

Ein Beitrag zum ganzheitlichen Engineering von elektrischen Servoantriebssystemen

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation
von

Dipl.-Ing. Maik Lorch

Tag der mündlichen Prüfung: 18.10.2013
Hauptreferent: Prof. Dr.-Ing. habil. Georg Bretthauer
Korreferent: Prof. Dr.-Ing. Albert Albers

Vorwort

Selbstverständlich konnte meine Arbeit nur entstehen, weil ich von vielen Menschen unterstützt und gefördert worden bin.

Allen jenen Menschen aus Familie, Freundes- und Kollegenkreis gilt mein ganz persönlich angetragener Dank.

Ein besonderer Dank gilt zudem meinem Doktorvater Herrn Prof. Dr.-Ing. habil. Georg Bretthauer für die jahrelange Unterstützung besonders in Form von zahlreichen fachlichen Anregungen und für die Gewährung eines großen wissenschaftlichen Freiraums, welcher die Erarbeitung der vorliegenden Arbeit erst ermöglichte.

Ebenfalls bedanken möchte ich mich bei Herrn Prof. Dr.-Ing. Albert Albers für die Übernahme des Korreferats und die Begutachtung meiner Arbeit.

Schließlich bedanke ich mich bei allen motivierten und kreativen Helfern, die Ihren Beitrag mit anfänglich nicht immer offensichtlich Erfolg versprechenden Versuchen, Programmierungen oder Entwicklungen geleistet haben.

Maik Lorch

Karlsruhe, Juni 2013

Inhaltsverzeichnis

1	Einleitung	7
1.1	Bedeutung	7
1.2	Entwicklungsstand	8
1.2.1	Servoantrieb	8
1.2.2	Servoantriebssysteme	11
1.2.3	Kinematische Zusammenhänge	13
1.2.4	Engineering von Antrieben und Antriebssystemen	15
1.2.5	Notwendige mathematische Methoden	21
1.3	Offene Probleme	25
1.4	Ziele	26
2	Neues ganzheitliches Konzept	30
2.1	Anforderungen	30
2.2	Konzeptübersicht	34
3	Neu- und Weiterentwicklungen zum Bewegungsentwurf	38
3.1	Berechnung von Bewegungsverläufen	39
3.1.1	Aufarbeitung von Funktionen für die Bewegungsberechnung	40

3.1.2	Neues Verfahren zur Bewegungsberechnung	42
3.2	Entwickelte Selektionsverfahren	47
3.2.1	Kinematische Selektion	48
3.2.2	Einfache Selektion	49
3.2.3	Multikriterielle Selektion	50
3.3	Implementierung des Bewegungsentwurfs	55
3.3.1	Interpolation von Bewegungen	56
3.3.2	Selektion von Bewegungen	57
3.3.3	Teilprogramm zum Bewegungsentwurf	58
3.4	Zusammenfassung der Neuerungen im Bewegungsentwurf	59
4	Neu- und Weiterentwicklungen zum Hardwareentwurf	61
4.1	Angepasste Modellierung der Mechanik	62
4.1.1	Formulierung eines modularisierten Gesamtmodells	62
4.1.2	Zweckmäßige Notation von Standardmodulen	64
4.2	Integration der Auslegungsrechnung	69
4.3	Automatisierte Auswahl von Antriebskomponenten	70
4.4	Implementierung des Hardwareentwurfs	73
4.4.1	Implementierung der Modellbildung	73
4.4.2	Implementierung der Auslegungsrechnung und Komponentenauswahl	74
4.5	Zusammenfassung der Neuerung im Hardwareentwurf	75
5	Neu- und Weiterentwicklungen zum Systementwurf	76
5.1	Mathematische Modellierung der Anforderungen	77
5.1.1	Beschreibung einer Bewegung	77

5.1.2	Innere Zusammenhänge des Antriebssystems	79
5.1.3	Synthese eines Petrinetzes	82
5.2	Entwürfe zur Optimierung des Gesamtsystems	85
5.2.1	Ermittlung eines Initialsystems	86
5.2.2	Ausgewogener Zeitverlust in einem zusammengesetzten Bewegungsabschnitt	87
5.2.3	Zeitminimierung	90
5.3	Implementierung des Systementwurfs	91
5.4	Zusammenfassung der Neuerungen im Systementwurf	92
6	Ergebnisse	94
6.1	Referenzdaten	94
6.1.1	Manueller Systementwurf für einen Längstaktmontageautomat	96
6.1.2	Entwurf für einen Längstaktmontageautomat mittels gängiger CAE-Software	100
6.2	Neuartiger Entwurf für einen Längstaktmontageautomat	101
6.2.1	Berechnung der Werte für den energiesparenden, ruckarmen und zeitoptimalen Entwurf	101
6.2.2	Vergleich der erstellten Entwürfe	109
6.3	Bewertung der entwickelten Algorithmen und deren Implementierung	111
6.4	Analyse und Bewertung der Methoden mittels Modellmaschine	116
6.4.1	Aufbau der Modellmaschine	116
6.4.2	Versuchsergebnisse	120
7	Zusammenfassung	130

8	Literaturverzeichnis	135
9	Anhang	144
9.1	Zur Erweiterung der VDI 2143 um die dritte Ableitung	144
9.2	Zur Ausarbeitung der Fuzzy-Regelsätze	149
9.2.1	Erster Regelsatz (Bewertung einer Bewegung)	149
9.2.2	Zweiter Regelsatz (Affinität einer Bewegung)	150
9.3	Zur Optimierung der Modellmaschine	152

1 Einleitung

1.1 Bedeutung

Der Umsatz der deutschen Elektroindustrie betrug im Jahre 2011 im Bereich „Elektrische Antriebe“ 10,7 Milliarden Euro [1]. Elektrische Antriebe werden zunehmend als geregelte Komponenten von mechatronischen Systemen [2, 3, 4, 5] eingesetzt und erfüllen somit Aufgaben in einem komplexen System, beispielsweise in einer Produktionsmaschine.

Hierin kommen eine Vielzahl von Antrieben zum Einsatz, die ausgelegt und deren Bewegungsaufgabe definiert werden muss. Aus historischen Gründen ist eine getrennte Betrachtung von Auslegungsrechnung und Entwurf einer Bewegungsführung üblich. Eine Betrachtung von Zusammenhängen zwischen den Bewegungen findet auf der Ebene des Engineerings von Antriebssystemen bis heute praktisch nicht statt.

Die separate Betrachtung steht der Findung von insgesamt optimierten Systemen im Wege und verhindert die Berücksichtigung von Einflüssen zwischen verschiedenen Entwicklungsschritten. Für komplexer werdende Antriebsaufgaben und bei wachsender Bedeutung von Effizienz und umfassender Systementwicklung ist deshalb ein einheitliches zielgerichtetes Engineering von Antriebssystemen anzustreben.

Ein einheitliches Engineering von Antriebssystemen wird derzeit nicht ausführlich durchgeführt, da Methoden fehlen, die eine Umsetzung einer derartigen Gestaltung von Antriebssystemen ermöglichen. Rechnerbasierte Möglichkeiten zum Engineering bestehen nur für die Teilaufgaben der Auslegung von Antrieben und für den Entwurf von Bewegungen. Es existieren nur teilweise umgesetzte Schnittstellen zwischen den genannten Teilsystemen und auch keine Algorithmen zur Optimierung der einzelnen Bewegungsachsen oder einer umfassend betrachteten Antriebslösung. Jedoch liegen zahlreiche Methoden und Algorithmen vor, mit denen eine Bahnplanung auch unter der Berücksichtigung der Kollisionsvermeidung realisiert werden kann [6, 7, 8, 9, 10, 11, 12]. In der Regel beziehen sie sich jedoch auf kinematische Ketten, wie sie in Robotersystemen vorkommen. Für die im Weiteren behandelten Einsatzgebiete elektrischer Antriebssysteme, insbesondere hinsichtlich expliziter Kollisionsbetrachtungen sowie ganzheitlicher Betrachtungsweise, sind keine geeigneten Algorithmen bekannt.

Um den komplexeren Systemen und zugleich steigenden Anforderungen gerecht zu werden, bedarf es einer einheitlichen Betrachtung beim Engineering von Antriebssystemen. Der steigende

Einsatz von mechatronischen Antriebslösungen und der wachsende Aufwand beim Engineering von Antriebssystemen untermauern die Erforderlichkeit.

Mit der vorliegenden Arbeit wird daher ein Beitrag zur Weiterentwicklung von Methoden geleistet, die es Ingenieuren ermöglichen, optimierte Maschinen mit Systemen von Servomotoren als Antriebe zu entwickeln. Sie beinhaltet die Forderung nach einer neuen Entwicklungsmethodik zum Entwurf von Antriebssystemen mit der Zielstellung der Aufwandsminimierung beim Engineering und der Gestaltung effizienter Systeme mit definierten optimalen Eigenschaften. Die abgeleitete neue Vorgehensweise wird legitimiert durch ein Konzept zur Umsetzung, welche beispielhaft anhand einer Software statuiert wird.

1.2 Entwicklungsstand

Eine erste wissenschaftliche Dokumentation eines Stellantriebes aus der ersten Hälfte des 16. Jahrhunderts lieferte bereits Georgius Agricola (1494-1555): „Wenn daher die eine Bulge schon fast bis zu dem geneigten Bretterboden herausgezogen ist, schließt er¹ die Öffnung, damit das Rad zum stehen kommt. Wenn die Bulge ausgegossen ist, macht er die andere Öffnung auf, damit die anderen Schaufeln, vom Stoße des Wassers getroffen, das Rad in der entgegengesetzten Richtung bewegen.“(Abbildung: 1.1)[13] Es wird also die Bewegung der Bulge² durch einen Arbeiter vorgegeben, der nur die Hilfsenergie zum Steuern der „Ventile“ aufbringen muss. Erste Motoren, die als Servo³ bezeichnet wurden, konnten ebenfalls eine Bewegung führen und waren zunächst derart ineffektiv, dass sie nur die Hilfsenergie aufbrachten. Bis heute werden komplexe Bewegungen über mechanische Wirkprinzipien wie Kurvenscheiben erzeugt [15, 16], doch eine Entwicklung hin zu extrem leistungsstarken Servosystemen mit Eigenschaften moderner mechatronischer Produkte ermöglicht eine neue Anwendungsqualität gesteuerter Bewegungsführungen. Im Folgenden sollen der aktuelle Entwicklungsstand und bestehende Herausforderungen aufgezeigt werden, die es im Rahmen des Engineerings von elektrischen Antriebssystemen zu meistern gilt.

1.2.1 Servoantrieb

Sind im historischen klassischen Maschinenbau Bewegungsführungen oder -aufgaben rein mechanisch realisiert worden, bedient sich der moderne Maschinenbau mechatronischer Methoden, welche Produkte mit vielfältigen Vorteilen hervorbringen [17, 18]:

¹der Maschinensitzer auf der Hängebühne

²modern: Förderkorb

³Ein Servo (lat. *servus*: der Sklave) ist zunächst nur ein Stellantrieb für die tatsächliche Stelleinrichtung [14]

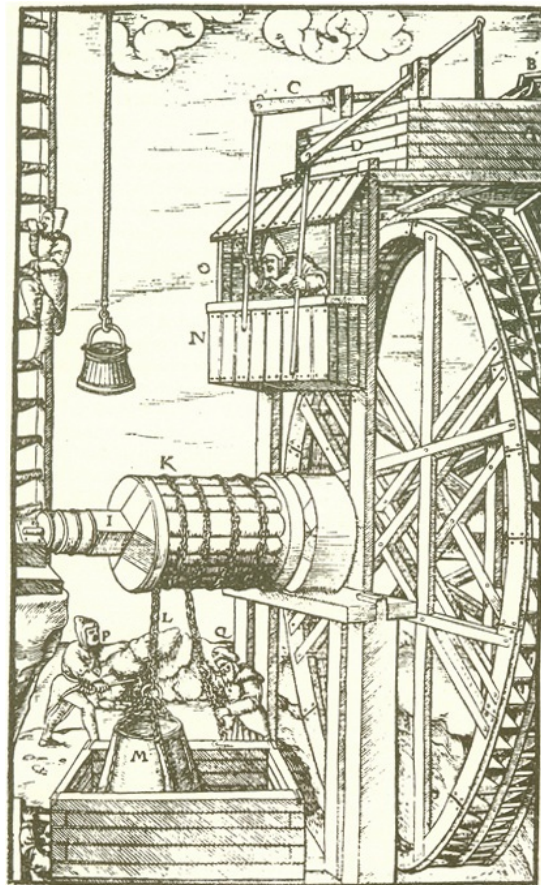


Abbildung 1.1: „Kerrad“ in der „De re metallica“ [13]

- Die Bewegung kann in der Betriebsphase einer Maschine über Software adaptiert werden.
- Es ist ein stark vereinfachter mechanischer Aufbau möglich, zum Beispiel durch den Verzicht auf besonders genau gefertigte Kurvenscheiben.
- Auf komplexe mechanische übertragenden Elemente wie Schrittgetriebe oder Königswellen, die mit geringen Toleranzen und komplexen schwingungsdynamischen Eigenschaften gefertigt werden müssen, kann verzichtet werden.
- Eine deutlich höhere Anzahl synchronisierter Einzelbewegungen ist erreichbar. [19].

Ein einzelner Servoantrieb (Abbildung 1.2) besteht im Wesentlichen aus einem elektromechanischen Wandler (Motor), einer elektrischen Komponente, die den Motor mit Energie versorgt und in den eine Regelung implementiert ist, sowie einem Geber⁴, der die Position und Geschwindigkeit vom Rotor des Motors feststellt.

⁴Ein Geber ist beispielsweise ein Sensor zur Erfassung eines Drehwinkels. Magnetische oder optische Messverfahren werden beispielsweise in [20] beschrieben.

Für die Motoren kommen hauptsächlich folgende technische Ausführungen zum Einsatz, die unterschiedliche Vorzüge genießen:

- Synchronmaschine [21]:
Die Synchronmaschine wird mit Wechselstrom betrieben, wobei die Frequenz die Rotationsgeschwindigkeit des magnetischen Feldes bestimmt, mit der wiederum der Läufer synchron läuft. Unter Last läuft der permanent magnetische Rotor dem Feld allerdings nach, was eine entsprechende Regelung erfordert [22]. Der Synchronmotor besitzt eine vergleichsweise kleine Massenträgheit, ist aber wegen des Magneten im Läufer auch kostspieliger.
- Asynchronmaschine [23, 21]:
Die Asynchronmaschine ist eine wechselstromerregte Maschine, in der das magnetische Feld des Rotors durch den Strom erzeugt wird, der vom umlaufenden Feld des Stators induziert wird. Um das Feld des Läufers zu induzieren, muss das Statorfeld asynchron zum Läufer drehen. Im Vergleich zur Synchronmaschine besitzt die Asynchronmaschine einen kleineren Wirkungsgrad, ist aber kostengünstiger in der Anschaffung.
- Gleichstrommotor [24, 21]:
Ein Gleichstrommotor wird mit Gleichstrom versorgt, seine Drehzahl und Richtung ist über die Spannung steuerbar. Er verfügt in der Regel über einen permanenten magnetischen Statorfluss. Das Drehmoment wird durch die Umschaltung des Ankerstroms über einen Kommutator erzeugt. Gleichstrommotoren werden nur sehr eingeschränkt für Neuentwicklungen verwendet [25]. Besondere Nachteile der Bauart sind die Bürsten am Kommutator, die verschleifen und deren Bürstenfeuer einen Einsatz in beispielsweise vor Feuer geschützten oder explosionsgefährdeten Umgebungen ausschließen.

Durch einen Servoregler, in den heutzutage häufig auch die Leistungselektronik integriert ist, werden Position und Geschwindigkeit einer Bewegungsachse im Rahmen physikalischer Grenzen regelbar. Führungen von Position und Geschwindigkeit erfolgen in der Regel durch Kaskadenregelungen [22, 21]. Weitere Regelgrößen können das Drehmoment bzw. der Strom des Motors sein [26, 27]. Servoregler für industrielle Anwendungen verfügen über zahlreiche Überwachungsfunktionen, Einrichtungen für die Maschinensicherheit, Schnittstellen für verschiedenste Architekturen in der Automation [28] und integrierte Steuerungen [29], z.B. nach IEC 61131.

Der beschriebene Aufbau eines Servoantriebs ermöglicht es, Bewegungen als Führungsgrößen zu entwickeln und von der Mechanik nachlaufen zu lassen. Dabei folgt der Servomotor der Führungsgröße mit einer Regelabweichung, teilweise Schleppfehler genannt. Ein Servoregler geht bei Überschreitung eines maximal zulässigen Schleppfehlers in einen Fehlerzustand. Zur Lösung der im vorliegenden Kontext abgehandelten Probleme kann folglich davon ausgegangen werden, dass die Regelabweichung vernachlässigbar ist und somit eine Bewegung exakt der erstellten Führungsgröße folgt.

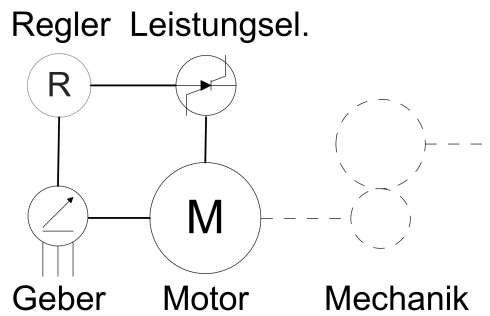


Abbildung 1.2: Aufbau eines Servoantriebs

1.2.2 Servoantriebssysteme

Ein Servoantriebssystem besteht aus mindestens zwei Servoantrieben, die gemeinsam auf eine Maschine einwirken. Im gegebenen Zusammenhang ist ein breites Spektrum von Maschinen abgedeckt, bei denen mehrere gesteuerte Bewegungen eingesetzt werden. Als typische Beispiele können Verpackungsmaschinen, Montagemaschinen, Transporteinrichtungen, Webmaschinen, Abfüllanlagen und viele andere genannt werden. Ab einem gewissen Abstraktionsgrad formulieren sich äquivalent darstellbare Herausforderungen. Eine Abgrenzung existiert zu Systemen, die kinematische Ketten bilden, wie beispielsweise ein Roboterarm. Solche Systeme, bei denen die Dynamik einer Achse eine Last an einer anderen induziert, werden im Rahmen der Arbeit nicht abgedeckt.

Ein herausgegriffenes Beispiel für die Anwendung eines elektrischen Servoantriebssystems stellt der Längstaktmontageautomat (Abbildung 1.3) dar, auf dem verschiedenste Produkte in mehreren Arbeitsschritten, verteilt auf nacheinander folgenden Stationen, montiert werden. Hierbei



Abbildung 1.3: Längstaktmontageautomat [30]

kommen beim Grundaufbau drei einzelne Antriebe zum Einsatz, die miteinander auf den Arbeitsablauf der Maschine einwirken. Ein Servoantrieb stellt die lineare Bewegung zum Trans-

port von Produkten dar, ein weiterer treibt die Arbeitsstationen an und ein dritter sorgt für die mechanische Fixierung der Werkstückträger während eines Arbeitsschrittes. Der Ablauf eines Maschinentaktes lässt sich wie folgt beschreiben:

1. Verfahren der Werkstückträger um eine definierte Strecke und Stillsetzen der Träger mit definierter Genauigkeit
2. Schließen der Fixierung zur Aufnahme von Bearbeitungskräften
3. Arbeiten am Werkstück, angetrieben durch eine kontinuierlich laufende Rotation mit einer Umdrehung pro Maschinentakt
4. Öffnen der Fixierung.

