcke der Nachfolgekörper immer über denen des Vorgängerkörpers erscheinen. Für einen Körper i und seinen Vorgänger $k = V(i)$ ergibt sich folgender Aufbau

$$\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots \\ \cdots & -\mathbf{E} & \mathbf{M}_i & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \cdots & \mathbf{0} & -\mathbf{E} & \Phi_i & \cdots & \mathbf{0} & \mathbf{G}_{ik} & \mathbf{0} & \cdots \\ \cdots & \Phi_i^T & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \\ \cdots & \mathbf{G}_{ik}^T & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{E} & \mathbf{M}_k & \mathbf{0} & \cdots \\ \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{E} & \Phi_k & \cdots \\ \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Phi_k^T & \mathbf{0} & \mathbf{0} & \cdots \\ \ddots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{l}_i \\ \mathbf{x}_i \\ \dot{\mathbf{s}}_i \\ \vdots \\ \mathbf{l}_k \\ \mathbf{x}_k \\ \dot{\mathbf{s}}_k \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{l}_{i,ext} \\ \mathbf{c}_i \\ \boldsymbol{\eta}_i \\ \vdots \\ \mathbf{l}_{k,ext} \\ -\mathbf{c}_k \\ \boldsymbol{\eta}_k \\ \vdots \end{bmatrix}. \quad (3.15)$$

Das entstandene Gleichungssystem besitzt schon annähernd eine obere Blockdreiecksform. Lediglich in den Spalten der Schnittlasten finden sich jeweils zwei Einträge in der unteren Dreieckshälfte. Um die Elimination dieser Einträge übersichtlicher zu gestalten, werden die relevanten Zeilen und Spalten des Gleichungssystems in folgendes Tableau geschrieben.

	\mathbf{l}_i	\mathbf{x}_i	$\dot{\mathbf{s}}_i$	\mathbf{x}_k	
I_i	$-\mathbf{E}$	\mathbf{M}_i^A	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{l}_{i,ext}^A$
A_i	$\mathbf{0}$	$-\mathbf{E}$	Φ_i	\mathbf{G}_{ik}	\mathbf{c}_i
D_i	Φ_i^T	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\boldsymbol{\eta}_i$
I_k	\mathbf{G}_{ik}^T	$\mathbf{0}$	$\mathbf{0}$	\mathbf{M}_k	$\mathbf{l}_{k,ext}$

Die erste Spalte enthält den Namen der Gleichung entsprechend Abb. 3.4 und die erste Zeile enthält die zugehörigen Variablen. Für Blattkörper, also Körper die keinen Nachfolger besitzen, gilt $\mathbf{M}_i^A = \mathbf{M}_i$ und $\mathbf{l}_{i,ext}^A = \mathbf{l}_{i,ext}$. Nun kann mithilfe der Gauß-Elimination das Tableau auf eine

3. Lösungsalgorithmen für Mehrkörpersysteme

obere Blockdreiecksform überführt werden.

	\mathbf{l}_i	\mathbf{x}_i	$\dot{\mathbf{s}}_i$	\mathbf{x}_k	
I_i	$-\mathbf{E}$	\mathbf{M}_i^A	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{l}_{i,ext}^A$
A_i	$\mathbf{0}$	$-\mathbf{E}$	Φ_i	\mathbf{G}_{ik}	\mathbf{c}_i
D_i	$\mathbf{0}$	$\mathbf{0}$	$\hat{\mathbf{M}}_i$	$\Phi_i^T \mathbf{M}_i^A \mathbf{G}_{ik}$	$\boldsymbol{\eta}_i + \Phi_i^T (\mathbf{l}_{i,ext}^A + \mathbf{M}_i^A \mathbf{c}_i)$
I_k	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{M}_k + \mathbf{G}_{ik}^T \mathbf{M}_i^A \mathbf{G}_{ik}$	$\mathbf{l}_{k,ext} + \mathbf{G}_{ik}^T \mathbf{l}_{i,ext}^a$

mit

$$\hat{\mathbf{M}}_i = \Phi_i^T \mathbf{M}_i^A \Phi_i \quad (3.16)$$

$$\mathbf{M}_i^a = \mathbf{M}_i^A - \mathbf{M}_i^A \Phi_i \hat{\mathbf{M}}_i^{-1} \Phi_i^T \mathbf{M}_i^A \quad (3.17)$$

$$\mathbf{l}_{i,ext}^a = \mathbf{l}_{i,ext} + \mathbf{M}_i^a \mathbf{c}_i - \mathbf{M}_i^A \Phi_i \hat{\mathbf{M}}_i^{-1} (\boldsymbol{\eta}_i + \Phi_i^T \mathbf{l}_{i,ext}^A).$$

Wird diese Elimination nun für alle Körper beginnend bei den Blattkörpern durchgeführt, so wird das gesamte Gleichungssystem in eine obere Blockdreiecksform überführt. Dabei werden in den Bewegungsgleichungen schrittweise die Schnittlasten der Vorgängerkörper eliminiert. Somit werden die Bewegungsgleichungen aller Körper sukzessive auf die Form von Blattkörpern gebracht.

$$\mathbf{M}_k^A \mathbf{x}_k = \mathbf{l}_{k,ext}^A + \mathbf{l}_k \quad (3.18)$$

mit

$$\mathbf{M}_k^A = \mathbf{M}_k + \sum_{i \in N(k)} \mathbf{G}_{ik}^T \mathbf{M}_i^a \mathbf{G}_{ik} \quad (3.19)$$

$$\mathbf{l}_{k,ext}^A = \mathbf{l}_{k,ext} + \sum_{i \in N(k)} \mathbf{G}_{ik}^T \mathbf{l}_{i,ext}^a. \quad (3.20)$$

Die \mathbf{M}_k^A beschreiben die kombinierten Trägheitseigenschaften des Körpers k und aller seiner Nachfolger. Im Gegensatz zu den $\bar{\mathbf{M}}_k$, vergleiche Gl. (3.12) und (3.17), wird in (3.19) der Korrekturterm $(\Phi_i^T \mathbf{M}_i^A)^T \hat{\mathbf{M}}_i^{-1} (\Phi_i^T \mathbf{M}_i^A)$ abgezogen. Dieser Term berücksichtigt den Einfluss der Gelenke bei der Berechnung der zusammengesetzten Massen, anstelle pauschal von einer festen Verbindung auszugehen, wie es beim $\mathcal{O}(n^3)$ -Algorithmus mit expliziter Beschreibung der Fall ist.

Das Gleichungssystem kann nun durch Rückwärtseinsetzen rekursiv gelöst werden. Es werden zuerst die Gelenkbeschleunigungen der Bindungen berechnet, die mit dem Inertialsystem verbunden sind. Mit diesen Werten kann die Beschleunigung der zugehörigen Körper berechnet werden. Ausgehend davon können die Gelenkbeschleunigungen der angeschlossenen Gelenke berechnet werden und von da an die Rekursion fortgeführt werden. Der gesamte Berechnungsablauf ist in Algorithmus 3.3 dargestellt.

Algorithmus 3.3 Expliziter $\mathcal{O}(n)$ -Algorithmus

Rückwärtsrekursion:

```

for i=n:-1:n
     $M_i^A = M_i + \sum_{k \in N(i)} G_{ki}^T M_k^a G_{ki}$ 
     $l_{i,ext}^A = l_{i,ext} + \sum_{k \in N(i)} G_{ki}^T l_{k,ext}^a$ 
     $\hat{M}_i = \Phi_i^T M_i^A \Phi_i$ 
     $M_i^a = M_i - (\Phi_i^T M_i)^T \hat{M}_i^{-1} (\Phi_i^T M_i)$ 
     $l_{i,ext}^a = l_{i,ext}^A - M_i^a c_i - M_i \Phi_i \hat{M}_i^{-1} (\eta_i + \Phi_i^T l_{i,ext}^A)$ 
end
    
```

end

 Vorwärtsrekursion ($x_0 = \mathbf{0}$)

```

for i=1:n
     $\hat{s}_i = \hat{M}_i^{-1} (\eta_i + \Phi_i^T (l_{i,ext} - M_i (G_{iV} x_V - c_i)))$ 
     $x_i = G_{iV} x_V + \Phi_i \cdot \hat{s}_i + c_i$ 
end
    
```

 end

Während der Berechnung aller Gelenkbeschleunigungen \hat{s}_i müssen folglich n_K Gleichungssysteme der Dimension $\dim(s_i)$, die der Anzahl der Freiheitsgrade entspricht, gelöst werden.

3.2.5. $\mathcal{O}(n)$ -ALGORITHMUS FÜR IMPLIZITE BESCHREIBUNG

Der $\mathcal{O}(n)$ -Algorithmus für eine implizite Gelenkbeschreibung wurde erstmals in [Bar96, Tas01] vorgestellt. Darin wird das Gleichungssystem (3.13) mithilfe einer verallgemeinerten Cholesky-Faktorisierung, der LDL^T -Faktorisierung, für schwach besetzte Matrizen gelöst. Statt die Gleichungen blockweise anzuordnen, werden sie entsprechend des Topologiegraphen angeordnet. Ähnlich dem Algorithmus für die explizite Beschreibung wird ausgehend von der Wurzel des aufspannenden Baumes eine *Tiefen-* oder *Breitensuche* ausgeführt und die Gleichungsblöcke in umgekehrter Reihenfolge in das Gleichungssystem eingefügt. Dadurch ergibt sich für einen Körper i und seinen Vorgänger $k = V(i)$

$$\begin{bmatrix}
 \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
 \cdots & M_i & -\bar{\Psi}_i & \cdots & \mathbf{0} & \mathbf{0} & \cdots \\
 \cdots & -\bar{\Psi}_i^T & \mathbf{0} & \cdots & \bar{\Psi}_i^T G_{ik} & \mathbf{0} & \cdots \\
 & \vdots & \vdots & & \vdots & \vdots & \\
 \cdots & \mathbf{0} & G_{ik}^T \bar{\Psi}_i & \cdots & M_k & -\bar{\Psi}_k & \cdots \\
 \cdots & \mathbf{0} & \mathbf{0} & \cdots & -\bar{\Psi}_k^T & \mathbf{0} & \cdots \\
 \ddots & \vdots & \vdots & & \vdots & \vdots & \ddots
 \end{bmatrix}
 \begin{bmatrix}
 \vdots \\
 \mathbf{x}_i \\
 \bar{\lambda}_i \\
 \vdots \\
 \mathbf{x}_k \\
 \bar{\lambda}_k \\
 \vdots
 \end{bmatrix}
 =
 \begin{bmatrix}
 \vdots \\
 l_{i,ext} \\
 \bar{\Psi}_i^T c_i \\
 \vdots \\
 l_{k,ext} \\
 \bar{\Psi}_k^T c_k \\
 \vdots
 \end{bmatrix}. \quad (3.21)$$

Im Unterschied zu [Bar96] soll die Herleitung des Algorithmus nicht über die Berechnung der LDL^T -Faktorisierung, sondern mithilfe der Gauß-Elimination erfolgen. Auf diese Weise wird die Ähnlichkeit zu der expliziten Version des vorherigen Abschnittes deutlicher. Die Blöcke auf der Hauptdiagonalen, die Verbindung zum Vorgänger sowie die rechte Seite werden kompakt als

3. Lösungsalgorithmen für Mehrkörpersysteme

	\mathbf{x}_i	$\bar{\boldsymbol{\lambda}}_i$	\mathbf{x}_k	
\mathbf{I}_i	\mathbf{M}_i^A	$-\bar{\boldsymbol{\Psi}}_i$	$\mathbf{0}$	$\mathbf{l}'_{i,ext}{}^A$
\mathbf{X}_i	$-\bar{\boldsymbol{\Psi}}_i^T$	$\mathbf{0}$	$\bar{\boldsymbol{\Psi}}_i^T \mathbf{G}_{ik}$	$\bar{\boldsymbol{\Psi}}_i^T \mathbf{c}_i$
\mathbf{I}_k	$\mathbf{0}$	$\mathbf{G}_{ik}^T \bar{\boldsymbol{\Psi}}_i$	\mathbf{M}_k	$\mathbf{l}'_{k,ext}$

dargestellt, wobei für Blattkörper $\mathbf{M}_i^A = \mathbf{M}_i$ und $\mathbf{l}'_{i,ext}{}^A = \mathbf{l}'_{i,ext}$ gilt. Wird der Block auf der Hauptdiagonalen nun in die Form einer oberen Dreiecksmatrix überführt, ergibt sich

	\mathbf{x}_i	$\bar{\boldsymbol{\lambda}}_i$	\mathbf{x}_k	
\mathbf{I}_i	\mathbf{M}_i^A	$-\bar{\boldsymbol{\Psi}}_i$	$\mathbf{0}$	$\mathbf{l}'_{i,ext}{}^A$
\mathbf{X}_i	$\mathbf{0}$	$\tilde{\mathbf{M}}_i^{-1}$	$-\bar{\boldsymbol{\Psi}}_i^T \mathbf{G}_{ik}$	$\bar{\boldsymbol{\Psi}}_i^T \left((\mathbf{M}_i^A)^{-1} \mathbf{l}'_{i,ext}{}^A + \mathbf{c}_i \right)$
\mathbf{I}_k	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{M}_k + \mathbf{G}_{ik}^T \mathbf{M}_i^a \mathbf{G}_{ik}$	$\mathbf{l}'_{k,ext} + \mathbf{G}_{ik}^T \mathbf{l}'_{i,ext}{}^a$

mit

$$\begin{aligned}
 \tilde{\mathbf{M}}_i^{-1} &= \bar{\boldsymbol{\Psi}}_i^T (\mathbf{M}_i^A)^{-1} \bar{\boldsymbol{\Psi}}_i & (3.22) \\
 \mathbf{M}_i^a &= \bar{\boldsymbol{\Psi}}_i \tilde{\mathbf{M}}_i \bar{\boldsymbol{\Psi}}_i^T \\
 \mathbf{l}'_{i,ext}{}^a &= \mathbf{M}_i^a \left((\mathbf{M}_i^A)^{-1} \mathbf{l}'_{i,ext}{}^A + \mathbf{c}_i \right).
 \end{aligned}$$

Mithilfe dieser Gleichungen lassen sich die expliziten Abhängigkeiten der Bewegungsgleichungen von den Koordinaten der Zwangslasten ihrer Nachfolger eliminieren und es folgt

$$\mathbf{M}_k^A \cdot \mathbf{x}_k = \mathbf{l}'_{k,ext}{}^A + \bar{\boldsymbol{\Psi}}_k \bar{\boldsymbol{\lambda}}_k \quad (3.23)$$

mit

$$\begin{aligned}
 \mathbf{M}_k^A &= \mathbf{M}_k + \sum_i \mathbf{G}_{ik}^T \mathbf{M}_i^a \mathbf{G}_{ik} \\
 \mathbf{l}'_{k,ext}{}^A &= \mathbf{l}'_{k,ext} + \sum_i \mathbf{G}_{ik}^T \mathbf{l}'_{i,ext}{}^a.
 \end{aligned}$$

Auf diese Weise können die Bewegungsgleichungen aller Körper rekursiv auf die Form eines Blattkörpers gebracht und damit die Matrix in eine obere Blocksdreiecksform überführt werden. Dieser Schritt wird oft als *Rückwärtsrekursion* bezeichnet.

Im Anschluss folgt die *Vorwärtsrekursion*. Sie beginnt mit den Gelenken, die direkt mit dem Inertialsystem verbunden sind. Für diese können die Koordinaten $\bar{\boldsymbol{\lambda}}_i$ der Schnittlasten berechnet und damit die Beschleunigungen \mathbf{x}_i der angeschlossenen Körper ermittelt werden. Dieses Vorgehen kann nun rekursiv für alle mit diesem Körper angeschlossenen Bindungen fortgesetzt werden. Sollen zusätzlich die Gelenkbeschleunigungen $\dot{\mathbf{s}}_i$ bestimmt werden, um z.B. direkt die Relativkinematik als Zustände für die numerische Integration zu verwenden, so muss im Anschluss Gl. (3.4a) für jedes Gelenk ausgewertet werden. Das gesamte Vorgehen ist im Algorithmus 3.4 nochmals zusammengefasst.

Algorithmus 3.4 Impliziter $\mathcal{O}(n)$ -Algorithmus

Rückwärtsrekursion:

```

for i=n:-1:1
     $M_i^A = M_i + \sum_{k \in N(i)} \mathbf{G}_{ki}^T M_k^a \mathbf{G}_{ki}$ 
     $\mathbf{l}'_{i,ext} = \mathbf{l}'_{i,ext} + \sum_{k \in N(i)} \mathbf{G}_{ki}^T \mathbf{l}_i^a$ 
     $M_i^a = \bar{\Psi}_k \left( \bar{\Psi}_i^T \left( M_i^A \right)^{-1} \bar{\Psi}_i \right)^{-1} \bar{\Psi}_k^T$ 
     $\mathbf{l}_i^a = M_k^a \left( \left( M_k^A \right)^{-1} \mathbf{l}'_{k,ext} + \mathbf{c}_k \right)$ 
end
    
```

end

 Vorwärtsrekursion ($\mathbf{x}_0 = \mathbf{0}$)

```

for i=1:n
     $\mathbf{l}_i = \bar{\Psi}_i \bar{\lambda}_i = M_i^a \mathbf{G}_{iV} \mathbf{x}_V - \mathbf{l}_i^a$ 
     $\mathbf{x}_i = \left( M_i^A \right)^{-1} \left( \mathbf{l}'_{i,ext} + \mathbf{l}_i \right)$ 
     $\dot{\mathbf{s}}_i = \Psi_i^T \left( \mathbf{x}_i - \mathbf{G}_{iV(i)} \mathbf{x}_{V(i)} + \mathbf{c}_i \right)$  (optional)
end
    
```

 end

Während der gesamten Berechnung der Beschleunigungen \mathbf{x}_i bzw. $\dot{\mathbf{s}}_i$ müssen somit $2 \cdot n_K$ Gleichungssysteme der Dimensionen 6 und $\dim(\bar{\lambda}_i) = 6 - \dim(\mathbf{s}_i)$ gelöst werden.

Es wird deutlich, dass dieser Algorithmus sehr stark der expliziten Form ähnelt. In der Tat ist es sogar möglich, sowohl implizit als auch explizit formulierte Gelenke mit demselben Algorithmus zu behandeln. Beide Algorithmen basieren auf dem Ansatz, die Bewegungsgleichungen der Einzelkörper auf die Form eines Blattkörpers zu überführen, indem die Schnittlasten der nachfolgenden Gelenke eliminiert werden, vergleiche Gl. (3.18) und (3.23). Daher sind die Massenmatrizen M_i^A identisch und die $\mathbf{l}'_{i,ext}$ und $\mathbf{l}^a_{i,ext}$ unterscheiden sich lediglich durch die Berücksichtigung der bekannten Koordinaten der Schnittlasten $\Psi_k \lambda_k$. Somit ergibt sich folgender Zusammenhang:

$$\begin{aligned}
 M_i^a &= M_i^A - M_i^A \Phi_i \hat{M}_i^{-1} \Phi_i^T M_i^A \\
 &= \bar{\Psi}_i \tilde{M}_i \bar{\Psi}_i^T
 \end{aligned} \tag{3.24}$$

$$\begin{aligned}
 \mathbf{l}'_{i,ext} &= \mathbf{l}_{i,ext} - M_i^a \mathbf{c}_i - M_i^A \Phi_i \hat{M}_i^{-1} \left(\boldsymbol{\eta}_i + \Phi_i^T \mathbf{l}_{i,ext} \right) \\
 &= M_i^a \mathbf{x}_i^i.
 \end{aligned} \tag{3.25}$$

Dies kann immer dann von Vorteil sein, falls nur eine implizite Gelenkbeschreibung vorliegt und die Berechnung von Φ sehr aufwendig wäre.

Die vorgestellte Permutation des Gleichungssystems entspricht der *Perfect Elimination Order* [DER86, Dav06], die für alle baumstrukturierten Systeme existiert. Eine Perfect Elimination Order bedeutet, dass die Matrixfaktoren der LDL^T -Faktorisierung keinen *Fill-in* besitzen. Als Fill-In werden von Null verschiedene Einträge in den Matrixfaktoren bezeichnet, die im Vergleich zu der Ausgangsmatrix hinzugekommen sind. Wird das Gleichungssystem (3.13) mithilfe eines Lösungsverfahrens für schwach besetzte Matrizen gelöst, so sucht der Löser zuerst nach einer geeigneten Permutation der Matrix, bevor er sie faktorisiert und das Gleichungssystem löst. Das am weitesten verbreitete Verfahren zum Finden einer geeigneten Permutation ist die *Minimum Degree* Sortierung, die als *Approximate Minimum Degree* laufezeitoptimiert implementiert wurde [ADD04]. Es lässt sich leicht zeigen, dass dieser Algorithmus immer die Perfect Elimination Order findet, sofern sie existiert. Daher beschreibt der hier vorgestellte Algorithmus den internen

Rechenablauf eines Löser für schwach besetzte Matrizen.

3.2.6. MODIFIZIERTER $\mathcal{O}(n)$ -ALGORITHMUS FÜR IMPLIZITE BESCHREIBUNG

Ähnlich dem modifizierten $\mathcal{O}(n^3)$ -Algorithmus für die implizite Beschreibungsform lässt sich ebenso der explizit formulierte $\mathcal{O}(n)$ -Algorithmus aus der impliziten Formulierung erzeugen. Dazu wird die Beschleunigung des Körpers i über der Basis $[\bar{\Phi}_i, \bar{\Phi}_i]$ mit den Koordinaten \bar{t}_i dargestellt

$$\mathbf{x}_i = \Phi_i \mathbf{t}_i + \bar{\Phi}_i \bar{t}_i.$$

Zusätzlich wird die Bewegungsgleichung auf die freien und gesperrten Richtungen projiziert. Somit ergibt sich in der kompakten Schreibweise

	$\bar{\lambda}_i$	\bar{t}_i	t_i	\mathbf{x}_k	
I_i^2	$-E$	$\bar{\Phi}_i^T M_i^A \bar{\Phi}_i$	$\bar{\Phi}_i^T M_i^A \Phi_i$	$\mathbf{0}$	$\bar{\Phi}_i^T l_{i,ext}^A$
X_i	$\mathbf{0}$	$-E$	$\mathbf{0}$	$\bar{\Psi}_i^T G_{ik}$	$-\bar{\Psi}_i^T c_i$
I_i^1	$\mathbf{0}$	$\Phi_i^T M_i^A \bar{\Phi}_i$	\hat{M}_i	$\mathbf{0}$	$\Phi_i^T l_{i,ext}^A$
I_k	$G_{ik}^T \bar{\Psi}_i$	$\mathbf{0}$	$\mathbf{0}$	M_k	$l'_{k,ext}$

Wird nun die Gauß-Elimination ausgeführt, so folgt

	$\bar{\lambda}_i$	\bar{t}_i	t_i	\mathbf{x}_k	
I_i^2	$-E$	$\bar{\Phi}_i^T M_i^A \bar{\Phi}_i$	$\bar{\Phi}_i^T M_i^A \Phi_i$	$\mathbf{0}$	$\bar{\Phi}_i^T l_{i,ext}^A$
X_i	$\mathbf{0}$	$-E$	$\mathbf{0}$	$\bar{\Psi}_i^T G_{ik}$	$-\bar{\Psi}_i^T c_i$
I_i^1	$\mathbf{0}$	$\mathbf{0}$	\hat{M}_i	$\Phi_i^T M_i^A \bar{\Phi}_i \bar{\Psi}_i^T G_{ik}$	$\Phi_i^T (l_{i,ext}^A - M_i^A \bar{\Phi}_i \bar{\Psi}_i^T c_i)$
I_k	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$M_k + G_{ik}^T M_i^A G_{ik}$	$l'_{k,ext} + G_{ik}^T l_{i,ext}^A$

mit

$$M_i^A = \bar{\Psi}_i \bar{\Phi}_i^T (M_i^a) \bar{\Phi}_i \bar{\Psi}_i^T$$

$$l_{i,ext}^A = \bar{\Psi}_i \bar{\Phi}_i^T (l_{i,ext}^a - (M_i^a) \bar{\Phi}_i \bar{\Psi}_i^T c_i - M_i^A \bar{\Phi}_i \hat{M}_i^{-1} \Phi_i^T l_{i,ext}^A).$$

3. Lösungsverfahren für Mehrkörpersysteme

Mithilfe der Gleichungen (3.24) lässt sich zeigen, dass $M_i'^a$ und M_i^a identisch sind, denn

$$\begin{aligned}
 M_i'^a &= \bar{\Psi}_i \bar{\Phi}_i^T (M_i^a) \bar{\Phi}_i \bar{\Psi}_i^T \\
 &= \bar{\Psi}_i \bar{\Phi}_i^T \left(\bar{\Psi}_i \tilde{M}_i \bar{\Psi}_i^T \right) \bar{\Phi}_i \bar{\Psi}_i^T \\
 &= \bar{\Psi}_i \left(\bar{\Phi}_i^T \bar{\Psi}_i \right) \tilde{M}_i \left(\bar{\Psi}_i^T \bar{\Phi}_i \right) \bar{\Psi}_i^T \\
 &= \bar{\Psi}_i \tilde{M}_i \bar{\Psi}_i^T \\
 &= M_i^a.
 \end{aligned}$$

Analog kann unter Nutzung von Gl. (3.25) gezeigt werden, dass $l_{i,ext}'^a = l_{i,ext}^a$ gelten muss.

Dieser Sachverhalt lässt sich an folgendem Beispiel veranschaulichen. Zur Darstellung der Tensoren wird eine Basis B gewählt, deren n_f erste Basisvektoren mit den Vektoren aus Φ zusammenfallen. Demnach lautet die Komponentendarstellung von Φ bezüglich der Basis B

$${}_B \Phi = \begin{bmatrix} E \\ \mathbf{0} \end{bmatrix}.$$

Ferner ist die Komponentendarstellung der Massenmatrix bezüglich der Basis B entsprechend partitioniert.

$${}_{BB} M = \begin{bmatrix} {}_{BB} M_{11} & {}_{BB} M_{12} \\ {}_{BB} M_{12}^T & {}_{BB} M_{22} \end{bmatrix}$$

Damit ergibt sich \hat{M}^{-1} zu

$${}_{BB} \hat{M}^{-1} = {}_{BB} M_{11}^{-1}$$

und die transformierte Massenmatrix M^a zu

$$\begin{aligned}
 M^a &= M - M \Phi \hat{M}^{-1} \Phi^T M \\
 {}_{BB} M_a &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}_{BB} M_{22} - {}_{BB} M_{12}^T \cdot {}_{BB} M_{11}^{-1} \cdot {}_{BB} M_{12} \end{bmatrix}. \tag{3.26}
 \end{aligned}$$

Es werden also die Komponenten, die mit den freien Richtungen zusammenhängen, eliminiert. Entsprechend der Forderung nach einer dualen Basis muss das Produkt $\bar{\Psi}_i \bar{\Phi}_i^T$ in Komponentendarstellung

$${}_{BB} \bar{\Psi}_i \cdot {}_{BB} \bar{\Phi}_i^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix}$$

ergeben. Und damit führt eine Multiplikation mit $\bar{\Psi}_i \bar{\Phi}_i^T$ von links und mit $\bar{\Phi}_i \bar{\Psi}_i^T$ von rechts zu einer Auswahl der rechten unteren Blockmatrix. Die restlichen Elemente ergeben sich zu null. Da ${}_{BB} M_a$ eben diese Struktur schon besitzt, kann die Multiplikation mit $\bar{\Psi}_i \bar{\Phi}_i^T$ entfallen.

Sofern die gewählte Basis mit den Vektoren aus Φ übereinstimmt, kann Gl. (3.26) benutzt werden, um eine effektive Implementierung zur Berechnung der M^a zu erhalten. Außerdem ist in diesem Falle ein generischer Permutationsalgorithmus in der Lage, die Gleichungen so zu sortieren, dass mit der impliziten Formulierung der Algorithmus für die explizite Beschreibung ausgeführt wird.

3.3. LÖSUNGsalgorithmen AM BEISPIEL EINER EBENEN PENDELKETTE

Zur Veranschaulichung der in den vorherigen Abschnitten vorgestellten Lösungsverfahren sollen diese anhand einer ebenen Pendelkette mit N Körpern, siehe Abb. 3.6, demonstriert werden. Dieses System ist interessant, da hier die Anzahl N der Körper und Gelenke leicht verändert werden kann. Ermöglicht wird somit die Untersuchung des Rechenzeitverhaltens bezüglich N . Obwohl die Simulation eines N -Pendels für sich wenig praxisrelevant ist, kann die Struktur des Modells doch in vielfältiger Form wiedergefunden werden. Zum einen wäre die Simulation von Seilen, wie sie z.B. bei großen Kränen vorgefunden werden, zu nennen. Ein weiteres Anwendungsbeispiel ist die Simulation von Ketten. Dies kann Traktionsketten in Fahrwerken, Antriebsketten in Getrieben oder sogar Ketten als Bestandteil des Arbeitswerkzeuges bei einem Kettenbagger umfassen. Eine andere potenzielle Anwendung stellt die Simulation von Bandanlagen dar, mit denen z.B. Abraum oder Kohle über mehrere Kilometer transportiert werden kann.

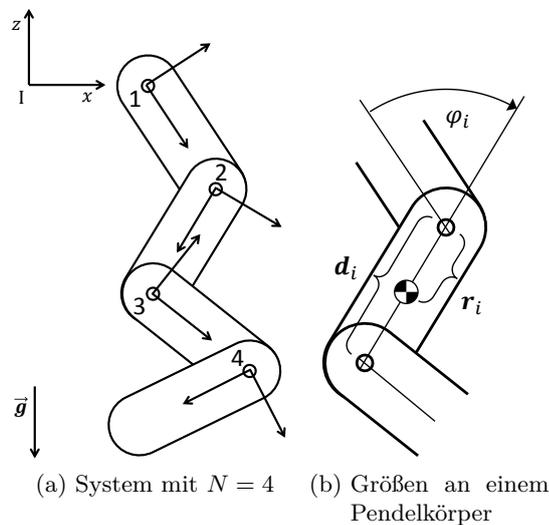


Abbildung 3.6.: Mehrfachpendel

3.3.1. ALLGEMEINE FESTLEGUNGEN

Zur Beschreibung der Bewegung wird ein Inertialsystem verwendet, siehe Abb. 3.6a, dessen x -Achse entlang des Horizonts und dessen z -Achse entgegen der Erdbeschleunigung \vec{g} zeigt. Die y -Achse ergibt sich aus der Forderung nach einem orthonormalen Rechtssystem. Die Bewegung des gesamten Pendels findet ausschließlich in der $x - z$ -Ebene des Inertialsystems statt. Deswegen können die translatorischen Komponenten entlang der y -Achse und die rotatorischen Komponenten um die x - und z -Achse vernachlässigt werden. Es müssen lediglich noch Vektoren mit drei anstelle von sechs Komponenten behandelt werden. Alle Vektoren werden bezüglich des Inertialsystems angeschrieben.

Alle Körper bilden zusammen eine Pendelkette und sind jeweils durch ein Drehgelenk mit dem Drehwinkel q_i und der Winkelgeschwindigkeit $s_i = \dot{q}_i$ miteinander verbunden. Die Drehachsen aller Drehgelenke liegen parallel zu der y -Achse des Inertialsystems. Entsprechend Abschnitt

2.2.5 ergibt sich der Vektor der freien Richtungen zu

$$\Phi = \frac{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}} := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

und die Vektoren der gesperrten Richtungen zu

$$\bar{\Psi} = \frac{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}} := \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Aufgrund der Parallelität von Drehachse und Drehgeschwindigkeit vereinfacht sich der geschwindigkeitsabhängige Term der Beschleunigung zu

$$\mathbf{c}_\Phi = \mathbf{0}.$$

Jeder Pendelkörper besitzt ein körperfestes Bezugssystem, das sich durch Drehung um die y -Achse aus dem Inertialsystem ergibt. Die x -Achse verläuft entlang der Verbindungslinie der beiden Drehgelenke. Die Länge der Verbindungslinie betrage d und der Schwerpunkt liege entlang der körperfesten x -Achse in der Entfernung r . Somit lauten die zugehörigen Abstandsvektoren, angeschrieben im Inertialsystem

$$\mathbf{d}_i = \begin{bmatrix} d_{i,x} \\ d_{i,z} \end{bmatrix} = \begin{bmatrix} \cos \varphi_i \\ -\sin \varphi_i \end{bmatrix} d$$

und

$$\mathbf{r}_i = \begin{bmatrix} r_{i,x} \\ r_{i,z} \end{bmatrix} = \begin{bmatrix} \cos \varphi_i \\ -\sin \varphi_i \end{bmatrix} r.$$

Die Ausrichtung des körperfesten Bezugssystems eines Pendelkörpers i , beschrieben durch die zugehörige Rotationsmatrix $\mathbf{A}_{i,I}$, lautet

$$\mathbf{A}_{i,I} = \begin{bmatrix} \cos \varphi_i & -\sin \varphi_i \\ \sin \varphi_i & \cos \varphi_i \end{bmatrix}$$

mit

$$\varphi_i = \sum_{k=1}^i q_k.$$

Die Transformationsmatrix zwischen den beiden Gelenkpunkten i und $i + 1$ lautet nach Gl. (2.10)

$$\mathbf{G}_{i+1,i} = \begin{bmatrix} 1 & 0 & 0 \\ d_{i,z} & 1 & 0 \\ -d_{i,x} & 0 & 1 \end{bmatrix}.$$

Bei der Transformation der Beschleunigungen muss der geschwindigkeitsabhängige Term, siehe Gl. (2.51),

$$\mathbf{c}_{i+1,G} = \begin{bmatrix} 0 \\ d_{i,x} \\ d_{i,z} \end{bmatrix} \dot{\varphi}_i^2$$

beachtet werden. Die Masse der Pendelkörper betrage m und das Trägheitsmoment bezüglich des Schwerpunktes um die y -Achse sei Θ . Es ergibt sich die Massenmatrix bezüglich des körperfesten Bezugssystems, angeschrieben im Inertialsystem, zu

$$\mathbf{M}_i = \begin{bmatrix} \Theta + m \cdot |\mathbf{r}_i|^2 & m \cdot r_{i,z} & -m \cdot r_{i,x} \\ m \cdot r_{i,z} & m & 0 \\ -m \cdot r_{i,x} & 0 & m \end{bmatrix}.$$

Der zugehörige Lastvektor, gemäß Gl. (2.25), setzt sich aus der Schwerkraft und den Zentrifugalkräften zusammen.

$$\mathbf{l}_{i,ext} = \begin{bmatrix} r_{i,x} \\ 0 \\ -1 \end{bmatrix} \cdot m \cdot g + \begin{bmatrix} 0 \\ r_{i,x} \\ r_{i,z} \end{bmatrix} \cdot m \cdot \dot{\varphi}_i^2$$

Diese Informationen sind ausreichend, um ausgehend von den Anfangswerten \mathbf{q}_0 und \mathbf{s}_0 die Bewegung des gesamten Pendels mittels numerischer Integration zu beschreiben. Dazu werden die Gelenkbeschleunigungen $\dot{\mathbf{s}}$ benötigt, die über die vorgestellten Algorithmen aus Kapitel 3.2 ermittelt werden können.