Hieraus lassen sich einfache Restriktionen ableiten. So darf die Fixierung erst greifen, wenn die Werkstückträger zum Stillstand gekommen sind. Des Weiteren muss die Fixierung abgeschlossen sein, bevor die Manipulation an den Produkten beginnt. Kompliziertere Restriktionen treten bereits auf, wenn die Maschine kategorisch ohne Kollisionsmöglichkeiten im Fehlerfall sein soll. Das ist eine typische Anforderung [31], die sich durch zu schützende wertvolle Werkzeuge legitimiert. So lautet die Restriktion, dass keine Werkzeuge in den Eingriff am Werkstück geraten dürfen, wenn durch einen Fehlerzustand die Fixierung nicht greift. Der Werkzeugantrieb muss noch rechtzeitig still zu setzen sein. Zusätzlich gilt dennoch ein Optimierungsziel nach möglichst hoher Taktzahl, welche massiv die Wertschöpfung einer Maschine für die Massenproduktion beeinflusst. Bereits die vorgestellten Zusammenhänge zwischen Bewegungsanforderungen an die einzelnen Achsen und deren Einfluss auf die anderen Achsen unter Forderung einer Zeitoptimierung und schließlich auf die Auslegung der Maschine lassen erkennen, dass nur eine gezielte ganzheitliche Analyse des Problems zum gewünschten Entwurf des Antriebssystems verhelfen kann.

Zeitliche Abhängigkeiten, konkurrierende Raumnutzung oder Forderungen nach Kollisionssicherheit treten in anderen Maschinenarten in prinzipiell gleicher Weise auf. Ebenso gleichartig sind die Architekturen von Antriebssystemen, auf die im folgenden Abschnitt näher eingegangen wird.

Systemarchitektur

Antriebssysteme, in der Anwendung wie im vorhergehenden Abschnitt beschrieben, können physikalisch und logisch diverse Strukturen aufweisen, besitzen aber methodische Gemeinsamkeiten. Da im weiteren Verlauf der Arbeit allgemeingültige Methoden anzustreben sind, werden nach dem Aufzeigen der Diversität von Systemarchitekturen allgemeingültige Gemeinsamkeiten formuliert.

Die Informationsverarbeitung von Antriebssystemen findet sowohl auf der Akteur- als auch auf

der Steuerungsebene der Automatisierungshierarchie statt [32]. Sie sind in grob unterschiedlichen Ausführungen im industriellen Einsatz.

Auf der Steuerungsebene können Antriebe komplett durch eine übergeordnete Steuerung betrieben werden oder über eigene Anteile an der Steuerungsstruktur verfügen. Bei kleineren Anwendungen können moderne Antriebsregler gar die komplette Prozessleitung übernehmen [33]. Die Bereitstellung von Führungsgrößen kann ebenfalls in der Steuerungsebene festgelegt sein, beispielsweise in Form von Parametern, die von der Aktorebene interpretiert werden.

In der Regel werden Führungsgrößen jedoch auf der Aktorebene verteilt gespeichert. Die Regelung und Leistungsbereitstellung der einzelnen Antriebe kann sowohl in einem oder mehreren zentral vorhandenen Geräten erfolgen oder auch dezentral direkt am Motor. Während in der ersten Variante Motorleistung und Gebersignal über weitere Strecken transportiert werden müssen, muss in der zweiten die Kommunikation zwischen den Antrieben über weitere Strecken erfolgen.

Der Informationsaustausch erfolgt jeweils über ein Bussystem, wobei übliche Systeme AS-I, Profibus, CAN-Bus [34] oder Interbus wie auch herstellerspezifische Systeme je nach konkretem Anwendungsfall und Antriebshersteller zum Einsatz kommen. Die Kommunikation der Antriebe miteinander ist in beiden Varianten der Prozessleitung notwendig, sobald Bewegungen hinreichend synchron ausgeführt werden müssen [35].

Abbildung 1.4 stellt das gemeinsame Modell eines Antriebssystems graphisch dar. Es besteht aus n Achsen, die wie im vorherigen Abschnitt beschrieben, aufgebaut sind. Wie auch immer Logik, Datenspeicherung und Kommunikation in einem Antriebssystem verteilt sind, muss es für Anwendungen mit mehreren Bewegungsachsen ein Signal zur Synchronisation geben. In der Praxis hängen die Führungsgrößen aller Achsen von einer Leit- oder Masterachse ab, die auch rein virtuell bestehen kann. Abhängig vom Winkel zwischen 0 und 2π werden Führungsgrößen an die jeweiligen Antriebsregler ausgegeben. Die Position der Masterachse wird über das angesprochene Bussystem verbreitet, ebenso werden Informationen über etwaige Fehlerzustände über das System weitergegeben. Als Grundlage für weitere Betrachtungen ist in Abbildung 1.4 ein allgemeingültiges Modell für Antriebssysteme dargestellt. Für die abhängigen Achsen muss die Führungsgröße (in der Regel Position und Geschwindigkeit) hinterlegt werden. Für die im Weiteren behandelte Anpassung der Führungsgrößen spielt deren physikalischer Speicherort jedoch keine Rolle. Zwischen den Achsen eines Antriebssystems können je nach Anwendungsfall diverse Abhängigkeiten unterschiedlicher Qualität bestehen. Die Behandlung innerer Zusammenhänge stellt sich als einer der Schwerpunkte beim ganzheitlichen Engineering heraus, deren Modellierung im Abschnitt 5.1.2 behandelt wird.

1.2.3 Kinematische Zusammenhänge

Die Kinematik (griech. kinesis = Bewegung) ist die Lehre vom geometrischen und zeitlichen Bewegungsablauf, ohne dass auf Kräfte als Ursache oder Wirkung der Bewegung eingegangen wird [36]. Im Wesentlichen sind folgende hinlänglich bekannten kinematischen Zusammenhänge [37, 38] für die vorliegende Arbeit von weiterer Bedeutung.

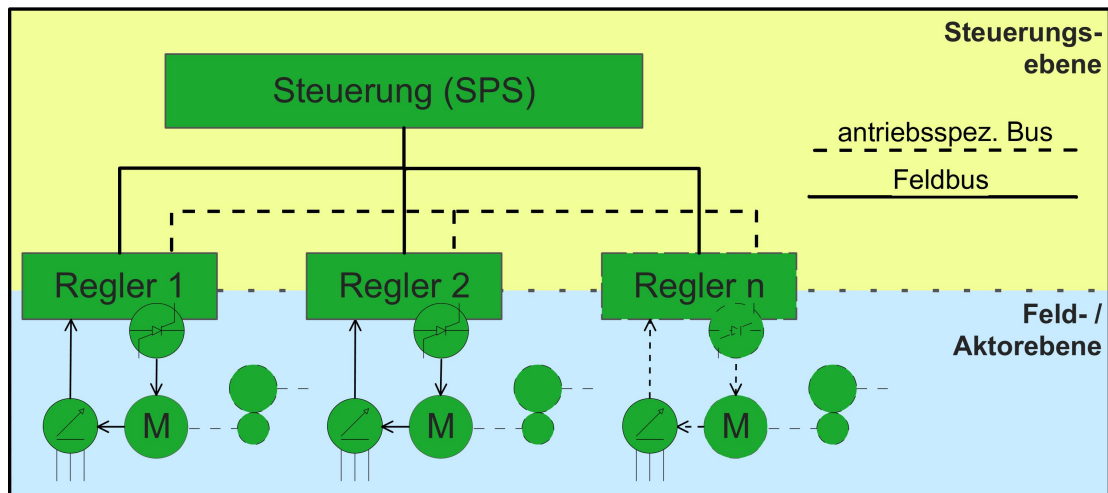


Abbildung 1.4: Antriebssystem

Zwischen dem zeitlichen Verlauf von Position s , Geschwindigkeit v , Beschleunigung a und Ruck r eines geometrischen Punktes im Raum besteht ein differentieller Zusammenhang. Reduziert auf den eindimensionalen Raum, resultierend aus der Rotation oder Translation eines elektrischen Antriebs, gilt folgender Zusammenhang zwischen den gegebenen Größen und deren Änderungen [36]:

$$\dot{s} = \dot{v} = \dot{a} = r \quad (1.1)$$

Der Ruck ist die zeitliche Änderung der Beschleunigung a , die wiederum bei konstanter Masse m nach dem 2. Newtonschen Grundgesetz proportional zur Kraft F ist:

$$F = m \cdot a \quad (1.2)$$

Somit gilt für die zeitliche Änderung des Kraftverlaufs:

$$\dot{F} = m \cdot \dot{a} \quad (1.3)$$

Durch die Einprägung einer steigenden oder fallenden Stromflanke (Sprung) in einen Motor resultiert, wegen des allgemein proportional angenommenem Verhältnisses von Strom und Drehmoment [39], eine entsprechende Flanke der Beschleunigung. Je steiler die Flanke ist, desto größer ist der Ruck und er wird mehr und mehr einem Dirac-Impuls gleichen [40]. Allgemein bekannt ist auch, dass durch einen Dirac-Impuls sämtliche Eigenfrequenzen eines Systems angeregt werden. Die durch einen Servoantrieb bewegte Mechanik wird bei hohem Ruck demnach zu Schwingungen angeregt. Derartige Schwingungen können zu hohe Schleppfehler auslösen, wenn sie nicht ausgeglet werden können (Instabilität). Sie können die Lebenszeit von Maschinenteilen reduzieren oder zu schlechterer Qualität in Bearbeitungsprozessen führen. Demnach ist es oftmals eine Anforderung, den Ruck in einer Bewegungsführung zu begrenzen. Das Ausregeln von Störungen durch das Schwingen der Mechanik kostet zudem Energie [41, 42], was zusätzlich für eine Ruckbegrenzung in Führungsgrößen spricht.

Weil zum Beschleunigen einer Masse Arbeit aufgebracht werden muss, folgt, dass bereits durch

den Verlauf der Beschleunigung Unterschiede in der Energieeffizienz einer Bewegung bestehen. Starkes Beschleunigen und späteres ebenfalls starkes Abbremsen sorgt für hohen Energieeinsatz. Unter Berücksichtigung von statischen Lasten und eventueller Wirkungsgrad behafteter Rückgewinnung der Bremsenergie verkompliziert sich die Energiebilanz einer Bewegung. Die Näherung an eine effiziente Bewegung kann dennoch durch Manipulation des reinen kinematischen Verlaufs einer Bewegung erreicht werden. Die mathematischen Beziehungen hierzu sind in der Herleitung eines Gütekriteriums im weiteren Verlauf der Arbeit beschrieben (siehe Abschnitt 3.2.2).

Wie kinematische Verläufe als Führungsgrößen erzeugt werden können, wird im Kapitel 3 dargestellt. Zunächst wird durch eine detailliertere Betrachtung eines bewegten mechanischen Systems auf mathematisch unumgängliche Bedingungen im Bewegungsentwurf hingewiesen.

Gleichung (1.1) ist formuliert um zu verdeutlichen, dass beim Entwurf einer Bewegung darauf zu achten ist, dass Zusammenhänge zwischen Strecke, Geschwindigkeit, Beschleunigung und resultierender Ruck einzuhalten sind. Die kinematischen Zusammenhänge müssen auch für die Schnittstelle zweier Einzelbewegungen eines Punktes zum Zeitpunkt t_s eingehalten werden. Die Formel legt die Bedingungen für eine rechts- und linksseitige Annäherung an den Punkt fest. Es gilt insbesondere, dass Strecke und Geschwindigkeit stetig zu verlaufen haben, der stetige Beschleunigungsverlauf ist im Sinne der Ruckbegrenzung wünschenswert aber aus physikalischer Sicht nicht obligatorisch. Ein Beschleunigungssprung ist durch einen Kraftsprung realisierbar. Die beschriebenen Anforderungen ergeben zusammenfassend:

$$\begin{aligned} \lim_{t \rightarrow t_s^-} s_1(t) &= \lim_{t \rightarrow t_s^+} s_2(t) \\ \lim_{t \rightarrow t_s^-} v_1(t) &= \lim_{t \rightarrow t_s^+} v_2(t) \\ \lim_{t \rightarrow t_s^-} a_1(t) & (=) \lim_{t \rightarrow t_s^+} a_2(t) \end{aligned} \quad (1.4)$$

Die Bedeutung eines stetigen Ruckverlaufs ist lediglich in besonderen Fällen von Bedeutung, die weiterhin keine Beachtung finden. Mathematische Methoden für die entsprechende Interpolation von Bewegungsvorgängen sind im Abschnitt 3.1.1 aufbereitet.

1.2.4 Engineering von Antrieben und Antriebssystemen

Methoden zur Entwicklung von Antriebssystemen können prinzipiell aus allgemeingültigen Entwicklungs- und Konstruktionsmethoden [43, 44, 45, 46] abgeleitet werden. Moderne Vorgehensweisen zur Entwicklung liefern Methoden, wie die „SPALTEN-Technik“ [47, 48] oder zur Qualitätssteigerung die „Quality Function Deployment (QFD)“ [49]. Einflüsse der Marktwirtschaft [50, 51] auf die Entwicklung von Produkten sind ebenso zu berücksichtigen. Das Engineering [15, 52] von Antriebssystemen umfasst im Wesentlichen folgende Grundaufgaben:

- Analyse und Dokumentation der Anforderungen

- Festlegung von Systemarchitekturen
- Entwurf von Bewegungen
- Auswahl von geeigneten Geräten
- Optimierung des Systems
- Realisierung und Inbetriebnahme des Systems.

Nach [53] gehören neben der Inbetriebnahme (teilweise Entwurf von Bewegungen) auch der Service zum Engineering von Antriebssystemen.

Zur Darstellung von Anforderungen an ein Antriebssystem und für die Beschreibung ereignisgesteuerter Bewegungsverläufe kann der hybride Funktionsplan [54, 25] verwendet werden. Die Bewegungen des Antriebssystems werden in sogenannte Bewegungszellen unterteilt und schließlich der Gesamtablauf aller Bewegungen an der Maschine strukturiert dargestellt [55]. Es existiert ein Softwaretool zur Arbeit mit dem hybriden Funktionsplan [56]. Hiermit lassen sich besonders der zeitliche Ablauf aller Maschinenbewegungen und kausale Zusammenhänge gut abbilden. Die Interpolation, die Geräteauswahl und die Systemoptimierung bleiben hierbei jedoch zunächst außen vor.

Entwicklungsprozess

Bei der Entwicklung von Antriebssystemen ist es sinnvoll und üblich, typische Entwicklungsstrategien aus den Konstruktionswissenschaften anzunehmen. Die VDI-Richtlinie [57] enthält allgemeingültige Vorgaben für Vorgehensweisen in der Entwicklung, sie ist für mechatronische Systeme weiter spezifiziert [58]. Viele weitere Methoden sind weitreichend dokumentiert, sie enthalten insgesamt betrachtet die in Abbildung 1.5 dargestellten Schritte für Antriebssysteme [53]. In der Planungsphase finden drei Schritte nacheinander statt, wobei die Funktionsfestlegung auf die durch die Antriebsauslegung legitimierten Möglichkeiten beschränkt ist. Eine Beachtung der Anforderungen durch die Funktionsfestlegung an die Antriebsauslegung findet nicht statt. Eine systematisch integrierte Vorgehensweise zur Systemoptimierung ist ebenfalls nicht vorgesehen.

Praktisch erfolgt das Engineering häufig durch die Anbieter von Systemtechnik für elektrische Antriebssysteme. Die Unternehmen stellen ihre Fachkenntnisse im Zuge ihrer Verkaufsstrategien zur Verfügung. Die Strategie wird durch die Bereitstellung von geschulten Vertriebs- oder Applikationsingenieuren vollzogen. Auch die Bereitstellung von Leitfäden und Software gehören zu den Geschäftsmodellen, womit die hohe Qualität und die große Leistungsfähigkeit von Engineeringtools zu Wettbewerbsvorteilen führt, denn schließlich wird dadurch auch die Qualität des entworfenen Antriebssystems beeinflusst [53, 59].

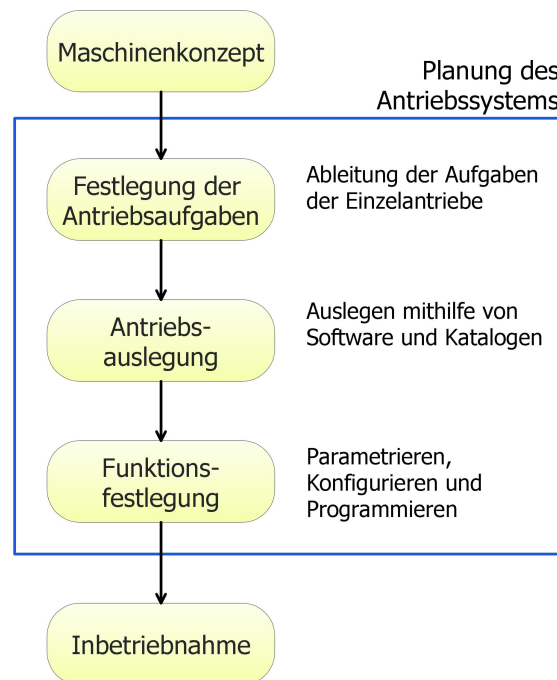


Abbildung 1.5: Ablauf des Engineerings von Antriebssystemen, entsprechend [53]

Bewegungsentwurf

Mathematische Funktionen zum Bewegungsentwurf sind sehr breit gefächert und werden je nach Anwendungsfall eingesetzt. Gemeinsam stellen alle Funktionen im Wesentlichen eine Interpolation oder eine Trajektorie zwischen zwei Randwerten dar, die in einem definierten zeitlichen Abstand liegen. Dabei lassen sich die Randwerte klassifizieren in Punkte der Rast „R“, Umkehrpunkte „U“, Punkte konstanter Geschwindigkeit „G“ und Punkte beschleunigter Bewegungen „B“ mit definierter Geschwindigkeit. Die kinematischen Eigenschaften können Tabelle 1.1 entnommen werden. Die Methoden zur Interpolation von Bewegungen zwischen Kombinationen

R	$v = 0$ $a = 0$
U	$v = 0$ $a \neq 0$
G	$v \neq 0$ $a = 0$
B	$v \neq 0$ $a \neq 0$

Tabelle 1.1: Kinematische Eigenschaften von Randwerten (entsprechend [60])

aus Randwerten reichen von aus mechanischen Kurvenscheiben abgeleiteten mathematischen

Gesetzen nach VDI-Richtlinie 2143 [60] über Erweiterungen [61] bis hin zu spezifischen anwendungsbezogenen Rechenvorschriften [62].

Ist es erforderlich, Bewegungen unter kinematischen Restriktionen und unter Maximen der Energieeinsparung, Schwingungsoptimierung oder Zeitverkürzung zu erzeugen, dann sind mathematisch geschlossene Lösungen nur begrenzt verfügbar [22]. Abhilfe dazu schaffen numerische Methoden zur Ermittlung optimierter Bewegungsverläufe, die im Rahmen der vorliegenden Arbeit entwickelt werden.