3.3.2. $\mathcal{O}(n^3)$ -ALGORITHMUS FÜR IMPLIZITE BESCHREIBUNGEN

Entsprechend der Herleitung in Abschnitt 3.2.2 wird aufgrund der speziellen Anordnung der Gleichungen und Variablen die Bewegungsgleichung nach den Beschleunigungen in Abhängigkeit der Zwangslasten umgestellt. Diese werden anschließend in die Beschleunigungszwangsbedingungen eingesetzt. Wie in Abb. 3.7a zu erkennen ist, hängt im Falle des Pendels die Beschleunigung des freigeschnittenen i -ten Körpers von den Zwangslasten i und $i + 1$ ab.

Die i -te Beschleunigungszwangsbedingung hängt wiederum von den Beschleunigungen der Körper $i - 1$ und i ab und ist demnach indirekt abhängig von den Zwangslasten $i - 1$, i und $i + 1$. Eine Ausführung für jede Beschleunigungszwangsbedingung entsprechend Algorithmus 3.2 ergibt im Falle des Pendels für \mathbf{M}^* in Blockdarstellung eine Bandstruktur, wie sie in Abb. 3.7b dargestellt ist.

Zur Berechnung der einzelnen Blöcke wird die Inverse der Massenmatrix eines einzelnen Körpers benötigt. Diese ergibt sich zu

$$\mathbf{M}_i^{-1} = \frac{1}{\Theta} \begin{bmatrix} 1 & -r_{i,z} & r_{i,x} \\ -r_{i,z} & \frac{\Theta}{m} + r_{i,z}^2 & -r_{i,x} \cdot r_{i,z} \\ r_{i,x} & -r_{i,x} \cdot r_{i,z} & \frac{\Theta}{m} + r_{i,x}^2 \end{bmatrix}.$$

3. Lösungsalgorithmen für Mehrkörpersysteme

Da die Beschleunigungszwangsbedingung i jedoch direkt im Körper Bezugssystem i ausgewertet wird, muss sie zusätzlich auf diesen Punkt transformiert werden.

$$\mathbf{G}_{i+1,i} \cdot \mathbf{M}_i^{-1} \cdot \mathbf{G}_{i+1,i}^T = \frac{1}{\Theta} \begin{bmatrix} 1 & -\tilde{r}_{i,z} & \tilde{r}_{i,x} \\ -\tilde{r}_{i,z} & \frac{\Theta}{m} + \tilde{r}_{i,z}^2 & -\tilde{r}_{i,x} \cdot \tilde{r}_{i,z} \\ \tilde{r}_{i,x} & -\tilde{r}_{i,x} \cdot \tilde{r}_{i,z} & \frac{\Theta}{m} + \tilde{r}_{i,x}^2 \end{bmatrix}$$

mit

$$\tilde{\mathbf{r}}_i = \mathbf{r}_i - \mathbf{d}_i$$

Interessanterweise bleiben die Ausdrücke erhalten und es wird lediglich der Abstand \mathbf{r}_i vom Körper Bezugspunkt i zum Schwerpunkt durch $\tilde{\mathbf{r}}_i = \mathbf{r}_i - \mathbf{d}_i$ ersetzt, der die Entfernung vom Körper Bezugspunkt $i + 1$ zum Schwerpunkt angibt. Durch Multiplizieren des Ausdruckes von links und rechts mit $\bar{\Psi}_i^T$ bzw. $\bar{\Psi}_i$ bleiben nur noch die Komponenten entlang der gesperrten Richtungen erhalten.

$$\bar{\Psi}_i^T \left(\mathbf{G}_{i+1,i} \cdot \mathbf{M}_i^{-1} \cdot \mathbf{G}_{i+1,i}^T \right) \bar{\Psi}_i = \frac{1}{\Theta} \begin{bmatrix} \frac{\Theta}{m} + \tilde{r}_{i,z}^2 & -\tilde{r}_{i,x} \cdot \tilde{r}_{i,z} \\ -\tilde{r}_{i,x} \cdot \tilde{r}_{i,z} & \frac{\Theta}{m} + \tilde{r}_{i,x}^2 \end{bmatrix}$$

Die Blockmatrizen in \mathbf{M}^* besitzen eben diese Gestalt.

Aufgrund der Bandstruktur der Matrix \mathbf{M}^* lässt sich zeigen, dass es für dieses Beispiel nicht sinnvoll wäre, das entstehende lineare Gleichungssystem mit einem Löser für dicht besetzte Matrizen zu berechnen. Würde jedoch ein Löser für schwach besetzte Matrizen zum Einsatz kommen, könnte das System in $\mathcal{O}(n)$ gelöst werden. Diese Aussage lässt sich jedoch nicht verallgemeinern, da z.B. ein Mehrkörpersystem bestehend aus einem Basiskörper der direkt mit einer beliebigen Anzahl anderer Körper verbunden ist, eine voll besetzte Matrix \mathbf{M}^* ergeben würde, deren Lösung aufgrund der Notwendigkeit ihrer Invertierung $\mathcal{O}(n^3)$ Operationen benötigt.

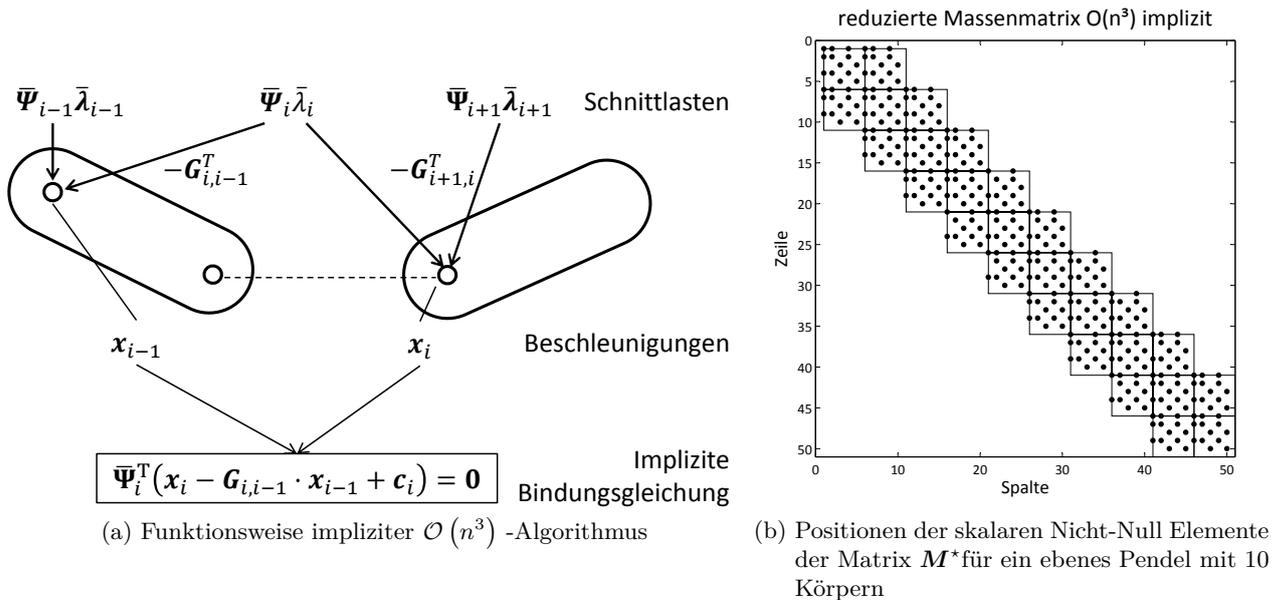


Abbildung 3.7.: Pendel mit $\mathcal{O}(n^3)$ -Algorithmus

3.3.3. $\mathcal{O}(n^3)$ -ALGORITHMUS FÜR EXPLIZITE BESCHREIBUNG

Zur Durchführung des $\mathcal{O}(n^3)$ -Algorithmus für eine explizite Beschreibung des MKS, entsprechend Algorithmus 3.1, gilt es zuerst die zusammengesetzten Massenmatrizen zu berechnen. Dazu muss die Massenmatrix mittels der Beziehung $\mathbf{G}_{i+1,i}^T \mathbf{M}_{i+1} \mathbf{G}_{i+1,i}$ transformiert werden. Es ergibt sich

$$\mathbf{G}_{i+1,i}^T \mathbf{M}_{i+1} \mathbf{G}_{i+1,i} = \begin{bmatrix} \Theta + m \cdot |\hat{\mathbf{r}}_{i+1}|^2 & m \cdot \hat{r}_{i+1,z} & -m \cdot \hat{r}_{i+1,x} \\ m \cdot \hat{r}_{i+1,z} & m & 0 \\ -m \cdot \hat{r}_{i+1,x} & 0 & m \end{bmatrix}$$

mit

$$\hat{\mathbf{r}}_{i+1} = \mathbf{r}_{i+1} + \mathbf{d}_i.$$

Durch die Transformation, analog zum vorigen Abschnitt, wird lediglich die Entfernung zum Schwerpunkt von \mathbf{r}_{i+1} auf $\hat{\mathbf{r}}_{i+1} = \mathbf{r}_{i+1} + \mathbf{d}_i$ geändert. $\hat{\mathbf{r}}_{i+1}$ gibt die Entfernung vom Körperbezugspunkt i zum Schwerpunkt des Körpers $i+1$ an. Durch Addition der Massenmatrix des i -ten Körpers ergibt sich

$$\begin{aligned} \bar{\mathbf{M}}_i &= \mathbf{M}_i + \mathbf{G}_{i+1,i}^T \bar{\mathbf{M}}_{i+1} \mathbf{G}_{i+1,i} \\ &= \begin{bmatrix} \bar{\Theta}_i + \bar{m}_i \cdot |\bar{\mathbf{r}}_i|^2 & \bar{m}_i \cdot \bar{r}_{i,z} & -\bar{m}_i \cdot \bar{r}_{i,x} \\ \bar{m}_i \cdot \bar{r}_{i,z} & \bar{m}_i & 0 \\ -\bar{m}_i \cdot \bar{r}_{i,x} & 0 & \bar{m}_i \end{bmatrix} \end{aligned}$$

mit

$$\begin{aligned} \bar{m}_i &= m + \bar{m}_{i+1} \\ \bar{\mathbf{r}}_i &= \frac{1}{\bar{m}_i} (m \cdot \mathbf{r}_i + \bar{m}_{i+1} \cdot \hat{\mathbf{r}}_{i+1}) \\ \bar{\Theta}_i &= \Theta + \bar{\Theta}_{i+1} + \frac{m \cdot \bar{m}_{i+1}}{\bar{m}_i} |\mathbf{r}_i - \hat{\mathbf{r}}_{i+1}|^2 \end{aligned}$$

und es folgen die bekannten Formeln zur Berechnung der Gesamtmasse \bar{m}_i , Schwerpunktlage $\bar{\mathbf{r}}_i$ und die auf den neuen Schwerpunkt bezogene Trägheit $\bar{\Theta}_i$ des Verbundkörpers, wie in Abb. 3.8 angedeutet.

Auf diese Weise besitzt die Massenmatrix $\bar{\mathbf{M}}_i$ des Verbundkörpers dieselbe Struktur, wie die Massenmatrix des Vorgängerkörpers \mathbf{M}_{i+1} . Der neue Verbundkörper kann in analoger Weise mit seinem Vorgänger verschmolzen werden.

Es lässt sich zeigen, dass sich die Bewegungsgleichung des i -ten Körpers als Überlagerung der Gesamtbeschleunigung \mathbf{x}_i des Verbundkörpers und der Relativbeschleunigungen \dot{s}_k der nachfolgenden Verbundkörper darstellen lässt.

$$\bar{\mathbf{M}}_i \cdot \mathbf{x}_i + \sum_{k=i+1}^n \mathbf{G}_{k,i}^T \bar{\mathbf{M}}_k \Phi_k \dot{s}_k = \mathbf{l}_i + \bar{\mathbf{l}}_{i,ext}$$

Mit dieser lässt sich die Gelenklast \mathbf{l}_i direkt berechnen. Die Beschleunigung \mathbf{x}_i hängt jedoch von den Gelenkbeschleunigungen

$$\mathbf{x}_i = \sum_{k=1}^i \mathbf{G}_{i,k} \Phi_k \dot{s}_k + \mathbf{c}$$

aller Vorgängergelenke ab. Somit gehen alle Gelenkbeschleunigungen \dot{s} in die Berechnung der

Gelenklast \mathbf{l}_i ein. Dies wiederum hat zur Folge, dass die reduzierte Massenmatrix $\hat{\mathbf{M}}$, deren i -te Zeile das Prinzip der virtuellen Leistung

$$\Phi^T \mathbf{l}_i = 0$$

repräsentiert, voll besetzt sein muss. Folglich benötigt die direkte Lösung dieses voll besetzten linearen Gleichungssystems mit n Unbekannten $\mathcal{O}(n^3)$ Operationen.

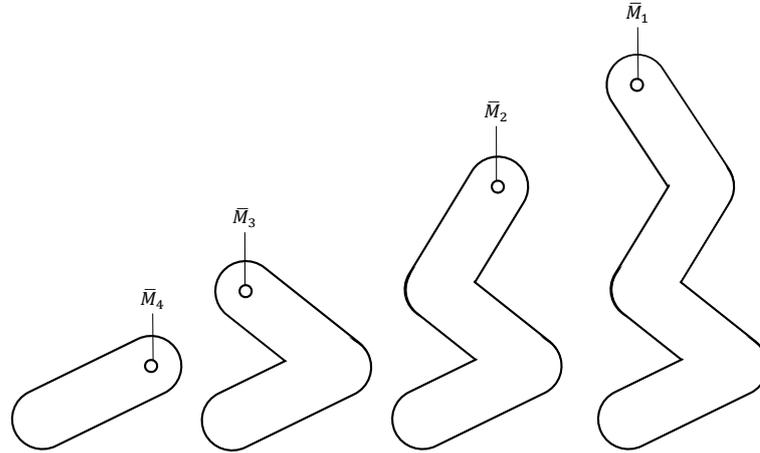


Abbildung 3.8.: Verbundkörper des Pendels

3.3.4. $\mathcal{O}(n)$ -ALGORITHMEN

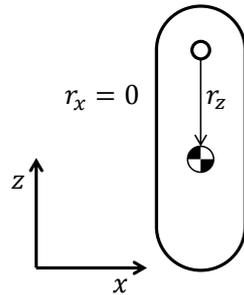
Für die Simulation eines Pendels mit einer hohen Anzahl an Körpern sind die $\mathcal{O}(n)$ -Algorithmen rechentechnisch am effizientesten. In [SH95] ist beschrieben, dass sich der Einsatz von $\mathcal{O}(n)$ -Algorithmen ab etwa sieben Pendelkörpern lohnt. Ähnlich wie bei dem $\mathcal{O}(n^3)$ -Algorithmus für explizite Beschreibungen von Mehrkörpersystemen werden dabei Verbundkörper gebildet. Jedoch wird bei der Berechnung der neuen Trägheitseigenschaften die Art des Gelenks mit berücksichtigt. Dazu muss zuerst der Wert $\Phi_i^T \mathbf{M}_i \Phi_i$ invertiert werden. Im Falle des ebenen Pendels extrahiert diese Operation lediglich das skalare Trägheitsmoment bezüglich der y -Achse aus der Massenmatrix.

Nun kann die korrigierte Massenmatrix \mathbf{M}_i^a des letzten Pendelkörpers, die an den Nachfolgekörper übergeben wird, unter der Verwendung von explizit formulierten Gelenken berechnet werden.

$$\begin{aligned} \mathbf{M}_i^a &= \mathbf{M}_i - \frac{1}{\Phi_i^T \mathbf{M}_i \Phi_i} \left(\Phi_i^T \mathbf{M}_i \right)^T \left(\Phi_i^T \mathbf{M}_i \right) \\ &= \frac{m}{\Theta + m \cdot |\mathbf{r}_i|^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Theta + m \cdot r_{i,x}^2 & m \cdot r_{i,x} r_{i,z} \\ 0 & m \cdot r_{i,x} r_{i,z} & \Theta + m \cdot r_{i,z}^2 \end{bmatrix} \end{aligned} \quad (3.27)$$

Es lassen sich zwei interessante Beobachtungen an diesem Ausdruck feststellen. Zum einen sind alle Komponenten, die mit der Rotation zusammenhängen, verschwunden. Dies ergibt sich unmittelbar aus der Eigenschaft des Drehgelenks, entsprechend dem Prinzip der virtuellen Leistung, kein Drehmoment in die nicht gesperrte Richtung zu übertragen. Zum anderen sind alle translatorischen Komponenten ungleich null. Damit entspricht die korrigierte Massenmatrix nicht mehr der Form eines üblichen Starrkörpers, für den sich $m\mathbf{E}_2$ für den translatorischen Teil der Massenmatrix ergeben müsste, wie es bei \mathbf{M}_i der Fall ist. Dieser Umstand kann an zwei Bei-

spielen verdeutlicht werden. Falls die Verbindung zwischen Gelenk- und Schwerpunkt parallel zur globalen z -Achse liegt, siehe Abb. 3.9, gilt $r_x = 0$. Wird der Körper nun in Richtung der z -Achse beschleunigt, so findet eine rein translatorische Beschleunigung statt und der zugehörige Eintrag in der Massenmatrix lautet deshalb m . Wird der Körper jedoch in x -Richtung beschleunigt, so findet eine rotatorische Ausweichbewegung des Körpers statt und es muss nicht jeder Massepunkt des Körpers gleich stark beschleunigt werden. Dies wiederum führt zu einer geringeren wahrgenommenen Trägheit bezüglich dieser Richtung. Im Falle einer verschwindenden Trägheit bezüglich des Schwerpunkts, etwa im Falle eines Massepunktes, wäre die wahrgenommene Trägheit in diesem Punkt sogar null. Sobald die Verbindungslinie zwischen Gelenk- und Schwerpunkt nicht parallel zu einer Achse steht, sind die Nebendiagonalen des translatorischen Teils der korrigierten Massenmatrix besetzt. Dies bedeutet, dass eine Beschleunigung entlang einer Achse eine Kraftwirkung senkrecht dazu hervorruft und dadurch die rotatorische Ausweichbewegung des Pendelkörpers verursacht wird.



$$M_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & m \frac{\Theta}{\Theta + m \cdot r_z^2} & 0 \\ 0 & 0 & m \end{bmatrix}$$

Abbildung 3.9.: Beispiel

Bei der anschließenden Transformation $\mathbf{G}_{i,i-1}^T \mathbf{M}_i^a \mathbf{G}_{i,i-1}$ wird der Bezugspunkt verschoben und es folgt eine symmetrische, aber voll besetzte Massenmatrix \mathbf{M}_{i-1}^A des Verbundkörpers unter Berücksichtigung des verbindenden Gelenks. Dieser Umstand führt zu komplexeren Ausdrücken der nachfolgenden \mathbf{M}_i^a , weshalb auf deren Angabe an dieser Stelle verzichtet werden soll. Deren Interpretation der richtungsabhängigen Trägheitseigenschaften sowie die Besetzung der Nebendiagonalen bleiben jedoch gültig.

Der $\mathcal{O}(n)$ -Algorithmus lässt sich ebenso auf implizit formulierte Gelenke anwenden, wobei diese Formulierung für das untersuchte Beispiel aufwendiger ist. Dieser Umstand soll anhand des letzten Körpers demonstriert werden. Dessen invertierte Massenmatrix lautet

$$\mathbf{M}_i^{-1} = \frac{1}{\Theta} \begin{bmatrix} 1 & -r_{i,z} & r_{i,x} \\ -r_{i,z} & r_T^2 + r_{i,z}^2 & -r_{i,x}r_{i,z} \\ r_{i,x} & -r_{i,x}r_{i,z} & r_T^2 + r_{i,x}^2 \end{bmatrix}$$

mit dem Trägheitsradius r_T

$$r_T^2 = \frac{\Theta}{m}.$$

Bei der Projektion auf die gesperrten Richtungen verbleiben lediglich die translatorischen Anteile

$$\Psi_i^T \mathbf{M}_i^{-1} \Psi_i = \frac{1}{m \cdot r_T^2} \begin{bmatrix} r_T^2 + r_{i,z}^2 & -r_{i,x}r_{i,z} \\ -r_{i,x}r_{i,z} & r_T^2 + r_{i,x}^2 \end{bmatrix}.$$

Wird diese Matrix invertiert, ergibt sich

$$\left(\Psi_i^T M_i^{-1} \Psi_i\right)^{-1} = \frac{m}{r_T^2 + |\mathbf{r}_i|^2} \begin{bmatrix} r_T^2 + r_{i,x}^2 & r_{i,x}r_{i,z} \\ r_{i,x}r_{i,z} & r_T^2 + r_{i,z}^2 \end{bmatrix}.$$

Die Rückprojektion auf alle Richtungen führt dazu, dass die rotatorischen Komponenten der korrigierten Massenmatrix mit Nullen aufgefüllt werden und es folgt das gleiche Ergebnis, wie in Gl. (3.27).

3.3.5. VERGLEICH DER RECHENZEITEN VON LÖSUNGsalGORITHMEN

Die vorgestellten Algorithmen wurden für das ebene Pendel in *Modelica* implementiert und mit Hilfe des Modelica Werkzeuges *Dymola* für verschiedene Anzahlen an Körpern simuliert. Detailliertere Erläuterungen zu der Implementierung folgen in Kapitel 4. Abb. 3.10 stellt die Rechenzeiten der $\mathcal{O}(n)$ -Algorithmen und des expliziten $\mathcal{O}(n^3)$ -Algorithmus für das ebene Pendel über der Anzahl n der Körper und Gelenke dar. Jeder der Algorithmen berechnet die exakten Gelenkbeschleunigungen. Unterschiede in den Ergebnissen sind höchstens aufgrund von Rundungsfehlern durch die begrenzte Rechengenauigkeit zu erwarten. Somit sind die Rechenzeitergebnisse direkt miteinander vergleichbar. Es ist zu sehen, dass der $\mathcal{O}(n)$ -Algorithmus, basierend auf der expliziten Formulierung etwa nur ein Fünftel der Ausführungszeit der impliziten Version benötigt. Dies liegt hauptsächlich daran, dass die verwendeten Drehgelenke nur eine freie Richtung besitzen. Folglich muss im Algorithmus basierend auf der expliziten Gelenkbeschreibung lediglich durch einen Skalar geteilt werden. Im Algorithmus mit impliziter Beschreibung müssen hingegen je eine 6×6 und eine 5×5 Matrix pro Gelenk für die gesperrten Richtungen invertiert werden. Der explizite $\mathcal{O}(n^3)$ -Algorithmus ist für eine geringe Anzahl an Körpern fast so schnell, wie der explizite $\mathcal{O}(n)$ -Algorithmus. Für größere n steigt die Rechenzeit wie erwartet stark an und ab etwa 80 Körpern ist er auch langsamer als der implizite $\mathcal{O}(n)$ -Algorithmus und ab dieser Stelle nicht mehr lohnenswert.

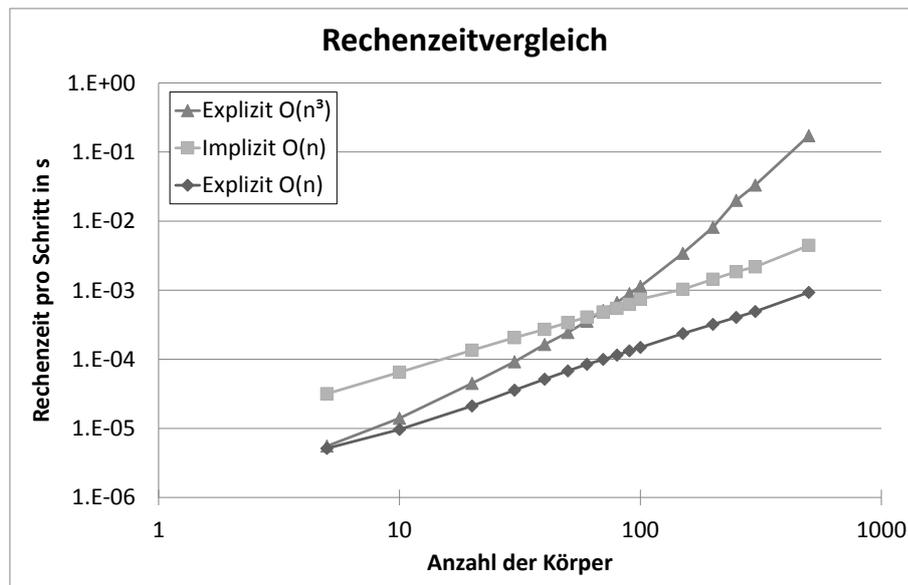


Abbildung 3.10.: Rechenzeitvergleich der Algorithmen

3.4. BERÜCKSICHTIGUNG KINEMATISCHER SCHLEIFEN

In den vorangegangenen Abschnitten wurden Algorithmen vorgestellt, mit denen die Beschleunigungen baumstrukturierter Mehrkörpersysteme berechnet werden können. An dieser Stelle folgt die Einbeziehung sekundärer Gelenke.

3.4.1. ANSÄTZE ZUR BERÜCKSICHTIGUNG KINEMATISCHER SCHLEIFEN

Um die sekundären Gelenke der kinematischen Schleifen zu berücksichtigen, existieren zwei generelle Ansätze. Zum einen können die um sekundäre Gelenke erweiterten Gleichungssysteme direkt mit Lösern für schwach besetzte Gleichungssysteme gelöst werden. In diesem Fall können die spezialisierten Algorithmen nicht mehr angewandt werden und es ist mit einer schlechteren Laufzeit zu rechnen. Daher soll im Folgenden ein weiterer Ansatz vorgestellt werden.

Alle Gleichungssysteme (3.8), (3.13), (3.21), (3.15) der hier diskutierten Algorithmen lassen sich, vorerst ohne Berücksichtigung der Permutation und mit entsprechenden Matrizen \mathbf{A} , \mathbf{B} , \mathbf{c} und \mathbf{d} , darstellen als

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{l} \\ \mathbf{d} \end{bmatrix}.$$

Nun können die Bewegungsgleichungen um die Schnittlasten der sekundären Gelenke erweitert werden. Wie sich im Verlauf zeigen wird, ist es vorteilhaft, eine implizite Formulierung der sekundären Gelenke zu verwenden, um möglichst wenig neue Variablen einzuführen. Daraus folgt

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{l} \\ \mathbf{d} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_s \\ \mathbf{0} \end{bmatrix} \bar{\lambda}_s$$

mit

$$\mathbf{W}_s = \mathbf{G}_s^T \bar{\Psi}_s,$$

wobei $\bar{\lambda}_s$ die Koordinaten der sekundären Zwangslasten beschreibt. Die Beschleunigung \mathbf{x} ergibt sich zu

$$\begin{aligned} \mathbf{x} &= \underbrace{\begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{l} \\ \mathbf{d} \end{bmatrix}}_{\mathbf{x}_p} + \underbrace{\begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{W}_s \\ \mathbf{0} \end{bmatrix}}_{\mathbf{X}} \bar{\lambda}_s \\ &= \mathbf{x}_p + \mathbf{X} \bar{\lambda}_s. \end{aligned} \tag{3.28}$$

Der erste Term steht für die Beschleunigung \mathbf{x}_p , die sich ohne die Berücksichtigung der sekundären Gelenke einstellen würde. Der zweite Term gibt an, wie sich die Koordinaten der sekundären Zwangslasten $\bar{\lambda}_s$ auf die Gesamtbeschleunigung auswirken und ist linear in selbigen. Um die Matrixinversion zur Berechnung der Matrix \mathbf{X} zu vermeiden, kann diese als Unbekannte des folgenden Gleichungssystems aufgefasst werden

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x} & | & \mathbf{X} \\ \mathbf{c} & | & \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{l} & | & \mathbf{W}_s \\ \mathbf{d} & | & \mathbf{0} \end{bmatrix} \tag{3.29}$$

Die Berechnung von \mathbf{X} lässt sich somit durch die mehrmalige Anwendung des jeweiligen Algorithmus realisieren, der zur Berechnung der Beschleunigungen im baumstrukturierten System verwendet wurde. Anstelle der externen Lasten \mathbf{l} müssen lediglich die Spalten der Matrix \mathbf{W}_s berücksichtigt werden. Der sich dabei ergebende Beschleunigungsvektor bildet die zugehörige Spalte der Matrix \mathbf{X} .

Nun kann der Ausdruck (3.28) zur Berechnung der Beschleunigung in Abhängigkeit der gesuchten $\bar{\lambda}_s$ in die impliziten Bindungsgleichungen der sekundären Gelenke eingesetzt werden. Es ergibt sich

$$\begin{aligned} \mathbf{W}_s^T \mathbf{x} &= \gamma \\ \mathbf{W}_s^T (\mathbf{x}_P + \mathbf{X} \bar{\lambda}_s) &= \gamma \\ (\mathbf{W}_s^T \mathbf{X}) \bar{\lambda}_s &= \gamma - \mathbf{W}_s^T \mathbf{x}_0. \end{aligned}$$

Mit dieser Beziehung können die Koordinaten $\bar{\lambda}_s$ der Zwangslasten der sekundären Gelenke berechnet und in Gl. (3.28) eingesetzt werden. Damit ergeben sich schlussendlich die gesuchten Beschleunigungen \mathbf{x} aller Körper.

Der damit verbundene Mehraufwand hängt stark von der jeweiligen Umsetzung der Algorithmen ab. Im Idealfall wird das Gleichungssystem aus (3.29) einmal faktorisiert und die Rückwärtselimination einmal für jede der $1 + \dim(\bar{\lambda}_s)$ Spalten ausgeführt. Dazu muss der jeweilige Algorithmus jedoch entsprechend vorbereitet sein. Eine solche Implementierung des $\mathcal{O}(n)$ Algorithmus für explizite Bindungsbeschreibungen wird in [Rei97] vorgestellt. Im ungünstigsten Fall muss der gesamte Algorithmus $1 + \dim(\bar{\lambda}_s)$ mal ausgeführt werden.

Zusätzlich muss die im Allgemeinen dicht besetzte Matrix $\mathbf{W}_s^T \mathbf{X}$ der Dimension $\dim(\bar{\lambda}_s)$ invertiert werden. Dies bedeutet einen zusätzlichen Aufwand von $\mathcal{O}(\dim(\bar{\lambda}_s)^3)$, der besonders bei vielen kinematischen Schleifen ins Gewicht fällt. Es sei darauf hingewiesen, dass es sich bei $\mathbf{W}_s^T \mathbf{X}$ um eine symmetrische Matrix handelt, wie an der Definition von \mathbf{X} aus Gl. (3.28) zu erkennen ist.

3.4.2. VERGLEICH DER LÖSUNGALGORITHMEN

Um den realen Mehraufwand der Berücksichtigung von kinematischen Schleifen abschätzen zu können, wurde das Beispiel der ebenen Pendelkette um kinematische Schleifen erweitert. Es wurde eine feste Anzahl N an Körpern verwendet. Um m kinematische Schleifen einzufügen, wurde das Pendel aller N/m Körper über einen drehbar gelagerten Stab am Inertialsystem befestigt. Es ergibt sich ein Verhalten, siehe Abb. 3.11, wie es bei einem schlaff durchhängenden Seil zu erwarten wäre. Als Körperanzahl wurde $N = 120$ gewählt. Dies garantiert zum einen eine ausreichend hohe Anzahl an Körpern und zum anderen besitzt diese Zahl eine große Zahl an Teilern m , so dass N/m eine Ganzzahl wird. Da es sich um ein ebenes Problem handelt, entspricht die Dimension der Koordinaten der Zwangslasten $\dim(\bar{\lambda}_s)$ gerade $2m$ und ist damit nicht mehr vernachlässigbar klein gegenüber N .

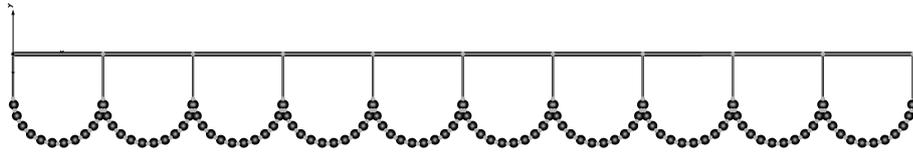


Abbildung 3.11.: Modell eines Pendels mit $m = 10$ kinematischen Schleifen

Abb. 3.12 zeigt die Rechenzeiten für die Einbindung kinematischer Schleifen, sowohl absolut als auch normiert auf die Rechenzeiten ohne Schleifen, des $\mathcal{O}(n^3)$ und des $\mathcal{O}(n)$ -Algorithmus für die explizite Bindungsformulierung für bis zu $m = 60$ Schleifen. Der $\mathcal{O}(n^3)$ -Algorithmus nimmt die $\bar{\lambda}_s$ als zusätzliche Unbekannte in das Gleichungssystem auf. Damit muss in jedem Schritt

3. Lösungsalgorithmen für Mehrkörpersysteme

ein System der Größe $N + 2m$ gelöst werden. Da der Aufwand mit der dritten Potenz wächst, ergibt sich der relative Mehraufwand η_3 in etwa zu

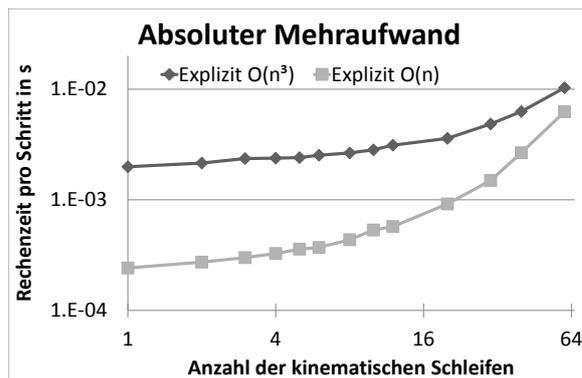
$$\begin{aligned}\eta_3 &= \frac{(N + 2m)^3}{N^3} \\ &= 1 + 6\frac{m}{N} + 12\left(\frac{m}{N}\right)^2 + 8\left(\frac{m}{N}\right)^3\end{aligned}$$

Im Falle des $\mathcal{O}(n)$ -Algorithmus sind Aufgaben verschiedener Komplexitätsklassen zu lösen [AC03], deren relative zeitliche Wichtung nicht bekannt ist. Daher ergibt sich für den relativen Mehraufwand η_1 des $\mathcal{O}(n)$

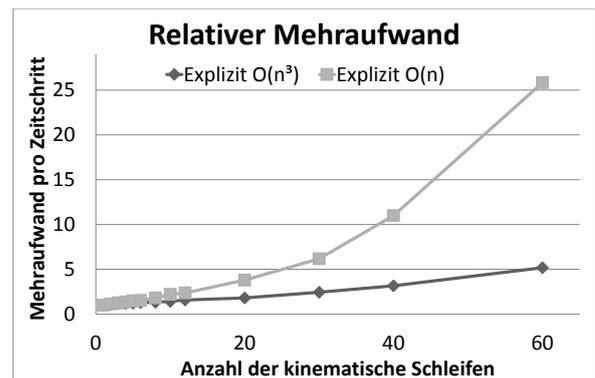
$$\begin{aligned}\eta_1 &= \frac{N + C_1mN + C_2m^3}{N} \\ &= 1 + C_1\frac{m}{N} + C_2\frac{m^3}{N}\end{aligned}$$

wobei die Parameter C eine unbekannte relative Wichtung zwischen den drei Bestandteilen des Aufwandes darstellen.

Für beide Algorithmen ist somit ein linearer Anstieg für $m \ll n$ zu erwarten, der für große m in ein Polynom dritter Ordnung bezüglich m übergeht. Es ist jedoch ersichtlich, dass der Term dritter Ordnung in η_1 deutlich dominanter ist, da er nur auf N anstelle von N^3 normiert wird. In Abb. 3.12b ist eben dieses Verhalten am Beispiel eines m -fach aufgehängten Pendels zu erkennen. Während der Aufwand des $\mathcal{O}(n)$ -Algorithmus nur etwa bis $m = 12$ linear und danach wie ein Polynom dritten Grades ansteigt, so wird der polynomielle Anteil bei dem $\mathcal{O}(n^3)$ -Algorithmus erst ab etwa $m = 60$ sichtbar. Es ist außerdem zu sehen, dass der Anstieg des linearen Abschnittes des $\mathcal{O}(n)$ - deutlich größer als der des $\mathcal{O}(n^3)$ -Algorithmus ist. Die Berücksichtigung der Schleifen in diesem $\mathcal{O}(n)$ -Algorithmus scheint jedoch effizient umgesetzt, so dass nur die Rückwärtselimination mehrmals ausgeführt werden muss. Andernfalls wäre ein deutlich stärkerer Anstieg zu verzeichnen gewesen.



(a) Absoluter Mehraufwand



(b) Relativer Mehraufwand

Abbildung 3.12.: Mehraufwand bei Berechnung mit Schleifen

3.5. ZUSAMMENFASSUNG DER LÖSUNGsalGORITHMEN EINES MEHRKÖRPERSYSTEMS

Ausgehend von dem großen schwach besetzten linearen Gleichungssystem (2.55) aus Kapitel 2 wurde dessen Lösung mit speziellen Algorithmen diskutiert. Zum besseren Verständnis wurde, wie bei schwach besetzten Gleichungssystemen üblich, in Abschnitt 3.1 zuerst die Struktur des Gleichungssystems analysiert und herausgefunden, dass der Systemgraph der Formulierung mit expliziten Bindungsgleichungen eine spezielle wiederkehrende Leiterstruktur besitzt, deren zwei Holme jeweils den Topologiegraphen enthalten. Diese Erkenntnis wird in Kapitel 4.2 eine zentrale Rolle spielen. Der Systemgraph der Formulierung mit impliziten Bindungsgleichungen besitzt abgesehen von den fehlenden Griffen die gleiche Leiterstruktur.

In Abschnitt 3.2 wurden sowohl bekannte als auch bisher unbekannte Lösungsalgorithmen für das Gleichungssystem hergeleitet. Diese Algorithmen wurden bezüglich ihrer zeitlichen Komplexität in die Gruppe der $\mathcal{O}(n^3)$ - und der $\mathcal{O}(n)$ -Algorithmen unterteilt. Unter der Gruppe der $\mathcal{O}(n^3)$ -Algorithmen sind alle diejenigen zusammengefasst, bei denen ein dicht besetztes lineares Gleichungssystem gelöst werden muss, dessen Größe sich proportional zur Summe der Anzahl der Gelenkfreiheitsgrade verhält. Sie werden $\mathcal{O}(n^3)$ genannt, da das direkte Lösen eines dicht besetzten Gleichungssystems mit n Unbekannten gerade $\mathcal{O}(n^3)$ Operationen benötigt. Die $\mathcal{O}(n)$ -Algorithmen gehen bei der Lösung rekursiv vor und erfordern für jedes Gelenk die Lösung eines Gleichungssystems, das nie mehr als sechs Unbekannte besitzt. Die benötigte Zeit für dessen Lösung ist damit unabhängig von der Systemgröße, muss jedoch n Mal ausgeführt werden und ergibt somit das zeitliche Verhalten $\mathcal{O}(n)$.

In der Literatur wird stark zwischen der impliziten und der expliziten Formulierung der Bindungsgleichungen unterschieden. Es wurde gezeigt, dass sich die klassischen Algorithmen hauptsächlich darin unterscheiden, ob die Bewegungsgleichung eines Körpers nach der Gelenklast (explizit) oder nach der Beschleunigung (implizit) umgestellt wird. Letzteres erfordert die Invertierung der im Allgemeinen zeitlich veränderlichen 6×6 Massenmatrix und führt zu Geschwindigkeitseinbußen gegenüber der expliziten Formulierung. Durch die Kenntnis der dualen Basen ist es jedoch möglich, die Formulierung mit impliziten Bindungsgleichungen in analoger Weise zur expliziten Form mit etwa demselben Aufwand zu lösen. Es wurde erkannt, dass für den $\mathcal{O}(n)$ -Algorithmus sogar eine gemischte Beschreibung der Bindungsgleichungen gefunden werden kann. Dies erlaubt die Einbeziehungen von komplizierten Gelenken, für die nur eine implizite Formulierung vorliegt, ohne eine veränderte Beschreibung der anderen Gelenke und damit ohne die Rechenzeit zu beeinträchtigen.

Am Beispiel einer ebenen Pendelkette in Abschnitt 3.3 wurden Eigenschaften und Zwischenergebnisse der vorgestellten Algorithmen diskutiert. Abgeschlossen wurde der Abschnitt mit einem Vergleich der Rechenzeiten pro Zeitschritt. Daraus ließen sich zwei zentrale Erkenntnisse ablesen:

- Algorithmen basierend auf der expliziten Formulierung der Bindungsgleichungen sind schneller als die der impliziten Formulierung.
- Für große Systeme mit vielen Körpern bzw. Gelenken sind die $\mathcal{O}(n)$ -Algorithmen deutlich schneller als die $\mathcal{O}(n^3)$ -Algorithmen.

Bezogen auf das Beispiel des ebenen Pendels aus Abschnitt 3.3 lohnt sich der Einsatz der impliziten Formulierung des $\mathcal{O}(n)$ -Algorithmus ab etwa 80 Körpern gegenüber dem expliziten $\mathcal{O}(n^3)$ -Algorithmus. Der Einsatz des expliziten $\mathcal{O}(n)$ -Algorithmus lohnt sich hingegen immer.

Die Einbeziehung der kinematischen Schleifen im $\mathcal{O}(n)$ -Algorithmus gestaltet sich jedoch deutlich schwieriger als in $\mathcal{O}(n^3)$ -Algorithmen, wie in Abschnitt 3.4 gezeigt wurde. Während im Falle des $\mathcal{O}(n^3)$ -Algorithmus lediglich die impliziten Bindungsgleichungen der sekundären Gelenke dem Gleichungssystem hinzugefügt werden, muss die Rückwärtselimination des $\mathcal{O}(n)$ -Algorithmus mehrmals und zusätzlich ein weiteres Gleichungssystem gelöst werden. Letzteres hat die Dimension der Summe aller Koordinaten der Zwangslasten der sekundären Gelenke,

$\dim(\bar{\lambda}_S)$. Anhand des erweiterten Pendelbeispiels konnte gezeigt werden, dass die Rechenzeit des expliziten $\mathcal{O}(n)$ -Algorithmus deutlich stärker unter der Einbeziehung von kinematischen Schleifen leidet als im Falle des expliziten $\mathcal{O}(n^3)$ -Algorithmus.

Gerade im Hinblick auf die steigende Leistungsfähigkeit moderner Rechnerarchitekturen, die es den Modellierern ermöglicht, immer größere Modelle zu erstellen, sollte zur Simulation von Mehrkörpersystemen im Allgemeinen der explizite $\mathcal{O}(n)$ -Algorithmus verwendet werden. Es muss jedoch festgestellt werden, dass der $\mathcal{O}(n)$ -Algorithmus vergleichsweise schlecht mit kinematischen Schleifen umgeht. Dies ist problematisch, da in Modellen von Baumaschinen meist mehrere kinematische Schleifen vorkommen. Als Beispiele seien hier das Viergelenk einer Baggerschau-felaufhängung oder die Viergelenke der Kinematik einer Betonpumpe genannt. Da die Berechnungsvorschriften des $\mathcal{O}(n)$ -Algorithmus durch komplizierte Umformungen aus den Bewegungs- und Bindungsgleichungen hervorgegangen sind, kann es unter Umständen sehr schwer sein, neue Modellelemente, wie z.B. ein Reifen mit massebehaftetem Gürtel oder flexible Körper, die keinen klassischen Starrkörper oder ein ideales Gelenk darstellen, zu berücksichtigen.

4. EFFIZIENTE BERECHNUNG VON MEHRKÖRPERSYSTEMEN

Bezug nehmend auf das Ziel der Arbeit muss basierend auf den bereits hergeleiteten Erkenntnissen im nächsten Schritt analysiert werden, welche der aufgeführten Probleme bei Anwendung der Methode der Mehrkörpersysteme auf die Simulation von Baumaschinen tatsächlich zu langsamen Rechenzeiten führt. Das bedeutet, es ist nun von Interesse, wie die zur Anwendung kommende Software Mehrkörpersysteme beschreibt und löst. Im Folgenden soll speziell auf die Simulation mithilfe des Sprachstandards Modelica eingegangen werden.

4.1. BERECHNUNG VON MEHRKÖRPERSYSTEMEN BASIEREND AUF MODELICA

Die *Modelica Association* entwickelt seit 1996 einen Sprachstandard zur werkzeugunabhängigen Beschreibung von Simulationsmodellen unter dem Namen *Modelica* [Ass12]. Ein Modelica Modell besitzt eine objektorientierte Struktur, deren Klassen Gleichungen und Variablen enthalten. Der objektorientierte Ansatz unterstützt die einfache Erstellung von Modellbibliotheken. Die bekannteste dieser Modellbibliotheken ist die *Modelica Standard Library* (MSL). Diese wird von der Modelica Association frei zur Verfügung gestellt und enthält eine Vielzahl vorgefertigter Bibliothekselemente aus verschiedenen technischen Domänen.

Ein Modelica Modell kann mithilfe eines *Modelica Compilers*, durch Überführung in eine Repräsentation aus Gleichungen und Variablen, in ein ausführbares Programm übersetzt und anschließend simuliert werden. Neben freien Modelica Compilern wie OpenModelica [FAP⁺06] und JModelica.org [ÅÅG⁺10] gehören SimulationX [UBB⁺09] und Dymola [BEMO02] zu den bekanntesten kommerziellen Vertretern. Modelica Compiler unterscheiden sich von anderen herkömmlichen Simulationswerkzeugen, wie z.B. MATLAB, durch ihre Fähigkeit, Gleichungen symbolisch manipulieren zu können. Daher wird im Kontext von Modelica streng zwischen einer Zuweisung und einer Gleichung unterschieden. Im Gegensatz zu einer Zuweisung, kann eine Gleichung nach jeder Variablen gelöst werden. In Kombination mit der Möglichkeit Gleichungen symbolisch differenzieren zu können, sind Modelica Compiler in der Lage selbst differenzial-algebraische Systeme mit höherem Index, robust zu lösen. Genauere Informationen über die Abläufe innerhalb eines Modelica Compilers können [Fri04, Fre13] entnommen werden.

Zur Simulation von Mehrkörpersystemen stellt die Modelica Standard Library die Bibliothek *Modelica.Mechanics.MultiBody* zur Verfügung. Sie umfasst verschieden geformte Starrkörper,

alle einfachen Gelenke sowie übliche Sensoren und Lastelemente. Die in den Bibliothekselementen enthaltenen Gleichungen sind vergleichbar mit denen aus Kapitel 2, wobei fast ausschließlich die explizite Formulierung der Bindungen genutzt wird. Nach der Indexreduktion mit anschließendem Matching [Pan88, Fre13] entsteht innerhalb des Modelica Compilers das große schwach besetzte lineare Gleichungssystem (2.55), mit dem Kapitel 2 endete.

Um den $\mathcal{O}(n)$ -Algorithmus, der sich im vorangegangenen Kapitel als äußerst lohnenswert herausstellte, für Modelica zugänglich zu machen, gibt es zwei Ansätze. Zum einen ist es denkbar, die MKS Bibliothek aus der MSL zu nutzen und das entstehende lineare Gleichungssystem innerhalb des Modelica Compilers analog zu Kapitel 3.2.4 zu behandeln. Alternativ kann eine spezielle Bibliothek erstellt werden, die direkt den $\mathcal{O}(n)$ -Algorithmus implementiert, so dass innerhalb des Modelica Compilers erst gar kein lineares Gleichungssystem entsteht.

Eine Umsetzung des $\mathcal{O}(n)$ -Algorithmus innerhalb eines Modelica Compilers wurde erstmals durch den Autor der vorliegenden Arbeit in [SFKB12] vorgestellt. Das Grundproblem liegt in der Abstrahierung der Mechanik auf einfache Gleichungen und Variablen. Dadurch sind innerhalb des Modelica Compilers die Informationen zu Körpern und Gelenken nicht mehr verfügbar. Dies erschwert die Vorsortierung der Gleichungen und Variablen für die Gauß-Elimination beträchtlich. Ein möglicher Weg zum Auffinden der Sortierung wird in Abschnitt 4.2 vorgestellt, indem aus dem Systemgraphen des Gleichungssystems der Topologiegraph rekonstruiert wird. Der implementierte Algorithmus wird am Beispiel eines ebenen Pendels getestet und die Rechenzeit mit der des Tearing, siehe Abschnitt 4.2.1, und der eines numerischen Lösungsverfahrens für große schwach besetzte Gleichungssysteme verglichen.

Eine Umsetzung des $\mathcal{O}(n)$ -Algorithmus als eigenständige Bibliothek wurde z.B. von der ITI GmbH für das auf Modelica basierende Simulationswerkzeug SimulationX [BL09] umgesetzt. Die Bibliothekselemente enthalten anstelle der physikalischen Zusammenhänge die Berechnungsvorschrift des Algorithmus direkt. Dies erschwert die Erweiterung solcher Bibliothek sogar für mechanisch versierte Anwender, wenn sie nicht mit allen Details der Implementierung des $\mathcal{O}(n)$ -Algorithmus vertraut sind. Abschnitt 4.3.2 stellt einen Lösungsweg vor, der es erlaubt komplexe mechanische Elemente als Subsysteme zu betrachten. Darin wird eine Methode präsentiert, wie die Berechnungsvorschrift gewonnen und als neues Element der Bibliothek hinzugefügt werden kann.

Im Rahmen der Arbeit wurden zwei praktisch relevante Subsysteme erstellt. Zum einen handelt es sich um eine kinematische Schleife, deren Modellierung mit den Standardelementen einer $\mathcal{O}(n)$ -Bibliothek bisher immer zu einer deutlich längeren Rechenzeit führte, wie in Abschnitt 3.4 angedeutet. Anhand zweier typischer Anwendungen aus dem Bereich der Baumaschinen wird in Abschnitt 4.3.3 gezeigt, dass mithilfe des neuen Subsystems die Rechenzeit deutlich verkürzt werden kann. Als zweites Subsystem wurde das Modell eines Gleichlaufgelenks implementiert. Dies war bisher nur als eine Aneinanderreihung zweier Kreuzgelenke möglich, die einer Zwangsbedingung unterliegen. Eine solche Lösung führte zu einem Gleichungssystem auf Compilerbene und damit zu einer Beeinträchtigung der Rechengeschwindigkeit. In Abschnitt 4.3.4 wird gezeigt, wie mithilfe der *dualen Basen* eine rein explizite Formulierung des Gelenks gewonnen und als Subsystem implementiert werden kann. Anhand von Messungen der Rechenzeit wird gezeigt, dass diese explizite Implementierung etwa doppelt so schnell ausgewertet werden kann, wie die bisher bekannten Beschreibungen eines Gleichlaufgelenkes.

4.2. $\mathcal{O}(n)$ -ALGORITHMUS FÜR MODELICA COMPILER

Bevor auf die Implementierung eines $\mathcal{O}(n)$ -Algorithmus für Modelica Compiler genauer eingegangen wird, soll der interne Programmablauf solch eines Compilers, bezogen auf die Bewegungsgleichungen eines Mehrkörpersystems, erläutert werden.

4.2.1. TEARING

Sobald ein Modelica Compiler auf ein schwach besetztes lineares oder nichtlineares Gleichungssystem stößt, wie z.B. Gl. (2.55), wird ein Verfahren names *Tearing* [EO94, Car00] angewandt. Tearing ist ein symbolisches Verfahren, um große schwach besetzte Gleichungssysteme der Form

$$\mathbf{0} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n, f : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (4.1)$$

in ein deutlich kleineres dicht besetztes System zu überführen, welches sich effizienter lösen lässt. Dazu werden die unbekanntes \mathbf{x} in n_t Tearingvariablen \mathbf{x}_t und n_a algebraische Variablen \mathbf{x}_a sowie die Systemgleichungen \mathbf{f} in n_t Residuen \mathbf{f}_t und n_a algebraische Beziehungen \mathbf{f}_a partitioniert. Gleichung (4.1) lässt sich dann als

$$\begin{aligned} \mathbf{0} &= \mathbf{f}_a(\mathbf{x}_t, \mathbf{x}_a) \\ \mathbf{0} &= \mathbf{f}_t(\mathbf{x}_t, \mathbf{x}_a) \end{aligned} \quad (4.2)$$

schreiben, wobei die Partitionierung so erfolgen soll, dass die Matrix der partiellen Ableitungen $\frac{\partial \mathbf{f}_a}{\partial \mathbf{x}_a}$ regulär ist. Ist dies der Fall, kann \mathbf{f}_a nach den algebraischen Variablen \mathbf{x}_a gelöst werden

$$\mathbf{x}_a = \tilde{\mathbf{f}}_a(\mathbf{x}_t) \quad (4.3)$$

und diese Beziehung in (4.2) eingesetzt werden. Es entsteht ein neues Gleichungssystem

$$\mathbf{0} = \mathbf{f}_t(\mathbf{x}_t, \tilde{\mathbf{f}}_a(\mathbf{x}_t)) = \tilde{\mathbf{f}}(\mathbf{x}_t),$$

das nur noch $n_t < n$ Unbekannte besitzt. Auf diese Weise kann die Anzahl der Unbekannten in der Praxis meist deutlich reduziert werden. Im Gegenzug ist das kleinere System meist nicht mehr schwach besetzt. Oft kann Gl. (4.3) durch symbolische Umformung gewonnen werden, ohne zusätzliche Gleichungssysteme numerisch lösen zu müssen.

Ziel aller Tearingverfahren ist es, eine möglichst geringe Anzahl an Tearingvariablen \mathbf{x}_t zu finden und eine Funktion \mathbf{f}_a zu erhalten, die sich möglichst leicht invertieren lässt. Da sich dieses Problem jedoch nicht in polynomieller Zeit lösen lässt [CG97], muss dafür auf spezielle Heuristiken ausgewichen werden. Ein Vergleich existierender Verfahren kann [Wau13] entnommen werden.

Empirische Untersuchungen in [Wau13] haben gezeigt, dass bessere Tearingverfahren im Falle des linearen Gleichungssystems eines Mehrkörpersystems meist die Gelenkbeschleunigungen $\dot{\mathbf{s}}_i$ und im Falle von kinematischen Schleifen die Koordinaten der Zwangslasten $\bar{\boldsymbol{\lambda}}$ als Tearingvariablen wählen. Es wird angenommen, dass das lineare System die Gestalt

$$\begin{bmatrix} -\mathbf{G}^T & \mathbf{M} & \mathbf{0} & \mathbf{G}_s^T \bar{\boldsymbol{\Psi}}_s \\ \mathbf{0} & -\mathbf{G} & \boldsymbol{\Phi} & \mathbf{0} \\ \boldsymbol{\Phi}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{\Psi}}_s^T \mathbf{G}_s & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{l} \\ \mathbf{x} \\ \dot{\mathbf{s}} \\ \bar{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{l}_{ext} \\ -\mathbf{c} \\ \mathbf{0} \\ -\bar{\boldsymbol{\Psi}}_s^T \mathbf{c}_s \end{bmatrix}$$

besitzt, die aus Gleichung (3.3) gewonnen werden kann. Dazu müssen die Gelenke in primäre und sekundäre Gelenke unterteilt werden, wobei alle Größen der sekundären Gelenke durch den nachgestellten Index s gekennzeichnet wurden. Die Matrix \mathbf{G} , welche die Topologie des aufspannenden Baums des Mehrkörpersystems beschreibt, kann leicht invertiert werden, siehe Abschnitt 3.1.2. Somit können die ersten beiden Zeilen genutzt werden, um die verbleibenden algebraischen Variablen $\mathbf{x}_a = [\mathbf{l}, \mathbf{x}]$ zu berechnen. Das verbleibende reduzierte lineare System ergibt sich dann zu

$$\begin{bmatrix} \hat{M} & \Phi^T \mathbf{G}^{-T} \mathbf{G}_s^T \bar{\Psi}_s \\ \bar{\Psi}_s^T \mathbf{G}_s \mathbf{G}^{-1} \Phi & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{s} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} \Phi^T \mathbf{G}^{-T} (\mathbf{l}_{ext} - \mathbf{M} \mathbf{G}^{-1} \mathbf{c}) \\ -\bar{\Psi}_s^T (\mathbf{c}_s + \mathbf{G}_s \mathbf{G}^{-1} \mathbf{c}) \end{bmatrix}.$$

Dies entspricht exakt dem expliziten $\mathcal{O}(n^3)$ -Algorithmus aus Abschnitt 3.2.1, ergänzt um kinematische Schleifen. Insofern ist Tearing nur für kleine Mehrkörpersysteme sinnvoll anwendbar, wie in Abschnitt 3.3.5 dargestellt wurde.

Folglich ist es möglich den expliziten $\mathcal{O}(n^3)$ -Algorithmus mithilfe eines generischen Verfahrens für schwach besetzte lineare Gleichungssysteme zu gewinnen. Dem Autor ist jedoch kein Verfahren aus der Literatur bekannt, das zu dem expliziten $\mathcal{O}(n)$ -Algorithmus führt. Daher ist es derzeit nicht möglich unter Nutzung der MKS Bibliothek der MSL große Mehrkörpersysteme im Hinblick auf die Rechenzeit effizient zu simulieren. Im Folgenden Abschnitt soll ein Weg aufgezeigt werden, mit dem diese Einschränkung aufgehoben werden kann.

4.2.2. ÜBERTRAGUNG DES $\mathcal{O}(n)$ -ALGORITHMUS AUF DEN SYSTEMGRAPHEN

Im Gegensatz zu spezialisierten Werkzeugen zur Simulation von Mehrkörpersystemen hat ein Modelica Compiler keinen Zugriff auf den Topologiegraphen und kennt folglich nicht die Vorgänger- und Nachfolgerbeziehungen innerhalb eines Mehrkörpermodells. Stattdessen stellt sich das Modell als eine Menge anonymer Gleichungen und Variablen dar, die per se z.B. nicht als Gelenkfreiheitsgrade oder Elemente der Massenmatrix zu erkennen sind. Daher ist es nicht sofort möglich, den Algorithmus aus Abschnitt 3.2.4 zu implementieren.

Aus demselben Abschnitt ist jedoch bekannt, dass sich der $\mathcal{O}(n)$ -Algorithmus durch eine Gauß-Elimination bei geeigneter Sortierung aller Variablen und Gleichungen herleiten lässt. Es handelt sich dabei um ein generisches Verfahren, das sich auf einen Modelica Compiler übertragen lässt. Das Problem reduziert sich somit auf das Finden der geeigneten Sortierung. Um diese genauer zu analysieren, soll der Systemgraph aus Kapitel 3.1.5 diskutiert werden. Das Finden der geeigneten Sortierung bedeutet, dass jedem Gleichungs- und Variablenknoten ein eindeutiger Zeilen- bzw. Spaltenindex zugeordnet wird, der den Aufbau des linearen Gleichungssystems bestimmt.

Abbildung 4.1 zeigt den Systemgraphen des Viergelenks. Im Gegensatz zu Abb. 3.4 wird das sekundäre Gelenk implizit dargestellt, d.h. es wurde die Projektion auf die Lastrichtungen $[\Psi_4, \bar{\Psi}_4]$ vorgenommen, um die Gelenkbeschleunigungen \dot{s}_4 und das Prinzip der virtuellen Leistung D_4 zu eliminieren, siehe Abschnitt 3.1.5. Es wurden die partiellen Ableitungen der jeweiligen Gleichung nach den Variablen in Form von Kantengewichten ergänzt. Die Kanten sind außerdem gerichtet und geben den Informationsfluss im $\mathcal{O}(n)$ -Algorithmus an. Eine Kante, die einen Gleichungsknoten verlässt bedeutet, dass diese Gleichung nach der angrenzenden Variablen gelöst wird. Kanten, die in einem Gleichungsknoten enden, zeigen an, dass zur Auswertung der Gleichung die verbundene Variable benötigt wird. Die fett gedruckten Verbindungslinien zeigen die Hauptdiagonale an, also Gleichungen und Variablen, die den gleichen Zeilen- bzw. Spaltenindex besitzen.

Es fällt auf, dass alle Einheitsmatrizen im System auf die Hauptdiagonalen gelegt wurden. Auf diese Weise können die Gleichungen ohne Inversion einer Matrix nach der entsprechenden Va-

riablen gelöst werden. Da jede Gleichung nur nach einer Variablen umgestellt werden kann, darf nur maximal eine Kante einen Gleichungsknoten verlassen. Im Gegensatz dazu darf für jede Variable nur eine bestimmende Gleichung existieren, weshalb jeder Variablenknoten nur maximal eine einlaufende Kante besitzen darf. Unter Beachtung dieser Regeln ist der Informationsfluss im Systemgraphen vollständig definiert. Variablenknoten, in die keine Kante einläuft, und Gleichungsknoten, die von keiner Kante verlassen werden, stellen die Tearingvariablen bzw. Residuen dar.

Der Informationsfluss erfolgt kreisförmig. Beginnend bei der Beschleunigung \mathbf{x}_0 des Inertialsystems können alle Beschleunigungen entsprechend der Topologie des Mehrkörpersystems berechnet werden, da die Gelenkbeschleunigungen, in ihrer Eigenschaft als Tearingvariable, als bekannt angenommen werden können. Mithilfe der Newton-Euler-Gleichungen I_i und der Koordinate $\bar{\lambda}_4$ (Tearingvariable) können alle Gelenklasten ausgehend von den Blättern des aufspannenden Baums hin zur Wurzel berechnet werden.

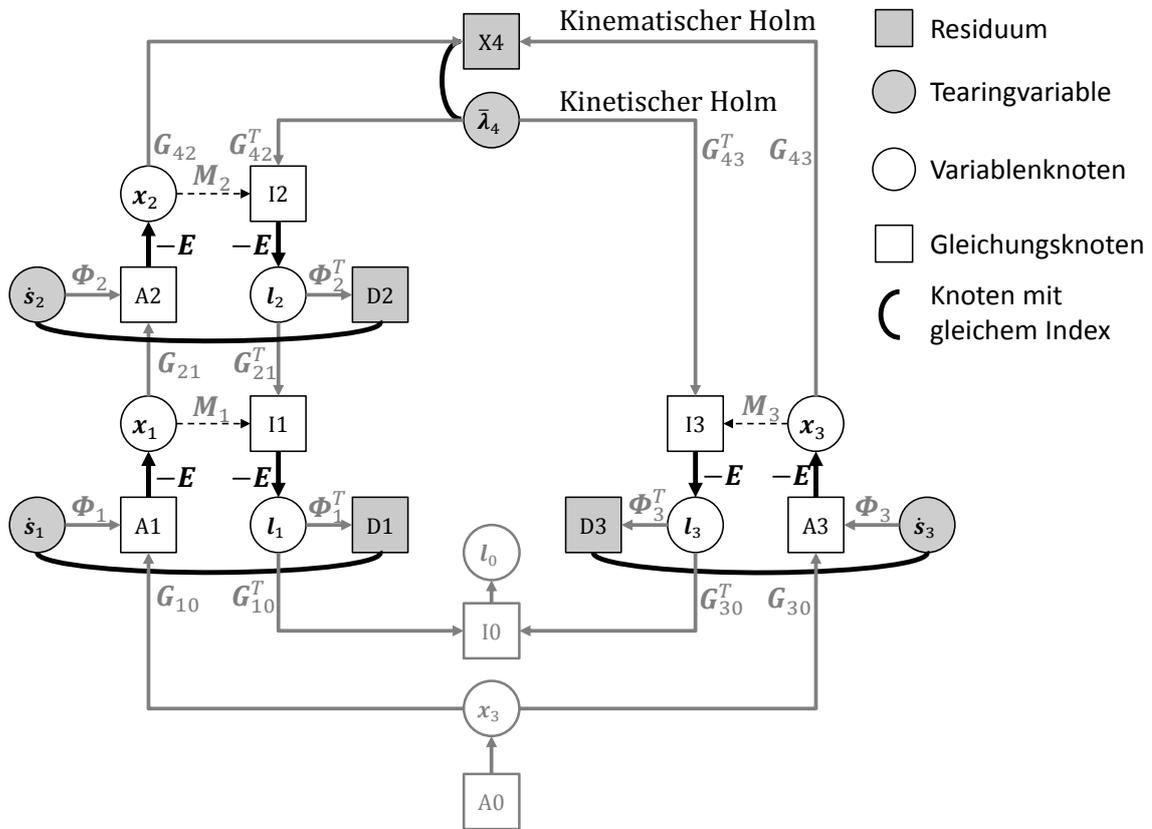


Abbildung 4.1.: Zuordnung von Gleichungen und Variablen im Systemgraphen eines Viergelenks

Da alle anderen Gleichungen und Variablen zugeordnet sind, müssen die Paare aus Tearingvariablen und Residuen eines Gelenks ebenfalls denselben Zeilen- und Spaltenindex besitzen. Zwischen diesen Knoten existiert jedoch keine Kante. Somit besitzt das lineare System auf der Hauptdiagonalen einen Nulleintrag. Folglich muss sichergestellt sein, dass sich während der Gauß-Elimination an dieser Stelle ein regulärer Eintrag ergibt.

Nachdem Paare mit gleichem Zeilen- und Spaltenindex identifiziert wurden, muss deren Platzierung untereinander diskutiert werden. Analog zu Kapitel 3.2.4 wird ausgehend von den Residuen der Blattkörper des aufspannenden Baums begonnen, das Paar $l_i - I_i$ und anschließend das Paar $\mathbf{x}_i - X_i$ zu nummerieren. Danach wird der Index für das Paar aus Tearingvariable und Residuum vergeben. Dies wird ausgehend von den Blattkörpern bis hin zur Wurzel rekursiv ausgeführt.

Die Paare aus Residuen und Tearingvariablen werden, wie in [BJO87] beschrieben, eingefügt, sobald der Körper erreicht wurde, an dem sich alle Stränge der kinematischen Schleife vereinigen.

Das Ergebnis der Nummerierung ist in Abb. 4.2 dargestellt.

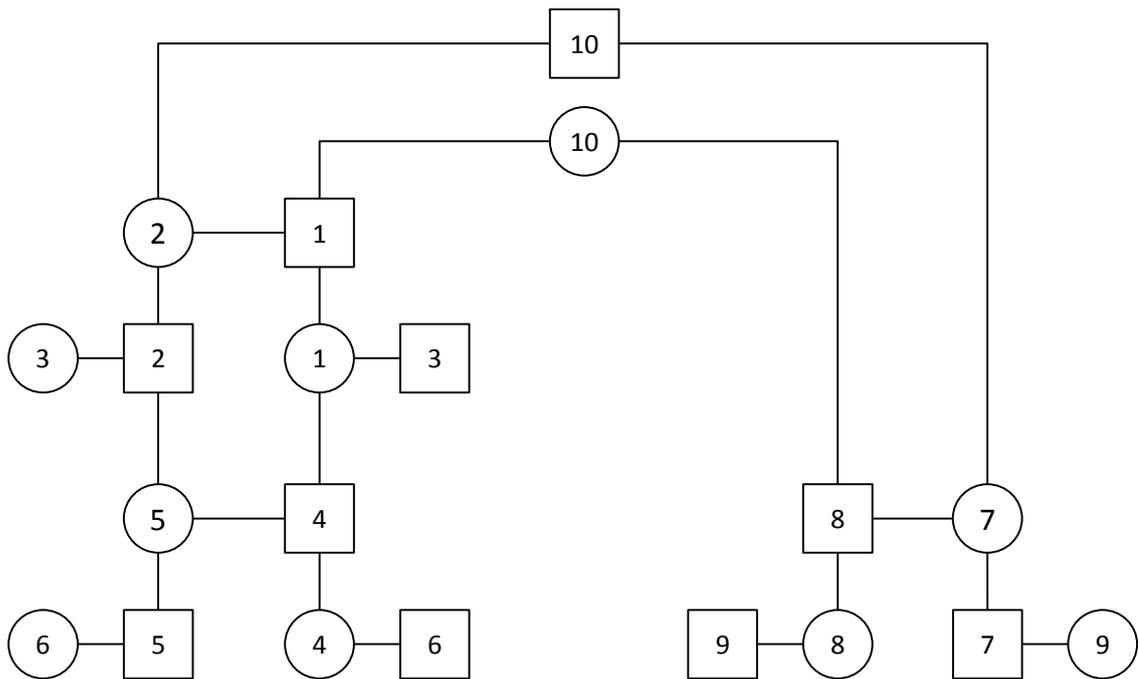


Abbildung 4.2.: nummerierter Systemgraph des Viergelenks

Das Vorgehen zur Übertragung eines $\mathcal{O}(n)$ -Algorithmus kann wie in diesem Abschnitt beschrieben in die folgenden Schritte unterteilt werden:

- Definition des Informationsflusses im betrachteten System
- Aufbau des Topologiegraphen
- Vergeben der Zeilen- und Spaltenindizes für den Aufbau des linearen Gleichungssystems
- Durchführen der symbolischen Gauß-Elimination

4.2.3. ÜBERTRAGUNG DES $\mathcal{O}(n)$ -ALGORITHMUS AUF MODELICA COMPILER

VORBEMERKUNG

Die Implementierung eines $\mathcal{O}(n)$ -Algorithmus in einen Modelica Compiler gestaltet sich schwierig, da der real im Compiler vorliegende Systemgraph nicht exakt die idealisierte Struktur des vorhergehenden Kapitels aufweist. Dies liegt zum einen daran, dass es in Modelica z.B. möglich ist, mehrere Körper oder Gelenke aneinanderzuheften und damit die alternierende Reihenfolge zwischen Körper und Gelenk gestört wird. Zum anderen werden die Gleichungen innerhalb eines Modelica Compilers auf skalarer Ebene behandelt. Dies bedeutet, dass eine sechsdimensionale Vektorgleichung, wie sie in dem idealisierten Systemgraphen benutzt wurden, in sechs skalare Gleichungen zerfällt. Dies ermöglicht es dem Modelica Compiler, zusätzliche Optimierungen auszuführen und beispielsweise einzelne Elemente von Vektoren einzusetzen oder aufgrund der Gleichheit zu einer anderen Variablen aus dem System zu entfernen. Der Systemgraph eines Viergelenkes innerhalb des Modelica Compilers ist in Abb. 4.3 dargestellt.