Auslegungsrechnung

Eine Auslegungsrechnung wird durchgeführt, um Geräte in geeigneter Dimension zum Einsatz zu bringen. Motoren müssen erforderliche Drehmomente und Drehzahlen für entworfene Bewegungsvorgänge und die verbundenen mechanischen Eigenschaften der Last bereitstellen. Gleichzeitig ist eine Überdimensionierung aus wirtschaftlichen Gründen und wegen der Effizienz eines Antriebs im Einsatz zu vermeiden. Zusätzlich hat die Dimensionierung Einfluss auf die Regelbarkeit eines Antriebs. Eine Auslegung ist insbesondere für den Motor und danach für die Leistungselektronik durchzuführen. Im Folgenden werden die wesentlichen Berechnungsgrundlagen aufgeführt.

Ein Motor wird gekennzeichnet durch das Nenndauerdrehmoment \bar{M}_{Motor} , eine Kennlinie, welche das maximale Drehmoment $M_{Motor}(n)$ in Abhängigkeit der Drehzahl n beschreibt und durch die Nenndrehzahl n_{Motor} (exemplarische Abbildung 1.6). Ebenfalls von Bedeutung ist die Massenträgheit des Rotors J_{Motor} , die für gute Regelbarkeit in einem bestimmten Verhältnis zur Trägheit der Mechanik stehen sollte.

Für die Dimensionierungsrechnung ist es notwendig, das Lastverhalten am Motor quantita-

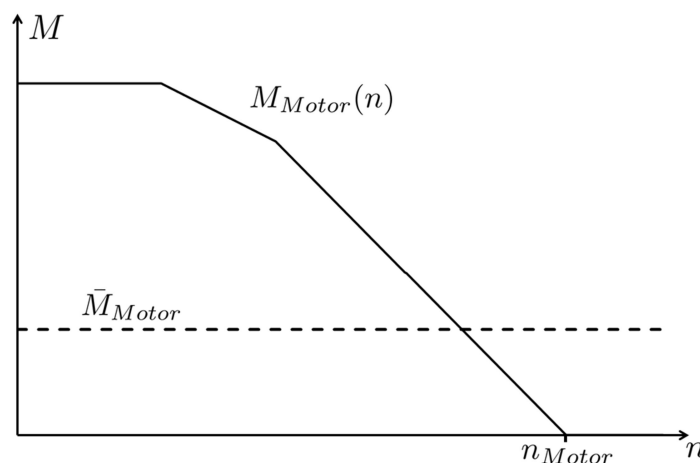


Abbildung 1.6: Exemplarische Motorkennlinie (entsprechend [63])

tativ zu bestimmen. Bei Kenntnis der angestrebten Bewegungsführung lassen sich erforderliche

Drehmomente und Drehzahlen ermitteln, wenn die mechanischen Eigenschaften der Übertragungsmechanik und kinematische Prozessgrößen (z.B. Bearbeitungskräfte) bekannt sind. Zur Modellierung von mechanischen Übersetzungen aller Art siehe Abschnitt 4.1. Letztlich lassen sich folgende Kennwerte einer Bewegung, die für die Motorauswahl von Bedeutung sind, an der Motorwelle ermitteln [21, 53, 59, 63]:

- die betragsmäßig maximal auftretende Drehzahl an der Motorwelle n_{max}
- das maximale Drehmoment M_{max} im Bewegungsverlauf aus dem Momentenverlauf M der Zusammensetzung aus statischem Anteil M_s , dynamischen Anteil M_d und einem verallgemeinerten Reibmoment \tilde{M}_R .

$$M = \left| M_s + M_d + \tilde{M}_R \right| \quad \text{mit} \quad M_d = J \frac{d\omega}{dt} \quad (1.5)$$

Das dynamische Drehmoment errechnet sich in Abhängigkeit des Betriebszustands, bremssender Motor M_b oder treibender Motor M_t , weil sich der Wirkungsgrad η kontrovers auswirkt, was die nachstehenden Gleichungen (nach [63]) verdeutlichen. Hierin gehen zudem die Massenträgheit J , die Winkelgeschwindigkeit ω , auch mit zeitlicher Ableitung $\dot{\omega}$, sowie das allgemeine Abtriebsmoment M_{ab} mit Übersetzungsverhältnis i ein.

$$\begin{aligned} M_t &= \eta \left(\frac{1}{i} M_{ab} + J \dot{\omega} \right) \\ M_b &= \frac{1}{\eta} \left(\frac{1}{i} M_{ab} + J \dot{\omega} \right) \end{aligned} \quad (1.6)$$

Um für weitere Ausführungen zu einer verallgemeinerten Notation beider Fälle zu gelangen, wird der abhängige Wirkungsgrad $\tilde{\eta}$ definiert [63]:

$$\tilde{\eta} = \begin{cases} \eta & \text{wenn Motor treibt } (\dot{\omega} > 0) \\ \eta^{-1} & \text{wenn Motor bremst } (\dot{\omega} \leq 0) \end{cases} \quad (1.7)$$

In einigen Fällen addiert sich zur bereits eingeführten dynamischen und zur statischen Belastung ein Reibmoment M_R , welches verallgemeinernd mit \tilde{M}_R angegeben ist. Seine Wirkung ist jeweils entgegen der Bewegungsrichtung gerichtet.

$$\tilde{M}_R = \begin{cases} M_R & \text{für } \omega \leq 0 \\ -M_R & \text{für } \omega > 0 \end{cases} \quad (1.8)$$

- das effektive (mittlere) Drehmoment \bar{M} aus der Taktzeit T_{Takt} der Maschine und dem Momentenverlauf entsprechend obiger Zusammensetzung

$$\bar{M} = \frac{1}{T_{Takt}} \int_0^{T_{Takt}} M \cdot dt \quad (1.9)$$

- die mittlere Drehzahl \bar{n} aus dem Drehzahlverlauf n über die Taktzeit

$$\bar{n} = \int_0^{T_{\text{Takt}}} n \cdot dt \quad (1.10)$$

Ein geeigneter Motor muss folgenden Restriktionen genügen (unter anderem [59, 64]):

$$\begin{aligned} n_{Motor} &\geq n_{max} \\ M_{Motor}(n) &\geq M_{max} \\ \bar{M}_{Motor} &\geq \bar{M} \\ k &= \frac{J_{Mechanik}}{J_{Motor}} \in [0, 5; 10] \end{aligned} \quad (1.11)$$

Die Motornendrehzahl n_{Motor} , das maximal erreichbare Motordrehmoment $M_{Motor}(n)$ in Abhängigkeit der Drehzahl sowie das effektive Motordrehmoment \bar{M}_{Motor} müssen über den entsprechenden Werten der Bewegung liegen. Dabei handelt es sich um die maximal auftretende Drehzahl n_{max} , dem maximal auftretenden Drehmoment M_{max} und dem effektiven Drehmoment \bar{M} aus der Kinematik der Bewegung. Das Verhältnis aus Massenträgheiten von Mechanik $J_{Mechanik}$ und Motor J_{Motor} muss je nach Anwendungsfall und verwendeter Motorbauart im genannten Intervall liegen [59]. Es ist zudem üblich, auch die Lage des Betriebspunktes $B(n_B | M_B)$ zu überprüfen, dessen Lage sich aus effektivem Drehmoment und mittlerer Drehzahl erschließt [63]:

$$\begin{aligned} n_B &= \bar{n} \\ M_B &= \bar{M} \end{aligned} \quad (1.12)$$

Der Betriebspunkt muss unterhalb der Kennlinie für das maximale Drehmoment liegen. Auf Basis des gewählten Motors ist es möglich, auch das Stellgerät bzw. die Leistungselektronik, sprich den Stromrichter, für den Antrieb zu dimensionieren. Neben der Nennspannung, welche mit der des Motors übereinstimmen muss, soll auch die Netzeingangsspannung mit der verfügbaren Netzspannung am Einsatzort des Antriebs übereinstimmen. Weitere Kenndaten eines Stellgeräts sind der Bemessungsstrom I_S , der Maximalstrom $I_{S,max}$ und der Grundlaststrom $I_{S,G}$. Die Werte sind vom maximal auftretendem Strom I_{max} und dem effektiven Strom I_{eff} unter Berücksichtigung eines Sicherheitsfaktors f zu unterschreiten [63]:

$$\begin{aligned} I_S &\geq I_{eff} \\ I_{S,max} &= I_S \cdot f \geq I_{max} \quad \text{mit } f \in [1, 5; 3] \end{aligned} \quad (1.13)$$

Dabei ergibt sich der Strom I zum Motor je nach Bauart und über die Motorkonstante k_M , so wie in Tabelle 1.2 zusammengefasst mit dem Motornennstrom I_{Motor} und dem Leerlaufstrom I_0 .

Gleichstrommotor	$I = I_{Motor} \frac{M}{M_{Motor}}$
Synchronmotor	$I = \frac{M}{k_M}$
Asynchronmotor	$I = \sqrt{I_0^2 + \frac{M^2}{M_{Motor}^2} I_{Motor}^2 I_0^2}$

Tabelle 1.2: Berechnung von Stromwerten nach Motorbauart [28]

Der effektive Motorstrom berechnet sich entsprechend Mittelwertsatz der Integralrechnung zu:

$$I_{eff} = \frac{1}{T_{Takt}} \int_0^{T_{Takt}} I \cdot dt \quad (1.14)$$

Bei detaillierten Betrachtungen können innere Größen des Stromrichters, beispielsweise die Zwischenkreiskapazität, dimensioniert werden. Auch die Auslegung des Bremswiderstands gehören zu den Aufgaben beim Aufbau eines Servoantriebs, die sich nach der Auswahl eines Stromrichters ergeben und daher keine Rolle im Rahmen des gesamten Systementwurfs spielen. Schließlich hängt die Auswahl eines Motors oder Reglers mit Leistungselektronik auch von vielen Umweltfaktoren des Einsatzortes ab. Es sind beispielsweise die Schutzklasse des Motors, das Prinzip der Kühlung und die Reglereigenschaften mit Schnittstellen oder Software zu bestimmen. Die Parameter haben jedoch keinen direkten Einfluss auf die kinematische Gestaltung eines Antriebssystems und werden im Weiteren nicht betrachtet.

1.2.5 Notwendige mathematische Methoden

Im Rahmen der Arbeit sind Aufgaben zu lösen, die den Einsatz mathematischer Methoden bedürfen. Mitunter wird sich der Theorie über Petrinetze [65, 66] sowie der Fuzzy-Logik [67, 68] bedient, deren Eigenschaften und relevanten Grundlagen nachfolgend beschrieben werden.

Petrinetze

Petrinetze dienen als Beschreibungsform für ereignisdiskrete Systeme. Sie wurden 1962 von Carl Adam Petri eingeführt [65] und nach ihm benannt. Sie sind seither für verschiedenartige Zwecke manipuliert und weiterentwickelt. Erste Beschreibungen von Petrinetzen für Anwendungen innerhalb von Ingenieurwissenschaften liefern [69, 70]. Inzwischen sind Petrinetze innerhalb der

Standardliteratur zur Automatisierungstechnik oder zu ereignisdiskreten System hinreichend beschrieben, um sie methodisch nutzen zu können [71, 72, 73, 74, 75]. Im Folgenden werden einige im weiteren Verlauf der Arbeit aufgegriffene Grundgedanken zu Petrinetzen erläutert.

Grundsätzlich ist ein Netz aus Stellen und Transitionen aufgebaut (siehe Abbildung 1.7). Stellen

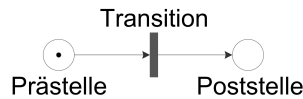


Abbildung 1.7: Grundeinheit eines Petrinetzes

können mit Marken versehen werden, was in einer kausalen Kette von Ereignissen bedeutet, dass eine bestimmte Bedingung erfüllt ist. Eine Marke fließt von einer Stelle zur nächsten, wenn die Transition feuert. Sie muss dazu entsprechend der folgenden Schaltregel aktiviert sein:

- die Prästelle ist markiert
- die Poststelle ist nicht markiert

Bei der Existenz von mehr als einer Prä- oder Poststelle an einer Transition ist sie aktiviert, sobald alle Prästellen markiert und alle Poststellen unmarkiert sind. Durch geschicktes Strukturieren der Petrinetze lassen sich Graphen für parallel laufende Prozesse oder Synchronisationsgraphen und viele weitere Abbildungen von diskreten Prozessen erstellen. Petrinetze können mit Eingängen und Ausgängen versehen werden. Dabei werden Eingänge auf Transitionen und Ausgänge nach Stellen geschaltet. Die Notation kann dabei auf diverse Weise erfolgen:

- Eine Möglichkeit bietet die Darstellung durch ein Quintupel \mathcal{N} , im Allgemeinen bestehend aus Mengen von Stellen \mathcal{P} , Transitionen \mathcal{T} , Präkanten \mathcal{F} , Postkanten \mathcal{W} und einem Anfangszustand des Netzes M_0 :

$$\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, M_0) \quad \text{mit: } \mathcal{F} \subseteq \mathcal{P} \times \mathcal{T} \text{ und } \mathcal{W} \subseteq \mathcal{T} \times \mathcal{P} \quad (1.15)$$

- Alternativ kann die Netzstruktur als Matrix \mathbf{N} abgebildet werden, in der die Elemente n_{ij} so definiert sind, dass $+1$ eine Kante von der Transition j zur Stelle i bedeutet beziehungsweise eine Kante von der Stelle i zur Transition j mit einer -1 notiert ist. Stellen und Transitionen ohne Verbindung werden mit 0 gekennzeichnet.
- Basierend auf der Matrix \mathbf{N} kann eine Zustandsraumdarstellung eines Petrinetzes angegeben werden, wenn es einen Schaltvektor $\vec{t}(k)$ zur Zeit k gibt [74]. Im Schaltvektor können jeweils nur aktivierte Transitionen \mathcal{T}_{akt} enthalten sein. Es ergeben sich dann die markierten Stellen $\vec{p}(k+1)$:

$$\vec{p}(k+1) = \vec{p}(k) + \mathbf{N}\vec{t}_S(k) \quad \text{mit: } \vec{t}(k) \in \mathcal{T}_{akt} \quad (1.16)$$

Eine entscheidende Fragestellung betrifft die Erreichbarkeit von Zuständen eines Petrinetzes, also welche Markierungsvektoren das Netz ausgehend von einer Anfangsmarkierung \vec{p}_0 annehmen kann. Hierzu können Erreichbarkeitsgraphen aufgestellt werden, die alle möglichen Markierungsfolgen aufzeichnen [74].

Petrinetze als ereignisdiskrete Systembeschreibung berücksichtigen keine zeitlichen Aspekte. Mathematisch wird lediglich festgehalten, ob eine Transition schaltet, nicht aber wann. Wie im vorliegenden Fall, wenn es um zeitliche Optimierungen oder um zeitabhängige Berechnungen von Hardwareentwurf und Bewegungsentwürfen geht, ist der zeitliche Ablauf im System jedoch relevant. Zur Modellierung derartiger kontinuierlicher Systeme können hybride oder zeitbehaf-tete Petrinetze verwendet werden. Eine Möglichkeit zeitkontinuierliche Aspekte zu integrieren ist in [76] gegeben.

Fuzzy-Logik

Die Fuzzy-Logik ist ein Teilgebiet der Computational Intelligence⁵ (CI) und beschäftigt sich mit einer, gegenüber der Booleschen Logik erweiterten, unscharfen Logik [77]. Sie wurde von Prof. Zadeh in den Sechzigerjahren entwickelt [78]. Mit der unscharfen Logik ist menschliches unscharfes oder linguistisch vorhandenes Wissen oder Regelkenntnisse mathematisch abbildbar. Die Fuzzy-Logik ermöglicht neben dem Aufbau von Regelungen⁶ auch eine Klassifizierung von Daten und schließlich beispielsweise die Selektion einer Lösung aus einer Menge von Möglichkeiten basierend auf wissensbasierter Bewertung (Abschnitt 3.2.3). Ein konventionelles Fuzzy System zur Entscheidungsfindung besteht aus einer Fuzzyfizierung, einem Regelsatz und einer Defuzzyfizierung.

Mithilfe von Zugehörigkeitsfunktionen werden vorliegende Daten mit einem Zugehörigkeitswert zu unscharfen Mengen versehen (Fuzzyfizierung). Typische Funktionen sind dabei die Singleton-, Dreiecks-, Trapez-, Gauß- oder Cosinusquadratfunktion [82, 83]. Im weiteren Verlauf der Arbeit zeigt sich, dass die anliegenden Daten anhand der ersten drei einfachen Funktionen bewertbar sind.

Die Basis eines Regelsatzes bilden linguistische Terme, da in ihnen das Vorwissen festgehalten ist. Sie lauten beispielsweise für die Analyse einer Bewegung:

```
WENN Geschwindigkeit hoch UND Beschleunigung hoch DANN Zeit  
klein
```

Zur Auswertung der Regeln ist es notwendig, die unscharfen Mengen in einer geeigneten Strategie und mit geeigneten Operatoren zu verknüpfen. Die Strategie zur Auswertung, die Inferenz,

⁵Abk. für Computational Intelligence, seit 1992 durch Bezdek geprägter Überbegriff für Fuzzy-Logik, Künstliche Neuronale Netze, Evolutionäre Algorithmen und Data Mining

⁶Fuzzy Control, z.B. [79, 80, 81]

unterteilt sich in drei Schritte:

1. Aggregation: Mit geeigneten Operatoren wird der WENN-Teil einer Regel ausgewertet.
2. Aktivierung: Hierbei werden der aggregierte WENN-Teil und der DANN-Teil (Konklusion) miteinander verknüpft. Damit wird festgelegt, in wie weit eine Regel aktiviert ist.
3. Akkumulation: Die aktivierten Regeln werden miteinander verbunden.

Als logische Operatoren zur Verknüpfung zweier unscharfer Mengen μ_1 und μ_2 stehen eine Vielzahl von mathematischen Formulierungen zur Verfügung, die wichtigsten Operatoren sind in Tabelle 1.3 zusammengefasst. Vornehmlich werden zur Aktivierung UND-Operatoren und in

UND	ODER
Minimum: $\mu = \min\{\mu_1; \mu_2\}$	Maximum: $\mu = \max\{\mu_1; \mu_2\}$
Produkt: $\mu = \mu_1 \cdot \mu_2$	algebraische Summe: $\mu = \mu_1 + \mu_2 - \mu_1 \cdot \mu_2$

Tabelle 1.3: Unscharfe Operatoren

der Akkumulation ODER-Operatoren eingesetzt.

Schließlich muss aus der unscharfen Menge, die nach der Regelauswertung vorliegt, ein Ausgangswert y gewonnen werden. Dazu stehen Methoden zur Defuzzyfizierung zur Verfügung, die sich danach unterscheiden, ob die zu defuzzyfizierende Menge analytisch oder numerisch (als Singletons) vorliegt, wie in der rechnergestützten Behandlung üblich. In einer Menge von Singletons $\mu(i)$ an den Stellen x_i kann das Maximum gesucht werden oder eine Schwerpunkt-methode angewendet werden:

$$y = \frac{\sum_i \mu(i)x_i}{\sum_i \mu(i)} \quad (1.17)$$

Für detaillierte Erläuterungen der Fuzzy-Logik und deren Anwendung muss im Rahmen der vorliegenden Arbeit auf einschlägige Grundlagenwerke verwiesen werden, z.B. [82, 84, 85, 80, 86]. Implementierungen von Fuzzy Systemen für die mit der Arbeit einhergehende Umsetzung in C++⁷ oder zur Entwicklung von Fuzzy- und Defuzzyfizierung sowie Inferenz sind unter [88, 89, 90] verfügbar.