Für einen Modelica Compiler besteht der Systemgraph aus anonymen Variablen und Gleichungen, so dass zuerst die Struktur des Systems aus dem Gleichungssystem erarbeitet werden muss.

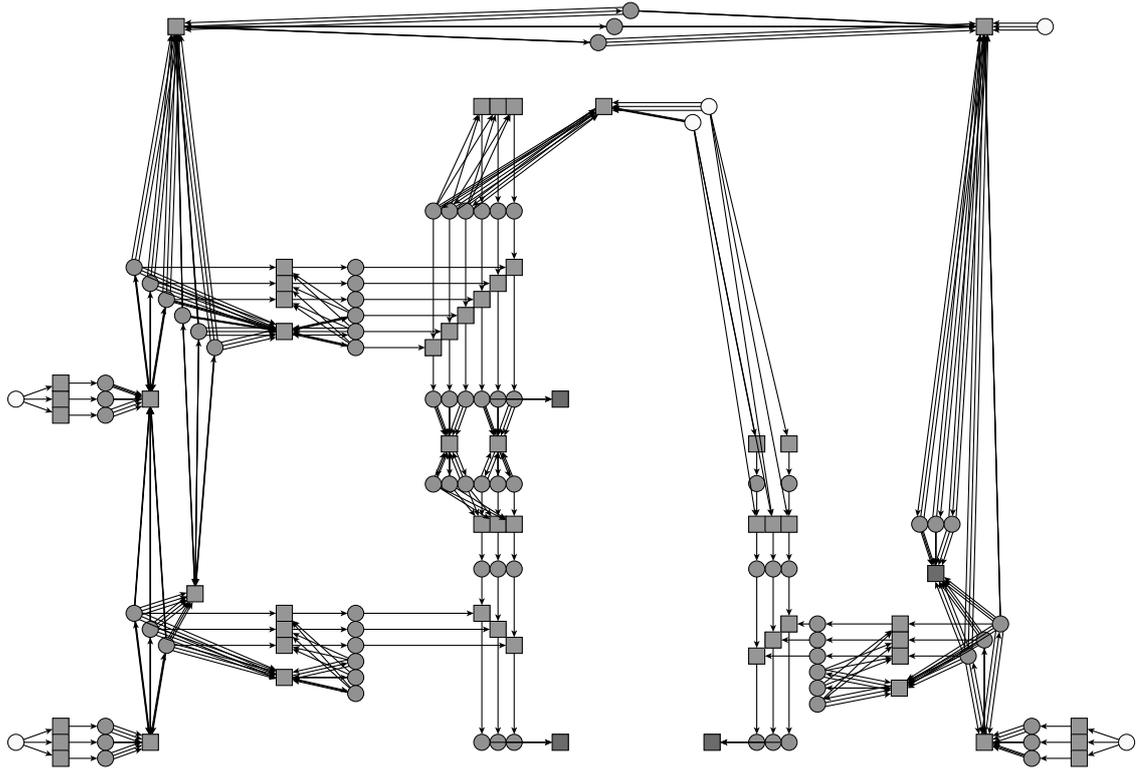


Abbildung 4.3.: Systemgraph des Viergelenks innerhalb des OpenModelica Compilers

HERSTELLEN DES INFORMATIONSFLUSSES

Es ist ausreichend, die Verteilung der Einheitsmatrizen E im Gleichungssystem zu kennen, um den Informationsfluss innerhalb des Systemgraphen festzulegen. Daher werden für jede Gleichung alle Variablen herausgesucht, deren Koeffizient eins oder minus eins ist. Mit diesen Informationen wird ein Matching [Fre13] ausgeführt, das solange iteriert, bis ein vollständiger und widerspruchsfreier Informationsfluss gefunden wurde. Damit ist es möglich, den Kanten innerhalb des Systemgraphen Richtungen zuzuweisen. Alle Indizes der Variablen, in deren Knoten keine Kante endet, werden in der Menge T der Tearingvariablen zusammengefasst. Als Tearingvariablen werden meist die Gelenkbeschleunigungen und die Koordinaten der Zwangskräfte gewählt. Alle Indizes der Gleichungen, in denen keine Kante beginnt, werden in der Menge R der Residuen zusammengefasst. Die Gleichungen für die Gelenklasten nach dem Prinzip der virtuellen Leistungen stellen meist die Residuen dar.

AUFBAU DES TOPOLOGIEGRAPHEN

Als nächstes gilt es, die Tearingvariablen entsprechend der Topologie des Mehrkörpersystems zu sortieren und jeder Tearingvariablen das Residuum desselben Gelenks zuzuweisen. Dazu wird für jeden Knoten k die Menge der Vorgänger-Tearingvariablen T_k bestimmt.

$$T_k = \{j \in T \mid \text{Es existiert ein Pfad von } j \text{ zu } k\}$$

Anschließend wird von jeder Tearingvariablen ausgehend der naechste Knoten l bestimmt, der auf dem kinematischen Holm liegt. Die Menge T_l gibt nun an, welche Tearingvariablen im aufspannenden Baum des Mehrkörpersystems naecher an der Wurzel liegen. Durch einen Vergleich der Mengen für alle Tearingvariablen lässt sich leicht der kinematische Baum ableiten. Eine Aufspaltung des kinematischen Baumes wird daran erkannt, dass zwei Knoten dieselben Vorgänger

besitzen. Die Tearingvariablen der sekundären Gelenke bilden immer Blattknoten und tauchen in verschiedenen Zweigen auf. Auf diese Weise können die kinematischen Schleifen erkannt werden. Wird von jeder Tearingvariable t eines sekundären Gelenks jeder Zweig des kinematischen Baums soweit verfolgt, bis eine gemeinsame Tearingvariable b gefunden wurde, so bildet diese den Basisknoten $b(t)$ der kinematischen Schleife.

Das gleiche Verfahren kann angewandt werden, um den kinematischen Baum der Residuen aufzubauen. Jedoch müssen in diesem Fall alle Kanten rückwärts durchlaufen werden. Während dieses Vorgangs können zu jedem Residuum alle erreichbaren Tearingvariablen festgestellt werden. Dies ermöglicht es, die beiden kinematischen Bäume gegeneinander abzugleichen und die Zuordnung der Tearingvariablen zu den Residuen festzulegen.

VERGEBEN VON ZEILEN- UND SPALTENINDIZES

Nachdem der kinematische Baum bekannt ist, können für alle Gleichungsknoten Zeilen- und für alle Variablenknoten Spaltenindizes zum Aufbau des linearen Gleichungssystems vergeben werden. Dazu wird bei den Residuen, die die Blattknoten im aufspannenden Baum darstellen, begonnen. Von diesen ausgehend werden alle Kanten in entgegengesetzter Richtung durchlaufen. Alle Gleichungs- und Variablenknoten die auf dem Weg zur zugehörigen Tearingvariablen passiert werden, werden getrennt voneinander in aufsteigender Reihenfolge nummeriert. Dies geschieht so, dass die Einheitsmatrizen immer auf der Hauptdiagonalen der Matrix angeordnet werden und keine Einträge in der unteren linken Dreiecksmatrix erfolgen. Anschließend wird der nächste freie Zeilen- und Spaltenindex für den Gleichungsknoten des aktuellen Residuums und der Tearingvariablen vergeben. Auf diese Weise kann nun mit dem nächstfolgenden Paar aus Tearingvariable und Residuum fortgefahren werden. Im Falle einer kinematischen Schleife, der die Tearingvariable t und das Residuum r zugeordnet ist, werden t und r indiziert, bevor der Basisknoten $b(t)$ abgearbeitet wird.

SYMBOLISCHE GAUSS-ELIMINATION

Nachdem jeder Variable ein Spaltenindex und jeder Gleichung ein Zeilenindex zugewiesen wurde, ist der Aufbau des linearen Gleichungssystems vollständig bekannt. Die hergestellte Permutation ist identisch mit der aus den Abschnitten 3.2.4 bzw. 3.2.6. Wie darin gezeigt wurde, kann der $\mathcal{O}(n)$ -Algorithmus nun durch eine Gauß-Elimination des linearen Gleichungssystems hergeleitet werden. Diese muss symbolisch durchgeführt werden, da sich die Koeffizienten des linearen Gleichungssystems über der Zeit ändern. Dabei sollte darauf geachtet werden geeignete Hilfsvariablen, wie sie z.B. die M_i^a und die \hat{M}_i darstellen, einzuführen, um die Komplexität der symbolischen Ausdrücke zu begrenzen. Im Ergebnis der symbolischen Gauß-Elimination erhält man eine Abfolge von Berechnungsvorschriften, mit denen die Lösung, in diesem Falle die Beschleunigungen und Bindungslasten, direkt berechnet werden können.

4.2.4. VERGLEICH DER RECHENZEITEN

Der vorgegebene Algorithmus wurde in den Open Source Modelica Compiler OpenModelica [FAP⁺06] implementiert. Im Folgenden werden die Zeiten für die Auswertung der rechten Seite für ein Pendel mit N Pendelkörpern und Gelenken untersucht. Dazu wurde ein explizites Euler-Verfahren mit fester Schrittweite eingesetzt. Zum Vergleich wurde dasselbe Modell mit Tearing und mit einem numerischen LU Lösungsverfahren für schwach besetzte Matrizen aus der UMFPACK [Dav04] Bibliothek berechnet.

Abb. 4.4 zeigt das Ergebnis zum Vergleich der Rechenzeiten für das Beispiel eines ebenen Pendels. Es ist zu erkennen, dass das in den Compiler integrierte $\mathcal{O}(n)$ -Verfahren (*onRelaxation*) in der

Tat linear mit der Anzahl N der Körper bzw. Gelenke skaliert, während wie erwartet Tearing mit $\mathcal{O}(n^3)$ skaliert. In diesem Beispiel ist es etwa ab $n = 18$ Körpern ratsam das neue $\mathcal{O}(n)$ -Verfahren einzusetzen. Auch das generische Lösungsverfahren UMFPACK skaliert linear mit der Anzahl der Körper. Allerdings benötigt es etwa die dreifache Rechenzeit des direkt in den Modelica Compiler integrierten Verfahrens. Dafür kann das generische Verfahren jedoch immer eingesetzt werden, auch wenn der Systemgraph von der speziellen Struktur abweichen würde, wie sie hier vorausgesetzt wurde. Dies wäre der Fall, wenn z.B. Coulombsche Reibung auftritt und damit Lasten im Systemgraphen berücksichtigt werden müssen, die selbst keine Gelenklasten sind, jedoch von diesen abhängen.

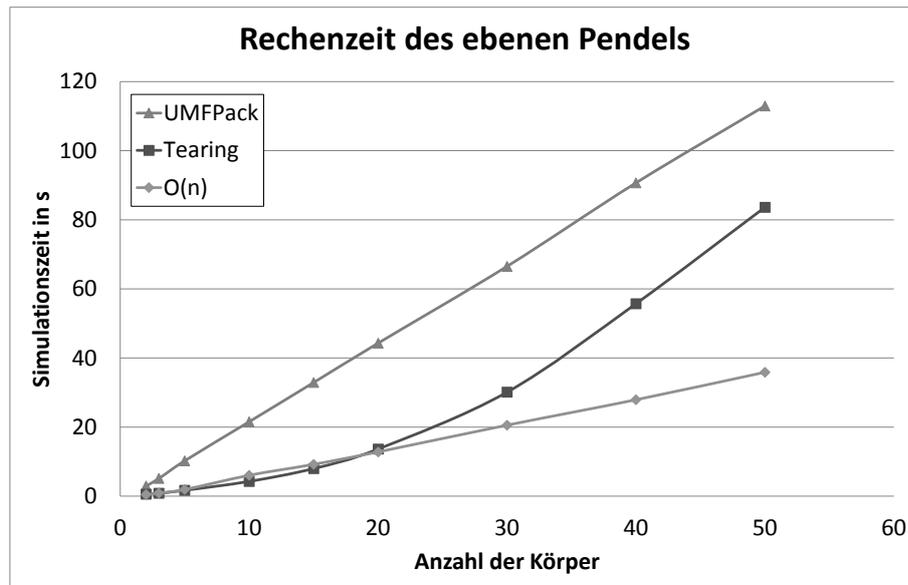


Abbildung 4.4.: Rechenzeitvergleich

4.3. $\mathcal{O}(n)$ -ALGORITHMUS FÜR BIBLIOTHEKSELEMENTE

Ein alternatives Vorgehen zu dem in Kapitel 4.2 vorgestellten Ansatz, den $\mathcal{O}(n)$ -Algorithmus in Modelica verfügbar zu machen, besteht darin, den Algorithmus selbst als eigenständige Modelica Bibliothek zu hinterlegen. Die Grundlage dafür wurde in [Rei97] geschaffen und in dem Simulationswerkzeug SimulationX, das selbst ein Modelica Compiler ist, bereits umgesetzt [BL09]. Im Folgenden sollen die wichtigsten Aspekte der Implementierung, die für das weitere Verständnis erforderlich sind, kurz umrissen werden.

4.3.1. $\mathcal{O}(n)$ -MKS-BIBLIOTHEK FÜR MODELICA

Die Bibliothekselemente einer Modelica Bibliothek tauschen über *Konnektoren* miteinander Informationen aus. Jeder Konnektor i besteht aus einer oder mehreren Potenzialgrößen p_i und Flussgrößen f_i .

Alle Knoten, die miteinander verbunden sind bilden zusammen ein *Connection set*. Für ein Connection set an dem die Konnektoren $i = 1 \dots n$ beteiligt sind, siehe Abb. 4.5, gilt die Gleichheit der Potenzialgrößen

$$p_1 = p_2 = \dots = p_n$$

und das Verschwinden der Summe der Flussgrößen

$$\sum_i f_i = 0. \quad (4.4)$$

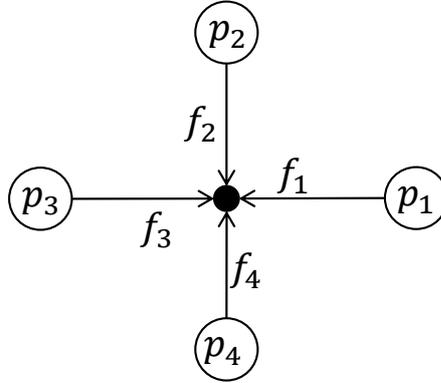


Abbildung 4.5.: Connection set

Im Sinne der Mechanik wird eine Verbindung zwischen zwei Konnektoren als eine starre abstandslose Verbindung interpretiert. Folglich sind die kinematischen Größen

- Pose \mathbf{z}
- Kinemate \mathbf{y}
- Beschleunigung \mathbf{x}

an dieser Stelle alle gleich und stellen somit die Potenzialgrößen des Konnektors dar. Die Last \mathbf{l} bildet die Flussgröße des Systems. Damit repräsentiert der Knotensatz (4.4) die Kräfte- und Momentenbilanz an der freigeschnittenen starren Verbindung. Anstelle jedoch die Last \mathbf{l} direkt zu übertragen, wird sie entsprechend der Bewegungsgleichung eines Blattkörpers k (3.18) als eine beschleunigungsproportionale Komponente \mathbf{M}_k^A und eine unabhängige Komponente \mathbf{l}_k^A übertragen. Damit gilt

$$\mathbf{l}_k = \mathbf{l}_{k,ext}^A - \mathbf{M}_k^A \cdot \mathbf{x}_k.$$

Entsprechend den Gleichungen (3.19) und (3.20) werden diese Größen ebenso wie Lasten in der starren Verbindung aufsummiert und an deren statt als Flußgrößen in dem Konnektor der $\mathcal{O}(n)$ -MKS-Bibliothek verwendet. Da zur Ausführung des $\mathcal{O}(n)$ -Algorithmus der aufspannende Baum des Topologiegraphen bekannt sein muß, erhält jedes Bibliothekselement eine primäre und beliebig viele sekundäre Schnittstellen. Das Bibliothekselement muss nun Gleichungen bereitstellen, mit denen es möglich ist, die kinematischen Größen der sekundären Konnektoren aus denen der primären zu berechnen. Darüber hinaus ist es notwendig, die kinetischen Größen \mathbf{M}_k^A und \mathbf{l}_k^A aus den kinetischen Größen der sekundären Konnektoren zu berechnen. Sobald der Modelica Compiler dieses System aus Gleichungen und Variablen sortiert hat und das System keine kinematischen Schleifen enthält, wird sich die Vorwärts- und Rückwärtskinematik, wie in Algorithmus 3.3 dargestellt, automatisch ergeben.

Auf diese Art ist es jedoch nicht möglich, ein Bibliothekselement mit mehr als einem primären Konnektor zu erstellen. Gäbe es ein Objekt mit zwei primären Konnektoren 1 und 2, so würde die Schnittlast des Konnektors 1

$$\mathbf{l} = \mathbf{M}_1^A \cdot \mathbf{x}_1 + \mathbf{M}_2^A \cdot \mathbf{x}_2 + \mathbf{l}^a$$

lauten und damit zusätzlich eine Abhängigkeit bezüglich der Beschleunigung des zweiten Konnektors besitzen. Dies kann jedoch durch die obige Konnektordefinition nicht dargestellt werden.

Stattdessen kann ein Element entworfen werden, das eine implizite Bindungsgleichung und nicht spezifizierte Zwangslasten λ_s enthält. Auf diese Weise entsteht ein lineares Gleichungssystem über dem $\mathcal{O}(n)$ -Algorithmus. Durch Anwenden des Tearingverfahrens aus Abschnitt 4.2.1 werden die λ_s als Tearingvariablen gewählt und können damit als bekannt angenommen werden. Über den Algorithmus wird danach so oft iteriert, bis das Residuum der impliziten Bindungsgleichungen ausreichend genau erfüllt ist. Dieses Vorgehen entspricht dem aus Abschnitt 3.4 und führt zu einem mehrmaligen Ausführen des $\mathcal{O}(n)$ -Algorithmus. Der genaue Mehraufwand lässt sich jedoch nur schwer abschätzen, da die Auswirkungen der symbolischen Optimierungen stark vom dem vorliegenden System abhängen. Die Modellierung der Systeme für Abb. 3.12 wurde auf diese Weise durchgeführt und es konnte gezeigt werden, dass der Modelica Compiler erkannt hat, dass die Faktorisierung nicht wiederholt werden muss.

Es ist jedoch nicht trivial, eigene Bibliothekselemente für diese Bibliothek zu erstellen, da immer mit den abstrakten Größen M_k^A und l_k^A gerechnet werden muss. Daher wurde im Rahmen dieser Arbeit die Methode der Subsysteme entworfen, mit der es möglich ist, alle auf Beschleunigungsebene linearen Beziehungen in ein Bibliothekselement einzubeziehen. Dies entspricht einer Übertragung des Macro-Joint Approaches aus [ER04] auf den $\mathcal{O}(n)$ -Algorithmus.

4.3.2. AUFBEREITUNG EINES SUBSYSTEMS

Es wird angenommen, dass das Subsystem über die primäre Schnittstelle 0 und über n_S sekundäre Schnittstellen $1 \dots n_S$ verfügen. Zusätzlich soll das Subsystem über beliebig viele interne Variablen $\mathbf{w} \in \mathbb{R}^{n_w}$ verfügen.

Zunächst müssen alle Gleichungen bereitgestellt werden, mit denen die Posen und die Kinematiken der sekundären Schnittstellen aus denen der primären Schnittstelle berechnet werden können.

Als nächstes muss das lineare Gleichungssystem auf Beschleunigungsebene gebildet werden. Dazu werden für jede sekundäre Schnittstelle i die Lastbilanz

$$\mathbf{l}_i = \mathbf{M}_i^a \cdot \mathbf{x}_i + \mathbf{l}_i^a \quad (4.5)$$

und die Berechnung der Beschleunigung der sekundären Schnittstelle, die potenziell von den inneren Variablen \mathbf{w} abhängen darf, aufgeführt.

$$\mathbf{x}_i = \mathbf{G}_{i0} \mathbf{x}_0 + \mathbf{\Phi}_{iw} \mathbf{w} + \mathbf{c}_i$$

Zusätzlich muss die Lastbilanz für die primäre Schnittstelle aufgestellt werden.

$$\mathbf{l}_0 = \sum_{i=1}^{n_S} \mathbf{G}_{i0}^T \mathbf{l}_i + \mathbf{F}_{0w} \mathbf{w}$$

Komplettiert wird das System durch n_w Gleichungen zur Bestimmung der inneren Variablen

$$\mathbf{W} \mathbf{w} + \sum_{i=0}^{n_S} (\mathbf{D}_{wi} \mathbf{l}_i + \mathbf{H}_{wi} \mathbf{x}_i) = \boldsymbol{\gamma}.$$

Da jedes Element als Subsystem betrachtet wird, muss es ebenso für ein einfaches Gelenk gelten.

In diesem Fall würde

$$\begin{aligned}
 n_w = n_S &= 1 \\
 \mathbf{w} &= s \\
 \Phi_{1w} &= \Phi \\
 D_{w1} &= \Phi^T \\
 \mathbf{H}_{w1} = \mathbf{H}_{w0} = D_{w0} = F_{0w} &= \mathbf{0}
 \end{aligned}$$

gelten.

Die genannten Gleichungen können nun wie folgt zu einem linearen Gleichungssystem zusammengefasst werden:

$$\underbrace{\begin{bmatrix} -\mathbf{E} & \mathbf{M}_i^a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{E} & \Phi_{iw} & \mathbf{0} \\ D_{wi} & \mathbf{H}_{wi} & \mathbf{W} & D_{w0} \\ \mathbf{G}_{i0}^T & \mathbf{0} & F_{0w} & -\mathbf{E} \end{bmatrix}}_A \begin{bmatrix} l_i \\ \mathbf{x}_i \\ \mathbf{w} \\ l_0 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} \\ -\mathbf{G}_{i0} \\ -\mathbf{H}_{w0} \\ \mathbf{0} \end{bmatrix}}_B \mathbf{x}_0 + \underbrace{\begin{bmatrix} -l_i^a \\ -\mathbf{c}_i \\ \gamma \\ \mathbf{0} \end{bmatrix}}_b. \quad (4.6)$$

Formal ergibt sich für die unbekannte Schnittlast l_0

$$l_0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix} \left(\mathbf{A}^{-1} \mathbf{B} \mathbf{x}_0 + \mathbf{A}^{-1} \mathbf{b} \right).$$

Die unbekannt kinetischen Größen \mathbf{M}_0^a und l_0^a der primären Schnittstelle 0 können nun leicht durch Vergleich mit Gl. (4.5) abgelesen werden. Für eine robuste und effiziente Implementierung ist jedoch von der Berechnung der Inversen \mathbf{A}^{-1} abzuraten. Stattdessen sollte folgendes lineares Gleichungssystem gelöst werden:

$$\begin{bmatrix} -\mathbf{E} & \mathbf{M}_1^a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{E} & \Phi_{iw} & \mathbf{0} \\ D_{wi} & \mathbf{H}_{wi} & \mathbf{W} & D_{w0} \\ \mathbf{G}_{i0}^T & \mathbf{0} & F_{0w} & -\mathbf{E} \end{bmatrix} \begin{bmatrix} l_{i,0} & \mathbf{X} \\ \mathbf{x}_{i,0} & \mathbf{Y} \\ \mathbf{w}_0 & \mathbf{Z} \\ l_0^a & \mathbf{M}_0^a \end{bmatrix} = \begin{bmatrix} -l_i^a & \mathbf{0} \\ -\mathbf{c}_i & -\mathbf{G}_{i0} \\ \gamma & -\mathbf{H}_{w0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (4.7)$$

Dies kann entweder numerisch geschehen oder, um nochmals Rechenzeit einzusparen, auf symbolischen Wege. Oft wird die Matrix \mathbf{W} nur schwach besetzt sein, in diesem Fall ist unbedingt auf die Permutation des Gleichungssystems zu achten, da dies die Effizienz sowie die Lösbarkeit stark beeinflussen kann.

4.3.3. IMPLEMENTIERUNG EINER KINEMATISCHEN SCHLEIFE ALS SUBSYSTEM

In diesem Abschnitt soll die vorgestellte Methode der Subsysteme genutzt werden, um die Berechnung der kinematischen Schleifen effizienter zu gestalten. Dazu wird eine gesamte kinematische Schleife als Subsystem aufgefasst. Das konkrete Vorgehen soll anhand eines ebenen Viergelenks, siehe Abb. (4.6), demonstriert werden. Das Prinzip kann in analoger Weise auf jede andere Schleife angewendet werden.

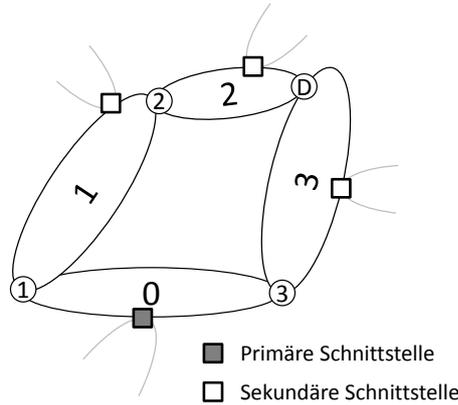


Abbildung 4.6.: Ebenes Viergelenk

BESCHLEUNIGUNGSEBENE

Es soll zuerst mit der Berechnung der Beschleunigungsebene begonnen werden. Daraus werden die kinetischen Größen gewonnen, die an die primäre Schnittstelle weitergegeben werden. Auf Lage- und Geschwindigkeitsebene ergeben sich zusätzlich ein nichtlineares und ein lineares Gleichungssystem, deren effiziente Lösung im Abschnitt 4.3.3 betrachtet wird.

Alle Körper der kinematischen Schleife werden als masselos angenommen. Um entsprechende Trägheiten zu berücksichtigen, können Körper an den sekundären Schnittstellen angeschlossen werden. Die sekundären Anschlüsse liegen, abweichend von der Abbildung, direkt auf den Gelenkpunkten, die gleichzeitig auch als Körperreferenzpunkte dienen. Die Schleife wird gedanklich am Gelenk D geschnitten, womit dieses zu einem sekundären Gelenk, siehe Abschnitt (3.1.4), wird. Die internen Gleichungen setzen sich unter anderem aus den Zusammenhängen der Beschleunigungen über den primären Gelenken zusammen.

$$\mathbf{x}_1 = \mathbf{G}_{10}\mathbf{x}_0 + \Phi_1\dot{s}_1 + \mathbf{c}_1$$

$$\mathbf{x}_2 = \mathbf{G}_{21}\mathbf{x}_1 + \Phi_2\dot{s}_2 + \mathbf{c}_2$$

$$\mathbf{x}_3 = \mathbf{G}_{30}\mathbf{x}_0 + \Phi_3\dot{s}_3 + \mathbf{c}_3$$

Zur Beschreibung des sekundären Gelenks D wird die Beschleunigung \mathbf{x}_D in dem Gelenkpunkt auf Körper 2 eingeführt und es folgt

$$\mathbf{x}_D = \mathbf{G}_{D2}\mathbf{x}_2 + \mathbf{c}_{D2} \quad (4.8)$$

$$\mathbf{x}_D = \mathbf{G}_{D3}\mathbf{x}_3 + \Phi_D\dot{s}_D + \mathbf{c}_{D3}. \quad (4.9)$$

Zusätzlich gilt für jedes Gelenk das Prinzip der virtuellen Leistung

$$\Phi_i^T \mathbf{l}_i = \mathbf{0}, \quad i = \{1, 2, 3, D\}. \quad (4.10)$$

Um die Anzahl der Variablen und Gleichungen zusätzlich zu reduzieren, wird \mathbf{x}_D durch Gleichsetzen der Gl. (4.8) und (4.9) eliminiert.

Das so entstandene Gleichungssystem ist nur schwach besetzt, daher sollte unbedingt die Permutation, also die Reihenfolge der Gleichungen und Variablen, günstig gewählt werden. Da das Subsystem selbst ein Mehrkörpersystem ist, kann zu diesem Zweck direkt auf die Erkenntnisse aus Kapitel 3 verwiesen werden. Demnach wäre es möglich, die Permutation entweder dem $\mathcal{O}(n)$ oder dem $\mathcal{O}(n^3)$ Formalismus nachzuempfinden. Gegen die Permutation des $\mathcal{O}(n)$ Formalismus spricht jedoch zum einen die geringe Größe des Subsystems und zum anderen lassen sich

kinematische Schleifen damit nicht effizient lösen, siehe Abschnitt 3.4. Außerdem müssten die auf die freien Bewegungsrichtungen der primären Gelenke projizierten Massen $\hat{M}_i = \Phi_i^T M_i^A \Phi_i$ stets invertierbar sein. Eine notwendige Bedingung dafür ist jedoch, dass an jeder sekundären Schnittstelle des Subsystems ein massebehaftetes Objekt angeschlossen wäre. Dies würde die Anwendbarkeit unnötig einschränken. Es unterstreicht nochmals den Fakt, dass die Permutation einen Einfluss auf die Lösbarkeit des Systems besitzt.

Das umsortierte lineare Gleichungssystem lautet somit:¹

$$\begin{bmatrix}
 & l_3 & l_2 & l_1 & x_3 & x_2 & x_1 & \dot{s}_3 & \dot{s}_2 & \dot{s}_1 & l_D & \dot{s}_D & l_0 \\
 \mathbf{I}_3 & -\mathbf{E} & \circ & \circ & \mathbf{M}_3 & \circ & \circ & \circ & \circ & \circ & \mathbf{G}_{D3}^T & \circ & \circ \\
 \mathbf{I}_2 & \circ & -\mathbf{E} & \circ & \circ & \mathbf{M}_2 & \circ & \circ & \circ & \circ & -\mathbf{G}_{D2}^T & \circ & \circ \\
 \mathbf{I}_1 & \circ & \mathbf{G}_{21}^T & -\mathbf{E} & \circ & \circ & \mathbf{M}_1 & \circ & \circ & \circ & \circ & \circ & \circ \\
 \mathbf{A}_3 & \circ & \circ & \circ & -\mathbf{E} & \circ & \circ & \Phi_3 & \circ & \circ & \circ & \circ & \circ \\
 \mathbf{A}_2 & \circ & \circ & \circ & \circ & -\mathbf{E} & \mathbf{G}_{21} & \circ & \Phi_2 & \circ & \circ & \circ & \circ \\
 \mathbf{A}_1 & \circ & \circ & \circ & \circ & \circ & -\mathbf{E} & \circ & \circ & \Phi_1 & \circ & \circ & \circ \\
 D_3 & \Phi_3^T & \circ \\
 D_2 & \circ & \Phi_2^T & \circ \\
 D_1 & \circ & \circ & \Phi_1^T & \circ \\
 \mathbf{A}_D & \circ & \circ & \circ & \mathbf{G}_{D3} & -\mathbf{G}_{D2} & \circ & \circ & \circ & \circ & \circ & \Phi_D & \circ \\
 D_D & \circ & \Phi_D^T & \circ & \circ \\
 \mathbf{I}_0 & \mathbf{G}_{30}^T & \circ & \mathbf{G}_{10}^T & \circ & -\mathbf{E}
 \end{bmatrix}
 \begin{bmatrix}
 l_{3,0} \\
 l_{2,0} \\
 l_{3,0} \\
 x_{3,0} \\
 x_{2,0} \\
 x_{1,0} \\
 \dot{s}_{3,0} \\
 \dot{s}_{2,0} \\
 \dot{s}_{1,0} \\
 l_{D,0} \\
 \dot{s}_D \\
 l_0^a \\
 M_0^a
 \end{bmatrix}
 =
 \begin{bmatrix}
 1 & x_0 \\
 l_{3,ext} & \mathbf{0} \\
 l_{2,ext} & \mathbf{0} \\
 l_{1,ext} & \mathbf{0} \\
 -c_3 & \mathbf{G}_{30} \\
 -c_2 & \mathbf{0} \\
 -c_1 & \mathbf{G}_{10} \\
 0 & \mathbf{0} \\
 0 & \mathbf{0} \\
 0 & \mathbf{0} \\
 -c_D & \mathbf{0} \\
 0 & \mathbf{0} \\
 0 & \mathbf{0}
 \end{bmatrix}
 \quad (4.11)$$

mit

$$c_D = c_{D3} - c_{D2}.$$

Die Gauß-Elimination wurde symbolisch vorgenommen und daraus die Systemgleichungen des Subsystems abgeleitet, siehe Anhang A.3. Um die Invertierbarkeit zu gewährleisten, wurden in dem Prozess die Gleichungen des sekundären Gelenks auf die gesperrten Richtungen $\bar{\Psi}_D$ projiziert und für die Schnittlasten l_D des sekundären Gelenks die Basis $[\Psi_D, \bar{\Psi}_D]$ eingeführt. Das hier vorgestellte System lässt sich durch entsprechende Wahl der gelenkabhängigen Größen Φ , $\bar{\Psi}$ und c auf verschiedene kinematische Schleifen übertragen. Umgesetzt wurde es für das ebene Viergelenk.

LAGE- UND GESCHWINDIGKEITSEBENE

Aufgrund der Betrachtung der gesamten Schleife als ein Objekt, ist es möglich, die Gleichungssysteme auf Lage- und Geschwindigkeitsebene mittels Coordinate Partitioning [WHB81] und der Verwendung analytischer Lösungen deutlich effizienter zu lösen, als dies ein generisches numerisches Verfahren erledigen könnte. Das Coordinate Partitioning beschreibt ein Verfahren mit dem es möglich ist, das differenzial-algebraische System (DAE) eines Mehrkörpersystems mit kinematischer Schleife auf ein gewöhnliches Differenzialgleichungssystem (ODE) zu reduzieren. Dazu wird der Zustandsvektor in eine Menge von abhängigen und unabhängigen Koordinaten partitioniert. Dieses Verfahren soll anhand des ebenen Viergelenks, siehe Abb. 4.7, erläutert werden.

¹Zur besseren Übersicht wurde eine Zeilen- und Spaltenüberschrift eingeführt.