Prinzipiell sind im weiteren Verlauf der Arbeit auftretende Problemstellungen auch mithilfe weiterer Teilgebiete der Computational Intelligence, zum Beispiel durch evolutionäre Algorithmen [91, 92, 93, 94, 95], lösbar.

⁷Die Programmiersprache wurde 1983 von Bjarne Stroustrup entwickelt und findet heute, auch durch später erfolgte Standardisierungen (beispielsweise nach ANSI), eine sehr breite Anwendung [87]

1.3 Offene Probleme

Im Markt befinden sich bereits diverse Möglichkeiten zur rechnergestützten Planung von Antriebssystemen und besonders zum Entwurf von Bewegungsvorgängen. Die Programme werden häufig von Systemanbietern von Antriebssystemen zur Verfügung gestellt. Auf vier exemplarische Planungstools, die im weiteren Verlauf geforderte Eigenschaften ansatzweise bereits besitzen, wird folgend verwiesen.

So existieren Programme, mit denen Bewegungen geplant werden, wie beispielsweise der Cam-Designer [96]. Durch die Angabe von Zwischenzuständen werden Bewegungsverläufe interpoliert und die kinematischen Größen berechnet. Es besteht oftmals die Möglichkeit mehrere Bewegungsachsen nebeneinander abzubilden. Die Produkte IndraSize [97] oder SizeMaxx [98] ermöglichen neben der Gestaltungsmöglichkeit von Bewegungsvorgängen auch die Möglichkeit aus dem jeweiligen Produktangebot der anbietenden Firmen eine passende Hardware auszuwählen. Dazu muss eine Auslegungsrechnung durchgeführt werden, die auf Kenndaten zum Lastverhalten und der Antriebsmechanik basiert. Darum sind ebenfalls Programme zur Modellierung von simplen Mechaniken integriert.

Die genannten Entwicklungstools sind jeweils exemplarisch für Software, welche die entsprechenden Aspekte behandelt. Zum Entstehungszeitpunkt der Arbeit ist die CAE-Software der Firma ControlEng mit dem Namen SERVOsoft (Version 3) [99] die am höchsten entwickelte auf dem Markt. Sie wird von zahlreichen Antriebsherstellern angewendet und vereint einige der vorgestellten Eigenschaften in einem Produkt.

Eine Möglichkeit zur Anpassung und Optimierung von Bewegungen basierend auf genetischen Algorithmen ist bekannt [100]. Weil die Abbildung von verbal vorhandenem Vorwissen komfortabler über andere Methoden (siehe Fuzzy-Logik aus Abschnitt 1.2.5) erreichbar ist, wird der Ansatz im Rahmen der vorliegenden Arbeit nicht weiter verfolgt.

Zur Verbesserung des Engineerings und zur weiteren Optimierung der entworfenen Antriebssysteme ergeben sich diverse mathematische Problemstellungen und ein Mangel an geeigneten Algorithmen. Um zu einem integrierten umfassenden Engineering zu gelangen, fehlt es an einer zweckdienlichen Vorgehensweise. Die offenen Probleme stellen sich im Einzelnen wie folgend dar:

- Die Auswahl von adäquaten Bewegungsgesetzen zur Interpolation von Bewegungsvorgängen ist aktuell manuell vom Anwender vorzunehmen und kann nur in einem durch den Menschen überschaubaren Rahmen erledigt werden. Dadurch ist die Qualität der Bewegung bezogen auf die Aufgabenstellung von Erfahrungswert des Anwenders abhängig. Eine automatische Selektion und auch Anpassung eines Bewegungsgesetzes hinsichtlich gestellter Optimierungskriterien ist notwendig, um zu leistungsfähigeren Antriebssystemen zu gelangen.
- Die Restriktionen, welche von ausgewählter Hardware resultieren, haben keinen algorithmischen Einfluss auf den Entwurf von Bewegungen. Eine Iteration zur Anpassung der Bewegungen muss vom Anwender händisch vorgenommen werden.

- Die entworfenen Bewegungen können nicht direkt als Sollgrößen für den Antriebsregler verwendet werden. Sie sind nicht an Abtastzeiten adaptierbar.
- Es sind keine Einflüsse der Achsen untereinander abbildbar. Wie im Beispiel des Längstaktmontageautomat deutlich wird, ist die Formulierung von kinematischen Abhängigkeiten zwischen den Achsen jedoch von großer Bedeutung.
- In Erweiterung zum letzten Punkt ist eine Beachtung von Havariefällen wünschenswert.
- Es existieren keinerlei Algorithmen zur Optimierung eines gesamten Antriebssystems hinsichtlich vorgegebener Kriterien.

Die Bedeutsamkeit der fehlenden Eigenschaften und Methoden für die Qualität des Engineerings und der geplanten Maschine werden im Rahmen der vorliegenden Arbeit verdeutlicht. Es ergeben sich die im nächsten Abschnitt formulierten Zielstellungen.

1.4 Ziele

Zum vollständigen Engineering von Servoantriebssystemen mit voneinander abhängigen Achsen unter der Zielstellung eines optimalen Systementwurfs müssen folgende Bedingungen gleichermaßen erfüllt werden:

- Die Bewegungsaufgabe der gesamten Maschine muss gelöst werden. Die Bewegungsführung jeder einzelnen Achse muss im Kontext die Anforderungen an den Prozess erfüllen.
- Die Beziehungen der Achsen untereinander müssen berücksichtigt werden. Dazu gehören auch Forderungen wie Kollisionsschutz im Fehlerfall.
- Es muss eine Auslegungsrechnung hinführend auf eine Hardware mit optimalem Wirkungsgrad durchgeführt werden oder es müssen Restriktionen aus verfügbarer Hardware oder mechanischer Grenzwerte aus dem Prozess (Maschine) eingehalten werden.
- Es muss eine energetisch, zeitlich oder schwingungsdynamisch optimale Bewegungsführung aller Antriebsachsen erreicht werden. In der Regel ist ein Kompromiss zwischen den Optimierungskriterien erforderlich.

Der Entwurf eines elektrischen Antriebssystems muss grundsätzlich sämtliche Kriterien parallel mit insgesamt höchstmöglichem Grad erfüllen.

Um die oben gestellte Forderungen ganzheitlich erfüllen zu können, reichen die bestehenden voneinander losgelösten Entwicklungsmethoden (siehe Abschnitt 1.2.4) nicht aus. Im Rahmen

der Arbeit wurde deshalb eine neue Methode aufgestellt. Den Vergleich zwischen dem alten Entwicklungsprozess und der neuen ganzheitlichen Methode stellt Abbildung 1.8 dar.

Soll ein Antriebssystem hin zu einer ökonomisch und ökologisch optimalen Gesamtlösung ge-

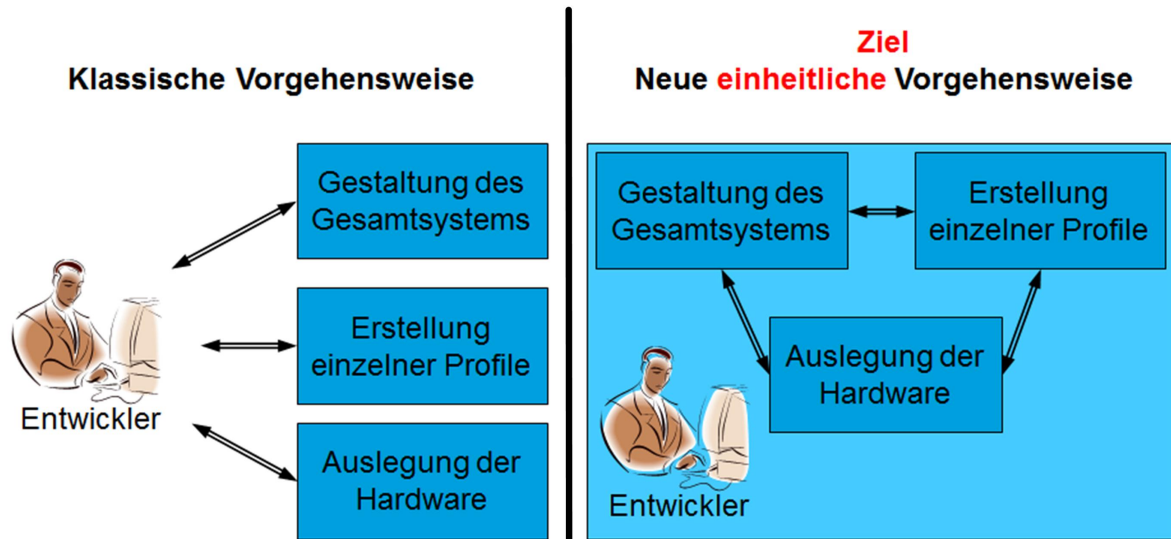


Abbildung 1.8: Vergleich zwischen alter und neuer Vorgehensweise

staltet werden, ist eine einheitliche Betrachtung von Auslegungsberechnung der Antriebe und der kinematischen Optimierung des gesamten Systems unerlässlich [101, 102].

Die These stützt sich auf das Argument, dass direkte technische Zusammenhänge zwischen den bisweilen isoliert betrachteten Aspekten eines Antriebssystems bestehen:

- Der Verlauf einer Bewegung impliziert kinematische Größen, die über die Auslegungsrechnung Einfluss auf die Auswahl von Hardware üben (Abschnitte 1.2.3 und 4.2).
- Die Gestaltung und Optimierung des gesamten Antriebssystems bedeutet eine unmittelbare Manipulation der Bewegungsabläufe (Abschnitt 5.2).
- Die Festlegung auf eine Hardware hat Restriktionen für den Bewegungsentwurf zur Folge.
- Anforderungen aus dem Prozess, beispielsweise Restriktionen oder Sollkurven einer Bewegung, bestimmen alle drei Aspekte.
- Ein optimiertes Antriebssystem ist nur entwickelbar, wenn sowohl der Entwurf von Bewegungen als auch die Auswahl von Hardware mit der Gesamtzielstellung abgeglichen werden.

Das Ziel ist somit eine Methode zu definieren, die unter Berücksichtigung sämtlicher Wechselwirkungen und Aspekte zu einer geeigneten Lösung gelangt. Dabei muss eine übergeordnete

Zielstellung, nämlich die der multikriteriellen Optimierung des gesamten Antriebssystems, erreicht werden. Es ist also eine Vorgehensweise zu entwickeln, in der die Abarbeitung einer Teilmethode, beispielsweise der Entwurf einer Bewegung, direkt durch die anderen Methoden, im Beispiel Systementwurf und Auslegung, beeinflussbar ist. Zusätzlich müssen Vorgehensweisen zur Optimierung der entstandenen Entwicklungsschritte mit globaler Zielstellung integriert werden.

Die geforderte Methode erfordert im Hinblick auf die Komplexität und der analytischen Variabilität der gestellten Aufgabe eine sehr große Menge von Rechenschritten. Unter der Erwartung von numerischen Berechnungsmethoden zur Anpassung des Antriebssystems an eine Vielzahl von Kriterien und Restriktionen ist eine rechnergestützte Umsetzung der Methode obligatorisch. Dem Applikationsingenieur⁸ muss ein Programm zur Seite gestellt werden, mit dem er die ihm gestellte Aufgabe zur Entwicklung eines Antriebssystems geeignet abbilden kann. Die Berechnung und Optimierung eines Systems unter genannten Aspekten soll algorithmisch in einer Weise erfolgen, dass alle Anforderungen automatisch berücksichtigt und eine optimierte Lösung entwickelt wird. Rekursive Methoden, die in einer großen Quantität auftreten, müssen weitgehend automatisch bearbeitet werden.

Aus den aufgezeigten offenen Problemen und aus der im letzten Absatz formulierten Zielstellung zum ganzheitlichen Engineering ergibt sich die Zielstellung für die vorliegende Arbeit.

- Ein übergreifendes Konzept zum ganzheitlichen Engineering muss entwickelt werden. Hierbei sind
 - zunächst detailliert die Anforderungen zu formulieren (Abschnitt 2.1)
 - und dann ein übergeordnetes Konzept abzuleiten (Abschnitt 2.2).
- Im Rahmen der weiteren Spezifizierung und der Umsetzung des Konzepts müssen Techniken entwickelt werden, mit denen der Entwurf von Bewegungen im Kontext zum ganzheitlichen Engineering erreichbar ist (Kapitel 3).
- Auch für die Auslegungsrechnung und Komponentenauswahl für elektrische Antriebssysteme müssen Lösungen aus der üblichen Vorgehensweise adaptiert werden, die dem neuartigen Konzept genügen (Kapitel 4).
- Um Systeme von Antriebssystemen zu entwerfen,
 - müssen sie in zur Implementierung geeigneter Weise abgebildet werden (Abschnitt 5.1)
 - und es müssen Optimierungsstrategien entwickelt werden (Abschnitt 5.2).
- Schließlich soll die Leistungsfähigkeit des Konzepts und seiner Umsetzung praktisch unter Beweis gestellt werden (Kapitel 6).

⁸In der Branche der Antriebstechnik übliche Bezeichnung für Entwicklungsingenieur

Dazu werden in Kapitel 2 ein Anforderungsportfolio und eine grundlegende algorithmische Lösungsstrategie entworfen. Drei wesentliche Teile des Konzepts sind der Bewegungsentwurf, der Hardwareentwurf und der Systementwurf. Im Kapitel 3 zum Bewegungsentwurf werden Methoden zur automatischen Berechnung von Bewegungsvorgängen mit beinhalteter Optimierung vorgestellt. Anschließend erfolgt in Kapitel 4 mit dem Thema Hardwareentwurf eine Aufarbeitung aller zur Auslegungsrechnung und Komponentenauswahl notwendigen Methoden. Die Ausarbeitung des neuen Gesamtkonzepts endet mit der Beschreibung der Algorithmen zum Systementwurf sowie der Optimierung der Antriebssysteme in Kapitel 5. Die drei spezifischen Kapitel zum neu entwickelten Konzept enden jeweils mit einem Einblick auf die jeweilige Implementierung der Algorithmen. In Kapitel 6 werden die erstellten Algorithmen und damit das neue Konzept anhand eines Beispiels aus der Fertigungstechnik und an einer physikalischen Modellmaschine validiert.

2 Neues ganzheitliches Konzept

2.1 Anforderungen

Nachdem im Abschnitt 1.4 die Zielstellung bezüglich eines neuartigen gesamtheitlichen Engineering von Antriebssystemen formuliert wurde, werden nachfolgend die konkreten Anforderungen zur Umsetzung definiert. Zu Beginn werden allgemeine Forderungen bestimmt, danach werden Anforderungen an den Entwurf von Bewegungsvorgaben formuliert. Ein weiterer Bestandteil des Konzepts zum ganzheitlichen Engineering ist die Auslegungsrechnung und die Auswahl von Hardware, die automatisch erfolgen sollen. Auch die Anforderungen an den Hardwareentwurf werden formuliert. Um das gesamte Antriebssystem und seine Abhängigkeiten modellieren zu können und zudem eine Optimierung der Gesamtzielstellung erreichbar zu machen, gelten zusätzlich diverse Vorgaben. Schließlich gibt es Anforderungen an die Algorithmen hinsichtlich der angestrebten Implementierung.

Zu den allgemeinen Anforderungen gehören:

- **Allgemeinheit des Konzepts**
Antriebssysteme können sehr verschiedenartig strukturiert und parametrisiert sein (siehe Abschnitt 1.2.2). Darum muss das Konzept in der Art formuliert sein, dass es möglichst allgemein für jede Ausführung von Antriebssystemen in möglichst vielen Maschinentypen, in denen sie angewendet werden, dienen kann.
- **Integration in bestehende Geschäftskultur und Branchentradition**
Die Auslegungsrechnung sowie das Bewegungsdesign und die Parametrierung von Antriebssystemen ist häufig eine Aufgabe, die von Vertriebs- und Applikationsingenieuren der Antriebshersteller erfüllt wird. Es ist somit eine entsprechende Gesamtlösung anzustreben, welche die technischen Fähigkeiten der anwendenden Personen, auch Entwickler mit kaufmännischem Hintergrund berücksichtigt. Komplexe Rechnungen, beispielsweise zur Auslegungsrechnung, sollen automatisch erfolgen.
- **Anerkennung vorliegender Konventionen und Beschreibungsformen**

Damit sich das neue Konzept etablieren kann, muss es bestehende einschlägige Begriffsdefinitionen, Herangehensweise und Datenstrukturen aufgreifen. Begriffe, die im Rahmen des Bewegungsdesigns geläufig sind, beispielsweise „Rast“ oder „Umkehr“, sind in Abschnitt 1.2.4 bereits erläutert worden. Auch die Strategie zum Erstellen von Bewegungen über die klassifizierten Randbedingungen müssen beibehalten werden, damit ein Grundverständnis des entwickelten Konzepts beim Anwender vorausgesetzt werden kann. Die formale Klassifizierung von Hardware über übliche Kennwerte muss aufgegriffen werden.

Zu den Anforderungen an den Entwurf von Bewegungen gehören:

- Berücksichtigung bestehender Normen und etablierter Berechnungsgrundlagen
Die eingangs erwähnten Bewegungsgesetze nach VDI-Richtlinie haben sich über viele Jahre etabliert und sind auch für die vorliegende Anwendung von Bedeutung. Um Kenntnisse über mathematisch geschlossen darstellbare Interpolationsmöglichkeiten zu nutzen, muss die Bewegungserstellung darauf aufbauen. Eine Möglichkeit, die zahlreichen neuartigen Bewegungsgesetze nachträglich zu integrieren, ist allerdings ebenso obligatorisch.
- Vollständig automatisierte Interpolation von Bewegungen
Im Sinne einer rekursiven Optimierung von Bewegungsvorgängen muss die Interpolation anhand eindeutig definierter Variablen ohne weitere Benutzereingriffe funktionieren.
- Interpolation hinsichtlich gegebener Optimierungsvorstellungen
Divers gelöste Bewegungsaufgaben sollen entsprechend gewünschter Maxime automatisch selektiert werden. Damit lassen sich bereits einzelne Bewegungsvorgänge, auch hinsichtlich eines übergeordneten Ziels, geschickt gestalten.
- Interpolation physikalisch und regelungstechnisch zweckmäßiger Sollwertvorgaben
Für kinematische Verläufe muss gewährleistet sein, dass kinematische Restriktionen bedingt durch Prozessgrößen oder ausgelegter Komponenten eingehalten werden. Die generierten Bewegungsverläufe sind Sollwertvorgaben für geregelte Servomotoren. Daher ist es wichtig, dass die Verläufe durch Folgeregelungen im Rahmen der erlaubten Schleppfehler¹ betrieben werden können. Beispielsweise ist zu beachten, dass Position und Geschwindigkeit stetige Verläufe, auch an den Übergängen von zwei Bewegungsabschnitten, aufzuweisen haben. Zusätzlich ist ein stetiger Beschleunigungsverlauf für die Ruckbegrenzung wünschenswert, wenn auch physikalisch nicht bindend notwendig. Die detaillierte Bestimmung von Regelungsparametern oder beispielsweise Strategien zur Schwingungskompensation sind sehr maschinen- und produktabhängig und können deshalb nicht Teil des entwickelten allgemeingültigen Konzepts zur Erstellung von Antriebssystemen sein.

Folgende Anforderungen an Auslegung und Komponentenauswahl sind zu berücksichtigen:

¹Der Schleppfehler ist die Regelabweichung der Position eines Antriebs

- **Berücksichtigung einschlägiger Regeln**
Wie bereits im Abschnitt 1.2.4 aufgezeigt, gibt es zahlreiche Regeln und Konventionen zur Auslegung von elektrischen Antriebskomponenten. Ziel im vorliegenden Fall muss eine Rechnung sein, die den gemeinsamen Regeln derart gerecht wird, dass eine Auslegung mit den typischerweise verfügbaren Daten möglich ist und gleichzeitig eine hinreichend genaue Modellierung des gesamten Antriebssystems vornehmbar ist, um eine legitime Hardwareauswahl treffen zu können.
- **Hinreichende Modellierung der Mechanik**
Eine Auslegungsrechnung sowie ein Systementwurf ist nicht möglich, ohne den Einfluss des Antriebsstranges auf die Bewegung und Eigenschaften des gesamten Systems zu kennen. Daher ist es unverzichtbar, die Mechanik des Antriebsstranges so zu modellieren, dass auf Übersetzungsverhältnisse oder Umsetzfaktoren von rotatorischen auf translatorische Bewegungen aber auch auf Massenträgheiten zurückgegriffen werden kann.
- **Wirtschaftliche Dimensionierung**
Die Auswahl von Komponenten muss in einer Weise getroffen werden, dass keine Überdimensionierung vorliegt, um Ressourcen zu schonen und wirtschaftlich sinnvolle Lösungen zu kalkulieren. Eine Unterdimensionierung stellt unter Anderem die Funktion, Sicherheit, Robustheit, Verfügbarkeit und die Langlebigkeit der Anwendung in Frage. Die Berücksichtigung kausaler Einsatzbedingungen von Komponenten, wie Umwelteinflüsse, Schutzklasse oder Arbeitsschutzbedingungen sowie Schnittstellen zur Prozessleitebene können kein Teil einer automatisierten Komponentenauswahl sein. Sie müssen von einer erfahrenen Fachkraft anwendungsspezifisch projiziert werden.
- **Breites Spektrum an einsetzbarer Hardware**
Das Konzept soll sich nicht auf die Lösungen einzelner Hersteller oder Produktreihen beschränken sondern für möglichst viele Produktkataloge gültig sein. Das ist erreichbar durch die bereits geforderte möglichst allgemeingültige Berechnung der Auslegung über weit verbreitete standardisierte Kennwerte von Antriebskomponenten.
- **Berücksichtigung des Anwenderwunsches**
Die automatische Auswahl von Komponenten ist derart einzuschränken, dass der Anwender des Konzepts in seinen Wünschen zur Auslegung nicht bevormundet wird. Es muss beispielsweise bei der Neugestaltung eines Bewegungsabschnittes abgefragt werden können, ob einem Upgrade der Komponenten vom Applikationsingenieur zugestimmt wird.

Bezüglich der Optimierung des Gesamtsystems gelten die folgenden Anforderungen:

- **Mathematisches Modell zur Beschreibung aller Zusammenhänge**
An das mathematische Modell besteht zum Einen der Anspruch, dass es in einer geeigneten Programmiersprache umgesetzt und implementiert werden kann, zum Anderen muss es auf konventionelle einfache und vollständige Weise formuliert sein, sodass alle Zusammenhänge erfasst sind und eine Optimierung des Antriebssystems möglich ist.

- **Strategien zur Optimierung des Antriebssystems**
Die Strategie zur Optimierung muss sich in der vorliegenden mathematischen Umgebung eingliedern lassen und zudem auf der ausgewählten Plattform implementierbar sein. Die Algorithmen sollen zu einem optimalen oder zumindest in der Art suboptimalen Antriebssystem gelangen, so dass ein Gewinn gegenüber konventioneller Methoden feststellbar ist. Dabei muss die Strategie dennoch für ein breites Spektrum von Problemstellungen und unterschiedlich gearteten Antriebssystemen funktionieren.
- **Zielgerichtete Vorgaben zur Interpolation von Bewegungsvorgängen**
Vorgaben für die Interpolation von Bewegungen müssen derart abgefragt werden, dass die Beschreibung aller Aufgabenstellungen möglich ist, aber auch so, dass ein definiertes Modell einer Bewegung jederzeit eindeutig bestimmbar ist. Dazu ist es notwendig, eine Konvention für die Beschreibung von Bewegungen zu entwickeln.

Die Anforderungen an die Algorithmen lauten:

- **Berechnungsaufwand**
Insgesamt ist die angestrebte Anwendung zeitlich unkritisch, da sie in der Planungsphase des Antriebssystems oder in einem Vorgang der Revision sowie der Produktumstellung angewendet wird, nicht aber im laufenden Betrieb einer Maschine. Es ist daher eine Komplexität der Algorithmen zulässig, die keine minutenlangen Wartezeiten im kreativen Prozess des Anwenders mit einem aktuellen durchschnittlichen Personalcomputer bedeuten.
- **Lauffähigkeit auf typischem PC**
Die Implementierung der Algorithmen soll in Zielrichtung auf eine breite Anwendbarkeit auf einer typischen Büroplattform und dem meistverbreitetem Betriebssystem in laufen². Der Besitz weiterer lizenzgeschützter kommerzieller Software muss für einen Endanwender verzichtbar sein.
- **Erweiterbarkeit**
Die Algorithmen müssen erweiterbar sein, um beispielsweise weitere Bewegungsgesetze, neue Optimierungsstrategien oder Anbindungen an Mehrkörpersimulationsplattformen sowie CAD-Systeme grundsätzlich integrieren zu können.
- **Genauigkeit**
Die berechneten Sollwertvorgaben für ein Antriebssystem müssen hinreichend genau sein, um die Antriebsaufgabe korrekt zu erfüllen und die einzelnen Servoachsen in einer ausreichenden Taktung mit hinreichend regelbaren Sollwerten führen zu können. Eine Größe stellt der geduldete Schleppfehler einer Achse dar, welcher von der Sollwertgenauigkeit nicht überboten werden muss. Der zulässige Schleppfehler kann in der Regel vom Applikationsingenieur projektiert werden. Typische Antriebsregler besitzen eine Genauigkeit von 16 bis 24 bit [33].

²Zum Entstehungszeitpunkt der Arbeit: Windows7 PC mit 4 GB Arbeitsspeicher und 64-bit Prozessor

- **Numerische Stabilität**
Die Algorithmen eines Tools müssen numerisch stabil sein. Optimierungsverfahren müssen durch ein Abbruchkriterium zu einem Abschluss kommen. Die Abbruchkriterien müssen der eingestellten Genauigkeit für die Berechnung angepasst werden.
- **Eingabefreundliche Benutzerschnittstelle**
Ein Antriebssystem wird besonders durch seine Vielzahl von kinematischen, kausalen und prozessbezogenen Zusammenhänge zu einem komplexen Konstrukt. Um zu einer guten Abbildung aller Zusammenhänge durch den Anwender zu gelangen und um ihm Zusammenhänge und Optimierungspotenzial vor Augen zu führen, bedarf es einer speziell dafür geeigneten Benutzerschnittstelle.
- **Fehlertoleranz**
Die Software muss stabil gegenüber falscher Bedienung und gegenüber konzeptioneller Entwicklungsfehler sein. Außerdem sind die errechneten Sollwertvorgaben auf Stetigkeit und Eignung zu überprüfen.

Nachdem die Ziele, die Methode zum ganzheitlichen Engineering und die abgeleiteten Anforderungen an eine rechnergestützte Umsetzung definiert sind, wird im Folgenden beispielhaft aufgezeigt, wie sich das Konzept algorithmisch umsetzen lässt. Im folgenden Abschnitt wird zuerst eine Übersicht über die zu entwickelnden Algorithmen und Teilsysteme gegeben.

2.2 Konzeptübersicht

Die algorithmische Umsetzung des Engineerings von Antriebssystemen ist in Abbildung 2.1 dargestellt.

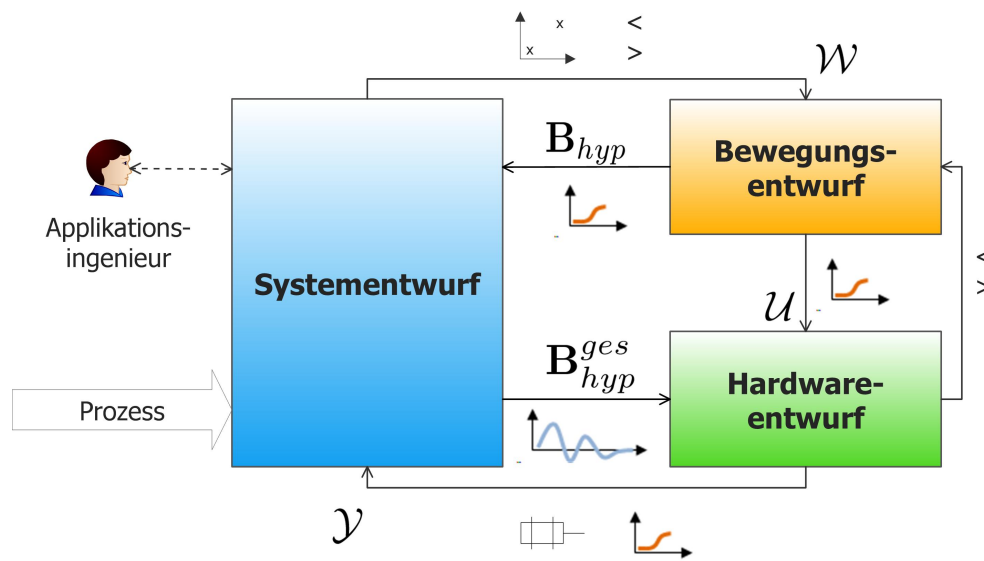


Abbildung 2.1: Gesamtübersicht

Die Algorithmen teilen sich in drei Teilsysteme, den Bewegungsentwurf, den Hardwareentwurf sowie den Systementwurf. Die Aufgaben grenzen sich dabei wie folgt ab:

1. Im Bewegungsentwurf werden anhand von Randinformationen (Bewegungspunkte), Restriktionen und Optimierungsvorgaben die Bewegungen ermittelt, die die Aufgabenstellung erfüllen.
2. Der Hardwareentwurf verfügt über ein mathematisches Modell der Mechanik des Antriebsstranges einer jeden Servoachse und die Eigenschaften der verfügbaren Komponenten. Somit kann mit geeigneter Auslegungsrechnung, bei Kenntnis eines Bewegungsverlaufes, eine Hardwareauswahl bestimmt und aus ihr resultierende Restriktionen festgelegt werden.
3. Der Systementwurf bildet den Kern der Benutzeranwendung. In dem Teilsystem werden alle gewünschten Bewegungsvorgänge vom Anwender wunschgemäß festgelegt und Optimierungsvorgaben festgehalten. Es besteht die Möglichkeit, Abhängigkeiten zwischen den Bewegungen der einzelnen Achsen abzubilden. Im Besonderen sind hier die Algorithmen zur Optimierung des gesamten Systems enthalten.

Die Zusammenarbeit der drei Teilsysteme und der auszutauschenden Signale wird im Folgenden detaillierter erläutert.

Der automatisierte Bewegungsentwurf bildet das Herz der algorithmischen Umsetzung des Konzepts. Durch eine Interpolation von Bewegungen anhand weniger Eingangsparameter, ohne weiteren Eingriff eines Applikationsingenieurs, ermöglicht er die Berechnung einer großen Anzahl von Bewegungen im Rahmen eines rekursiven Optimierungsalgorithmus. Zur

restriktions- und optimierungsbehafteten Interpolation notwendige Parameter sind dabei in der Signalmenge \mathcal{W} zusammengefasst.

Die Eingangssignale des Bewegungsentwurfs stammen aus den Algorithmen des Systementwurfs und beinhalten die im Diagramm graphisch symbolisierten Randpunkte einer Teilbewegung, sprich kinematische Randbedingungen für Strecke, Geschwindigkeit und Beschleunigung. Hinzu kommt die Zeitdifferenz und die durch Verhältniszeichen symbolisierten kinematischen Restriktionen. Eine mathematische Notation der Eingangssignale und wie daraus eine gewünschte Bewegung bestimmt wird, wird Kapitel 3 beschrieben. Als Ausgangsmenge \mathcal{U} liefert der Bewegungsentwurf jedoch in jedem Fall einen numerisch beschriebenen Bewegungsabschnitt.

Des Weiteren besitzt der Bewegungsentwurf Eigenschaften zur Berechnung von unoptimierten grober abgeschätzten hypothetischen Bewegungsabläufen \mathbf{B}_{hyp} , die zur Erzeugung von einem zu optimierenden Initialsystem notwendig sind. Zusätzlich steht dem Bewegungsentwurf ein zurückgeführtes Signal von kinematischen Restriktionen des Hardwareentwurfs zur Verfügung, dessen Herkunft nun aufgezeigt wird.

Der Hardwareentwurf ermöglicht eine, ebenfalls weitreichend automatisch getroffene, Auswahl von Hardware. Die physikalischen Grenzen der ausgewählten Komponenten bedingen kinematische Restriktionen, die an den Bewegungsentwurf wie an den Systementwurf weitergegeben werden. Um die Folgen eines erstellten Bewegungsabschnittes, bekannt aus dem Bewegungsentwurf, in die Hardwareauswahl einfließen lassen zu können, ist die Kenntnis über den gesamten Bewegungsablauf eines Maschinenzyklus obligatorisch. Er steht jedoch teilweise nur unoptimiert bzw. hypothetisch zur Verfügung und ist mit \mathbf{B}_{hyp}^{ges} notiert.

Ein Modell des Antriebsstranges liefert das Drehmoment und die Drehzahl am Motor über die Zeit in Abhängigkeit der kinematische Größen Geschwindigkeit und Beschleunigung sowie eines Lasteinflusses am wirkenden Ende des Antriebsstranges. Mit den genannten ermittelten Größen kann eine Auslegungsrechnung erfolgen, womit typische zur Auswahl von Komponenten geeignete Kennzahlen verfügbar sind.

Anschließend erfolgt eine Auswahl aus dem Vorrat an Komponenten in rekursiven Schritten. Eine Notation der Signale, der Berechnungsvorschriften sowie der Auswahlalgorithmen wird in Kapitel 4 beschrieben. Der Hardwareentwurf fasst die Ergebnisse der Auswahl und die entstandenen kinematischen Restriktionen in der Menge \mathcal{Y} zusammen.

Der Systementwurf als Kern der Benutzeranwendung ermöglicht die Abbildung von Bewegungsplänen von vielen Bewegungsachsen und deren Abhängigkeiten untereinander. Im Rahmen von Optimierungsstrategien zum Erlangen einer gewünschten Gesamtanwendung ruft er dabei den Bewegungsentwurf mit den bereits genannten Signalen auf und erhält dabei letztlich neue kinematische Restriktionen. Die Abbildung des gesamten Antriebssystems sowie deren inneren Zusammenhänge und Abhängigkeiten erfolgt dabei über dazu entworfene Petrinetze. Der Systementwurf bildet zudem die Basis für eine Benutzeranwendung. Durch ihn werden kinematische Prozessgrößen³, Randbedingungen von Strecken, Geschwindigkeiten,

³beispielsweise Zwangsbewegungen

Beschleunigungen und Zeiten, auch in Abhängigkeit zwischen den Achsen, modelliert. Auch die Optimierungskriterien für den gesamten Prozess, bestimmter Achsen oder einzelner Bewegungsabschnitte gibt der Entwickler vor. Zuletzt kommen auch die prozessbedingten oder anderweitig gewünschten Restriktionen vom Applikationsingenieur als Daten in den Systementwurf. Zudem fordert das System die Legitimation der Neuauswahl von größer oder kleiner dimensionierten Komponenten im Rahmen der Systemoptimierung. Der detaillierte Aufbau des Systementwurfes und entwickelter Optimierungsstrategien sind im Kapitel 5 weitergehend erläutert.

3 Neu- und Weiterentwicklungen zum Bewegungsentwurf

Den ersten Schwerpunkt der Arbeit bildet die Erläuterung der neuen Methoden zum Bewegungsentwurf. In drei Abschnitten werden das Vorgehen zur Berechnung von Bewegungsverläufen und die Selektion von Bewegungen gemäß Optimierungsvorgaben sowie die zugehörige Implementierung der neuen Algorithmen in der Software zum Engineering beschrieben.

Der Bewegungsentwurf ist der neu entwickelte Systemteil, der die Bewegungsvorgänge berechnet. Im Folgenden wird die Bestimmung eines einzelnen Bewegungsabschnittes betrachtet. Die Eingangssignale, die durch den System- und Hardwareentwurf verfügbar sind, werden in der Menge \mathcal{W} zusammengefasst, die sich wie folgt zusammensetzt:

$$\mathcal{W} := \{\vec{x}_0; \vec{x}_z; \vec{o}; \mathbf{K}_{ultimo}; T\}, \quad (3.1)$$

wobei mit \vec{x}_0 und \vec{x}_z die Start- und Zielzustände eines Bewegungsabschnitts gekennzeichnet sind. Hinzu kommen die für den Abschnitt relevanten Optimierungskriterien \vec{o} und die aus diversen Restriktionen kombinierten ultimativen kinematischen Grenzwerte in der Matrix \mathbf{K}_{ultimo} sowie die Bewegungszeit T . Die Menge \mathcal{U} fasst die Ausgangssignale des Bewegungsentwurfs zusammen.

$$\mathcal{U} := \{\mathbf{B}; \mathbf{K}_B\} \quad (3.2)$$

Sie enthält die Bewegung \mathbf{B} als Matrix mit Zeit-, Positions-, Geschwindigkeits-, Beschleunigungs- und Ruckwerten als Spalten und eine Matrix mit den kinematischen Minimal- und Maximalwerten der Bewegung \mathbf{K}_B . Hinzu kommen die Mengen \mathcal{F} und \mathcal{A} aller zur Verfügung stehenden Funktionen und Algorithmen zur Berechnung von Bewegungsverläufen sowie deren Fähigkeiten. Sie werden im folgenden Abschnitt detailliert definiert.