Die Nullen wurden als \circ markiert und im Vektor der Unbekannten alle nicht benötigten Ergebnisse mit einem \cdot gekennzeichnet

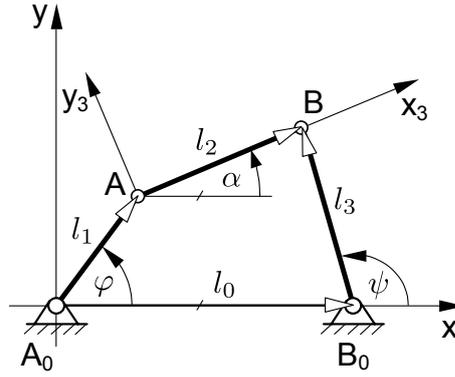


Abbildung 4.7.: Skizze eines ebenen Viergelenks, nach [Mod13]

Ein ebenes Viergelenk verfügt über einen Freiheitsgrad. Daraus folgt, dass die drei Gelenkwinkel $q_1 = \varphi$, $q_2 = \alpha$ und $q_3 = \psi$ nicht voneinander unabhängig sein können. Die Lage des Gelenkpunktes D ist über zwei Wege erreichbar, und muss in beiden Fällen gleich sein, also

$$l_1 \cos \varphi + l_2 \cos \alpha = l_0 + l_3 \cos \psi \quad (4.12a)$$

$$l_1 \sin \varphi + l_2 \sin \alpha = l_3 \sin \psi. \quad (4.12b)$$

Dieses Gleichungssystem lässt sich nach ψ in Abhängigkeit von φ auflösen zu [Mod13]

$$\psi_{1/2} = 2 \arctan \frac{B \pm \sqrt{A^2 + B^2 - C^2}}{A - C} \quad (4.13)$$

mit

$$\begin{aligned} A &= 2l_3 (l_0 - l_1 \cos \varphi) \\ B &= -2l_1 l_3 \sin \varphi \\ C &= l_0^2 + l_1^2 - l_2^2 + l_3^2 - 2l_0 l_1 \cos \varphi. \end{aligned}$$

Beide Lösungen der Gl. (4.13) sind gültig und beschreiben jeweils eine Konfiguration in welcher der Punkt B entweder überhalb bzw. unterhalb der Verbindungslinie $\overline{AB_0}$ liegt.

Nachdem eine Lösung ψ ausgewählt wurde, ergibt sich α zu

$$\alpha = \arctan \frac{l_3 \sin \psi - l_1 \sin \varphi}{l_0 + l_3 \cos \psi - l_1 \cos \varphi}. \quad (4.14)$$

Zur Vermeidung von Unstetigkeitsstellen und zur eindeutigen Bestimmung des Quadranten ist es jedoch besser die Hilfsfunktion

$$\alpha = \text{atan2}(l_3 \sin \psi - l_1 \sin \varphi, l_0 + l_3 \cos \psi - l_1 \cos \varphi)$$

zu benutzen. Mithilfe dieser Beziehungen lassen sich nun die Gelenkwinkel in einen unabhängigen Winkel $u = \varphi$ und zwei abhängige Winkel $v = [\alpha, \psi]^T$ unterteilen. Gleichungen (4.13) und (4.14) geben eine Vorschrift an, mit der es möglich ist, die abhängigen Größen v aus u zu berechnen. Üblicherweise würde sonst das Gleichungssystem (4.12) als nichtlineares Gleichungssystem iterativ gelöst werden. Dies hat sich als deutlich ineffizienter als die Benutzung der analytischen Lösung herausgestellt [SBK10]. Das Verfahren kann jedoch nur funktionieren, wenn sichergestellt ist, dass es zu jedem Wert des unabhängigen Winkels u genau eine Lösung

für v gibt. Eine notwendige Bedingung, im Falle des Vieregelns, ist daher die Festlegung auf ein Vorzeichen in Gl. (4.13). Diese Forderung ist jedoch nicht hinreichend. Abhängig von den Längenverhältnissen kann es vorkommen, dass Gelenk 1 umläuft während Gelenk 3 eine Schwingung vollführt. In diesem Fall ist es möglich von φ auf ψ zu schließen, jedoch nicht umgekehrt. Nimmt das Vieregelnk eine Stellung ein, in der ein Übergang von der positiven zur negativen Lösung von Gl. (4.13) möglich ist, so tritt an dieser Stelle eine Singularität auf. Folglich sollte die Wahl des unabhängigen Winkels dem Modellierer vorbehalten sein. Andererseits kann auch eine automatische, ggf. auch während der Simulation variierende, Auswahl getroffen werden, wie es von [MOE00] vorgestellt wurde. Für die Simulation von Vieregelnen in Baumaschinen kann das Coordinate Partitioning jedoch immer eingesetzt werden, da die Freiheit von Singularitäten sowie die Eindeutigkeit der Lage der Körper konstruktiv gefordert wird.

Die Ableitung von Gl. (4.12) definiert den Zusammenhang der Gelenkwinkelgeschwindigkeiten $s_i = \dot{q}_i$. Das Gleichungssystem kann aufgeschrieben werden als

$$\mathbf{y}_3 = \mathbf{G}_{30}\mathbf{y}_0 + \Phi_3 s_3 \quad (4.15a)$$

$$\mathbf{y}_2 = \mathbf{G}_{21}\mathbf{y}_1 + \Phi_2 s_2 \quad (4.15b)$$

$$\mathbf{y}_1 = \mathbf{G}_{10}\mathbf{y}_0 + \Phi_1 s_1 \quad (4.15c)$$

$$\mathbf{0} = \bar{\Psi}_D^T (\mathbf{G}_{D3}\mathbf{y}_3 - \mathbf{G}_{D2}\mathbf{y}_2) \quad (4.15d)$$

Werden alle Gleichungen ineinander eingesetzt, so taucht der Koeffizient $(\mathbf{G}_{D3}\mathbf{G}_{30} - \mathbf{G}_{D2}\mathbf{G}_{21}\mathbf{G}_{10})$ vor \mathbf{y}_0 auf. Dieser vereinfacht sich zu null, sofern die Zwangsbedingung (4.12) exakt erfüllt ist. Da die Gleichungen des Systems (4.15) in abgeleiteter Form auch in (4.11) vorkommen und diese sich nur in der rechten Seite unterscheiden, ist es möglich, die Faktorisierung der Koeffizientenmatrix von der Geschwindigkeitsebene aus ohne Mehraufwand aus denen der Beschleunigungsebene zu gewinnen.

Das Vieregelnk wurde in SimulationX implementiert. Die so entstehenden Rechenzeiten werden im folgenden Abschnitt mit anderen kommerziell und frei verfügbaren Varianten verglichen.

VERGLEICH DER RECHENZEITEN

Im Folgenden sollen die Rechenzeiteinsparungen bei der Verwendung einer kinematischen Schleife, in Form eines Subsystems untersucht werden. Dazu wurde das Vieregelnk mit all den im vorherigen Abschnitt vorgestellten Optimierungen als Subsystem in SimulationX implementiert. Intern setzt die Software einen objektorientierten $\mathcal{O}(n)$ -Algorithmus ein. Um die Rechenzeit abschätzen zu können, wurde als numerischer Zeitintegrator das Euler-Vorwärtsverfahren mit einer festen Schrittweite genutzt und die Simulationsdauer durch die Anzahl der Schritte geteilt. Alle Ergebnisse wurden auf die Zeit des Vieregelns als Subsystem normiert.

Zuerst wurde zu Testzwecken ein einzelnes ebenes Vieregelnk aufgebaut und mit den folgenden Simulationsbibliotheken mechanischer Systeme verglichen:

- Planar Mechanics von ITI [SGR11],
- Planar Linkages des IFMT, TU Chemnitz [EMB⁺10],
- Planar Mechanics des DLR-RM [Zim12],
- MBS Mechanics von ITI [WBN06],
- Modelica.Mechanics.MultiBody der Modelica Standard Library (MSL) 3.2.1 [OEM03],
- Modelica.Mechanics.MultiBody der MSL 3.2.1 mit analytischer Lösung (JointUSR) [Woe88]

Das Ergebnis ist in Abb. 4.8 dargestellt. Es ist deutlich zu erkennen, dass das optimierte Subsystem etwa zehnmals schneller rechnet als alle anderen Bibliotheken, mit denen die kinematische

Schleife traditionell modelliert wurde. Aufgrund des verwendeten Indexreduktionsalgorithmus in SimulationX, wird jedes dieser Modelle als gewöhnliche Differentialgleichung behandelt und damit in Minimalkoordinaten berechnet. Somit können sich Genauigkeitsunterschiede nur bei der Berechnung der (minimalen) Gelenkbeschleunigungen ergeben. Das nichtlineare Gleichungssystem auf Lageebene wird, abhängig vom Modell, entweder analytisch oder numerisch mit einem modifizierten Newtonverfahren gelöst. Somit kann es in den Gelenkwinkeln zwar zu sehr geringen jedoch meist irrelevanten Abweichungen kommen. Auf Geschwindigkeits- und Beschleunigungsebene lösen alle Modelle das gleiche lineare Gleichungssystem, ggf. mit unterschiedlichen Permutationen, und gelangen somit zu gleichen Ergebnissen. Abweichungen können aufgrund der Winkelabweichungen und den Rundungsfehlern entstehen, führten jedoch zu keinen nennenswerten Genauigkeitseinbußen.

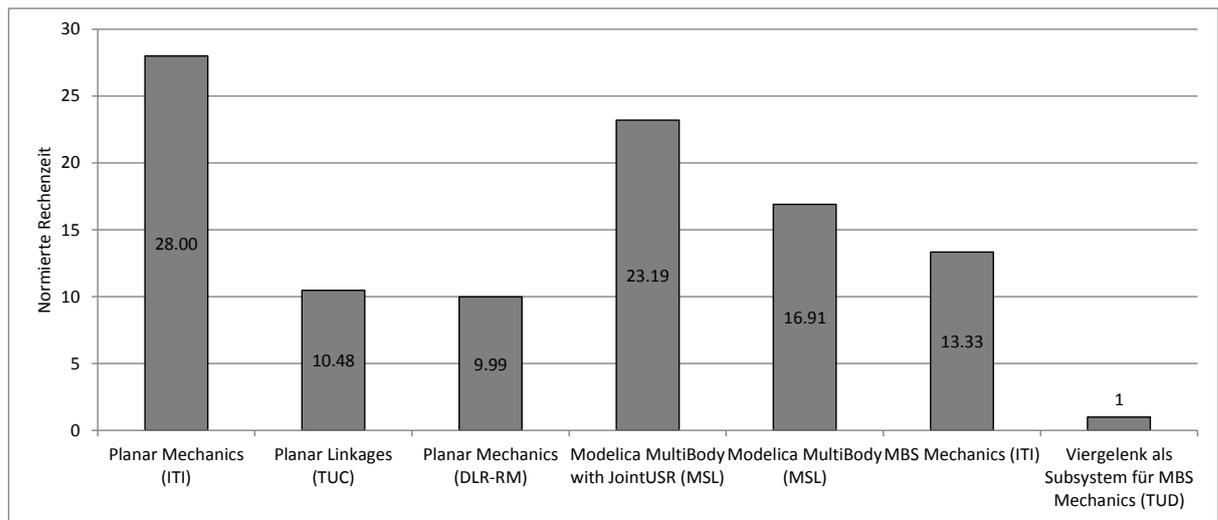


Abbildung 4.8.: Normierte Rechenzeit eines Viergelenks mit verschiedenen Modelica Mechanikbibliotheken mit SimulationX 3.6

Im folgenden Schritt werden praxisrelevante Vergleiche vorgestellt. Als Vergleichsobjekt dient das vereinfachte mechanische Modell eines Masts einer Autobetonpumpe. Dieses setzt sich aus fünf Armen zusammen, die miteinander mit Drehgelenken verbunden sind. Jedes Gelenk wird über einen Hydraulikzylinder aktuiert. Mithilfe von Viergelenken ist es möglich die lineare Bewegung der Zylinder auf einen Winkelbereich von mehr als 180° zwischen den Armen der Autobetonpumpe zu übersetzen. Dies ist notwendig, um den Mast zusammenfallen zu können. Die Hydraulikzylinder wurden der Einfachheit halber durch Feder-/Dämpfersysteme mit geeigneter Steifigkeit und Dämpfung ersetzt. Abb. 4.9b zeigt die Rechenzeiten für eine Modellierung der kinematischen Schleifen mit Zwangsbedingungen, mit dem analytischen Viergelenk als Subsystem und mit deaktivierten Zwangsbedingungen. Obwohl im letzteren Fall unsinnige Ergebnisse errechnet werden, ist es jedoch ein gutes Maß für die minimal erreichbare Rechenzeit. Da jedes der Viergelenke über zwei Zwangskräfte verfügt, verwundert es nicht, dass etwa die achtfache Zeit zur Berechnung des Gesamtsystems benötigt wird. Das analytische Viergelenk rechnet jedoch nur etwa 5% langsamer als die offene kinematische Kette und ist somit fast achtmal schneller als die Variante mit Zwangsbedingungen. Wie die nächste Untersuchung zeigen wird, sind diese Ergebnisse nicht auf alle in der Praxis vorkommenden Modelle übertragbar. Das Mastmodell verfügt über sehr viele kinematische Schleifen bei gleichzeitig wenigen Körpern und anderen Modellelementen und ist daher besonders günstig für die vorgestellte Lösung.

Aus diesem Grund wurde zusätzlich das Modell eines Baggararms mit anspruchsvoller Hydraulik untersucht, indem lediglich ein Viergelenk vorkommt. Ursprünglich verfügte das Modell über vier zusätzliche Zwangsbedingungen, da die Mechanik jedes Hydraulikzylinders als solche mo-

delliert wurde. Deren Entfernung reduzierte die Rechenzeit um etwa 30%, siehe Abb. 4.10b. Die anschließende Verwendung des analytischen Viergelenks für die verbleibende kinematische Kette der Löffelaufhängung führte nochmals zu einer Rechenzeitersparnis von gut 30%.

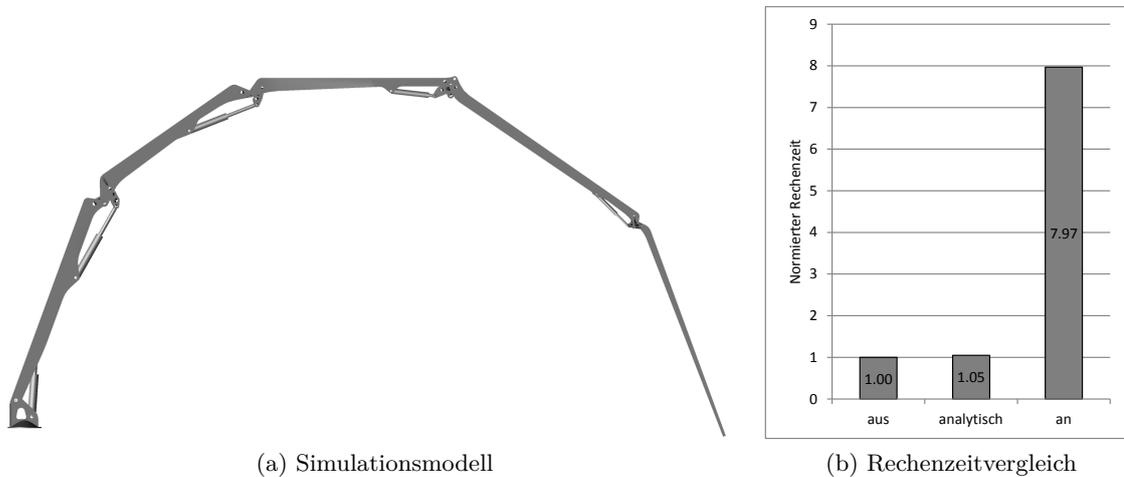


Abbildung 4.9.: Mast einer Autobetonpumpe

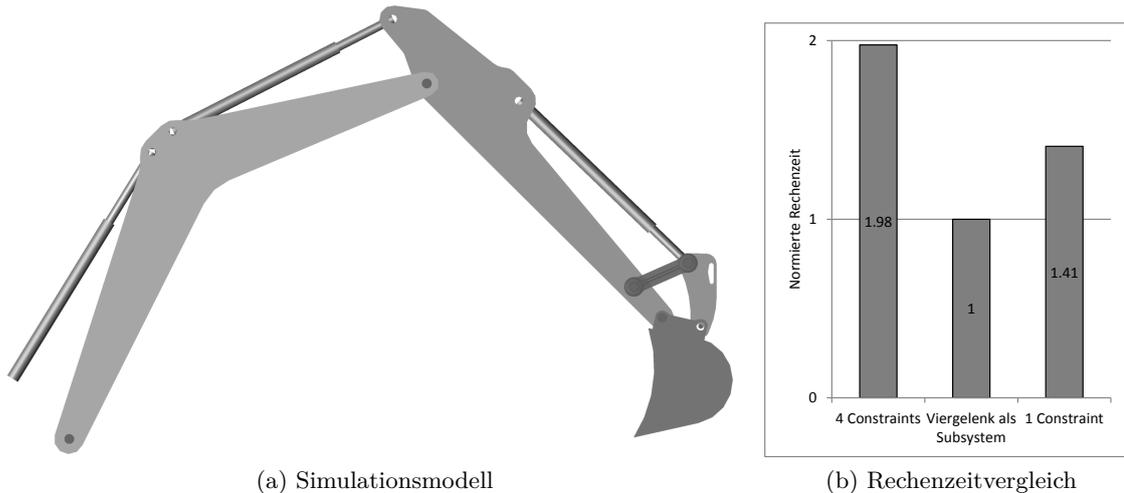


Abbildung 4.10.: Baggerarm

4.3.4. IMPLEMENTIERUNG EINES GLEICHLAUFGELENKES ALS SUBSYSTEM

In diesem Abschnitt soll die vorgestellte Methode der Subsysteme genutzt werden, um die Funktionalität eines Gleichlaufgelenkes in einem Element der MKS-Bibliothek von SimulationX zu modellieren. Das Gleichlaufgelenk findet vor allem in frontgetriebenen Kraftfahrzeugen Verwendung. Für diese Klasse von Fahrzeugen ist es wichtig, dass die Antriebsleistung des Motors gleichmäßig auf die Vorderräder übertragen wird. Aufgrund der Federung und Lenkung führen diese jedoch große Relativbewegungen bezüglich des Getriebeausgangs aus. Dabei treten große Beugewinkel in den Antriebswellen zwischen Rad und Getriebe auf. Dies verbietet den Einsatz von Kreuzgelenken, da sich unter solchen Bedingungen ein stark variierendes Drehzahlübersetzungsverhältnis einstellen würde. Stattdessen kommen sogenannte Gleichlaufgelenke (auch homokinetische Gelenke genannt) zum Einsatz, die diesen Nachteil, entsprechend ihres Namens, nicht aufweisen [SBH10].

Es gibt eine Vielzahl an Ausführungen von Gleichlaufgelenken. Zu den bekanntesten Vertretern gehören:

- Rzeppa Gelenk, Abb. 4.11a
- Tracta Gelenk
- Weiss Gelenk
- Doppeltes Kreuzgelenk
- Zahnkupplung.

Oft werden auch Tripod Gelenke, siehe Abb. 4.11b, ebenfalls als Gleichlaufgelenke bezeichnet. Sie stellen jedoch keinen exakten Gleichlauf sicher, da das innere Joch eine geringe Nutation ausführt [UP98]. Dieser Effekt ist jedoch nicht stark ausgeprägt. Daher werden die Gelenke in Kraftfahrzeugen verwendet, da sie im Unterschied zu reinen Gleichlaufgelenken zusätzlich Längenunterschiede in axialer Richtung ausgleichen können.

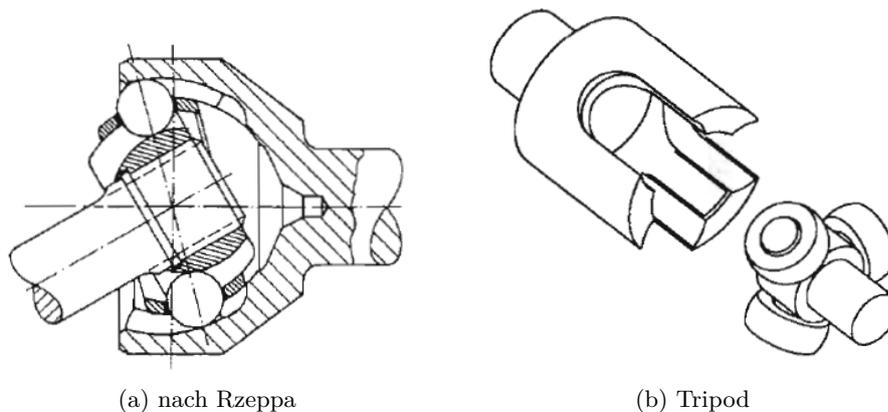


Abbildung 4.11.: Ausführungen von Gleichlaufgelenken (aus Schmitz 1994)

Im Folgenden werden drei Beschreibungsformen für die Funktionsweise eines Gleichlaufgelenkes basierend auf

- Eulerparameter,
- drei Drehgelenken und
- zwei symmetrischen angeordneten Kreuzgelenken

hergeleitet und für einen Vergleich der jeweils benötigten Rechenzeiten implementiert.

BESCHREIBUNG

Ideale Gleichlaufgelenke weisen ein konstantes Drehzahlverhältnis der An- und Abtriebswelle auf, sofern die Körper A und B keine Relativbewegung zueinander ausführen (siehe Abb. 4.12) und damit der Beugewinkel konstant bleibt. Bei sich änderndem Beugewinkel der Achsen tritt in der Regel eine leichte Ungleichförmigkeit in der Abtriebsdrehgeschwindigkeit auf, die jedoch deutlich geringer ist als es bei herkömmlichen Kreuzgelenken der Fall wäre. Abb. 4.12 zeigt ein abstrahiertes Gleichlaufgelenk, das zwar in der Praxis nicht verwendet wird, jedoch zur Veranschaulichung der Funktionsweise sehr gut geeignet ist. Eine zentrale Rolle spielt dabei die *homokinetische Ebene* [Hun90]. Zur Beschreibung dieser Ebene werden jeweils ein Punkt auf der Eingangs- und Ausgangswelle gesucht, die den gleichen Abstand zum Kugelgelenk S1 besitzen. Die homokinetische Ebene enthält alle Punkte, deren Abstand zu diesen beiden Punkten gleich ist. In [MM08] wird mithilfe von Myard's Theorem gezeigt, dass die untersuchten Gelenkformen (two rotoidal couples and one planar couple) die Eigenschaft des Gleichlaufs besitzen, sofern sie

symmetrisch zur homokinetischen Ebene aufgebaut sind. Dies hat zur Folge, dass die *relative motion instantaneous screw axis* \vec{u} , die im Folgenden als Knickachse bezeichnet wird, innerhalb der homokinetischen Ebene liegen muss [PVS07]. Die Knickachse entspricht jedoch genau der Achse, die laut Gleichung (A.13), in den Eulerparametern \mathbf{p} enthalten ist. Durch Nutzung der Eulerparameter \mathbf{p} zur Parametrierung der relativen Pose lässt sich die Zwangsbedingung auf Lageebene [Lac07, Tas01] formulieren als

$$\begin{bmatrix} 0 & \vec{n}_{a1} \end{bmatrix} \cdot \mathbf{p} = \begin{bmatrix} 0 & \vec{n}_{b1} \end{bmatrix} \cdot \mathbf{p} = 0. \quad (4.16)$$

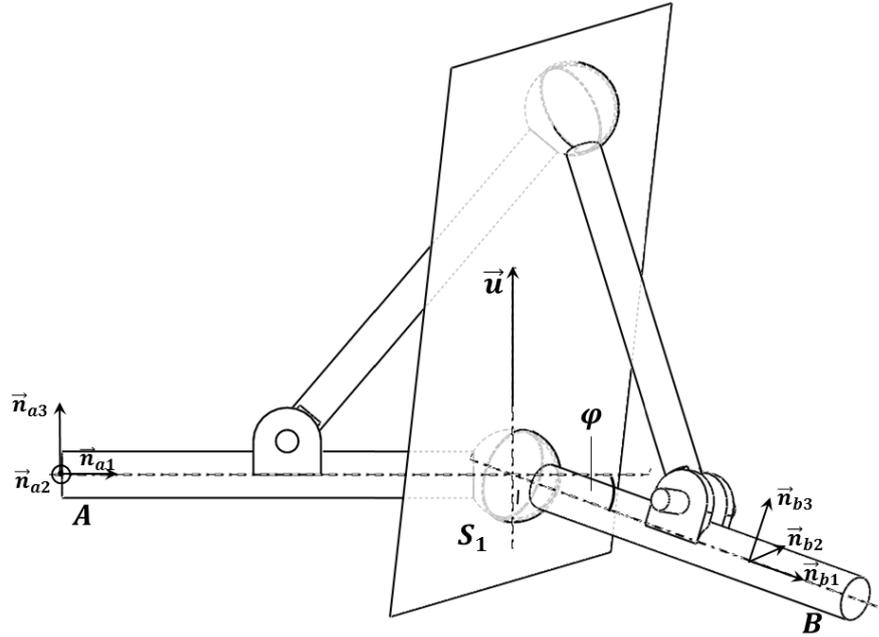


Abbildung 4.12.: Abstrahiertes Gleichlaufgelenk (nach [Sch94])

Die Knickachse \vec{u} kann nur in der homokinetischen Ebene liegen, wenn sie senkrecht auf den Wellenachsen \vec{n}_{a1} und \vec{n}_{b1} steht. Diese Erkenntnis lässt sich verwenden, um ein Gleichlaufgelenk aus einer Abfolge von drei Drehgelenken zu konstruieren, siehe Abb. 4.13. Das erste Drehgelenk führt eine Drehung mit dem Winkel θ um die Wellenachse \vec{n}_{a1} aus. Würde die zweite Drehachse senkrecht zu \vec{n}_{a1} gewählt, lässt sich für jede mögliche Knickachse \vec{u} ein Winkel θ finden, mit dem eine Drehung um \vec{u} erfolgen kann. Anschließend wird die Drehung des ersten Drehgelenks durch eine Drehung um \vec{n}_{b1} mit dem Winkel $-\theta$ kompensiert. Diese Beschreibung ist jedoch nur eingeschränkt nutzbar, da sie eine Singularität aufweist, sobald die Achsen parallel zueinander liegen. Dies wird daran sichtbar, dass in dieser Konfiguration eine Winkelgeschwindigkeit $\dot{\theta}$ zu keiner Bewegung von massebehafteten Komponenten führt und folglich mit keiner kinetischen Energie verbunden ist. Daher muss in diesem Fall die reduzierte Massenmatrix \hat{M} einen Rangabfall aufweisen.

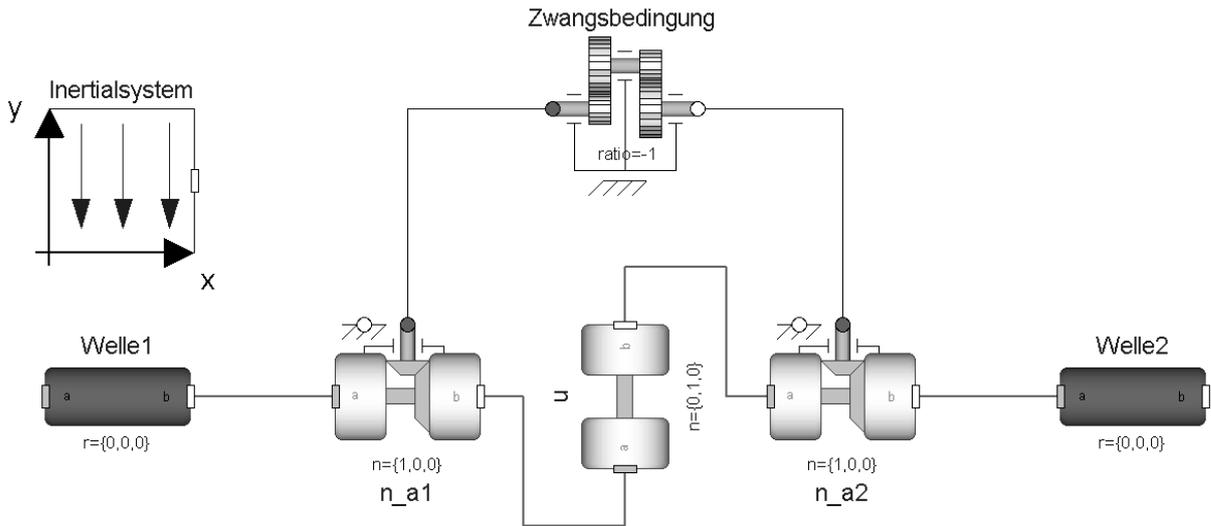


Abbildung 4.13.: Gleichlaufgelenk, modelliert mit Modelica

Eine dritte Beschreibung resultiert aus der Beobachtung, dass das doppelte Kreuzgelenk, sofern symmetrisch bezüglich der homokinetischen Ebene, ebenfalls ein Gleichlaufgelenk darstellt. Dies wurde in [MM08] ausgenutzt, um ebenfalls ein Gleichlaufgelenk mithilfe von drei Drehgelenken darzustellen. Dazu wurde die Gelenkwelle zwischen den beiden Kreuzgelenken entfernt, so dass die inneren Drehgelenke der Kreuzgelenke aufeinander fallen. Die Symmetrie bedingt, dass die beiden äußeren und die beiden inneren Drehgelenke jeweils um den gleichen Winkel verdreht sind. Somit erfolgt die erste Drehung mit dem Winkel $\frac{\theta}{2}$ um \vec{n}_{a1} , anschließend mit φ um \vec{n}_b , gefolgt von einer Drehung um \vec{n}_{a2} erneut mit dem Winkel $\frac{\theta}{2}$, siehe Abb. 4.14.

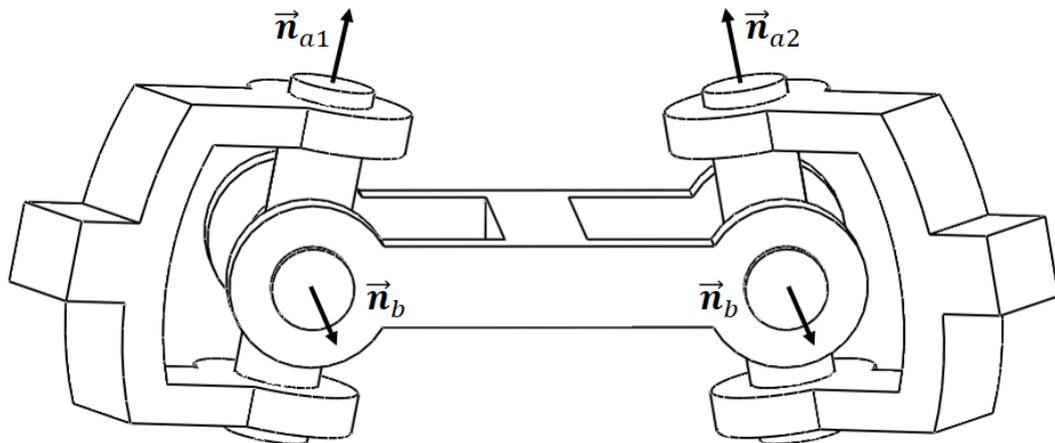


Abbildung 4.14.: Modell eines Gleichlaufgelenks als doppeltes Kreuzgelenk

Jede dieser drei Beschreibungsarten verfügt über Zwangsbedingungen. Entweder Gleichung (4.16) oder die Forderung, dass die Drehwinkel zweier achsparallelen Drehgelenke gleich sein bzw. die Drehwinkel zweier senkrecht zur Wellenachse stehenden Drehgelenke sich aufheben sollen. Dies führt zu einem deutlichen Mehraufwand beim Einsatz mit den im vorigen Abschnitt vorgestellten $\mathcal{O}(n)$ -Formalismen für Mehrkörpersysteme. Im Folgenden wird nun vorgestellt, wie mithilfe der Dualen Basen bzw. Modes of Motion, eine effiziente explizite Formulierung gewonnen werden kann.

BESTIMMUNG DER DUALEN BASEN (EULERPARAMETER)

Durch Ableitung der Gleichung (4.16) bezüglich der Basis der Welle folgt

$$\begin{bmatrix} 0 & \vec{n}_{a1} \end{bmatrix} \cdot \dot{\mathbf{p}} = 0.$$

Wird nun die Beziehung (A.16), die den Zusammenhang zwischen der Ableitung der Eulerparameter und der relativen Winkelgeschwindigkeit darstellt, eingesetzt, folgt für die Zwangsbedingung auf Lageebene

$$\vec{n}_{a1} \cdot \left(p_s \vec{E} - \vec{p}_v \right) \cdot \vec{\omega}_{rel} = 0. \quad (4.17)$$

Durch Ausnutzung der Zwangsbedingung (4.16) kann diese Gleichung so umgeformt werden, dass die Symmetrie bezüglich der homokinetischen Ebene erkennbar wird.

$$\left(\vec{A} \left(\vec{u}, \frac{\varphi}{2} \right) \cdot \vec{n}_{a1} \right) \cdot \vec{\omega}_{rel} = 0 \quad (4.18)$$

Da der mit dem Winkel $\frac{\varphi}{2}$ um \vec{u} gedrehte Vektor \vec{n}_{a1} genau senkrecht auf der homokinetischen Ebene steht, muss demnach der relative Drehgeschwindigkeitsvektor $\vec{\omega}_{rel}$ vollständig in der homokinetischen Ebene enthalten sein. Diese Forderung ist direkt mit der Gleichlauf eigenschaft verbunden. Sind die Drehgeschwindigkeiten $\vec{\omega}_1$ und $\vec{\omega}_2$ der beiden Achsen mit diesen ausgerichtet und im Betrag gleich, also

$$\begin{aligned} \vec{\omega}_1 &= \omega \cdot \vec{n}_{a1} \\ \vec{\omega}_2 &= \omega \cdot \vec{n}_{b1}, \end{aligned}$$

dann muss die Differenzdrehgeschwindigkeit in der homokinetischen Ebene liegen, siehe Abb. 4.12.

$$\vec{\omega}_{rel} = \omega \cdot (\vec{n}_{b1} - \vec{n}_{a1})$$

Somit garantiert die Zwangsbedingung (4.18) und damit auch (4.16) die Gleichlauf eigenschaft des Gelenkes.

Aus der Gleichung (4.18) lässt sich sofort die Matrix der gesperrten Richtungen $\vec{\Psi}$ ablesen.²

$$\vec{\Psi} = \vec{A} \left(\vec{u}, \frac{\varphi}{2} \right) \cdot \vec{n}_{a1}$$

Aufgrund der günstigen Darstellung lässt sich daraus leicht die Matrix $\vec{\Phi}$ der freien Richtungen ableiten.

$$\vec{\Phi} = \vec{A} \left(\vec{u}, \frac{\varphi}{2} \right) \cdot [\vec{n}_{a2}, \vec{n}_{a3}]$$

Da die \vec{n}_{ai} ein orthonormales Rechtssystem bilden, gilt außerdem $\vec{\Psi} = \vec{\Phi}$ und $\vec{\Phi} = \vec{\Psi}$. Der Rotationstensor \vec{A}_h , der die Lage der homokinetischen Ebene beschreibt, kann direkt aus den

²Die translatorischen Anteile sind der Übersichtlichkeit halber nicht explizit aufgeführt.

Eulerparametern \mathbf{p} mithilfe der Beziehung

$$\begin{aligned}\vec{\mathbf{A}}_h &= \vec{\mathbf{A}}\left(\vec{\mathbf{u}}, \frac{\varphi}{2}\right) \\ &= \mathbf{E} + \sin \frac{\varphi}{2} \tilde{\mathbf{u}} + \left(1 - \cos \frac{\varphi}{2}\right) \tilde{\mathbf{u}} \tilde{\mathbf{u}} \\ &= \mathbf{E} + \vec{\mathbf{p}}_v + \kappa \vec{\mathbf{p}}_v \vec{\mathbf{p}}_v\end{aligned}$$

und

$$\kappa = \frac{1 - p_s}{\vec{\mathbf{p}}_v \cdot \vec{\mathbf{p}}_v}$$

gewonnen werden. Für parallele Achsen, also $\varphi = 0$, nimmt der Nenner den Wert null an, siehe Gl. (A.13). Eine analytische Untersuchung des Grenzwertes für $\varphi \rightarrow 0$ nach dem Satz von L'Hospital [Heu09] ergibt

$$\lim_{\varphi \rightarrow 0} \kappa = \frac{1}{2}$$

und somit einen wohldefinierten endlichen Wert. Der Rotationsgeschwindigkeitstensor $\vec{\boldsymbol{\omega}}_h$ der homokinetischen Ebene ergibt sich aus den Eulerparametern \mathbf{p} und deren zeitlicher Ableitungen $\dot{\mathbf{p}}$ zu

$$\vec{\boldsymbol{\omega}}_h = \dot{\vec{\mathbf{p}}}_v - \kappa \vec{\mathbf{p}}_v \dot{p}_s + \kappa (\vec{\mathbf{p}}_v \times \dot{\vec{\mathbf{p}}}_v).$$

Der nichtlineare Anteil \mathbf{c}_Φ bei der Berechnung der Beschleunigungen, siehe Gl. (2.52), ergibt sich zu

$$\begin{aligned}\mathbf{c}_\Phi &= \left(\frac{d^O}{dt} \Phi\right) \mathbf{s} \\ &= \begin{bmatrix} \vec{\boldsymbol{\omega}}_h \times \Phi \\ \vec{\mathbf{0}} \end{bmatrix} \mathbf{s}.\end{aligned}$$

DOPPELTES KREUZGELENK

Zwei symmetrisch angeordnete Kreuzgelenke, siehe Abb. 4.14, stellen ein ideales Gleichlaufgelenk dar, sofern die Drehwinkel der äußeren Drehgelenke immer gleich sind und die inneren Drehgelenke keinen Abstand zueinander besitzen [MM08]. Damit fallen die inneren Drehgelenke aufeinander und können als ein einziges aufgefasst werden. Die den Drehachsen $\vec{\mathbf{n}}_{a1}$, $\vec{\mathbf{n}}_b$ und $\vec{\mathbf{n}}_{a2}$ zugehörigen Drehwinkel sollen im Folgenden mit φ_{a1} , φ_b und φ_{a2} bezeichnet werden. Aufgrund der Symmetrie der beiden Kreuzgelenke gilt

$$\vec{\mathbf{n}}_{a2} = \vec{\mathbf{A}}(\vec{\mathbf{n}}_b, \varphi_b) \cdot \vec{\mathbf{n}}_{a1}.$$

Um die Funktion eines Gleichlaufgelenks zu erfüllen, muss zusätzlich die Zwangsbedingung

$$\varphi_{a1} = \varphi_{a2} \tag{4.19}$$

gelten. Aus dieser Anordnung lässt sich der relative Drehgeschwindigkeitsvektor $\vec{\boldsymbol{\omega}}_{rel}$ unter Ausnutzung der Ableitung von Gl. (4.19) direkt bestimmen.

$$\vec{\boldsymbol{\omega}}_{rel} = (\vec{\mathbf{n}}_{a1} + \vec{\mathbf{n}}_{a2}) \dot{\varphi}_{a1} + \vec{\mathbf{n}}_b \dot{\varphi}_b \tag{4.20}$$

Daraus ergeben sich die freien Bewegungsrichtungen Φ , sowie der Vektor s der unabhängigen Gelenkgeschwindigkeiten.

$$\begin{aligned}\Phi &= \begin{bmatrix} \vec{n}_{a1} + \vec{n}_{a2}, & \vec{n}_b \end{bmatrix} \\ s &= \begin{bmatrix} \dot{\varphi}_{a1} \\ \dot{\varphi}_b \end{bmatrix}\end{aligned}$$

Das System der Dualen Basen lässt sich nun wie folgt vervollständigen:

$$\begin{aligned}\Psi &= \begin{bmatrix} \frac{1}{|\vec{n}_{a1} + \vec{n}_{a2}|^2} (\vec{n}_{a1} + \vec{n}_{a2}), & \vec{n}_b \end{bmatrix} \\ \bar{\Psi} &= (\vec{n}_{a1} + \vec{n}_{a2}) \times \vec{n}_b \\ \bar{\Phi} &= \frac{1}{|\bar{\Psi}|^2} \bar{\Psi}.\end{aligned}$$

Der nichtlineare Anteil c_Φ ergibt sich zu

$$\begin{aligned}c_\Phi &= \left(\frac{d^O}{dt} \Phi \right) s \\ &= \begin{bmatrix} (\vec{\omega}_1 \times \vec{n}_{a1} + \vec{\omega}_{a2} \times \vec{n}_{a2}), & \vec{\omega}_b \times \vec{n}_b \\ \vec{0} & \vec{0} \end{bmatrix} s\end{aligned}$$

mit

$$\begin{aligned}\vec{\omega}_b &= \vec{\omega}_1 + \vec{n}_{a1} \dot{\varphi}_{a1} \\ \vec{\omega}_{a2} &= \vec{\omega}_b + \vec{n}_b \dot{\varphi}_b.\end{aligned}$$

Entgegen der Behauptung in [MM08] konnte mithilfe der Dualen Basen eine explizite Beschreibung des doppelten Kreuzgelenks gefunden werden, deren Zwangsbedingungen vollständig eliminiert wurden.

Um sicherzugehen, dass die hier stark verkürzte Rechnung korrekt ist, wurden die Gleichungen nochmals mithilfe der Methode der Subsysteme hergeleitet und verglichen, siehe A.4. Es zeigten sich keine Unterschiede.

VERGLEICH DER RECHENZEITEN

Die hier vorgestellten Beschreibungen der Gleichlaufgelenke wurden unter Nutzung der Modellierungssprache Modelica für die $\mathcal{O}(n)$ -Bibliothek von SimulationX implementiert und verglichen. Dazu wurden die verschiedenen Varianten des Gleichlaufgelenks jeweils einmal einzeln simuliert und zusätzlich als letztes Element in ein Pendel mit zehn Elementen integriert. Abb. 4.15 zeigt die erzielten Simulationszeiten, bezogen auf das explizite doppelte Kreuzgelenk aus dem vorherigen Abschnitt 4.3.4. Die Formulierung mit Eulerparametern stellt auf Lageebene ein differenziell-algebraisches System mit vier Zuständen und zwei Nebenbedingungen auf. Der Modelica Compiler überführt dieses System durch Indexreduktion auf eine gewöhnliche Differentialgleichung. Die Wahl der beiden unabhängigen Zustandsgrößen kann jedoch im Allgemeinen erst zur Laufzeit erfolgen. Daher wird die Methode der Dynamic State Selection angewandt. Diese benötigt jedoch zusätzlichen Rechenaufwand, da zu jedem Zeitschritt geprüft werden muss, ob es eventuell eine günstigere Zustandswahl gibt. In der Beispielrechnung ist es jedoch auch

möglich, eine feste Zustandswahl vorzunehmen, um die Rechenzeit zusätzlich zu reduzieren. Trotzdem rechnet die auf Eulerparametern basierende Formulierung deutlich langsamer als das doppelte Kreuzgelenk. Dies liegt vorwiegend daran, dass aufgrund der speziellen Funktionsweise der Modelica Compiler zu jedem Zeitschritt das nichtlineare Gleichungssystem $\mathbf{p}^T \mathbf{p} - 1 = 0$ nach einem der vier Eulerparameter aus \mathbf{p} aufgelöst werden muss. Zwischen den Formulierungen, basierend auf dem doppelten Kreuzgelenk, zeigt sich deutlich, dass die in dieser Arbeit vorgestellte explizite Version bis zu zweimal schneller als die implizite Formulierung aus [MM08] ist. Darüber hinaus ist die explizite Formulierung uneingeschränkt für die Echtzeitsimulation geeignet, da weder lineare noch nichtlineare Systeme gelöst werden müssen.

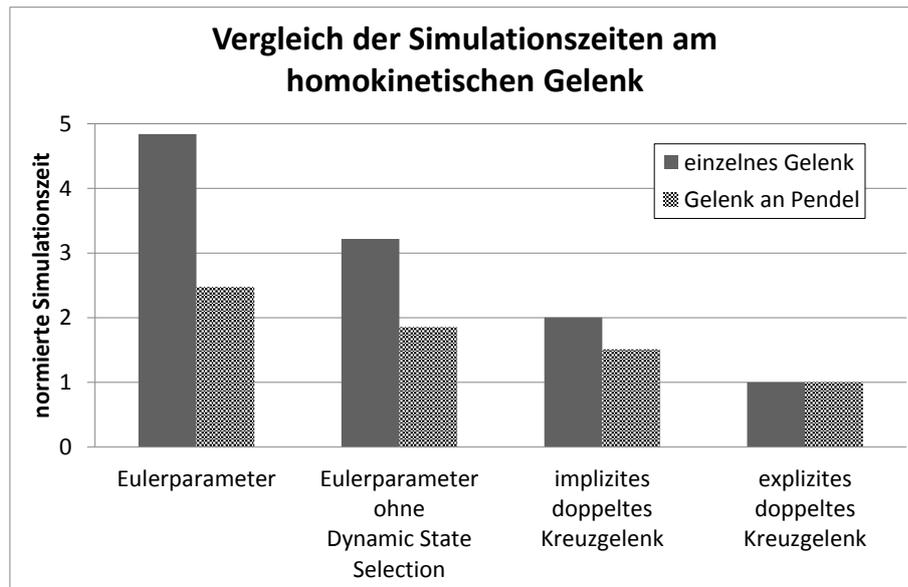


Abbildung 4.15.: Vergleich der Rechenzeiten der verschiedenen Implementierungen des homokinischen Gelenks

5. ZUSAMMENFASSUNG UND AUSBLICK

Die vorliegende Arbeit entstand während der vierjährigen Tätigkeit des Autors an der Professur für Baumaschinen- und Fördertechnik der Technischen Universität Dresden. Sie stellt die im Laufe der wissenschaftlichen Tätigkeit gesammelten Erfahrungen und erarbeiteten Lösungen für eine beschleunigte Simulation komplexer technischer Systeme, wie Baumaschinen sie darstellen, vor. Der Fokus liegt dabei vor allen Dingen auf der Reduzierung der Berechnungszeiten für das mechanische Teilsystem eines domänenübergreifenden Maschinenmodells. Wie in Kapitel 1 erläutert, werden diese vorzugsweise unter Nutzung des objektorientierten Sprachstandards Modelica mit der Methode der Mehrkörpersysteme simuliert.

Es wurde erarbeitet, wie durch die Methode der MKS das zeitliche Verhalten dreidimensionaler mechanischer Systeme berechnet wird. Anders als in der einschlägigen Literatur wurde an dieser Stelle darauf geachtet, die zur Lösung beitragenden Fachbereiche der Mechanik und der linearen Algebra getrennt voneinander zu betrachten. Die Mechanik eines MKS wurde in Kapitel 2 hergeleitet, wohingegen in Kapitel 3 auf die Lösungsalgorithmen, welche auf erstere angewendet werden, eingegangen wurde. Dort hat das erarbeitete mechanische Verständnis des Vorgängerkapitels entscheidend dazu beigetragen, die in der Regel rein mathematisch hergeleiteten Algorithmen mithilfe der Graphentheorie auf eine gemeinsame Basis, nämlich das Gleichungssystem (2.55), zu stellen.

Es wurde gezeigt, dass alle bekannten Algorithmen durch Anwendung der Gauß-Elimination aus diesem hergeleitet werden können. Auf diese Weise war es möglich den Aufwand der verschiedenen Algorithmen miteinander zu vergleichen. Am Beispiel eines planaren Pendels wurde bestätigt, dass $\mathcal{O}(n)$ -Algorithmen für große Systeme rechenzeiteffizienter sind als die $\mathcal{O}(n^3)$ -Algorithmen. Ferner konnte gezeigt werden, dass für dieses Beispiel der explizit formulierte $\mathcal{O}(n)$ -Algorithmus sogar für kleine Systeme am schnellsten rechnet.

In Kapitel 2 wurde bereits gezeigt, dass sich die implizite Bindungsbeschreibung aus der expliziten durch Projektion auf die gesperrten Lastrichtungen ableiten lässt. Dieses Erkenntnis konnte in Kapitel 3 auf die darauf basierenden Lösungsalgorithmen übertragen werden. Die Anwendung von $\mathcal{O}(n^3)$ -Algorithmen erlaubt eine einfache Mischung von explizit und implizit beschriebenen Gelenken. $\mathcal{O}(n)$ -Algorithmen skalieren für große Systeme zwar besser, schränken jedoch bei der Anwendung ein. So lassen sich z.B. implizit und explizit beschriebene Gelenke nicht ohne weiteres zusammenbringen. In Abschnitt 3.2.5 wurde jedoch eine Möglichkeit gefunden, wie selbst implizit beschriebene Gelenke in den expliziten $\mathcal{O}(n)$ -Algorithmus eingebunden werden können. Dies kann sehr wertvoll sein, wenn bestimmte komplexere Bindungen nur in einer impliziten Beschreibung vorliegen.

Sämtliche Aussagen gelten jedoch nur für Systeme ohne kinematische Schleifen. Da jedes Modell einer Baumaschine meist mehrere kinematische Schleifen aufweist, wurde deren Einfluss auf die Berechnungszeiten eines MKS in Abschnitt 3.4 diskutiert. Sowohl anhand einer theoretischen Betrachtung als auch einer praktischen Untersuchung wurden die Auswirkungen hervorgehoben, welche sich durch die Einbeziehung von kinematischen Schleifen im System ergeben. So konnte gezeigt werden, dass sich die Rechengeschwindigkeit des favorisierten $\mathcal{O}(n)$ -Algorithmus dadurch überproportional stark verringert.

In Abschnitt 4.1 wurde erarbeitet, dass innerhalb von Modelica Werkzeugen zwei Strategien verfolgt werden, um eine dreidimensionale Mechanik zu berechnen.

- Die frei verfügbare Modelica.Mechanics.Multibody Bibliothek wird als Grundlage herangezogen und mit dem Tearing Verfahren durch den Compiler mit einem expliziten $\mathcal{O}(n^3)$ -Algorithmus gelöst.
- Eine eigenständige MKS-Bibliothek wird genutzt, deren Elemente direkt die Berechnungsvorschrift des $\mathcal{O}(n)$ -Algorithmus enthalten.

In Bezug auf die Simulation von Baumaschinen ergeben sich damit folgende Nachteile:

- Die Nutzung der MSL führt zu einem Berechnungsaufwand von $\mathcal{O}(n^3)$ und ist daher für große Systeme ungeeignet.
- Die Nutzung von $\mathcal{O}(n)$ -Bibliotheken eignet sich zwar für große Systeme; es ist aber nicht trivial, diese um eigene Modelle komplexerer Körper oder Bindungen wie z.B. dem Gleichlaufgelenk zu erweitern. Die Einbeziehung kinematischer Schleifen oder anderer Zwangsbedingungen führt außerdem zu einer deutlichen Reduzierung der Rechengeschwindigkeit.

Diese beiden Probleme wurden in Kapitel 4 der vorliegenden Arbeit gelöst. Während MKS Bibliotheken mit $\mathcal{O}(n)$ bereits kommerziell verfügbar sind, wurde erstmals vom Autor der explizite $\mathcal{O}(n)$ -Algorithmus direkt in einen Modelica Compiler implementiert, siehe Abschnitt 4.2. Die Umsetzung erfolgte im freizugänglichen OpenModelica Compiler [FAP⁺06]. Am ebenen Pendel konnte gezeigt werden, dass in der Tat der Berechnungsaufwand damit nur noch linear mit der Anzahl der Körper steigt und dieser Ansatz im Vergleich zur reinen numerischen Lösung nur etwa ein Drittel der Zeit benötigt.

Das ebenfalls auf Modelica basierende Simulationswerkzeug SimulationX hat bereits einen effizienten $\mathcal{O}(n)$ -Algorithmus in seinen Bibliothekselementen implementiert. Dessen rechentechnische Vorzüge gehen allerdings bei der Einbindung von kinematischen Schleifen verloren. Ein weiterer Nachteil liegt darin, dass komplexere Körper- und Gelenkkombinationen, selbst von mechanisch versierten Anwendern, nicht ohne Spezialwissen über die Implementierung von $\mathcal{O}(n)$ -Algorithmen der Bibliothek hinzugefügt werden können.

Aus diesem Grund wurde im zweiten Teil der Arbeit die Methode der Subsysteme vorgestellt, mit der es für Anwender möglich wird, die Gleichungen für neue Bibliothekselemente der $\mathcal{O}(n)$ -MKS-Bibliothek selbst herzuleiten. Dies wurde verwendet, um eine gesamte kinematische Schleife als eigenes Subsystem zu implementieren und speziell auf den Einsatzzweck der Simulation von Baumaschinen zu optimieren. Damit war es möglich, am Beispiel des mechanischen Modells eines Verteilermasts einer Betonpumpe die Rechengeschwindigkeit zu verachtfachen. Sogar das repräsentative Modell der Arbeitsausrüstung eines Baggers, das zwar über eine detaillierte Beschreibung der Hydraulik aber über eine geringere Anzahl an kinematischen Schleifen verfügt, konnte doppelt so schnell berechnet werden.

In Abschnitt 4.3.4 wurde mit der Methode der Subsysteme ein weiteres Anwenderproblem behoben. Recherchen des Autors hatten gezeigt, dass es bisher kein gebrauchsfertiges Modell eines Gleichlaufgelenkes gibt, das ohne Zwangsbedingungen auskommt und somit effizient in die $\mathcal{O}(n)$ -MKS-Bibliothek integriert werden kann. Mithilfe der dualen Basen war es möglich, eine explizite Formulierung für dessen Funktionsweise zu erhalten, die etwa doppelt so schnell ausgewertet werden kann, wie die aus der Literatur bekannte Formulierung über das doppelte Kreuzgelenk und einer Zwangsbedingung.

Mit den vorgestellten Lösungen wurden Voraussetzungen dafür geschaffen, Simulationsmodelle von Baumaschinen den in Kapitel 1 vorgestellten Anwendungen zugänglich zu machen. Aus den gesammelten Erkenntnissen lassen sich jedoch noch weitere Verbesserungspotenziale ableiten.

Die in dieser Arbeit vorgestellte Methode der Subsysteme wurde exemplarisch für das ebene Viergelenk implementiert und bewertet. Sie kann jedoch prinzipiell auf alle Formen von kinematischen Schleifen angewandt werden. Speziell die Radaufhängungen von Kraftfahrzeugen stellen aufgrund ihrer komplexen schleifenbehafteten Geometrie ein interessantes Anwendungsgebiet dar.

Mithilfe der Abstraktion der Mehrkörperalgorithmen auf eine Gauß-Elimination eines linearen Gleichungssystems wäre es wünschenswert, in weiterführenden Arbeiten zu untersuchen, inwiefern es möglich ist, Effekte wie z.B. Coulomb'sche Reibung zu berücksichtigen. Deren Einbeziehung würde dazu führen, dass das große schwach besetzte lineare Gleichungssystem (2.55) um einen neuen Gleichungstyp erweitert würde.

Eine andere interessante Erweiterung würde eine effiziente Berücksichtigung von Kontakten innerhalb von Modelica Modellen darstellen. Während elastische Kontakte zwar einfach als Lastelemente implementiert werden können, sorgen sie oft für numerische Schwierigkeiten und in der Folge zu inakzeptablen Rechenzeiten. Hier könnte es lohnenswert sein, einen impulsbasierten Ansatz zu verfolgen. Dabei entsteht ein ähnliches lineares Gleichungssystem, für dessen Lösung die hier vorgestellten Algorithmen und Methoden benutzt werden können.

Um eine weitere Beschleunigung der Simulation zu erreichen, sollte neben der Optimierung der Hydraulik der Einsatz von speziellen numerischen Lösern bis hin zur Co-Simulation untersucht werden. Speziell der Einsatz von Multirate-Verfahren würde es erlauben, schnelle Subsysteme mit anderen Schrittweiten oder gar Lösungsverfahren zu berechnen als langsame Teilsysteme. Da im Falle von Baumaschinen das schnelle Teilsystem meist durch die Hydraulik repräsentiert wird, deren Berechnungsaufwand deutlich unter der der Mechanik liegt, bietet dies ein großes Potenzial zur Rechenzeitreduzierung.

Ein anderer Weg eine weitere Beschleunigung der Simulation zu erreichen, liegt in deren Parallelisierung. Diese sollte über die Ausführung paralleler Zweige des Topologiegraphen, wie z.B. in [JBL02] vorgestellt, hinausgehen, um auch die Berechnung langer Ketten beschleunigen zu können.

LITERATURVERZEICHNIS

- [ÅÄG⁺10] ÅKESSON, J.; ÅRZÉN, K.-E.; GÄFVERT, M.; BERGDAHL, T.; TUMMESCHEIT, H.: Modeling and optimization with Optimica and JModelica.org - Languages and tools for solving large-scale dynamic optimization problems. In: *Computers & Chemical Engineering* 34 (2010), Nr. 11, S. 1737–1749
- [AC03] ANDERSON, K. S.; CRITCHLEY, J. H.: Improved Order-N' Performance Algorithm for the Simulation of Constrained Multi-Rigid-Body Dynamic Systems. In: *Multibody system dynamics* 9 (2003), Nr. 2, S. 185–212
- [ADD04] AMESTOY, P. R.; DAVIS, T. A.; DUFF, I. S.: Algorithm 837: AMD, an approximate minimum degree ordering algorithm. In: *ACM Transactions on Mathematical Software (TOMS)* 30 (2004), Nr. 3, S. 381–388
- [Åma11] ÅMAN, R.: *Methods and Models for Accelerating Dynamic Simulation of Fluid Power Circuits*. Lappeenranta University of Technology, 2011
- [Arm79] ARMSTRONG, W. W.: Recursive solution to the equations of motion of an n-link manipulator. In: *Proc. 5th World Congress on Theory of Machines and Mechanisms* Bd. 2, 1979, S. 1343–1346
- [Ass12] ASSOCIATION, Modelica: *The Modelica Language Specification Version 3.3*. May, 2012
- [Bap10] BAPAT, R. B.: *Graphs and matrices*. London; New York; New Delhi: Springer; Hindustan Book Agency, 2010. – ISBN 9781848829800
- [Bar96] BARAFF, D.: Linear-Time Dynamics using Lagrange Multipliers. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, S. 137–146
- [Bau72] BAUMGARTE, J.: Stabilization of constraints and integrals of motion in dynamical systems. In: *Computer methods in applied mechanics and engineering* 1 (1972), Nr. 1, S. 1–16
- [BEMO02] BRÜCK, D.; ELMQVIST, H.; MATTSSON, S. E.; OLSSON, H.: Dymola for multi-engineering modeling and simulation. In: *Proceedings of Modelica*, 2002
- [BJO86] BRANDL, H.; JOHANNI, R.; OTTER, M.: A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix. In: *IFAC/IFIP/IMACS Symposium on Theory of Robots*, 1986, S. 95–100
- [BJO87] BRANDL, H.; JOHANNI, R.; OTTER, M.: An algorithm for the simulation of multibody systems with kinematic loops. In: *Proceedings of the IFToMM Seventh World Congress on the Theory of Machines and Mechanisms*, 1987, S. 407–411

- [BL09] BEUTLICH, T.; LIEN, R.: Multidisciplinary Simulation of Active Heave Compensators for Offshore Module Handling Systems. Warsaw, Juni 2009
- [BO03] BEATER, P.; OTTER, M.: Multi-Domain Simulation: Mechanics and Hydraulics of an Excavator. In: *Paper presented at the 3rd International Modelica Conference*, 2003
- [Bud90] BUDO, A.: *Theoretische Mechanik*. 12. Aufl. Wiley-Vch, 1990. – ISBN 3326000782
- [Bur10] BURGERMEISTER, B.: *Linear-implizite Zeitintegrationsverfahren für differentiell-algebraische Systeme in der Mehrkörperdynamik*, Martin-Luther-Universität Halle-Wittenberg, Naturwissenschaftliche Fakultät III, Diss., Mai 2010
- [Car00] CARPANZANO, E.: Order reduction of general nonlinear DAE systems by automatic tearing. In: *Mathematical and Computer Modelling of Dynamical Systems* 6 (2000), Nr. 2, S. 145–168
- [CG97] CARPANZANO, E.; GIRELLI, R.: The tearing problem: definition, algorithm and application to generate efficient computational code from DAE systems. In: *Proceedings of 2nd Mathmod Vienna*, 1997
- [Dav04] DAVIS, T. A.: Algorithm 832: UMFPACK V4.3 - an unsymmetric-pattern multifrontal method. In: *ACM Trans. Math. Softw.* 30 (2004), Juni, Nr. 2, S. 196–199. – ISSN 0098–3500
- [Dav06] DAVIS, T. A.: *Direct methods for sparse linear systems*. Bd. 2. Society for Industrial and Applied Mathematics, 2006
- [DER86] DUFF, I. S.; ERISMAN, A. M.; REID, J. K.: *Direct methods for sparse matrices*. Clarendon Press Oxford, 1986
- [Die10] DIESTEL, R.: *Graphentheorie*. Heidelberg: Springer, 2010. – ISBN 9783642149115
- [EMB⁺10] EBERT, F.; MAHN, M.; BERGER, M.; RODIONOW, P.; SCHREIBER, U.: Simulation of Planar Linkages in SimulationX with Modular Library Elements. In: *Proceedings of the 13th ITI Symposium*. Dresden, Germany, 2010
- [EO94] ELMQVIST, H.; OTTER, M.: Methods for tearing systems of equations in object-oriented modeling. In: *ESM* Bd. 94, 1994, S. 1–3
- [ER04] EICHBERGER, A.; RULKA, W.: Process save reduction by macro joint approach: The key to real time and efficient vehicle simulation. In: *Vehicle System Dynamics* 41 (2004), Nr. 5, S. 401–413
- [Erl04] ERLEBEN, K.: *Stable, Robust, and Versatile Multibody Dynamics Animation*, University of Copenhagen, Department of Computer Science, Diss., 2004
- [ESF98] EICH-SOELLNER, E.; FÜHRER, C.: *Numerical methods in multibody dynamics*. Bd. 45. Teubner Stuttgart, 1998
- [FAP⁺06] FRITZSON, P.; ARONSSON, P.; POP, A.; LUNDEVALL, H.; NYSTROM, K.; SALDAMLI, L.; BROMAN, D.; SANDHOLM, A.: OpenModelica-A free open-source environment for system modeling, simulation, and teaching. In: *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, 2006, S. 1588–1595
- [Fea83] FEATHERSTONE, R.: The calculation of robot dynamics using articulated-body inertias. In: *The International Journal of Robotics Research* 2 (1983), Nr. 1, S. 13
- [Fea87] FEATHERSTONE, R.: *Robot Dynamics Algorithm*. Kluwer Academic Publishers, Norwell, MA, 1987
- [Fea07] FEATHERSTONE, R.: *Rigid Body Dynamics Algorithms*. 1. Aufl. Springer, Dezember 2007. – ISBN 0387743146
- [Fre13] FRENKEL, J.: *Entwicklung eines Modelica Compiler BackEnds für große Modelle*. Dresden, TU Dresden, Diss., Juli 2013

- [Fri04] FRITZSON, P.: *Principles of object-oriented modeling and simulation with Modelica 2.1*. Piscataway, NJ: IEEE Press [u.a.], 2004. – ISBN 0471471631
- [Fri11] FRITZSON, P.: *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*. John Wiley & Sons, 2011
- [Heu09] HEUSER, Harro: *Lehrbuch der Analysis. Teil 1*. Auflage: 17., durchges. Aufl. 2009. Wiesbaden: Vieweg+Teubner Verlag, 2009. – ISBN 9783834807779
- [Hun90] HUNT, K. H.: *Kinematic geometry of mechanisms*. Cambridge Univ Press, 1990
- [HW10] HAIRER, E.; WANNER, G.: *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. 1. Aufl. 1996. 2. Druck. Springer, März 2010. – ISBN 3642052207
- [JB93] DE JALON, J. G.; BAYO, E.: *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. 1. Springer, Dezember 1993. – ISBN 0387940960
- [JBL02] JARAMILLO-BOTERO, A.; LORENTE, A. I.: A Unified Formulation for Massively Parallel Rigid Multibody Dynamics of $O(\log 2n)$ Computational Complexity. In: *Journal of Parallel and Distributed Computing* 62 (2002), Juni, Nr. 6, S. 1001–1020. – ISSN 0743–7315
- [KFSJ09] KUNZE, G.; FRENKEL, J.; SCHUBERT, C.; JANKOV, K.: Using Modelica for interactive simulations of technical systems in a virtual reality environment. In: *Proceedings of the 7th International Modelica Conference, Como (Italy)*, 2009
- [KRW⁺14] KOCH, O.; RICHTER, S.; WEBER, J.; SCHUBERT, C.; KUNZE, G.; GRÄTZ, U.; NEUBERT, T.: Real-time models for hardware-in-the-loop simulation of hydraulic drive and control systems. In: *9th International Fluid Power Conference*. Aachen, März 2014
- [KSEL10] KUNZE, G.; SCHUBERT, C.; ESSWEIN, E.; LEHRMANN, S.: Software Architecture for Interactive Simulation of Mobile Machinery. In: *Proceedings of the 1st Commercial Vehicle Technology Symposium (CVT 2010)*. Kaiserslautern: Shaker Verlag, 2010. – ISBN 9783832290405, S. 417–426
- [KSF10] KUNZE, G.; SCHUBERT, C.; FRENKEL, J.: Einsatz der Modellierungssprache MODELICA für die interaktive Simulationen von Baumaschinen in virtuellen Umgebungen. In: *Wissensportal: www.baumaschine.de* (2010), Januar
- [Lac07] LACOURSIÈRE, C.: *Ghosts and Machines: Regularized Variational Methods for Interactive Simulations of Multibodies with Dry Frictional Contacts*, Department of Computing Science, Umea University, Sweden, Diss., Juni 2007
- [LCE10] LEBEDEV, L. P.; CLOUD, M. J.; EREMEYEV, V. A.: *Tensor Analysis with Applications in Mechanics*. World Scientific Pub Co Pte Ltd, Juni 2010. – ISBN 9814313122
- [LNPE92] LUBICH, C.; NOWAK, U.; POHLE, U.; ENGSTLER, C.: *MEXX: Numerical Software for the Integration of Constrained Mechanical Multibody Systems*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1992
- [MM08] MASARATI, P.; MORANDINI, M.: An ideal homokinetic joint formulation for general-purpose multibody real-time simulation. In: *Multibody System Dynamics* 20 (2008), Nr. 3, S. 251–270
- [MMS05] MAGNUS, K.; MÜLLER-SLANY, H. H.: *Grundlagen der Technischen Mechanik*. 7., durchges. und erg. Aufl. 2005. Vieweg+Teubner, Oktober 2005. – ISBN 3835100076
- [Mod13] MODLER, K.-H. *Umdruck zur Vorlesung Getriebelehre*. Oktober 2013
- [MOE00] MATTSSON, S. E.; OLSSON, H.; ELMQVIST, H.: Dynamic selection of states in dynamola. In: *Modelica Workshop 2000 Proceedings*, 2000, S. 61–67
- [MS93] MATTSSON, S. E.; SÖDERLIND, G.: Index reduction in differential-algebraic equations using dummy derivatives. In: *SIAM Journal on Scientific Computing* 14 (1993), S. 677–677

- [Nik05] NIKRAVESH, P. E.: An Overview of Several Formulations for Multibody Dynamics. In: *Product Engineering*. Springer, 2005, S. 189–226
- [OEM03] OTTER, M.; ELMQVIST, H.; MATTSSON, S. E.: The new modelica multibody library. In: *Proceedings of the 3rd International Modelica Conference*, 2003
- [Pan88] PANTELIDES, C. C.: The consistent initialization of differential-algebraic systems. In: *SIAM Journal on Scientific and Statistical Computing* 9 (1988), Nr. 2, S. 213–231
- [PG96] PFEIFFER, F.; GLOCKER, C.: *Multibody Dynamics with Unilateral Contacts*. 1. Aufl. Wiley-VCH, Juni 1996. – ISBN 0471155659
- [Pie11] PIETRUSZKA, W. D.: *Matlab Und Simulink In Der Ingenieurpraxis: Modellbildung, Berechnung und Simulation*. 3. Vieweg+Teubner Verlag, November 2011
- [PVS07] PENNESTRÌ, E.; VALENTINI, P. P.; STEFANELLI, R.: *Optimal Design and Multibody Analysis of Rzeppa Pilot-Lever Joint*. 2007
- [Rei97] REIN, U.: *Effiziente objektorientierte Simulation von Mehrkörpersystemen mit dem rekursiven Formalismus*, Universität Stuttgart, Inst. A für Mechanik, Diss., 1997
- [RS88] ROBERSON, R. E.; SCHWERTASSEK, R.: *Dynamics of Multibody Systems*. 1. Aufl. Springer, April 1988. – ISBN 3540174478
- [RS10] RILL, G.; SCHAEFFER, T.: *Grundlagen und Methodik der Mehrkörpersimulation: mit Anwendungsbeispielen*. 1. Aufl. Vieweg+Teubner, August 2010. – ISBN 3834808881
- [SBH10] SCHRAMM, D.; BARDINI, R.; HILLER, M.: *Modellbildung Und Simulation Der Dynamik Von Kraftfahrzeugen*. Springer, 2010. – ISBN 9783540893158
- [SBK10] SCHUBERT, C.; BEITELSCHMIDT, M.; KUNZE, G.: Handling kinematic loops of mobile machinery in real-time applications Submitted to Proc. In: *Appl. Math. Mech.(PAMM), Karlsruhe* (2010)
- [Sch94] SCHMITZ, T.: *Modellbildung und Simulation der Antriebsdynamik von Personenkraftwagen*. Duesseldorf: VDI-Verl., 1994. – ISBN 3183224127
- [SE11] SCHIEHLEN, W.; EBERHARD, P.: *Technische Dynamik Rechnergestützte Modellierung Mechanischer Systeme im Maschinen- und Fahrzeugbau*. Vieweg + Teubner, 2011. – ISBN 9783834814920
- [SFKB12] SCHUBERT, C.; FRENKEL, J.; KUNZE, G.; BEITELSCHMIDT, M.: Achieving O (n) Complexity for Models from Modelica.Mechanics.MultiBody. In: *Proceedings of the 9th International MODELICA Conference*. München, September 2012. – ISBN 9789175198262
- [SGR11] SCHREIBER, U.; GRÜTZNER, S.; RODIONOW, P.: Flat but Big – The new Planar Mechanics. In: *Proceedings of the 14th ITI Symposium*. Dresden, 2011
- [SH95] STELZLE, W.; HILLER, M.: A comparative study of recursive methods. In: *Archive of applied mechanics* 66 (1995), Nr. 1-2, S. 9–19
- [Sha10] SHABANA, A. A.: *Dynamics of Multibody Systems*. 3. Cambridge University Press, August 2010. – ISBN 0521154227
- [SNK11] SCHUBERT, C.; NEIDHOLD, T.; KUNZE, G.: Experiences with the new FMI Standard Selected applications at Dresden University. In: *Proceedings of the 8th International Modelica Conference*. Dresden, März 2011, S. 94
- [Tas01] TASORA, A. A.: An optimized lagrangian multiplier approach for interactive multibody simulation in kinematic and dynamical digital prototyping. In: *International Symposium on Computer Simulation in Biomechanics*, 2001, S. 4–6
- [UBB+09] UHLIG, A.; BEUTLICH, T.; BLOCHWITZ, T.; KURZBACH, G.; NÄHRING, T.: Modellierung und Simulation mit Modelica in SimulationX. Dresden, März 2009

- [UP98] URBINATI, F.; PENNESTRÌ, E.: Kinematic and Dynamic Analyses of the Tripode Joint. In: *Multibody System Dynamics* 2 (1998), Dezember, Nr. 4, S. 355–367. – ISSN 1384–5640, 1573–272X
- [Wau13] WAURICH, V.: *Untersuchung und Bewertung von Tearing-Algorithmen*, TU Dresden, Diplomarbeit, April 2013
- [WBN06] WEISS, J.; BEUTLICH, T.; NÄHRING, T.: Die neue Mehrkörpermechanik-Bibliothek in SimulationX. In: *Proceedings of the 9th ITI Simulation Workshop*. Dresden, Germany, 2006
- [Weh88] WEHAGE, R. A.: Application of matrix partitioning and recursive projection to O(n) solution of constrained equations of motion. In: *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, v15-2* (1988), S. 221–230
- [WHB81] WEHAGE, R.; HAUG, E. J.; BECK, R. R.: Generalized coordinate partitioning in dynamic analysis of mechanical systems. In: *Dissertation Abstracts International Part B: Science and Engineering* 42 (1981), Nr. 1, S. 1981
- [Woe88] WOERNLE, C.: *Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbedingungen in kinematischen Schleifen mit Anwendung bei der Rückwärts-transformation für Industrieroboter*. VDI-Verlag, 1988
- [Zim12] ZIMMER, D.: A planar mechanical library for teaching modelica. In: *review for the Proceedings of the 9th International Modelica Conference*. München, 2012

A. ANHANG

A.1. GRUNDLAGEN DER TENSORRECHNUNG

Im Rahmen dieser Arbeit wird ein Vektor, bzw. ein Tensor erster Stufe, als ein geometrisches Gebilde in Form einer gerichteten Strecke mit einer bestimmten Länge verstanden. Üblicherweise werden dreidimensionale Vektoren daher als Pfeil dargestellt.