Die Berechnung der Bewegungsverläufe muss auf Basis der in \mathcal{W} gegebenen Größen erfolgen. Dazu werden zuerst die Randbedingungen \vec{x}_0 und \vec{x}_z normiert zu \vec{x}_{RN} und eine Diskretisierung vorgenommen (diskrete Randbedingungen \vec{x}_D), bevor eine Vorauswahl geeigneter mathematischer Vorlagen getroffen wird. Mit allen in der Vorauswahl enthaltenen mathematischen Bewegungsgesetzen \mathcal{F}_A wird eine Schar von Bewegungsverläufen \mathbf{B}_{Schar} berechnet und anschließend eine Selektion auf Basis der Optimierungskriterien durchgeführt. Optional wird

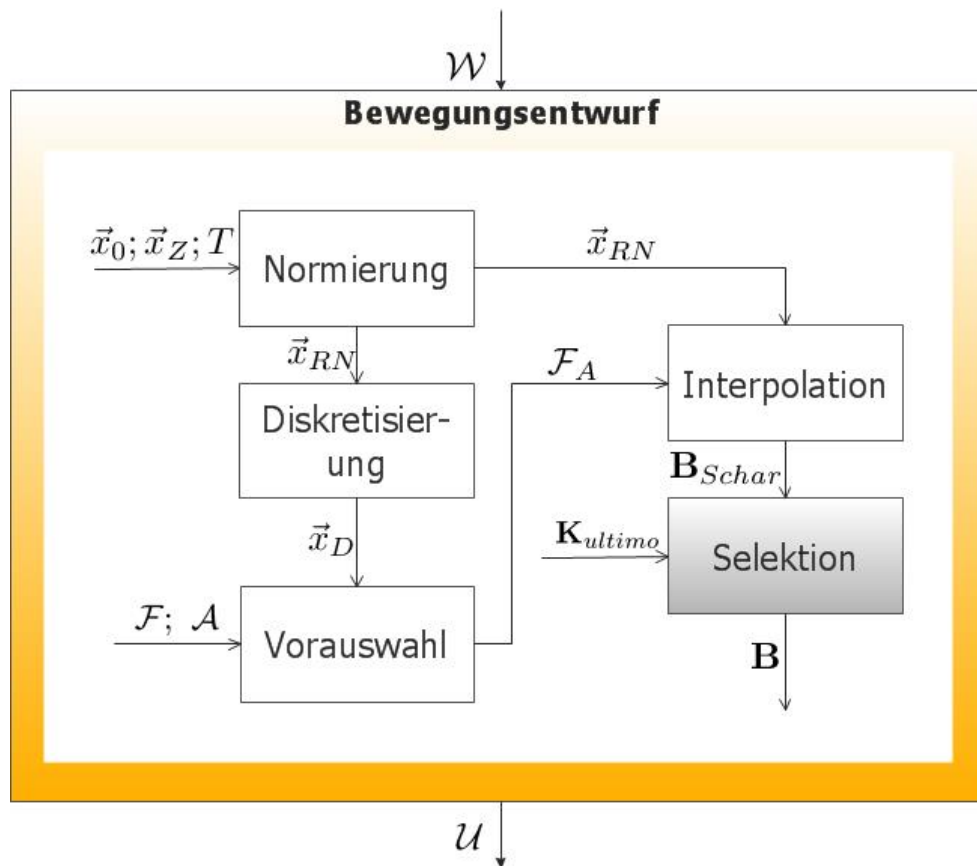


Abbildung 3.1: Bewegungsentwurf

zuvor eine Selektion unter Einbeziehung der kinematischen Restriktionen durchgeführt, um anschließend erst aus der erfolgten Teilmenge eine weitere Selektion nach Optimierungskriterien vorzunehmen. Abbildung 3.1 verdeutlicht das beschriebene Verfahren.

Im folgenden Abschnitt wird die Berechnung von Bewegungsverläufen detaillierter erläutert. Die Algorithmen zur Selektion werden danach gesondert behandelt.

3.1 Berechnung von Bewegungsverläufen

Die Berechnung von Bewegungsverläufen wird in zwei Abschnitten erläutert. Um die Bewegungsverläufe ermitteln zu können, müssen zuerst die gemäß VDI-Richtlinie 2143 [60] verwendeten Beziehungen eingeführt und dem neuen Konzept dienlich aufgearbeitet werden. Im zweiten Abschnitt wird der eigentliche Algorithmus zur Interpolation von Bewegungen schrittweise hergeleitet.

3.1.1 Aufarbeitung von Funktionen für die Bewegungsberechnung

Abbildung 3.2 zeigt einen Vergleich verschiedener Bewegungsgesetze anhand dessen exemplarischen¹ qualitativen Verläufen.

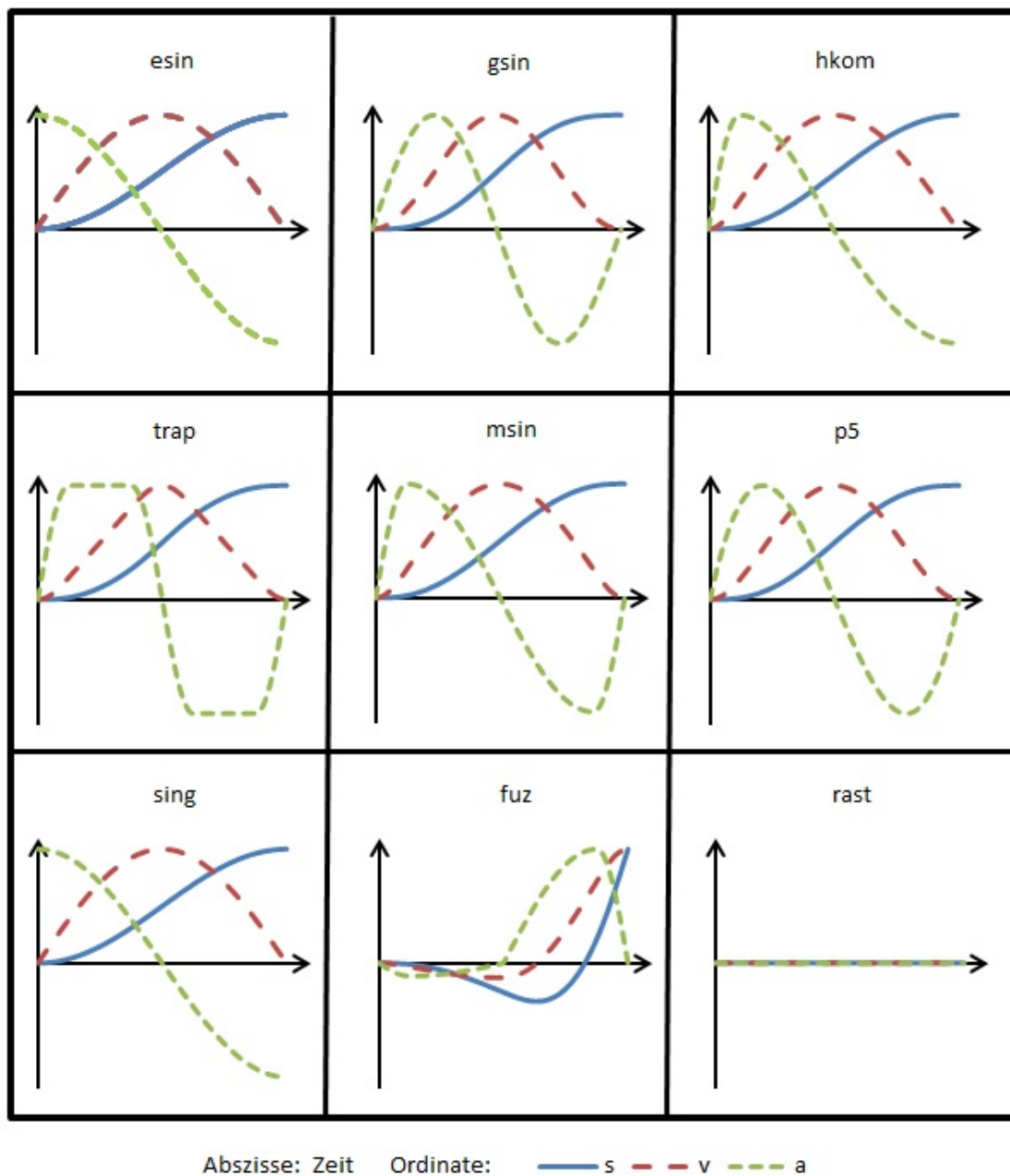


Abbildung 3.2: Vergleich von Bewegungsgesetzen aus [60, 103]

¹Es ist jeweils nur ein Fall und eine Parameteradaption dargestellt, siehe auch Tabelle 3.1

Die Diagramme enthalten jeweils den zeitlichen Verlauf der Strecke s und deren beiden Ableitungen, die Geschwindigkeit v und die Beschleunigung a . Mit Ausnahme der Rast ($rast$) und dem Fuzzygestützten Verlauf (fuz) sind Rast in Rast Bewegungen dargestellt. Das ist eine Bewegung vom Stillstand in eine andere Position im Stillstand. Mit dem Fuzzygestützten Verlauf sind besonders komplexe Bewegungen möglich, weil durch Fuzzy-Logik adaptierbare Zwischenpositionen angefahren werden.

Die Graphen der Geneigten Sinuslinie ($gsin$) und des Polynoms fünften Grades ($p5$) unterscheiden sich nicht augenscheinlich, sie lassen sich aber unterschiedlich an explizite Bewegungsaufgaben anpassen und haben eine leicht unterschiedliche Beschleunigungscharakteristik, wodurch die Geneigte Sinuslinie beispielsweise ruckärmer ist. Auch die anscheinend gleich verlaufenden Einfache Sinuslinie ($esin$) und Sinus-Gerade-Kombination ($sing$) unterscheiden sich durch einen Bereich konstanter Geschwindigkeit, über den nur die Sinus-Gerade-Kombination verfügt.

Im Rahmen der vorliegenden Arbeit wurde mit der folgenden Menge \mathcal{F} an Interpolationen gearbeitet²:

$$\mathcal{F} := \{ \vec{f}_{esin}; \vec{f}_{gsin}; \vec{f}_{hkom}; \vec{f}_{trap}; \vec{f}_{msin}; \vec{f}_{p5}; \vec{f}_{sing}; \vec{f}_{fuz}; \vec{f}_{rast} \} \quad (3.3)$$

Es handelt sich jeweils um einen Vektor \vec{f} von Funktionen, da jeweils analytische Vorschriften für die Ableitungen des Positionsverlaufs enthalten sind. Die Menge \mathcal{A} fasst Matrizen \mathbf{A} , bestehend aus Spalten möglicher diskreter Randbedingungen \vec{a}_i , zusammen. Die Tabelle 3.1 stellt eine Einordnung der genannten Bewegungsgesetze in den Kontext des neuen Konzepts dar. Die Bewegungsgesetze sind entsprechend ihrer Eignungen für verschiedene Kombinationen von Randbedingungen erfasst. Dabei sind Fälle, bei denen bestimmte Bewegungsgesetze über Parameter adaptiert werden können mit (λ) gekennzeichnet. Formell ist die Klassifizierung durch eine Matrix \mathbf{A} festgeschrieben. Die Matrizen sind in einer Menge \mathcal{A} zusammengefasst.

$$\mathcal{A} := \{ \mathbf{A}_{esin}; \mathbf{A}_{gsin}; \mathbf{A}_{hkom}; \mathbf{A}_{trap}; \mathbf{A}_{msin}; \mathbf{A}_{p5}; \mathbf{A}_{sing}; \mathbf{A}_{fuz}; \mathbf{A}_{rast} \} \quad (3.4)$$

Eine Matrix \mathbf{A} setzt sich damit zusammen aus:

$$\mathbf{A} = [\vec{a}_i]_{1 \times i_{max}} \quad \text{mit } i = 1; \dots; i_{max} \quad (3.5)$$

Dabei entspricht jedes i einer möglichen Kombination von Start- und Zielbedingungen, und somit i_{max} der Anzahl der lösbaren Bewegungskombinationen eines Bewegungsgesetzes. Zur Erweiterung des Bewegungsentwurfes um neue mathematische Vorlagen und Algorithmen müssen die Mengen \mathcal{F} und \mathcal{A} entsprechend erweitert werden.

Die zitierte VDI-Richtlinie stellt Formeln für die analytische Berechnung von Bewegungen bis zur zweiten zeitlichen Ableitung bereit. Die für das neue Konzept notwendige dritte Ableitung aller Bewegungsformen kann, wie folgend exemplarisch an der Einfachen Sinuslinie dargestellt,

² $esin$ = Einfache Sinuslinie, $gsin$ = Geneigte Sinuslinie, $hkom$ = Harmonische Kombination, $trap$ = Modifiziertes Beschleunigungstrapez, $msin$ = Modifizierte Sinuslinie, $p5$ = Polynom 5. Grades, $sing$ = Sinus-Gerade-Kombination, fuz = Fuzzygestützter Verlauf, $rast$ = Rast

diskrete Randbedingungen	R	G	U	B
Rast (R)	esin, gsin, msin, p5, trap	msin(λ), p5	p5, trap(λ), hkom(λ)	p5
Geschwindigkeit (G)	...	msin, p5	hkom(λ)	p5
Umkehr (U)	esin, p5, sing(λ)	p5
Bewegung (B)	p5

Tabelle 3.1: Fähigkeiten der Bewegungsgesetze, durch Analyse von [60]

berechnet werden:

$$\begin{aligned}
 s(t) &= -\frac{1}{2}\cos(\pi t) + \frac{1}{2} & (3.6) \\
 v(t) &= \frac{1}{2}\pi\sin(\pi t) \\
 a(t) &= \frac{1}{2}\pi^2\cos(\pi t) \\
 r(t) &= -\frac{1}{2}\pi^3\sin(\pi t)
 \end{aligned}$$

Eine vollständige Ergänzung aller in der VDI 2143 enthaltenen Bewegungsgesetze wurde mithilfe des Computeralgebrasystems MAPLE erstellt. Die vollständigen Ergebnisse für die jeweiligen Formeln des Ruckverlaufs sind im Anhang 9.1 dokumentiert. Sämtliche im Voraus genannten Bewegungsgesetze sind, einschließlich aller drei Ableitungen, zur Umsetzung des neuen Konzepts implementiert [104] und anschließend umfangreich evaluiert [105] worden.

3.1.2 Neues Verfahren zur Bewegungsberechnung

Um einer Interpolation von Bewegungen durchführen zu können, sind die folgenden Schritte, die bereits im Ablaufdiagramm aus Abbildung 3.1 zugeordnet sind, notwendig:

1. Normierung der Randbedingungen
2. Diskretisierung
3. Vorauswahl

4. Interpolation

5. Rücknormierung.

Die Verfahrensschritte werden nachfolgend beschrieben.

Normierung der Randbedingungen

Zur Berechnung der Bewegungsvorgänge ist es notwendig, die Randbedingungen zu normieren. Die Normierung orientiert sich an der VDI-Richtlinie 2143 [60], Abschnitt 5. Somit kann der Vektor der normierten Randbedingungen \vec{x}_{RN} des neuen Konzepts folgendermaßen formuliert werden:

$$\Delta s = s_z - s_0 \quad (3.7)$$

$$\vec{x}_{RN} = \begin{pmatrix} \frac{v_0 T}{\Delta s} \\ \frac{a_0 T^2}{\Delta s} \\ 1 \\ \frac{v_z T}{\Delta s} \\ \frac{a_z T^2}{\Delta s} \end{pmatrix}, \text{ falls } \Delta s \neq 0 \quad (3.8)$$

$$\vec{x}_{RN} = \begin{pmatrix} v_0 T \\ a_0 T^2 \\ 0 \\ v_z T \\ a_z T^2 \end{pmatrix}, \text{ falls } \Delta s = 0 \quad (3.9)$$

Die aus der VDI-Richtlinie bekannte Berechnung ist um eine Fallunterscheidung erweitert, damit auch Vorgänge ohne Positionsdivergenz Δs zwischen Start- und Zielposition (s_0 und s_z) algorithmisch erfasst werden können. Es werden zudem die Geschwindigkeiten und Beschleunigungen der Startbedingung v_0 und a_0 sowie der Zielbedingung v_z und a_z und die Fahrzeit T zur Normierung herangezogen.

Diskretisierung

Im Rahmen der Diskretisierung werden die Randbedingungen, also Startbedingung \vec{x}_0 und Zielbedingung \vec{x}_z , diskretisiert zu \vec{x}_D . Für das neue Verfahren zum Bewegungsentwurf ist die entwickelte Diskretisierung notwendig, um die Art der Anforderung klassifizieren und mit den Eigenschaften von den im vorigen Abschnitt aufgearbeiteten Funktionen abgleichen zu können. Folgend ist neben den Vektoren der Start- und Zielbedingung (\vec{x}_0 und \vec{x}_z) auch die Zusammen-

setzung des Vektors für die diskreten Randbedingungen \vec{x}_D notiert:

$$\vec{x}_0 = \begin{pmatrix} s_0 \\ v_0 \\ a_0 \end{pmatrix}; \vec{x}_z = \begin{pmatrix} s_z \\ v_z \\ a_z \end{pmatrix}; \vec{x}_D = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad (3.10)$$

Die Komponenten nehmen, mit Ausnahme von x_2 , entsprechend der Zuordnungsvorschrift $d(e)$ den Wert Eins an, wenn die zugehörige Randbedingung e (siehe Zuordnung aus Gleichung 3.13) sich von Null unterscheidet. Anderenfalls wird die Komponente mit der Null bewertet. Die Komponente x_2 bildet einen Sonderfall in der Zuordnungsvorschrift und bewertet analog den Betrag der Positionsdifferenz (entspricht Δs) zwischen s_z und s_0 unter Berücksichtigung einer Toleranz. Die Toleranz ist kleiner als die halbe zulässige Regelabweichung des Antriebsreglers, sprich dem halben vom Applikationsingenieur vorgegebenen maximalen Schleppfehler ϵ_S , um sicherzustellen, dass keine Positionsdifferenz vom Regler auszuregeln ist.

$$x_2 = \begin{cases} 0 & \text{für } \|s_z - s_0\| < 0,5\epsilon_S \\ 1 & \text{sonst} \end{cases} \quad (3.11)$$

$$d(e) = \begin{cases} 0 & \text{für } e = 0 \\ 1 & \text{sonst} \end{cases} \quad (3.12)$$

$$x_0; x_1; x_3; x_4 \xleftarrow{d(e)} v_0; a_0; v_z; a_z \quad (3.13)$$

Mit dem Vektor \vec{x}_D steht somit eine Typisierung der Bewegungsaufgabe zu Verfügung, welche die geeigneten Berechnungsmöglichkeiten eingrenzt. Mithilfe der diskreten Randbedingungen kann dann eine Vorauswahl aus den verfügbaren Bewegungsgesetzen getroffen werden.

Vorauswahl

Aus dem Vorrat an Funktionen \mathcal{F} wird eine Vorauswahl

$$\mathcal{F}_A \subseteq \mathcal{F} \quad (3.14)$$

getroffen, die nur diejenigen Funktionen enthält, welche, bezogen auf die diskreten Randbedingungen und die Klassifizierung der jeweiligen Funktion, geeignet ist, die Bewegungsaufgabe zu lösen. Das heißt eine Funktion \vec{f} ist als \vec{f}_A zugehörig zu \mathcal{F}_A , wenn ihre Matrix aus \mathcal{A} einen Vektor \vec{a}_i enthält, der gleich \vec{x}_D ist. Für die Menge der ausgewählten Funktionen ergibt sich somit:

$$\mathcal{F}_A = \{\vec{f}_A \mid \vec{f}_A \in \mathcal{F} \wedge \vec{x}_D \in \mathbf{A}\} \quad (3.15)$$

Durch die Vorauswahl ist somit gewährleistet, dass nur Funktionen verwendet werden, deren Analysis sich eignet, um die vorliegende Bewegungsaufgabe, also die Interpolation zu lösen.

Interpolation

Die Aufgabe der Interpolation liegt darin, eine Schar von Bewegungsvorgängen \mathbf{B}_{Schar} (Abbildung 3.3) zu berechnen, aus der anschließend eine vorgabengerechte optimale Bewegung selektiert wird (Abschnitt 3.2). Eine Schar von Bewegungsvorgängen umfasst die Bewegungsfunktionen aller mathematischen Vorlagen, die unter der Vorauswahl \mathcal{F}_A zusammengefasst sind. Einige Bewegungen lassen sich zudem über einen Parameter λ manipulieren (Tabelle 3.1), in der beispielhaften Abbildung durch die Legende mit „sing 0.1“ bis „sing 0.9“ bezeichnet.