Die Orientierung eines Koordinatensystems B , sei durch drei Basisvektoren $\vec{e}_{B,i}$, $i = 1, 2, 3$ gegeben. Ein beliebiger Vektor \vec{v} kann als Summe der Projektionen auf diese Basisvektoren dargestellt werden, siehe Abb. A.1

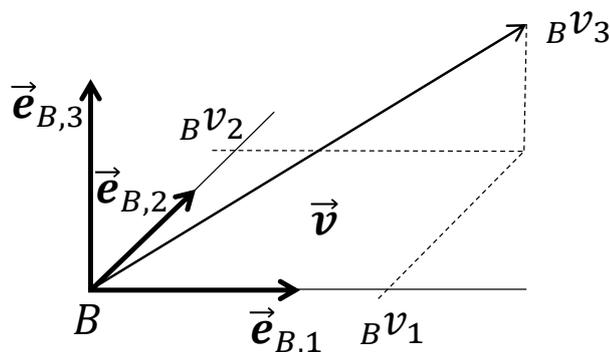


Abbildung A.1.: Projektion des Vektors \vec{v} auf eine Basis $\vec{e}_{B,i}$

$$\vec{v} = \sum_{i=1}^3 Bv_i \vec{e}_{B,i} = Bv_i \vec{e}_{B,i} \quad (\text{A.1})$$

Letztere Schreibweise verwendet die Einstein'sche Summenkonvention und wird Tensorarstellung genannt. Durch Festlegung einer Basis können Vektoren als Spaltenmatrix dargestellt werden

$$B\mathbf{v} = \begin{bmatrix} Bv_1 \\ Bv_2 \\ Bv_3 \end{bmatrix},$$

wobei die Koeffizienten der $\vec{e}_{B,i}$ in die Spalte i eingetragen werden. Diese Bv_i werden als *Koor-*

dirate des Vektors \vec{v} bezüglich der Basis B bezeichnet. Die verwendete Basis wird als Subskript dem Symbol vorangestellt.

Im Falle eines kartesischen Koordinatensystems gilt für das innere bzw. Skalarprodukt zweier Basisvektoren

$$\vec{e}_{B,i} \cdot \vec{e}_{B,j} = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{sonst} \end{cases}. \quad (\text{A.2})$$

Mithilfe dieser Eigenschaft lassen sich die Komponenten eines Vektors wie folgt bestimmen:

$${}^B v_i = \vec{e}_{B,i} \cdot \vec{v}. \quad (\text{A.3})$$

Unter Nutzung der Tensorschreibweise kann das innere Produkt zweier Vektoren \vec{a} und \vec{b} geschrieben werden als

$$\begin{aligned} \vec{a} \cdot \vec{b} &= ({}^B a_i \vec{e}_{B,i}) \cdot ({}^B b_j \vec{e}_{B,j}) \\ &= {}^B a_i \cdot {}^B b_j \cdot (\vec{e}_{B,i} \cdot \vec{e}_{B,j}) \\ &= {}^B a_i \cdot {}^B b_j \cdot \delta_{ij} \\ &= {}^B a_i \cdot {}^B b_i \\ &= {}^B \mathbf{a}^T \cdot {}^B \mathbf{b}. \end{aligned}$$

Somit wurde der Zusammenhang des inneren Produkts zweier Vektoren auf das Matrixprodukt ihrer Komponentendarstellungen zurückgeführt. Die Darstellung als inneres Tensorprodukt hat jedoch den Vorteil, dass sie unabhängig von dem verwendeten Bezugssystem ist und damit die Notation auf das Wesentliche reduziert wird.

Eine Dyade $\vec{\vec{D}}$ kann mithilfe des äußeren bzw. Vektorprodukts \otimes aus zwei Vektoren konstruiert werden.

$$\begin{aligned} \vec{\vec{D}} &= \vec{a} \otimes \vec{b} \\ &= ({}^B a_i \vec{e}_{B,i}) \otimes ({}^C b_j \vec{e}_{C,j}) \\ &= {}^B a_i \cdot {}^C b_j \cdot (\vec{e}_{B,i} \otimes \vec{e}_{C,j}) \\ &= {}^{BC} d_{ij} \cdot (\vec{e}_{B,i} \otimes \vec{e}_{C,j}) \end{aligned}$$

Sie repräsentiert einen Tensor zweiter Stufe mit dem Rang eins. Das Tensorprodukt ist nicht kommutativ und die neun Dyaden $\vec{e}_{B,i} \otimes \vec{e}_{C,j}$ bilden die Basis des Raumes der kartesischen Tensoren.

Jeder Tensor zweiter Stufe $\vec{\vec{T}}$ kann auch als Operator betrachtet werden. So bildet

$$\vec{y} = \vec{\vec{T}} \cdot \vec{x}$$

einen Vektor \vec{x} auf einen anderen Vektor \vec{y} ab. Ein besonderer Tensor ist der Einheitstensor $\vec{\vec{E}}$. Er bildet jeden Vektor auf sich selbst ab

$$\vec{x} = \vec{x} \cdot \vec{\vec{E}} = \vec{\vec{E}} \cdot \vec{x}$$

und ist definiert als

$$\vec{\vec{E}} = \vec{e}_i \otimes \vec{e}_i.$$

In der Tat ergibt

$$\begin{aligned}
 \vec{\mathbf{E}} \cdot \vec{\mathbf{x}} &= (\vec{\mathbf{e}}_i \otimes \vec{\mathbf{e}}_i) \cdot (x_j \vec{\mathbf{e}}_j) \\
 &= x_j \vec{\mathbf{e}}_i (\vec{\mathbf{e}}_i \cdot \vec{\mathbf{e}}_j) \\
 &= x_j \vec{\mathbf{e}}_i \delta_{ij} \\
 &= x_j \vec{\mathbf{e}}_j \\
 &= \vec{\mathbf{x}}.
 \end{aligned}$$

Analog zur Darstellung eines Vektors als Spaltenmatrix, kann ein Tensor zweiter Stufe als Matrix dargestellt werden. Dazu müssen zwei Basen B und C gewählt werden. Der Tensor kann nun als

$$\vec{\mathbf{T}} = BCt_{ij} \cdot (\vec{\mathbf{e}}_{B,i} \otimes \vec{\mathbf{e}}_{C,j})$$

mit der zugehörigen Matrixdarstellung

$$\mathbf{T}_{BC} = \begin{bmatrix} BCt_{11} & BCt_{12} & BCt_{13} \\ BCt_{21} & BCt_{22} & BCt_{23} \\ BCt_{31} & BCt_{32} & BCt_{33} \end{bmatrix}$$

angeschrieben werden, wobei die Koeffizienten der $\vec{\mathbf{e}}_{B,i} \otimes \vec{\mathbf{e}}_{C,j}$ in die Zeile i und Spalte j eingetragen werden. Die Koordinaten des Tensors $\vec{\mathbf{T}}$ können direkt über die Vorschrift

$$BCt_{ij} = \vec{\mathbf{e}}_{B,i} \cdot \vec{\mathbf{T}} \cdot \vec{\mathbf{e}}_{C,j}$$

ermittelt werden.

Das Kreuzprodukt zweier Vektoren

$$\vec{\mathbf{c}} = \vec{\mathbf{a}} \times \vec{\mathbf{b}}$$

kann ebenfalls als Operator aufgefasst werden, indem der Term $\vec{\mathbf{a}} \times$ einen Vektor $\vec{\mathbf{b}}$ auf einen Vektor $\vec{\mathbf{c}}$ abbildet. Entsprechend den Eigenschaften des Kreuzprodukts, definiert $\vec{\mathbf{a}} \times$ eine lineare Abbildung und es existiert ein Tensor $\vec{\mathbf{a}}$ zweiter Stufe, der die selben Abbildungseigenschaften besitzt.

$$\vec{\mathbf{c}} = \vec{\mathbf{a}} \times \vec{\mathbf{b}} = \vec{\mathbf{a}} \cdot \vec{\mathbf{b}} \quad (\text{A.4})$$

Soll ein Vektor in Komponentendarstellung ${}_B \mathbf{v}$ bezüglich einer anderen Basis C dargestellt werden, so kann dies mithilfe einer Rotationsmatrix \mathbf{A}_{CB} geschehen.

$${}_C \mathbf{v} = \mathbf{A}_{CB} \cdot {}_B \mathbf{v}$$

In Komponentenschreibweise kann die Multiplikation mit der gleichen Matrix \mathbf{A}_{CB} jedoch auch als Drehung des Vektors ${}_B \mathbf{v}$ interpretiert werden. Die Tensoren zu diesen beiden Operationen unterscheiden sich allerdings voneinander. Im Falle eines Basiswechsel wird der Vektor $\vec{\mathbf{v}}$ nicht verändert. Damit handelt es sich bei der Rotationsmatrix \mathbf{A}_{CB} um die Matrixdarstellung des Einheitstensors $\vec{\mathbf{E}}$ bezüglich zweier verschiedener Basen B und C . Soll der Vektor $\vec{\mathbf{v}}$ gedreht werden, so kann jede Drehung gemäß dem EULERSchen Theorem immer als eine normierte Drehachse $\vec{\mathbf{u}}$ zusammen mit einem Drehwinkel φ dargestellt werden. Der zugehörige Drehtensor

$\vec{\vec{A}}$ kann über die RODRIGUES-Formel berechnet werden.

$$\vec{\vec{A}} = \vec{\vec{E}} + \sin \varphi \vec{\vec{u}} + (1 - \cos \varphi) \vec{\vec{u}} \vec{\vec{u}}$$

Die Rotationsmatrix ergibt sich in diesem Fall als Matrixschreibweise des Tensors $\vec{\vec{A}}$ bezüglich zweier gleicher Basen. Die entgegengesetzte Rotation wird durch den transponierten Tensor repräsentiert, was wiederum einem Vorzeichenwechsel des Drehwinkels entspricht. Deshalb muss

$$\vec{\vec{A}}^T \cdot \vec{\vec{A}} = \vec{\vec{A}} \cdot \vec{\vec{A}}^T = \vec{\vec{E}} \quad (\text{A.5})$$

gelten. Soll die zeitliche Änderung eines Vektors $\frac{d}{dt} \vec{v}$ bestimmt werden, so kann diese Operation direkt auf Gleichung (A.1) angewandt werden und es ergibt sich

$$\frac{d}{dt} \vec{v} = {}_B \dot{v}_i \vec{e}_{B,i} + {}_B v_i \frac{d}{dt} \vec{e}_{B,i}. \quad (\text{A.6})$$

Um auf die Tensorarstellung zu gelangen, wird eine Darstellung von $\frac{d}{dt} \vec{e}_{B,i}$ bezüglich der Basis $\vec{e}_{B,i}$ benötigt. Die Änderung der Basisvektoren $\vec{e}_{B,i}$ hängt jedoch davon ab, von welcher Basis aus sie beobachtet werden. Sobald die Basis eines Beobachters festgelegt ist, kann eine *ruhende Basis* O definiert werden, als eine Basis deren Basisvektoren aus Sicht eines Beobachters konstant bezüglich der Zeit sind. Um Verwechslungen vorzubeugen, sollte immer kenntlich gemacht werden, bezüglich welcher Basis eine Ableitung ausgeführt wird. Daher wird die ruhende Basis immer als hochgestellter Index an den Ableitungsoperator angefügt. Folglich erfüllen die Basisvektoren einer ruhenden Basis O die Eigenschaft

$$\frac{d^O}{dt} \vec{e}_{O,i} = \vec{0}$$

Jede andere Basis, die bezüglich O keine Rotation ausführt, bildet ebenso eine ruhende Basis. Translatorische Bewegungen bezüglich O sind zulässig, da sich zwar der Ursprung der Basisvektoren aber nicht deren Ausrichtung ändert.

Werden nun die Basisvektoren $\vec{e}_{B,i}$ mithilfe eines Rotationstensors $\vec{\vec{A}}_{BO}(t)$ bezüglich der ruhenden Basis O beschrieben als

$$\vec{e}_{B,i} = \vec{\vec{A}}_{BO}(t) \cdot \vec{e}_{O,i}, \quad (\text{A.7})$$

so lässt sich eine Darstellung der zeitlichen Änderung der $\vec{e}_{B,i}$ berechnen.

$$\begin{aligned} \frac{d^O}{dt} \vec{e}_{B,i} &= \frac{d^O}{dt} \left(\vec{\vec{A}}_{BO} \cdot \vec{e}_{O,i} \right) \\ &= \frac{d^O}{dt} \vec{\vec{A}}_{BO} \cdot \vec{e}_{O,i} + \vec{\vec{A}}_{BO} \cdot \frac{d^O}{dt} \vec{e}_{O,i} \\ &= \frac{d^O}{dt} \vec{\vec{A}}_{BO} \cdot \left(\vec{\vec{A}}_{BO}^T \cdot \vec{e}_{B,i} \right) \\ &= \left(\frac{d^O}{dt} \vec{\vec{A}}_{BO} \cdot \vec{\vec{A}}_{BO}^T \right) \cdot \vec{e}_{B,i} \\ &= \vec{\vec{\Omega}}_{BO} \cdot \vec{e}_{B,i} \end{aligned}$$

Dabei wurde die folgende Abkürzung eingeführt:

$$\overset{\rightrightarrows}{\Omega}_{BO} = \frac{d^O \overset{\rightrightarrows}{A}_{BO}}{dt} \cdot \overset{\rightrightarrows}{A}_{BO}. \quad (\text{A.8})$$

Leitet man Gl. (A.5) nach der Zeit t ab, so sieht man, dass es sich bei $\overset{\rightrightarrows}{\Omega}_{BO}$ um einen antisymmetrischen Tensor handelt. Zu jedem antisymmetrischen Tensor $\overset{\rightrightarrows}{\Omega}_{BO}$ lässt sich ein Vektor $\vec{\omega}_{BO}$ mit der Eigenschaft

$$\overset{\rightrightarrows}{\Omega}_{BO} \cdot \vec{v} = \vec{\omega}_{BO} \times \vec{v}$$

finden [LCE10]. Der Vektor kann bezüglich einer kartesischen Basis B dargestellt werden als

$$\begin{aligned} \vec{\omega}_{BO} &= {}_{BB}\Omega_{BO,32}\vec{e}_{B,1} + {}_{BB}\Omega_{OB,13}\vec{e}_{B,2} + {}_{BB}\Omega_{OB,21}\vec{e}_{B,3} \\ &= \text{vect} \left(\overset{\rightrightarrows}{\Omega}_{BO} \right) \end{aligned} \quad (\text{A.9})$$

bzw.

$$\overset{\rightrightarrows}{\Omega}_{BO} = \vec{\omega}_{BO}.$$

Die Funktion $\text{vect}(\cdot)$ definiert folglich die Umkehrfunktion zum Tilde-Operator aus Gl. (A.4). Somit gilt

$$\frac{d^O}{dt} \vec{e}_{B,i} = \vec{\omega}_{BO} \times \vec{e}_{B,i}, \quad (\text{A.10})$$

wobei $\frac{d^O}{dt}$ anzeigt, dass die Basisvektoren $\vec{e}_{O,i}$ als ruhend angenommen wurden. Die geometrische Interpretation der Gleichung (A.10) zeigt, dass es sich bei $\vec{\omega}_{BO}$ um den Vektor der relativen Drehgeschwindigkeit zwischen den Basen B und O handelt. Durch Einsetzen der Gl. (A.10) in (A.6) ergibt sich die CORIOLIS Gleichung

$$\frac{d^O}{dt} \vec{v} = {}^B\dot{\vec{v}} + \vec{\omega}_{BO} \times \vec{v} \quad (\text{A.11})$$

unter Verwendung des Symbols ${}^B\dot{\vec{v}}$ für die komponentenweise Ableitung bezüglich der Basis B

$${}^B\dot{\vec{v}} = {}_B\dot{v}_i \vec{e}_{B,i} = \frac{d^B}{dt} \vec{v}.$$

Diese Größe beschreibt die zeitliche Änderung eines Vektors \vec{v} wie sie von einem Beobachter wahrgenommen würde, der sich mit dem Koordinatensystem B mitbewegt [RS88]. Sie wird daher als relative oder auch beobachtete Ableitung bezeichnet.

Der große Vorteil der Darstellung von Gleichung (A.11) mit Tensoren ist, dass sie dann unabhängig von einem Bezugssystem gilt. Sie kann jedoch leicht in die Matrixschreibweise durch Anwendung von Gleichung (A.3) überführt werden und man erhält für beliebige Basen B und C

$${}_C \left(\frac{d^O}{dt} \vec{v} \right) = {}_C ({}^B\dot{\vec{v}}) + {}_C \omega_{BO} \times {}_C \vec{v} \quad (\text{A.12})$$

mit ${}_C ({}^B\dot{\vec{v}})$ als Komponentendarstellung des Vektors ${}^B\dot{\vec{v}}$ bezüglich der Basis C .

Eine Drehung kann alternativ über vier Eulerparameter bzw. als Einheitsquaternion \mathbf{p} dargestellt werden. Dies setzt sich aus einem skalaren Teil p_s und einem vektoriellen Teil \vec{p}_v zusammen und

ist definiert als

$$\mathbf{p} = \begin{bmatrix} p_s \\ \vec{\mathbf{p}}_v \end{bmatrix} = \begin{bmatrix} \cos \frac{\varphi}{2} \\ \sin \frac{\varphi}{2} \cdot \vec{\mathbf{u}} \end{bmatrix}. \quad (\text{A.13})$$

Die Größe \mathbf{p} stellt eine Spaltenmatrix dar, deren Einträge Tensoren sind und es gelten die üblichen Rechenregeln für Matrizen.

Aufgrund der Forderung, dass $\vec{\mathbf{u}}$ ein Einheitsvektor sein muss, ergibt sich die Zwangsbedingung

$$\mathbf{p}^T \cdot \mathbf{p} = 1. \quad (\text{A.14})$$

mithilfe der Eulerparameter lässt sich der Drehtensor noch einfacher darstellen als

$$\vec{\mathbf{A}} = \vec{\mathbf{E}} + 2p_s \vec{\mathbf{p}}_v + 2\vec{\mathbf{p}}_v \vec{\mathbf{p}}_v.$$

Die Winkelgeschwindigkeit ω_{BO} ergibt sich durch Anwendung von Gl. (A.8) zu

$$\vec{\omega}_{BO} = \vec{\mathbf{u}}\dot{\varphi} + \sin \varphi \dot{\vec{\mathbf{u}}} + (1 - \cos \varphi) (\vec{\mathbf{u}} \times \dot{\vec{\mathbf{u}}}).$$

Die gleiche Beziehung kann mithilfe der Eulerparameter, analog zu [RS10], kurz geschrieben werden als

$$\vec{\omega}_{BO} = 2\mathbf{P} \cdot \dot{\mathbf{p}} \quad (\text{A.15})$$

mit

$$\mathbf{P} = \begin{bmatrix} -\vec{\mathbf{p}}_v, & p_s \vec{\mathbf{E}} + \vec{\mathbf{p}}_v \end{bmatrix}.$$

Unter Ausnutzung der zeitlichen Ableitung von (A.14) lässt sich der Zusammenhang zwischen der Änderung der Eulerparameter und dem Drehgeschwindigkeitsvektor bestimmen.

$$\dot{\mathbf{p}} = \frac{1}{2} \mathbf{P}^T \cdot \vec{\omega}_{BO} \quad (\text{A.16})$$

Die inverse Drehung lässt sich durch das konjugierte Quaternion $\bar{\mathbf{p}}$ beschreiben.

$$\bar{\mathbf{p}} = \begin{bmatrix} p_s \\ -\vec{\mathbf{p}}_v \end{bmatrix} \quad (\text{A.17})$$

Dies kann als Änderung des Vorzeichens des Drehwinkels φ anschaulich interpretiert werden, da es einen Vorzeichenwechsel des Koeffizienten $\sin \frac{\varphi}{2}$ verursacht. Sollen zwei Drehungen \mathbf{p}_1 und \mathbf{p}_2 nacheinander ausgeführt werden, so wird dies durch das Produkt der beiden Quaternionen ausgedrückt. Es ist definiert als

$$\mathbf{p}_{12} = \mathbf{p}_1 \star \mathbf{p}_2 = \begin{bmatrix} p_{1,s} \cdot p_{2,s} - \vec{\mathbf{p}}_{1,v} \cdot \vec{\mathbf{p}}_{2,v} \\ p_{1,s} \cdot \vec{\mathbf{p}}_{2,v} + p_{2,s} \cdot \vec{\mathbf{p}}_{1,v} + \vec{\mathbf{p}}_{1,v} \times \vec{\mathbf{p}}_{2,v} \end{bmatrix}. \quad (\text{A.18})$$

Aufgrund des Kreuzproduktes ist die Multiplikation von Quaternionen, ebenso wie die Drehungen im dreidimensionalen Raum, nicht kommutativ.

A.2. DUALE BASIS EINER BINDUNG

A.2.1. EINFÜHRUNG

Für das spätere Arbeiten mit den Bewegungsgleichungen eines Mehrkörpersystems ist es hilfreich analog zu [RS88] zwei spezielle Basen ϕ_i , $i = 1 \dots 6$ für die Kinemate \mathbf{y}_{VN}^V und ψ_i , $i = 1 \dots 6$ für die Gelenklasten \mathbf{l}_N einzuführen. Die gewählten Basisvektoren sollen die Eigenschaft einer *dualen Basis* erfüllen, d. h.

$$\phi_i^T \cdot \psi_j = \delta_{ij}. \quad (\text{A.19})$$

Werden die ϕ_i so gewählt, dass sie selbst eine orthonormale Basis bilden, so fallen ϕ_i und die ψ_i , aufgrund von Bedingung (A.2), zusammen. Die ϕ_i und ψ_i sollen als Basis zur Darstellung der \mathbf{y}_{VN}^V bzw. \mathbf{l}_N mit den Koordinaten s_i und λ_i benutzt werden.

$$\mathbf{y}_{VN}^V = \sum_{i=1}^6 s_i \phi_i \quad (\text{A.20})$$

$$\mathbf{l}_N = \sum_{i=1}^6 \lambda_i \psi_i \quad (\text{A.21})$$

Um eine kompaktere Darstellung zu erlangen, werden die ersten n_f und die verbleibenden Basisvektoren zu Matrizen zusammengefasst.

$$\begin{aligned} \Phi &= [\phi_i], i = 1 \dots n_f \\ \bar{\Phi} &= [\phi_i], i = (n_f + 1) \dots 6 \\ \Psi &= [\psi_i], i = 1 \dots n_f \\ \bar{\Psi} &= [\psi_i], i = (n_f + 1) \dots 6 \end{aligned}$$

Die Komponentendarstellung lautet mit den entsprechenden Koordinatenvektoren dann

$$\mathbf{y}_{VN}^V = \Phi \mathbf{s} + \bar{\Phi} \bar{\mathbf{s}} \quad (\text{A.22})$$

$$\mathbf{l}_N = \Psi \boldsymbol{\lambda} + \bar{\Psi} \bar{\boldsymbol{\lambda}}. \quad (\text{A.23})$$

Die beiden Basen sollen nun so gewählt werden, dass sie mit Blick auf die Gleichungen (2.33) und (2.37) günstige Eigenschaften besitzen. Dazu werden die Vektoren aus Φ so gewählt, dass sie eine Basis des Nullraums $\ker(\mathbf{W}^T)$ bilden, d.h.

$$\mathbf{W}^T \Phi = \mathbf{0}. \quad (\text{A.24})$$

Mithilfe dieser Bedingung vereinfacht sich Gl. (A.22) eingesetzt in (2.32) zu

$$\mathbf{W}^T \bar{\Phi} \cdot \bar{\mathbf{s}} = \boldsymbol{\zeta}'. \quad (\text{A.25})$$

Aufgrund der Definition der Φ muss die Matrix $\mathbf{W}^T \bar{\Phi}$ invertierbar sein und man erhält somit eine Gleichung zur Berechnung der Koordinaten $\bar{\mathbf{s}}$. Für die verbleibenden Koordinaten \mathbf{s} ergeben sich vorerst keine Bedingungen. Dadurch ergibt sich eine Interpretation der Vektoren ϕ_i . Werden die zugehörigen Koordinaten s_i durch die Bindung nicht eingeschränkt, so werden die Bewegungen entlang der zugehörigen ϕ_i ebenfalls nicht eingeschränkt und man spricht von den *freien Richtungen*. Ergibt sich durch Gleichung (A.25) eine Zeitfunktion für ein s_i so gehört das

zugehörige ϕ_i zu den *kinematisch angeregten* Richtungen. Alle verbleibenden ϕ_i , deren Koordinaten sich durch Gl. (A.25) zu Null ergeben, stellen die *gesperrten Richtungen* dar. In diesem Kontext spricht [RS88] auch von den *Modes of Motion*.

Aufgrund des mit Gl. (2.18) bereits angedeuteten speziellen Zusammenhangs zwischen Lasten und Geschwindigkeiten, lässt sich ein ähnliches Ergebnis für die duale Basis der Kräfte gewinnen. Die Vektoren in $\bar{\Psi}$ werden so gewählt, dass sie eine Basis des Nullraums $\ker(\mathbf{J}^T)$ bilden. Damit muss gelten

$$\mathbf{J}^T \bar{\Psi} = \mathbf{0} \quad (\text{A.26})$$

und damit vereinfacht sich Gl. (A.23) in (2.37) eingesetzt zu

$$\mathbf{J}^T \Psi \cdot \lambda = \eta'. \quad (\text{A.27})$$

Aufgrund der Definition der $\bar{\Psi}$ muss die Matrix $\mathbf{J}^T \Psi$ invertierbar sein und man erhält somit eine Gleichung zur Berechnung der Koordinaten λ . Für die verbleibenden Koordinaten $\bar{\lambda}$ ergibt sich keine Bedingung und sie sind vorerst frei wählbar.

A.2.2. WAHL DER DUALEN BASEN

Die Wahl der dualen Basen ϕ_i und ψ_i ist nicht eindeutig und hängt maßgeblich davon ab, in welcher Form die Bindung beschrieben ist.

Im Falle der explizit formulierten Bindung (2.33) ist die Matrix \mathbf{J} bekannt. Um der Bedingung (A.24) zu genügen, wird Φ so gewählt, dass es eine Basis des $\text{span}(\mathbf{J})$ bildet, z.B. $\Phi = \mathbf{J}$. Bedingung (A.26) wird erfüllt, indem $\bar{\Psi}$ eine Basis des $\ker(\mathbf{J}^T)$ bildet. Die verbleibenden Basisvektoren werden entsprechend den Gl. (A.19) gewählt.

Ist die Bindung hingegen implizit formuliert (2.33), so ist die Matrix \mathbf{W} bekannt. Werden $\bar{\Psi}$ als Basis des $\text{span}(\mathbf{W})$, z.B. $\bar{\Psi} = \mathbf{W}$, und Φ als eine Basis des $\ker(\mathbf{W}^T)$ gewählt, so sind die Bedingungen (A.24) und (A.26) erfüllt. Die verbleibenden Basisvektoren werden dann entsprechend der Bedingung (A.19) gewählt.

Der vorgestellte Ansatz ist etwas allgemeiner gefasst als in [RS88] vorgeschlagen und lässt mehr Freiheiten zu. Diese können beispielsweise genutzt werden, um die ϕ_i als orthonormale Basis zu konstruieren, da dies im Allgemeinen zu einer effizienteren Implementierung führt.

Somit ist es immer möglich, zu einer gegebenen Bindungsbeschreibung die dualen Basen ϕ_i und ψ_i zu bestimmen. Daher wird im Folgenden davon ausgegangen, dass sie bekannt sind. Damit können die Gleichungen (2.32) und (2.33) auf die neuen Basen

$$\mathbf{y}_{VN}^V = \Phi \mathbf{s} + \xi \quad (\text{A.28})$$

$$\bar{\Psi}^T \cdot \mathbf{y}_{VN}^V = \zeta \quad (\text{A.29})$$

umgeschrieben werden, wobei die neuen generalisierten Geschwindigkeiten \mathbf{s} Linearkombinationen der alten \mathbf{s}' darstellen. ζ geht ebenfalls über eine passende Linearkombination aus ζ' hervor. Mit dieser Festlegung lässt sich aufgrund der Definition der dualen Basis (A.19) die Gleichung (A.25) noch weiter vereinfachen

$$\bar{\mathbf{s}} = \zeta \quad (\text{A.30})$$

Ebenso kann Gl. (2.37) umgeschrieben werden.

$$\Phi^T \mathbf{l}_N = \eta \quad (\text{A.31})$$

Diese vereinfacht sich mit der Darstellung (A.21) für \mathbf{l}_N zu

$$\boldsymbol{\lambda} = \boldsymbol{\eta} = \boldsymbol{\Phi}^T \mathbf{l}_{N,A} \quad (\text{A.32})$$

und zeigt, dass die $\boldsymbol{\lambda}$ ausschließlich durch die eingepprägten Kräfte $\mathbf{l}_{N,A}$ verursacht werden. Demnach parametrisieren $\bar{\boldsymbol{\Psi}}$ und $\bar{\boldsymbol{\lambda}}$ die Zwangskräfte $\mathbf{l}_{N,C}$. In der Literatur werden die $\bar{\boldsymbol{\lambda}}$ auch Lagrange-Multiplikatoren genannt.

Die Eigenschaft der dualen Basis kann außerdem genutzt werden, um Koordinaten zu extrahieren. Wird z.B. Gl. (A.28) mit $\bar{\boldsymbol{\Psi}}^T$ multipliziert, so folgt

$$\bar{\boldsymbol{\Psi}}^T \cdot \mathbf{y}_{VN}^V = \bar{\boldsymbol{\Psi}}^T \boldsymbol{\xi} = \bar{\mathbf{s}}.$$

Dieselbe Gleichung auf $\boldsymbol{\Psi}^T$ projiziert ergibt

$$\boldsymbol{\Psi}^T \cdot \mathbf{y}_{VN}^V = \mathbf{s}.$$

Diese Eigenschaft erweist sich in Kapitel 3 als sehr nützlich.

A.2.3. KINEMATISCHE DIFFERENZIALGLEICHUNG

Um später das Differenzialgleichungssystem des gesamten Mehrkörpersystems lösen zu können, muss der differenzielle Zusammenhang zwischen den Ableitungen der Gelenkkordinaten $\dot{\mathbf{q}}$ und den Gelenkgeschwindigkeiten \mathbf{s} gefunden werden. Dazu wird Gl. (2.27) nach der Zeit abgeleitet.

$$\begin{aligned} \frac{d^V}{dt} \mathbf{z}_{VN} &= \frac{\partial \mathbf{z}_{VN}}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{z}_{VN}}{\partial t} \\ &= \mathbf{J}_q \cdot \dot{\mathbf{q}} + \boldsymbol{\xi}_q \end{aligned}$$

Der differenzielle Zusammenhang zwischen der Ableitung der Pose $\frac{d^V}{dt} \mathbf{z}_{VN}$ und der Kinematik \mathbf{y}_{VN}^V ist durch Gl. (2.6) und der Bezug zu \mathbf{s} durch Gl. (2.45) gegeben. Setzt man all diese Beziehungen ineinander ein, erhält man

$$\begin{aligned} \mathbf{H} \cdot \frac{d^V}{dt} \mathbf{z}_{VN} &= \boldsymbol{\Phi} \cdot \mathbf{s} + \boldsymbol{\xi} \\ \bar{\boldsymbol{\Psi}} \cdot \mathbf{H} \cdot (\mathbf{J}_q \cdot \dot{\mathbf{q}} + \boldsymbol{\xi}_q) &= \mathbf{s} \\ \mathbf{H}_q \cdot \dot{\mathbf{q}} + \mathbf{h}_q &= \mathbf{s}. \end{aligned}$$

Im Falle voneinander abhängiger \mathbf{q} , kann \mathbf{H}_q nicht direkt invertiert werden und obige Gl. muss zusammen mit den Nebenbedingungen $\boldsymbol{\chi}(\mathbf{q})$, siehe Gl. (2.28),

$$\mathbf{H}_q \cdot \dot{\mathbf{q}} + \mathbf{h}_q = \mathbf{s} \quad (\text{A.33a})$$

$$\mathbf{0} = \boldsymbol{\chi}(\mathbf{q}) \quad (\text{A.33b})$$

als differenzial-algebraisches System (DAE) gelöst werden.

A.3. HERLEITUNG DES SUBSYSTEMS DES VIERGELENKS

A.3.1. REDUKTION DES SYSTEMS

Ausgehend von dem Gleichungssystem (4.11) soll an dieser Stelle die symbolische Gauß-Elimination ausgeführt werden.

Zuerst wird die Bindungsgleichung des Gelenks D und dessen Prinzip der virtuellen Leistung in die implizite Form, durch Nutzung der dualen Basis $[\Psi_D, \bar{\Psi}_D]$, überführt. Es folgt zum Einen eine Beziehung zur Berechnung der Gelenkbeschleunigung

$$\dot{s}_D = \Psi_D^T (\mathbf{G}_{D2}\mathbf{x}_2 - \mathbf{G}_{D3}\mathbf{x}_3 - \mathbf{c}_D)$$

und zum Anderen der Wert der Koordinaten λ_D der freien Lastrichtungen des Gelenks D

$$\lambda_D = 0.$$

Durch Entfernen dieser Beziehungen aus dem Gleichungssystem, ergibt sich folgendes reduziertes System

$$\begin{bmatrix} l_3 & l_2 & l_1 & x_3 & x_2 & x_1 & \dot{s}_3 & \dot{s}_2 & \dot{s}_1 & \bar{\lambda}_D & l_0 \\ I_3 & -E & o & o & M_3 & o & o & o & o & G_{D3}^T \bar{\Psi}_D & o \\ I_2 & o & -E & o & o & M_2 & o & o & o & -G_{D2}^T \bar{\Psi}_D & o \\ I_1 & o & G_{21}^T & -E & o & o & M_1 & o & o & o & o \\ A_3 & o & o & o & -E & o & \Phi_3 & o & o & o & o \\ A_2 & o & o & o & o & -E & G_{21} & o & \Phi_2 & o & o \\ A_1 & o & o & o & o & o & -E & o & o & \Phi_1 & o \\ D_3 & \Phi_3^T & o & o & o & o & o & o & o & o & o \\ D_2 & o & \Phi_2^T & o & o & o & o & o & o & o & o \\ D_1 & o & o & \Phi_1^T & o & o & o & o & o & o & o \\ \bar{A}_D & o & o & o & \bar{\Psi}_D^T G_{D3} & -\bar{\Psi}_D^T G_{D2} & o & o & o & o & o \\ I_0 & G_{30}^T & o & G_{10}^T & o & o & o & o & o & o & -E \end{bmatrix} \underbrace{\begin{bmatrix} l_{3,0} & l_{3,x} \\ l_{2,0} & l_{2,x} \\ l_{3,0} & l_{3,x} \\ x_{3,0} & x_{3,x} \\ x_{2,0} & x_{2,x} \\ x_{1,0} & x_{1,x} \\ \dot{s}_{3,0} & \dot{s}_{3,x} \\ \dot{s}_{2,0} & \dot{s}_{2,x} \\ \dot{s}_{1,0} & \dot{s}_{1,x} \\ \bar{\lambda}_{D,0} & \bar{\lambda}_{D,x} \\ l_0^a & M_0^a \end{bmatrix}}_Y = \begin{bmatrix} 1 & x_0 \\ l_{3,ext} & 0 \\ l_{2,ext} & 0 \\ l_{1,ext} & 0 \\ -c_3 & G_{30} \\ -c_2 & 0 \\ -c_1 & G_{10} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\bar{\Psi}_D^T c_D & 0 \\ 0 & 0 \end{bmatrix}$$

Nach Reduktion der ersten drei Spalten l_3 , l_2 und l_1 folgt

$$\begin{bmatrix} l_3 & l_2 & l_1 & x_3 & x_2 & x_1 & \dot{s}_3 & \dot{s}_2 & \dot{s}_1 & \bar{\lambda}_D & l_0 \\ I_3 & -E & o & o & M_3 & o & o & o & o & G_{D3}^T \bar{\Psi}_D & o \\ I_2 & o & -E & o & o & M_2 & o & o & o & -G_{D2}^T \bar{\Psi}_D & o \\ I_1 & o & o & -E & o & G_{21}^T M_2 & M_1 & o & o & -G_{D1}^T \bar{\Psi}_D & o \\ A_3 & o & o & o & -E & o & o & \Phi_3 & o & o & o \\ A_2 & o & o & o & o & -E & G_{21} & o & \Phi_2 & o & o \\ A_1 & o & o & o & o & o & -E & o & o & \Phi_1 & o \\ D_3 & o & o & o & \Phi_3^T M_3 & o & o & o & o & \Psi_{D3} & o \\ D_2 & o & o & o & o & \Phi_2^T M_2 & o & o & o & \Psi_{D2} & o \\ D_1 & o & o & o & o & \Phi_1^T G_{21}^T M_2 & \Phi_1^T M_1 & o & o & \Psi_{D1} & o \\ \bar{A}_D & o & o & o & \bar{\Psi}_D^T G_{D3} & -\bar{\Psi}_D^T G_{D2} & o & o & o & o & o \\ I_0 & o & o & o & G_{30}^T M_3 & G_{20}^T M_2 & G_{10}^T M_1 & o & o & o & -E \end{bmatrix} Y = \begin{bmatrix} 1 & x_0 \\ l_{3,ext} & 0 \\ l_{2,ext} & 0 \\ l_{12,ext} & 0 \\ -c_3 & G_{30} \\ -c_2 & 0 \\ -c_1 & G_{10} \\ \Phi_3^T l_{3,ext} & 0 \\ \Phi_2^T l_{2,ext} & 0 \\ \Phi_1^T l_{12,ext} & 0 \\ -\bar{\Psi}_D^T c_D & 0 \\ l_{0,ext} & 0 \end{bmatrix}$$

mit

$$\begin{aligned}
 \mathbf{G}_{D1} &= \mathbf{G}_{D2}\mathbf{G}_{21} \\
 \mathbf{G}_{20} &= \mathbf{G}_{21}\mathbf{G}_{10} \\
 \mathbf{l}_{12,ext} &= \mathbf{l}_{1,ext} + \mathbf{G}_{21}^T \mathbf{l}_{2,ext} \\
 \Psi_{D3} &= \Phi_3^T \mathbf{G}_{D3}^T \bar{\Psi}_D \\
 \Psi_{D2} &= -\Phi_2^T \mathbf{G}_{D2}^T \bar{\Psi}_D \\
 \Psi_{D1} &= -\Phi_1^T \mathbf{G}_{D1}^T \bar{\Psi}_D \\
 \mathbf{l}_{0,ext} &= \mathbf{G}_{30}^T \mathbf{l}_{3,ext} + \mathbf{G}_{10}^T \mathbf{l}_{12,ext}.
 \end{aligned}$$

Durch Reduktion der nächsten drei Spalten \mathbf{x}_3 , \mathbf{x}_2 und \mathbf{x}_1 ergibt sich

$$\left[\begin{array}{c|cccccc|cccc|c} & \mathbf{l}_3 & \mathbf{l}_2 & \mathbf{l}_1 & \mathbf{x}_3 & \mathbf{x}_2 & \mathbf{x}_1 & \dot{s}_3 & \dot{s}_2 & \dot{s}_1 & \bar{\lambda}_D & \mathbf{l}_0 \\ \hline \mathbf{I}_3 & -\mathbf{E} & \circ & \circ & \mathbf{M}_3 & \circ & \circ & \circ & \circ & \circ & \mathbf{G}_{D3}^T \bar{\Psi}_D & \circ \\ \mathbf{I}_2 & \circ & -\mathbf{E} & \circ & \circ & \mathbf{M}_2 & \circ & \circ & \circ & \circ & -\mathbf{G}_{D2}^T \bar{\Psi}_D & \circ \\ \mathbf{I}_1 & \circ & \circ & -\mathbf{E} & \circ & \mathbf{G}_{21}^T \mathbf{M}_2 & \mathbf{M}_1 & \circ & \circ & \circ & -\mathbf{G}_{D1}^T \bar{\Psi}_D & \circ \\ \mathbf{A}_3 & \circ & \circ & \circ & -\mathbf{E} & \circ & \circ & \Phi_3 & \circ & \circ & \circ & \circ \\ \mathbf{A}_2 & \circ & \circ & \circ & \circ & -\mathbf{E} & \mathbf{G}_{21} & \circ & \Phi_2 & \circ & \circ & \circ \\ \mathbf{A}_1 & \circ & \circ & \circ & \circ & \circ & -\mathbf{E} & \circ & \circ & \Phi_1 & \circ & \circ \\ \hline \mathbf{D}_3 & \circ & \circ & \circ & \circ & \circ & \circ & \hat{\mathbf{M}}_{33} & \circ & \circ & \Psi_{D3} & \circ \\ \mathbf{D}_2 & \circ & \hat{\mathbf{M}}_{22} & \hat{\mathbf{M}}_{21} & \Psi_{D2} & \circ \\ \mathbf{D}_1 & \circ & \hat{\mathbf{M}}_{21}^T & \hat{\mathbf{M}}_{21} & \Psi_{D1} & \circ \\ \bar{\mathbf{A}}_D & \circ & \circ & \circ & \circ & \circ & \circ & \Psi_{D3}^T & \Psi_{D2}^T & \Psi_{D1}^T & \circ & \circ \\ \hline \mathbf{I}_0 & \circ & \circ & \circ & \circ & \circ & \circ & \Gamma_3^T & \Gamma_2^T & \Gamma_1^T & \circ & -\mathbf{E} \end{array} \right] \mathbf{Y} = \left[\begin{array}{c|c} 1 & \mathbf{x}_0 \\ \hline \mathbf{l}_{3,ext} & \mathbf{0} \\ \mathbf{l}_{2,ext} & \mathbf{0} \\ \mathbf{l}_{12,ext} & \mathbf{0} \\ -\mathbf{c}_3 & \mathbf{G}_{30} \\ -\mathbf{c}_2 & \mathbf{0} \\ -\mathbf{c}_1 & \mathbf{G}_{10} \\ \Phi_3^T \bar{\mathbf{l}}_{3,ext} & \Gamma_3 \\ \Phi_2^T \bar{\mathbf{l}}_{2,ext} & \Gamma_2 \\ \Phi_1^T \bar{\mathbf{l}}_{12,ext} & \Gamma_1 \\ -\bar{\Psi}_D^T \bar{\mathbf{c}}_D & \mathbf{0} \\ \bar{\mathbf{l}}_{0,ext} & \bar{\mathbf{M}}_0 \end{array} \right] \quad (\text{A.34})$$

mit

$$\begin{aligned}
 \bar{\mathbf{M}}_3 &= \mathbf{M}_3 \\
 \bar{\mathbf{M}}_2 &= \mathbf{M}_2 \\
 \bar{\mathbf{M}}_1 &= \mathbf{M}_1 + \mathbf{G}_{21}^T \bar{\mathbf{M}}_2 \mathbf{G}_{21} \\
 \bar{\mathbf{M}}_0 &= \mathbf{G}_{30}^T \bar{\mathbf{M}}_3 \mathbf{G}_{30} + \mathbf{G}_{10}^T \bar{\mathbf{M}}_1 \mathbf{G}_{10} \\
 \hat{\mathbf{M}}_{33} &= \Phi_3^T \bar{\mathbf{M}}_3 \Phi_3 \\
 \hat{\mathbf{M}}_{22} &= \Phi_2^T \bar{\mathbf{M}}_2 \Phi_2 \\
 \hat{\mathbf{M}}_{21} &= \Phi_2^T \bar{\mathbf{M}}_2 \mathbf{G}_{21} \Phi_1 \\
 \hat{\mathbf{M}}_{11} &= \Phi_1^T \bar{\mathbf{M}}_1 \Phi_1
 \end{aligned}$$

und

$$\begin{aligned}
 \bar{\mathbf{c}}_1 &= \mathbf{c}_1 \\
 \bar{\mathbf{c}}_2 &= \mathbf{c}_2 + \mathbf{G}_{21}\bar{\mathbf{c}}_1 \\
 \bar{\mathbf{c}}_3 &= \mathbf{c}_3 \\
 \bar{\mathbf{c}}_D &= \mathbf{c}_D + \mathbf{G}_{D3}\mathbf{c}_3 - \mathbf{G}_{D2}\mathbf{c}_2 - \mathbf{G}_{D1}\mathbf{c}_1 \\
 \bar{\mathbf{l}}_{3,ext} &= \mathbf{l}_{3,ext} - \mathbf{M}_3\bar{\mathbf{c}}_3 \\
 \bar{\mathbf{l}}_{2,ext} &= \mathbf{l}_{2,ext} - \mathbf{M}_2\bar{\mathbf{c}}_2 \\
 \bar{\mathbf{l}}_{1,ext} &= \mathbf{l}_{1,ext} - \mathbf{M}_1\bar{\mathbf{c}}_1 \\
 \bar{\mathbf{l}}_{12,ext} &= \bar{\mathbf{l}}_{1,ext} + \mathbf{G}_{21}^T\bar{\mathbf{l}}_{2,ext} \\
 \bar{\mathbf{l}}_{0,ext} &= \mathbf{G}_{30}^T\bar{\mathbf{l}}_{3,ext} + \mathbf{G}_{10}^T\bar{\mathbf{l}}_{12,ext}
 \end{aligned}$$

sowie

$$\begin{aligned}
 \mathbf{G}_{D0} &= \mathbf{G}_{D3}\mathbf{G}_{30} \\
 \mathbf{\Gamma}_3 &= \mathbf{\Phi}_3^T\bar{\mathbf{M}}_3\mathbf{G}_{30} \\
 \mathbf{\Gamma}_2 &= \mathbf{\Phi}_2^T\bar{\mathbf{M}}_2\mathbf{G}_{20} \\
 \mathbf{\Gamma}_1 &= \mathbf{\Phi}_1^T\bar{\mathbf{M}}_1\mathbf{G}_{10}.
 \end{aligned}$$

Das so entstandene Gleichungssystem entspricht dem expliziten $\mathcal{O}(n^3)$ Algorithmus, siehe Abschnitt 3.2.1, erweitert um die kinematische Schleife sowie der Abhängigkeit von \mathbf{x}_0 .

Das verbleibende Gleichungssystem kann nun entweder numerisch oder symbolisch gelöst werden. Im Folgenden sollen beide Möglichkeiten betrachtet werden.

A.3.2. NUMERISCHE LÖSUNG

Aus der letzten Zeile der Gleichung (A.34) lassen sich folgende zwei Gleichungen für die gesuchten Größen \mathbf{l}_0^a und \mathbf{M}_0^a ableiten

$$\begin{aligned}
 \mathbf{l}_0^a &= \mathbf{\Gamma}^T\mathbf{x} - \bar{\mathbf{l}}_{0,ext} \\
 \mathbf{M}_0^a &= \mathbf{\Gamma}^T\mathbf{X} - \bar{\mathbf{M}}_0
 \end{aligned}$$

mit

$$\mathbf{x} = \begin{bmatrix} \dot{s}_{3,0} \\ \dot{s}_{2,0} \\ \dot{s}_{1,0} \\ \bar{\lambda}_{D,0} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \dot{s}_{3,x} \\ \dot{s}_{2,x} \\ \dot{s}_{1,x} \\ \bar{\lambda}_{D,x} \end{bmatrix}$$

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}_3 \\ \mathbf{\Gamma}_2 \\ \mathbf{\Gamma}_1 \\ \mathbf{0} \end{bmatrix}.$$

Die Größen \mathbf{x} und \mathbf{X} ergeben sich als Lösung der linearen Gleichungssysteme

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{A.35a}$$

$$\mathbf{A}\mathbf{X} = \mathbf{\Gamma} \tag{A.35b}$$

mit

$$\mathbf{A} = \begin{bmatrix} \hat{M}_{33} & \circ & \circ & \Psi_{D3} \\ \circ & \hat{M}_{22} & \hat{M}_{21} & \Psi_{D2} \\ \circ & \hat{M}_{21}^T & \hat{M}_{11} & \Psi_{D1} \\ \Psi_{D3}^T & \Psi_{D2}^T & \Psi_{D1}^T & \circ \end{bmatrix} \tag{A.36}$$

$$\mathbf{b} = \begin{bmatrix} \Phi_3^T \bar{l}_{3,ext} \\ \Phi_2^T \bar{l}_{2,ext} \\ \Phi_1^T \bar{l}_{12,ext} \\ -\bar{\Psi}_D^T \bar{c}_D \end{bmatrix}.$$

Es bietet sich an, die beiden Gleichungssysteme (A.35a) und (A.35b) gemeinsam zu lösen. In diesem Fall muss die Matrix \mathbf{A} nur einmal faktorisiert werden. Die Rücksubstitution muss danach für jede Spalte der rechten Seiten \mathbf{b} und $\mathbf{\Gamma}$ einmal ausgeführt werden.

A.3.3. SYMBOLISCHE LÖSUNG

Anstelle das System (A.35) numerisch zu lösen, kann es ebenso gut symbolisch gelöst werden. Dies hat folgende Vorteile:

1. Es erfolgt kein Aufruf eines numerischen Lösers, der zusätzlichen Overhead verursacht.
2. Die verbleibenden Nullen im System können ausgenutzt werden.
3. Zwischenergebnisse können zur Lösung des linearen Gleichungssystems auf Geschwindigkeitsebene verwendet werden.

Das zu lösende Gleichungssystem auf Geschwindigkeitsebene ergibt sich aus den Bindungsgleichungen auf Geschwindigkeitsebene und lautet

$$\begin{bmatrix} \Psi_{D3}^T & \Psi_{D2}^T & \Psi_{D1}^T \end{bmatrix} \begin{bmatrix} s_3 \\ s_2 \\ s_1 \end{bmatrix} = \mathbf{0}.$$

Findet nun eine Partitionierung der Gelenkgeschwindigkeiten in die unabhängigen $\dot{u} = s_1$ und abhängigen $\dot{v} = [s_2, s_3]^T$ statt, so muss obiges System wie folgt betrachtet werden:

$$\Psi_{D32}^T \dot{v} = -\Psi_{D1}^T \dot{u} \quad (\text{A.37})$$

mit

$$\Psi_{D32} = \begin{bmatrix} \Psi_{D3} \\ \Psi_{D2} \end{bmatrix}.$$

Dieses Gleichungssystem ist jedoch auch auf Beschleunigungsebene enthalten. Nach einer erneuten Permutierung des Gleichungssystems (A.35) ergibt sich die Form

$$\begin{bmatrix} \Psi_{D32} & \tilde{M}_{32} & \tilde{M}_{321} \\ \circ & \Psi_{D32}^T & \Psi_{D1}^T \\ \Psi_{D1} & \tilde{M}_{321}^T & \hat{M}_{11} \end{bmatrix} \begin{bmatrix} \bar{\lambda}_{D,0} & \bar{\lambda}_{D,x} \\ \dot{s}_{32,0} & \dot{s}_{32,x} \\ \dot{s}_{1,0} & \dot{s}_{1,x} \end{bmatrix} = \begin{bmatrix} \tau_{32} & \Gamma_{32} \\ \gamma & \mathbf{0} \\ \tau_1 & \Gamma_1 \end{bmatrix}$$

mit

$$\begin{aligned} \dot{s}_{32,0} &= \begin{bmatrix} \dot{s}_{3,0} \\ \dot{s}_{2,0} \end{bmatrix} \\ \dot{s}_{32,x} &= \begin{bmatrix} \dot{s}_{3,x} \\ \dot{s}_{2,x} \end{bmatrix} \\ \gamma &= -\bar{\Psi}_D^T \bar{c}_D \\ \tau_1 &= \Phi_1^T \bar{l}_{12,ext} \\ \tau_{32} &= \begin{bmatrix} \Phi_3^T \bar{l}_{3,ext} \\ \Phi_2^T \bar{l}_{2,ext} \end{bmatrix} \\ \Gamma_{32} &= \begin{bmatrix} \Gamma_3 \\ \Gamma_2 \end{bmatrix} \\ \tilde{M}_{32} &= \begin{bmatrix} \hat{M}_{33} & \circ \\ \circ & \hat{M}_{22} \end{bmatrix} \\ \tilde{M}_{321} &= \begin{bmatrix} \circ \\ \hat{M}_{21} \end{bmatrix}. \end{aligned}$$

Nun kann mit der Gauß-Elimination in gewohnter Weise begonnen werden

$$\begin{bmatrix} \Psi_{D32} & \tilde{M}_{32} & \tilde{M}_{321} \\ \circ & \Psi_{D32}^T & \Psi_{D1}^T \\ \circ & \tilde{M}_{321}^T & \hat{M}_{11} \end{bmatrix} \begin{bmatrix} \bar{\lambda}_{D,0} & \bar{\lambda}_{D,x} \\ \dot{s}_{32,0} & \dot{s}_{32,x} \\ \dot{s}_{1,0} & \dot{s}_{1,x} \end{bmatrix} = \begin{bmatrix} \tau_{32} & \Gamma_{32} \\ \gamma & \mathbf{0} \\ \tau'_1 & \Gamma'_1 \end{bmatrix}$$

mit

$$\tilde{\mathbf{M}}_{321}'^T = \tilde{\mathbf{M}}_{321}^T - \Psi_{D1} \Psi_{D32}^{-1} \tilde{\mathbf{M}}_{32} \quad (\text{A.38a})$$

$$\tilde{\mathbf{M}}_{11}' = \hat{\mathbf{M}}_{11} - \Psi_{D1} \Psi_{D32}^{-1} \tilde{\mathbf{M}}_{321} \quad (\text{A.38b})$$

$$\tau_1' = \tau_1 - \Psi_{D1} \Psi_{D32}^{-1} \tau_{32} \quad (\text{A.38c})$$

$$\Gamma_1' = \Gamma_1 - \Psi_{D1} \Psi_{D32}^{-1} \Gamma_{32}. \quad (\text{A.38d})$$

Die Form nach dem letzten Schritt lautet

$$\begin{bmatrix} \Psi_{D32} & \tilde{\mathbf{M}}_{32} & \tilde{\mathbf{M}}_{321} \\ \circ & \Psi_{D32}^T & \Psi_{D1}^T \\ \circ & \circ & \tilde{\mathbf{M}}_{11}'' \end{bmatrix} \begin{bmatrix} \bar{\lambda}_{D,0} & \bar{\lambda}_{D,x} \\ \dot{s}_{32,0} & \dot{s}_{32,x} \\ \dot{s}_{1,0} & \dot{s}_{1,x} \end{bmatrix} = \begin{bmatrix} \tau_{32} & \Gamma_{32} \\ \gamma & \mathbf{0} \\ \tau_1'' & \Gamma_1' \end{bmatrix}$$

mit

$$\tilde{\mathbf{M}}_{11}'' = \tilde{\mathbf{M}}_{11}' - \mathbf{M}_{321}'^T \Psi_{D32}^{-T} \Psi_{D1}^T \quad (\text{A.39a})$$

$$\tau_1'' = \tau_1' - \mathbf{M}_{321}'^T \Psi_{D32}^{-T} \gamma. \quad (\text{A.39b})$$

Die Matrix Ψ_{D32} ist im Falle des analytischen Viergelenks lediglich von der Dimension 2×2 . Somit kann die Inverse mit wenigen Operationen mithilfe der Vorschrift

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

berechnet werden. Die Inverse Ψ_{D32}^{-1} wird sowohl in den Gleichungen (A.38) benötigt, als auch in transponierter Form $\Psi_{D32}^{-T} = \left(\Psi_{D32}^{-1}\right)^T$ in den Gleichungen (A.39) und zusätzlich zur Lösung des linearen Gleichungssystems (A.37) auf Geschwindigkeitsebene.

A.4. HOMOKINETISCHES GELENK ALS SUBSYSTEM

Wird das homokinetische Gelenk als doppeltes Kreuzgelenk betrachtet, so ergibt sich eine Abfolge von drei Drehgelenken $i = 1, 2, 3$, die der Zwangsbedingung $\varphi_1 = \varphi_2$ unterliegen. Die Verbindung der drei Drehgelenke erfolge mit masselosen Stäben der Länge null und somit $\mathbf{M} = \mathbf{0}$. Die Bewegungsgleichungen der Verbindungskörper degenerieren somit zu einfachen Lastbilanzen $\mathbf{l}_{i-1} = \mathbf{l}_i$. Da die Verbindungskörper die Länge null besitzen, vereinfachen sich zusätzlich die Transformationsmatrizen zu Einheitsmatrizen, also $\mathbf{G}_{i,i-1} = \mathbf{E}$. Somit lauten die Bindungsgleichungen der drei Drehgelenke

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \Phi_i \dot{s}_i + \mathbf{c}_i,$$

wobei die äußeren Drehachsen senkrecht auf der inneren stehen, also $\Phi_2^T \Phi_i = 0$, $i = 1, 3$. Das Prinzip der virtuellen Leistung muss für die beiden äußeren Drehgelenke um ein Zwangsmoment $\bar{\lambda}$ erweitert werden und lautet somit

$$\begin{aligned} \Phi_1^T \mathbf{l}_1 &= \bar{\lambda} \\ \Phi_2^T \mathbf{l}_2 &= 0 \\ \Phi_3^T \mathbf{l}_3 &= -\bar{\lambda}. \end{aligned}$$

Die zugehörige Zwangsbedingung lautet auf Beschleunigungsebene

$$\dot{s}_1 = \dot{s}_3.$$

Werden all diese Gleichungen in das Schema von Gl. (4.7) gebracht, folgt nach entsprechender Sortierung

$$\begin{bmatrix} -\mathbf{E} & \circ & \circ & \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ \\ \circ & 1 & \circ & \circ & \circ & \circ & \circ & -1 & \circ \\ \Phi_3^T & \circ & 1 & \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \Phi_3 & \circ & -\mathbf{E} & \mathbf{E} & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & -\mathbf{E} & \mathbf{E} & \Phi_2 & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & -\mathbf{E} & \circ & \Phi_1 & \circ \\ \Phi_2^T & \circ \\ \Phi_1^T & \circ & -1 & \circ & \circ & \circ & \circ & \circ & \circ \\ \mathbf{E} & \circ & -\mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{l}_0 \\ \dot{s}_{3,0} \\ \bar{\lambda}_0 \\ \mathbf{x}_{3,0} \\ \mathbf{x}_{2,0} \\ \mathbf{x}_{1,0} \\ \dot{s}_{2,0} \\ \dot{s}_{1,0} \\ \mathbf{l}_0^a \\ \mathbf{M}_0^a \end{bmatrix} = \begin{bmatrix} -\mathbf{l}_3^a & \mathbf{0} \\ 0 & \mathbf{0} \\ 0 & \mathbf{0} \\ -\mathbf{c}_3 & \mathbf{0} \\ -\mathbf{c}_2 & \mathbf{0} \\ -\mathbf{c}_1 & \mathbf{E} \\ 0 & \mathbf{0} \\ 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Nach den ersten drei Schritten der Gauß-Elimination ergibt sich

$$\begin{bmatrix} -\mathbf{E} & \circ & \circ & \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ \\ \circ & 1 & \circ & \circ & \circ & \circ & \circ & -1 & \circ \\ \circ & \circ & 1 & \Phi_3^T \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & -\mathbf{E} & \mathbf{E} & \circ & \circ & \Phi_3 & \circ \\ \circ & \circ & \circ & \circ & -\mathbf{E} & \mathbf{E} & \Phi_2 & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & -\mathbf{E} & \circ & \Phi_1 & \circ \\ \circ & \circ & \circ & \Phi_2^T \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & (\Phi_1 + \Phi_3)^T \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \mathbf{M}_3^a & \circ & \circ & \circ & \circ & -\mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{l}_0 \\ \dot{s}_{3,0} \\ \bar{\lambda}_0 \\ \mathbf{x}_{3,0} \\ \mathbf{x}_{2,0} \\ \mathbf{x}_{1,0} \\ \dot{s}_{2,0} \\ \dot{s}_{1,0} \\ \mathbf{l}_0^a \\ \mathbf{M}_0^a \end{bmatrix} = \begin{bmatrix} -\mathbf{l}_3^a & \mathbf{0} \\ 0 & \mathbf{0} \\ -\Phi_3^T \mathbf{l}_3^a & \mathbf{0} \\ -\mathbf{c}_3 & \mathbf{0} \\ -\mathbf{c}_2 & \mathbf{0} \\ -\mathbf{c}_1 & \mathbf{E} \\ -\Phi_2^T \mathbf{l}_3^a & \mathbf{0} \\ -(\Phi_1 + \Phi_3)^T \mathbf{l}_3^a & \mathbf{0} \\ -\mathbf{l}_3^a & \mathbf{0} \end{bmatrix}.$$

Werden die nächsten drei Schritte ausgeführt, kann mit

$$\begin{aligned} \mathbf{c} &= \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3 \\ \Phi_{13} &= \Phi_1 + \Phi_3 \end{aligned}$$

$$\begin{bmatrix} -\mathbf{E} & \circ & \circ & \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & 1 & \circ & \circ & \circ & \circ & \circ & \circ & -1 & \circ \\ \circ & \circ & 1 & \Phi_3^T \mathbf{M}_3^a & \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & -\mathbf{E} & \mathbf{E} & \circ & \circ & \circ & \Phi_3 & \circ \\ \circ & \circ & \circ & \circ & -\mathbf{E} & \mathbf{E} & \Phi_2 & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & -\mathbf{E} & \circ & \circ & \Phi_1 & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ & \Phi_2^T \mathbf{M}_3^a \Phi_2 & \Phi_2^T \mathbf{M}_3^a \Phi_{13} & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ & \Phi_{13}^T \mathbf{M}_3^a \Phi_2 & \Phi_{13}^T \mathbf{M}_3^a \Phi_{13} & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ & \mathbf{M}_3^a \Phi_2 & \mathbf{M}_3^a \Phi_{13} & -\mathbf{E} & \circ \end{bmatrix} \begin{bmatrix} l_0 \\ \dot{s}_{3,0} \\ \bar{\lambda}_0 \\ x_{3,0} \\ x_{2,0} \\ x_{1,0} \\ \dot{s}_{2,0} \\ \dot{s}_{1,0} \\ l_0^a \\ \mathbf{M}_0^a \end{bmatrix} = \begin{bmatrix} -l_3^a \\ 0 \\ -\Phi_3^T l_3^a \\ -c_3 \\ -c_2 \\ -c_1 \\ -\Phi_2^T (l_3^a + \mathbf{M}_3^a \mathbf{c}) \\ -\Phi_{13}^T (l_3^a + \mathbf{M}_3^a \mathbf{c}) \\ -l_3^a - \mathbf{M}_3^a \mathbf{c} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{E} \\ \Phi_2^T \mathbf{M}_3^a \\ \Phi_{13}^T \mathbf{M}_3^a \\ \mathbf{M}_3^a \end{bmatrix}$$

ermittelt werden. Wird nun

$$\Phi = \begin{bmatrix} \Phi_2 \\ \Phi_{13} \end{bmatrix}$$

$$\dot{s} = \begin{bmatrix} \dot{s}_2 \\ \dot{s}_1 \end{bmatrix}$$

definiert und der Einfachheit halber nur noch die untersten vier Gleichungen und Variablen betrachtet, so ergibt sich

$$\begin{bmatrix} \Phi^T \mathbf{M}_3^a \Phi & \circ \\ \mathbf{M}_3^a \Phi & -\mathbf{E} \end{bmatrix} \begin{bmatrix} \dot{s}_0 \\ l_0^a \end{bmatrix} = \begin{bmatrix} -\Phi^T (l_3^a + \mathbf{M}_3^a \mathbf{c}) \\ -l_3^a - \mathbf{M}_3^a \mathbf{c} \end{bmatrix} \begin{bmatrix} \Phi^T \mathbf{M}_3^a \\ \mathbf{M}_3^a \end{bmatrix}.$$

Dieses System lässt sich leicht lösen mit $\hat{\mathbf{M}}_3 = \Phi^T \mathbf{M}_3^a \Phi$

$$\begin{bmatrix} \hat{\mathbf{M}}_3 & \circ \\ \circ & -\mathbf{E} \end{bmatrix} \begin{bmatrix} \dot{s}_0 \\ l_0^a \end{bmatrix} = \begin{bmatrix} -\Phi^T (l_3^a + \mathbf{M}_3^a \mathbf{c}) \\ -l_3^a - \mathbf{M}_3^a \mathbf{c} + \mathbf{M}_3^a \Phi \hat{\mathbf{M}}_3^{-1} \Phi^T (l_3^a + \mathbf{M}_3^a \mathbf{c}) \end{bmatrix} \begin{bmatrix} \Phi^T \mathbf{M}_3^a \\ \mathbf{M}_3^a - (\Phi^T \mathbf{M}_3^a)^T \hat{\mathbf{M}}_3^{-1} \Phi^T \mathbf{M}_3^a \end{bmatrix}.$$

Aus der letzten Zeile lässt sich nun ablesen

$$\mathbf{M}_0^a = \mathbf{M}_3^a - (\Phi^T \mathbf{M}_3^a)^T \hat{\mathbf{M}}_3^{-1} \Phi^T \mathbf{M}_3^a$$

$$l_0^a = -l_3^a - \mathbf{M}_0^a \mathbf{c} + \mathbf{M}_3^a \Phi \hat{\mathbf{M}}_3^{-1} \Phi^T l_3^a.$$

Dies und alle anderen Gleichungen entsprechen exakt denen aus Algorithmus 3.3. Folglich verhält sich das homokinetische Gelenk wie ein klassisches Gelenk mit

$$\Phi = \begin{bmatrix} \Phi_2 \\ \Phi_1 + \Phi_3 \end{bmatrix}$$

$$\dot{s} = \begin{bmatrix} \dot{s}_2 \\ \dot{s}_1 \end{bmatrix}$$

$$\mathbf{c} = \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3.$$

Aus Sicht der Dualen Basen war dieses Ergebnis im Nachgang jedoch zu erwarten.