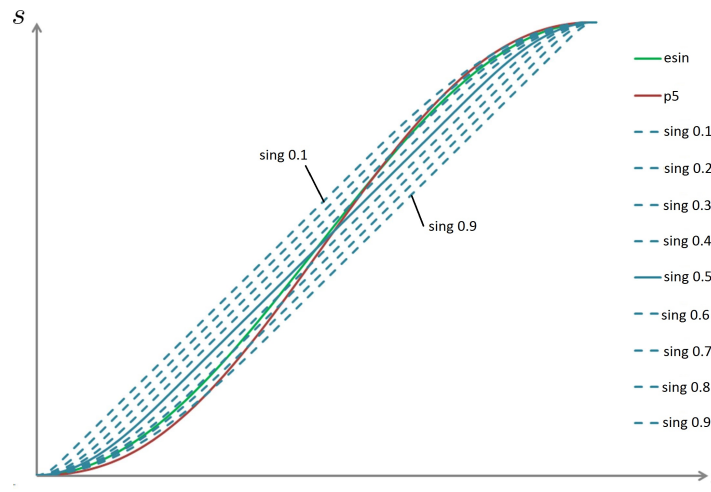


Abbildung 3.3: Schar von Strecken (Umkehr-Umkehr)

Hierbei beinhaltet die berechnete Schar Kurven für n verschiedene Werte des Parameters λ . Eine Bestimmung der Abstufung der Scharkurven kann über folgende Gleichung errechnet werden:

$$\lambda_i = i \frac{1}{n} \text{ mit } i = 1; \dots; (n - 1) \quad (3.16)$$

Mathematische Vorlagen für Bewegungsvorgänge, wie auch die aus der verwendeten VDI-Richtlinie, erlauben neben der Manipulation über den vorgestellten Parameter auch eine gezielte Anpassung der Bewegungsverläufe an kinematische Vorgaben. Derartige Vorgaben können explizite Grenzwerte für Geschwindigkeiten oder Beschleunigungen sein. Bei genauerer Analyse der mathematischen Modelle ist jedoch ersichtlich, dass derartige Parameter eindeutig analytisch mit λ zusammenhängen. Wegen der im Rahmen der vorliegenden Arbeit übergeordneten Optimierungs- und Selektionsstrategie wird die analytische Anpassung an kinematische Parameter nicht verwendet. Indirekt ist sie über die numerische Quantisierung von λ berücksichtigt.

Um eine Schar von Bewegungsvorgängen zu erstellen, werden sämtliche vorausgewählte Funktionen \vec{f}_A zu äquidistanten zeitlichen Schritten T_A ausgewertet. Im normierten Bereich entspricht das einer Auswertung in

$$n = \frac{T}{T_A} + 1 \quad (3.17)$$

äquidistanten Schritten im Bereich $[0; 1]$. Eine einzelne normierte Bewegung \mathbf{B}_N der Schar stellt sich als eine $(n \times 5)$ Matrix dar, in der die Spalten die Werte der Zeit \vec{t} , Strecke \vec{s} , Geschwindig-

keit \vec{v} , Beschleunigung \vec{a} und Ruck \vec{r} enthalten.

$$\mathbf{B}_N = f(\vec{f}_A, \lambda) = \left(\vec{t} \quad \vec{s} \quad \vec{v} \quad \vec{a} \quad \vec{r} \right)_{(n \times 5)} \quad (3.18)$$

Abbildung 3.4 zeigt den qualitativen Verlauf \vec{s} einer solchen Bewegung mit den Ableitungen \vec{v} , \vec{a} und \vec{r} nach der Zeit t .

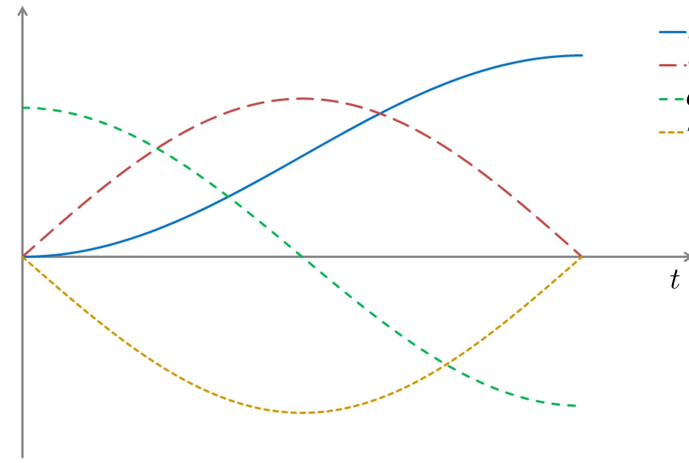


Abbildung 3.4: Graph einer Bewegung mit Ableitungen

Der neuartige Bewegungsentwurf sieht die Selektion einer Bewegung aus der erstellten Schar von Bewegungen vor. Hierzu, sowie zur Verwendung der Kurven als Führungsgrößen für die Antriebsregler, ist eine Rücknormierung und Transformation der Größen notwendig.

Rücknormierung

Eine normierte Bewegung lässt sich wieder Rücknormieren auf die Absolutbewegung \mathbf{B} (Erweiterte Formel zur Rücknormierung aus [60]):

$$\mathbf{B} = \mathbf{B}_N \cdot \begin{pmatrix} \Delta T & 0 & 0 & 0 & 0 \\ 0 & (\Delta s + s_0) & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta s}{T} & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta s}{T^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{\Delta s}{T^3} \end{pmatrix} \quad (3.19)$$

Dazu wird die Matrix der normierten Bewegung mit der in Gleichung notierten Matrix multipliziert. Durch die Addition von s_0 in der zweiten Spalte findet zudem eine Transformation der Bewegung zur Anfangsposition statt.

Für die im weiteren Verlauf der Arbeit entwickelten Selektions- und Optimierungsverfahren sind die Extrema der Ableitungen des Bewegungsverlaufs von Bedeutung. Die Matrix \mathbf{K}_B enthält die

Minimal- und Maximalwerte:

$$\mathbf{K}_B = \begin{pmatrix} \min(\vec{s}) & \max(\vec{s}) \\ \min(\vec{v}) & \max(\vec{v}) \\ \min(\vec{a}) & \max(\vec{a}) \\ \min(\vec{r}) & \max(\vec{r}) \end{pmatrix} \quad (3.20)$$

Bevor eine einzelne Bewegung vom Bewegungsentwurf ausgegeben werden kann, muss aus der eventuell vorhandenen Schar erst eine Bewegung ausgewählt werden. Strategien zur Selektion werden im folgenden Abschnitt erläutert.

3.2 Entwickelte Selektionsverfahren

Grundsätzlich läuft die Selektion wie im Diagramm in Abbildung 3.5 dargestellt ab. Zunächst werden aus der Schar von Bewegungsverläufen diejenigen entfernt, die nicht den kinematischen Restriktionen genügen. Danach wird je Optimierungsvorgaben eines von zwei neu entwickelten Verfahren zur Selektion eingesetzt.

Bei der kinematischen Selektion kann es im Bewegungsentwurf erstmals zu dem Zustand kom-

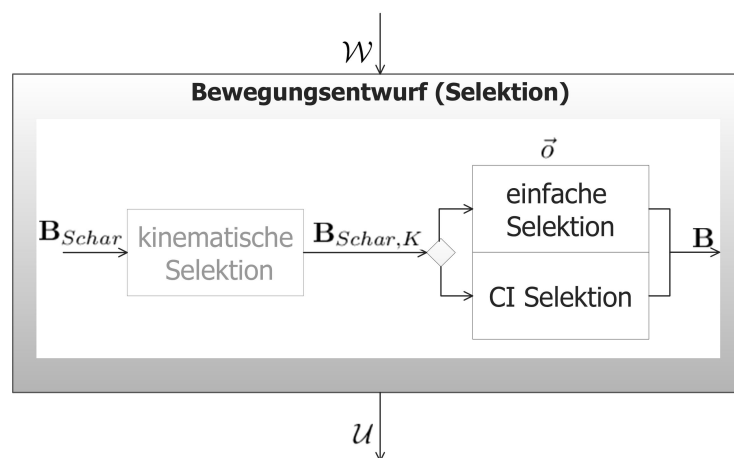


Abbildung 3.5: Übersicht über das Selektionsverfahren

men, dass keine Lösung für eine vorliegende Bewegungsaufgabe gefunden werden kann. Um bestimmen zu können, welche kinematischen Forderungen eine Bewegung erfordert, kann die kinematische Selektion optional übergangen werden. Aus der verbleibenden Schar $\mathbf{B}_{Schar,K}$ muss schließlich eine einzelne Bewegung \mathbf{B} selektiert werden. Die „einfache Selektion“ selektiert anhand eines diskreten eindeutigen Optimierungszieles, während die „CI Selektion“ so entwickelt ist, dass eine multikriterielle Optimierung möglich wird.

3.2.1 Kinematische Selektion

Die kinematische Selektion schränkt die Schar \mathbf{B}_{Schar} auf Bewegungen ein, die die Grenzwerte aus den Restriktionen K_{ultimo} des betroffenen Bewegungsabschnitts einhalten. Die Formeln zur Berechnung von K_{ultimo} sind in Kapitel 5 angegeben. Für die kinematische Selektion wird daher hergeleitet, dass eine Bewegung \mathbf{B} genau dann zu $\mathbf{B}_{Schar,K}$ gehört, wenn gilt:

$$\mathbf{K}_{ultimo} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \leq \left(\min(\vec{s}) \quad \min(\vec{v}) \quad \min(\vec{a}) \quad \min(\vec{r}) \right)^T \quad (3.21)$$

$$\wedge \mathbf{K}_{ultimo} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \geq \left(\max(\vec{s}) \quad \max(\vec{v}) \quad \max(\vec{a}) \quad \max(\vec{r}) \right)^T$$

Das heißt, die in K_{ultimo} enthaltenen Minima müssen allesamt kleiner sein als die aus dem Verlauf der betrachteten Bewegung. Umgekehrt müssen die Maxima aus den ultimativen Restriktionen größer sein als jene aus der betrachteten Bewegung. Somit ist sichergestellt, dass über den gesamten Bewegungsverlauf die kinematischen Restriktionen erfüllt sind. In Abbildung 3.6 ist eine exemplarische Schar von Bewegungen dargestellt. Mit den waagerechten rot gepunkteten Linien sind zudem die für die jeweilige Ableitung geltenden kinematischen Grenzen aus den Restriktionen aufgetragen.

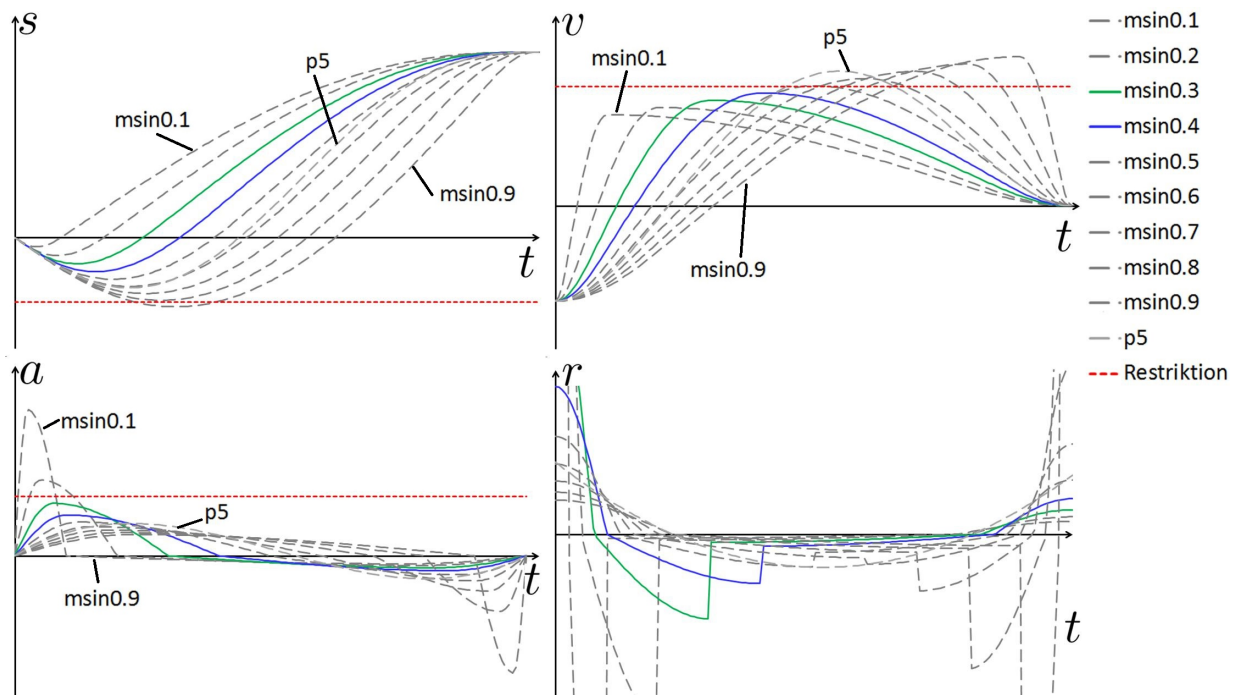


Abbildung 3.6: Selektion nach kinematischen Restriktionen

Nur die Bewegungen, die mit dem „Modifizierten Sinus (msin)“ und den Parametern $\lambda = 0,3$ (msin 0.3) sowie $\lambda = 0,4$ (msin 0.4) berechnet wurden, halten alle geforderten Grenzen ein.

Durch die Kumulation der drei Diagramme für s , v und a wird die kinematische Selektion graphisch interpretierbar. Zudem wird an dem Diagramm deutlich, dass durch Beschleunigungsverläufe, die nicht stetig differenzierbar sind Ruckverläufe mit Sprüngen entstehen und dass scheinbar geringe Änderungen im Beschleunigungsverlauf zu Änderungen von Größenordnungen im Ruckverlauf führen können. Die schwere Einschätzung der Verläufe ist einer der Gründe, weshalb auf Werte zur Begrenzung vom Ruck innerhalb des neuen Konzepts verzichtet wird. Ein weiterer Grund ist, dass der Ruck für den Applikationsingenieur eine vergleichsweise unanschauliche Größe ist. Der Verlauf ist jedoch für die weiteren nachfolgend beschriebenen Selektionsverfahren von Bedeutung.

3.2.2 Einfache Selektion

Die entwickelte Vorgehensweise zum Bewegungsentwurf sieht die Auswahl einer Bewegung aus einer Mannigfaltigkeit hinsichtlich Optimierungskriterien vor. Aus einer Schar von Bewegungen kann mittels Verlustfunktionen die Bewegung mit geringstem Energieverbrauch, kleinstem maximal auftretendem Ruck oder minimal zu erwartendem Zeitaufwand ermittelt werden. Die Einfache Selektion ist zweckmäßig und solide algorithmisch umsetzbar, jedoch lässt sie keine Kompromisse zwischen den Optimierungskriterien zu. Da nicht selten eine Optimierung eines Prozesses einseitigen Ansprüchen genügen soll, ist das Verfahren für bestimmte Aufgabenstellungen zweckmäßig und wurde deshalb mathematisch umgesetzt.

Hierbei gibt es drei voneinander unabhängig betrachtete Optimierungsziele

- minimaler Energieverbrauch,
- kleinstes Maximum in Ruckverlauf
- sowie größtes Potenzial für Verkürzung der Bewegungsdauer,

für die Verlustfunktionen entwickelt wurden. Der Energieverlust einer Bewegung ist allgemein herleitbar aus der Definition der Arbeit $W = F \cdot s$ und dem 2. Newtonschen Grundgesetz $F = m \cdot a$. Demnach kann für die Energie einer Bewegung um Δs die Energie ΔW angenommen werden.

$$\Delta W = ma \cdot \Delta s \quad (3.22)$$

Wenn die Masse während der Bewegung konstant bleibt, und dass alle Bewegungen der Schar sich auf die gleiche Masse beziehen, was in allen Betrachtungen im Rahmen der Arbeit der Fall ist, dann kann die Masse als unveränderliche Proportionalitätskonstante angesehen werden und es gilt:

$$\Delta W \sim a \Delta s \quad (3.23)$$

Somit kann der Energieverbrauch einer (Teil-)Bewegung im Vergleich zu einer anderen über das Produkt aus Beschleunigung und Streckenänderung abgeschätzt werden. Bei numerisch gegebener

nen Bewegungen \mathbf{B} wird somit folgende Verlustfunktion V_E als Kennzahl für den Energieverbrauch definiert:

$$V_E = \sum_{i=2}^n \left| \frac{1}{2} (a(i) + a(i-1))(s(i) - s(i-1)) \right| \quad (3.24)$$

Hierbei wird der Mittelwert zweier benachbarter Beschleunigungswerte $a(i-1)$ und $a(i)$ gebildet und mit dem zurückgelegten Weg zwischen $s(i-1)$ und $s(i)$ multipliziert. Die Verlustfunktion bildet die Summe der Produkte, die proportional zum Energieaufwand sind. Die Bewegung aus der Schar mit dem kleinsten Wert für V_E liefert die Lösung mit dem geringsten Energieaufwand. Für den zweiten Optimierungsfall ist die Bewegung mit dem geringsten maximalen Ruck zu selektieren, weshalb der Verlust V_R einer Bewegung das Betragsmaximum des Rucks ist:

$$V_R = \max(|r(i)|)_i \quad (3.25)$$

Die Bewegung aus der Schar mit dem kleinsten Wert für V_R liefert die Lösung mit dem kleinsten Betragsmaximum im Ruckverlauf und somit diejenige mit der tendenziell kleinsten Schwingungsanregung für die Mechanik (siehe Abschnitt 1.2.3).

Im Zuge der Optimierung des gesamten Antriebssystems wird für eine Bewegung iterativ die Zeit verkürzt, um eine zeitoptimale Lösung zu finden (siehe Abschnitt 5.2). Eine Einschränkung der Zeitverkürzung erfolgt letztlich über die Restriktion von Geschwindigkeit und Beschleunigung. Soll also eine Bewegung mit hohem Potenzial für eine zeitliche Optimierung selektiert werden, dann muss sie besonders geringe Betragsmaxima für Beschleunigung und Geschwindigkeit aufweisen. Die Überlegung wird als Verlustfunktion V_t formuliert. Über die Werte i einer Bewegung und die Bewegungen j einer Schar gilt:

$$V_t = \frac{\max(|v(i)|)_i}{\max(|v_j(i)|)_{i,j}} \cdot \frac{\max(|a(i)|)_i}{\max(|a_j(i)|)_{i,j}}, \quad (3.26)$$

wobei ein Produkt von den Koeffizienten der Betragsmaxima der untersuchten Bewegung und dem globalen Betragsmaximum aller Bewegungen der untersuchten Schar für die Geschwindigkeit und die Beschleunigung gebildet wird. Die Bewegung mit der kleinsten Zahl aus V_t deutet auf das höchste verfügbare zeitliche Optimierungspotenzial.

Im Gegensatz zur eindeutigen Optimierung mittels der Einfachen Selektion kann mit im folgenden Abschnitt beschriebenen CI-Selektion auch multikriteriell optimiert werden.

3.2.3 Multikriterielle Selektion

Die multikriterielle Selektion (CI-Selektion) basiert auf einer Auswertung der Schar von Bewegungen mittels unscharfer Logik.³ Im Gegensatz zur Einfachen Selektion erlaubt sie einen Kompromiss zwischen den drei teilweise gegensätzlichen Optimierungskriterien. Im Folgenden wird dargestellt, wie der neu entwickelte Algorithmus zur Selektion in fünf Schritten aufgebaut ist.

³Die Hinzunahme Künstlicher Neuronaler Netze wurde untersucht [106], sie konnte sich jedoch praktisch nicht durch geeignete Auswahl von Bewegungen bewähren.

1. Klassifizierung eines Profils anhand von Kennzahlen
2. Fuzzifizierung der Kennzahlen, der Optimierungskriterien und der Zwischenergebnisse
3. Regelauswertung in zwei Schritten anhand zuvor definierter linguistischer Terme
4. Defuzzifizierung
5. Auswahl einer Bewegung.

Die Erläuterung der Vorgehensweise beginnt mit der Klassifizierung der Bewegungen. Dazu sind Kennwerte hergeleitet, die denen der Einfachen Selektion ähneln. Die Betragsmaxima der einzelnen kinematischen Größen werden in Relation zueinander gesetzt, folgende Gleichung gibt detaillierteren Aufschluss darüber:

$$\bar{v}_{max} = \begin{cases} \frac{|v_{max}| - |v_{minmax}|}{|v_{maxmax}| - |v_{minmax}|} & \text{für } |v_{maxmax}| \geq |v_{minmin}| \\ \frac{|v_{min}| - |v_{maxmin}|}{|v_{minmin}| - |v_{maxmin}|} & \text{für } |v_{maxmax}| < |v_{minmin}| \end{cases} \quad (3.27)$$

Darin sind v_{max} die maximale bzw. v_{min} die minimale auftretende Geschwindigkeit einer Bewegung. Das Minimum aller Maxima ist mit v_{minmax} bzw. das Maximum aller Minima mit v_{maxmin} symbolisiert und demnach stehen v_{minmin} und v_{maxmax} für die absoluten Extrema aller Schar-kurven. In Abbildung 3.7 sind die verwendeten Größen veranschaulicht, für die blau markierte Bewegung wird der Kennwert \bar{v}_{max} bestimmt.

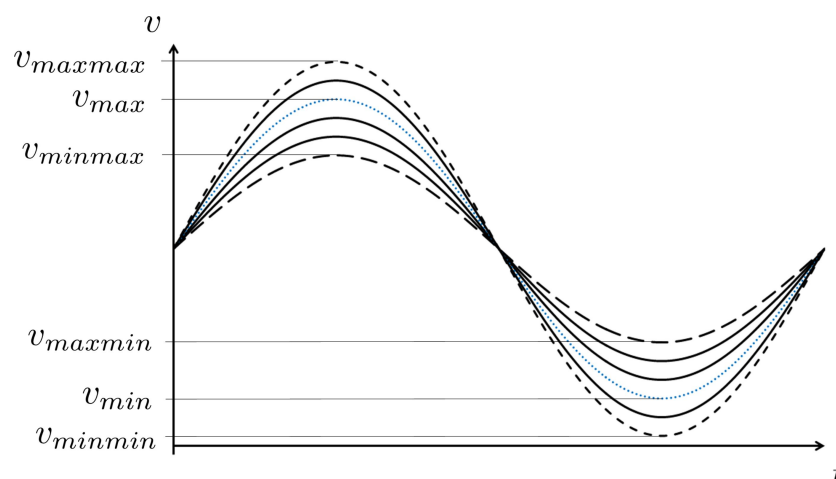


Abbildung 3.7: Größen zur Bestimmung der Kennwerte für die Klassifizierung

Die Fallunterscheidung sowie die Beträge in der notierten Formel sind notwendig, um den Kennwert unabhängig von der vertikalen Lage zur Zeitachse zu machen. Durch die Anwendung der Formel werden alle Kurven aus der Schar im Wertebereich $[0; 1]$ geordnet. Analog zum Geschwindigkeitskennwert \bar{v}_{max} aus Gleichung (3.27) werden auch die Kennwerte \bar{a}_{max} und \bar{r}_{max}

ermittelt:

$$\bar{a}_{max} = \begin{cases} \frac{|a_{max}| - |a_{minmax}|}{|a_{maxmax}| - |a_{minmax}|} & \text{für } |a_{maxmax}| \geq |a_{minmin}| \\ \frac{|a_{min}| - |a_{maxmin}|}{|a_{minmin}| - |a_{maxmin}|} & \text{für } |a_{maxmax}| < |a_{minmin}| \end{cases} \quad (3.28)$$

$$\bar{r}_{max} = \begin{cases} \frac{|r_{max}| - |r_{minmax}|}{|r_{maxmax}| - |r_{minmax}|} & \text{für } |r_{maxmax}| \geq |r_{minmin}| \\ \frac{|r_{min}| - |r_{maxmin}|}{|r_{minmin}| - |r_{maxmin}|} & \text{für } |r_{maxmax}| < |r_{minmin}| \end{cases} \quad (3.29)$$

In Abbildung 3.8 ist der Algorithmus dargestellt, mit dem ein Profil ausgewählt wird. Zunächst werden in einem ersten Regelsatz alle Bewegungen der vorliegenden Schar klassifiziert. Dazu wird festgestellt, inwieweit eine Bewegung energie-, ruck- oder zeitoptimal ist. Anhand der Information wird in einem zweiten Regelsatz festgelegt, wie stark eine jede Bewegung dem fuzzifizierten Anwenderwunsch entspricht.

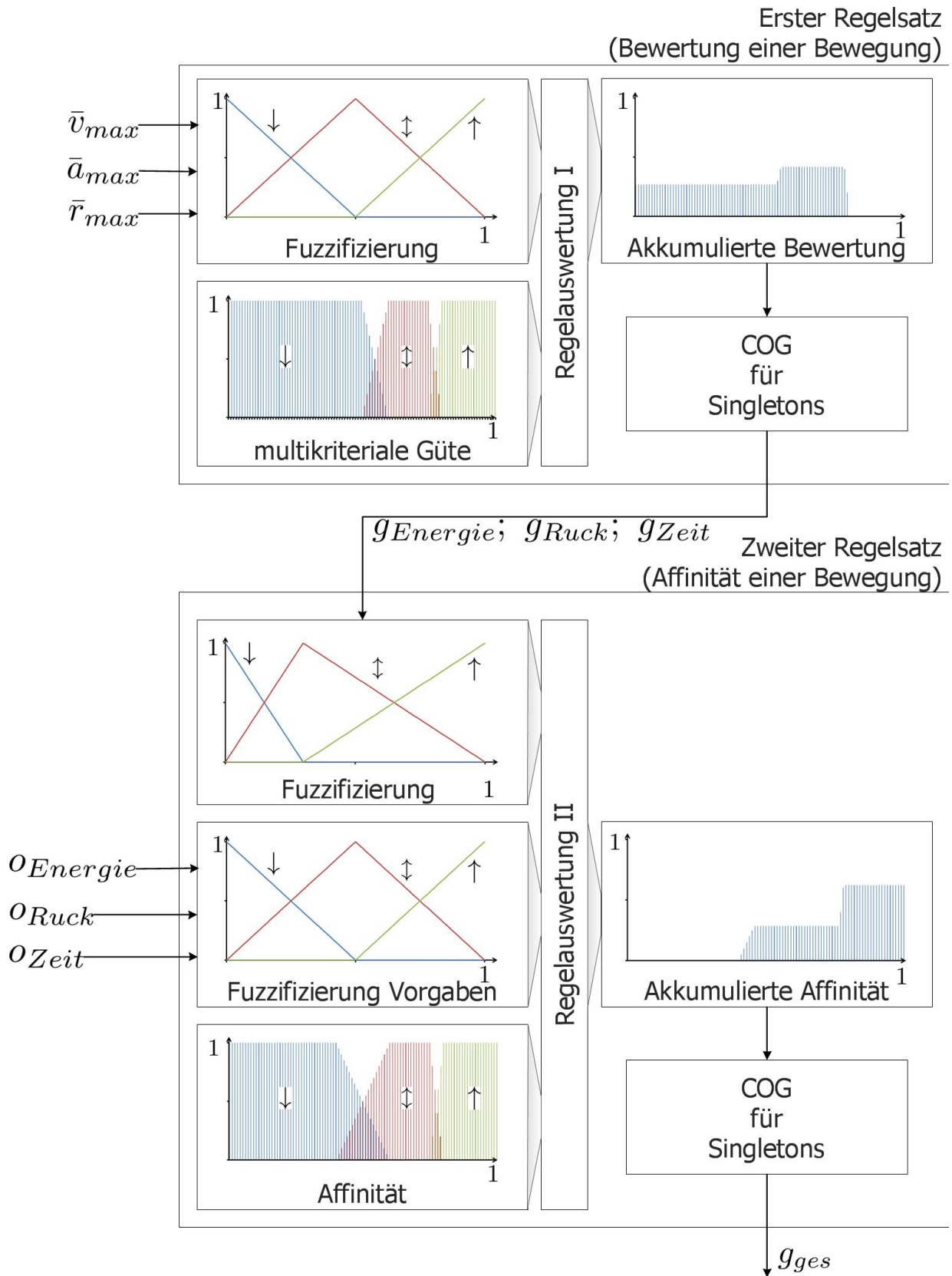


Abbildung 3.8: Selektion über zwei Fuzzy Regelsätze

Erster Regelsatz (Bewertung einer Bewegung)

Erst werden die normierten Kennwerte fuzzifiziert, was mithilfe von Dreiecksfunktionen für die linguistischen Werte⁴ *klein*, *neutral* und *groß* erfolgt. Trapezfunktionen der Konklusion ordnen die Güte der Optimierungskriterien⁵ in *klein*, *groß* und *sehr groß* ein und liegen in diskreter Form, sprich als eine Folge von Singletons, vor. Linguistische Terme bilden die Regeln für den ersten Satz, welche der nachstehenden exemplarischen Regel entsprechen:

WENN Ruck *klein* UND Beschleunigung *groß* DANN Ruckoptimal *sehr groß*

Die Aggregation erfolgt hierbei mittels des Minimumoperators, womit auch die Aktivierung umgesetzt wird.

$$\mu_{Agg} = \min\{\mu_1; \dots; \mu_n\} \quad (3.30)$$

Die Akkumulation wird in zwei Schritten durchgeführt und bedient sich des Maximumoperators.

$$\mu_{Akk\ 1/2} = \max\{\mu_1; \dots; \mu_n\} \quad (3.31)$$

Im ersten Schritt werden alle Regeln mit gleicher Konklusion vereint und anschließend ebenfalls das Maximum aller Regeln bezüglich eines Optimierungskriteriums zusammengefasst. Für alle drei Optimierungskriterien entstehen Zugehörigkeiten in Form von Folgen mit n Singletons s_i über einem Definitionsbereich $x_i \in [0; 1]$. Anschließend erfolgt mit der Schwerpunktmethod für Singletons (COG⁶) eine Defuzzifizierung. Als Ergebnis werden Werte für die relative Güte g eines Profils für drei Optimierungskriterien ($g_{Energie}$, g_{Ruck} , g_{Zeit}) erhalten. Die folgende Gleichung stellt die Berechnungsvorschrift für die Defuzzifizierung dar, wie sie zur Bestimmung der drei Gütewerte angewandt wird:

$$g = \frac{\sum_{i=1}^n s_i x_i}{\sum_{i=1}^n s_i} \quad (3.32)$$

Zweiter Regelsatz (Affinität einer Bewegung)

Die Gütewerte werden sogleich wieder durch entsprechend definierte Dreieckfunktionen fuzzifiziert. Es soll damit anhand des zweiten Regelsatzes festgelegt werden, wie affin eine Bewegung zum gestellten Optimierungswunsch ist. Dreieckfunktionen fuzzifizieren auch die durch den Anwender prozentual festgelegten Optimierungskriterien $o_{Energie}$, o_{Ruck} und o_{Zeit} , die als Komponenten des bereits eingeführten Vektors des Optimierungsziels \vec{o} einer Bewegung zu sehen sind. Als Folge von Singletons sind die Zugehörigkeiten der Affinitäten definiert. Die Affinität, die Güten und die Optimierungskriterien werden linguistisch ebenfalls mit *klein*,

⁴klein ↓ neutral ↓ groß ↑

⁵klein ↓ groß ↑ sehr groß ↑↑

⁶Abk. engl. Center Of Gravity

groß und sehr groß bewertet. Der Regelsatz enthält beispielhaft dargestellte Regeln:

WENN Kriterium Energieoptimal groß UND Energieoptimal groß DANN
Affinität groß

Ausführlicher formuliert die Regel, dass ein großer Wunsch des Anwenders nach Energieoptimalität bei einer Bewegung, die mit hoher Energieoptimalität bewertet wurde zu einer hohen Eignung (Affinität) der bewerteten Bewegung führt.

Die Regeln des zweiten Satzes werden analog zum ersten ausgewertet, nämlich mit dem Minimumoperator in der Aggregation und der Aktivierung sowie dem Maximumoperator in der Akkumulation. Letztlich wird mittels der Schwerpunktmethod für Singletons für jede Bewegung einer vorliegenden Schar eine Kennzahl für die Güte des Profils g_{ges} festgelegt.

Es wird diejenige Bewegung mit der maximalen Güte selektiert. Eine vollständige Aufstellung der Regelsätze und der Zugehörigkeitsfunktionen, die in der beispielhaften Umsetzung der Algorithmen erfolgreich selektieren, sind im Anhang der Arbeit formuliert.

Die Anwendung der vergleichsweise⁷ einfachen Operatoren in der Regelauswertung sowie der wenig komplexen Zugehörigkeitsfunktionen garantieren Algorithmen mit geringer Komplexität, was im Hinblick auf die numerische Optimierungsstrategie mit einer hohen Zahl von Durchläufen des Fuzzy-Algorithmus sinnvoll ist. Eine feinere Selektion durch präzisere Methoden ist infolge der eher geringen Quantität der Kurvenscharen ohnehin nicht zu erwarten. Daher lassen sich die Algorithmen auch einfach durch einige Experiment einstellen, so dass sie zuverlässig passende Bewegungen selektieren.

Grundsätzlich ist ein Vorteil der CI-Selektion gegenüber der Einfachen Selektion die Fähigkeit zu Kompromissen zwischen den Optimierungskriterien. Mit dem Streben des Algorithmus nach zuvor gewichteter Ausgewogenheit liefert er beim Wunsch nach einer sehr einseitigen Optimierung jedoch nur einer suboptimale Selektion im Vergleich zur Einfachen Selektion. Es lässt sich daher durch einfache exemplarische Selektionen leicht zeigen, dass auch die Einfache Selektion berechtigterweise als Teil des Bewegungsentwurfs unverzichtbar ist. Ein weiterer Gewinn durch die CI-Selektion ist festzustellen, weil durch sie komplexere Zusammenhänge in der Bewertung einer Bewegung abgebildet werden können. Es werden beispielsweise die Kennwerte verschiedener kinematischer Größen miteinander in Verbindung gebracht.

3.3 Implementierung des Bewegungsentwurfs

In diesem Abschnitt wird die objektorientierte Implementierung des Bewegungsentwurfs in die exemplarische Software aufgezeigt. Die mit der Arbeit einhergehende Entwicklung der Software

⁷Beginnend bei Grundlagen der Fuzzy-Logik stehen bereits weitaus komplexere Operatoren (vgl. Einstein, Hamacher) und Zugehörigkeitsfunktionen (vgl. Gauss, Exponentialfunktionen) zur Verfügung.

ist mithilfe der Hochsprache C++ realisiert. Als Entwicklungsumgebung steht das offene kostenfreie Qt [107, 108, 109] zur Verfügung, das unter Anderem umfangreiche Entwurfsmöglichkeiten für eine GUI⁸ zur Verfügung stellt. Alternativen auf Basis anderer Softwaretechnologien kommen grundsätzlich auch in Frage [110], sind aber aus Gründen der Kompatibilität für die vorliegende Implementierung nicht ausgewählt.

3.3.1 Interpolation von Bewegungen

Es besteht die Anforderung, die Algorithmen zum Bewegungsentwurf jederzeit um weitere mathematische Methoden zur Interpolation (Gl. 3.3) von Bewegungsvorgängen erweitern zu können.

Die genannte Anforderung ist systematisch durch den Entwurf einer Mutterklasse „Bewegungsgesetz“ (UML-Klassendiagramm⁹ aus Abbildung 3.9) mit virtuellen Eigenschaften erfüllt, von der sämtliche Klassen zur Interpolation von Bewegungen erben und somit gezwungen sind, vorgegebene Methoden und Datentypen umzusetzen.

Die Mutterklasse enthält eine Variable zur Bezeichnung eines Bewegungsgesetzes und die Ma-

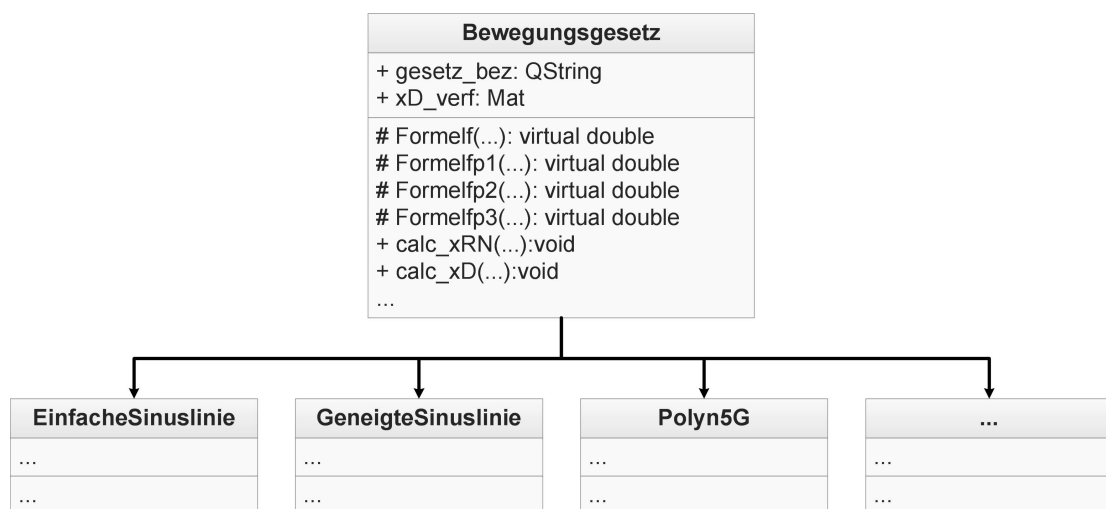


Abbildung 3.9: Klassendiagramm zur Interpolation von Bewegungen

trix „xD_verf“, die von den erbbenden Klassen mit der jeweiligen Eignung initialisiert wird. Die Methoden „Formelf“ bis „Formelfp3“ liefern die normierten Funktionswerte für die Strecke und deren Ableitungen zu einem normierten Zeitpunkt. Um auch den Ruck analytisch zu berechnen, wurden die Bewegungsgesetze nach der VDI-Richtlinie 2143 entsprechend erweitert, die Ergebnisse sind im Anhang angegeben. Weitere nennenswerte Methoden sind „calc_xRN“ zur Berechnung der normierten Randbedingungen entsprechend (3.7) bis (3.9) sowie „calc_xD“ zur Diskretisierung nach (3.10) bis (3.13).

⁸engl. Graphical User Interface = Graphische Benutzeroberfläche

⁹UML=Unified Modeling Language, beispielsweise [111, 112]

3.3.2 Selektion von Bewegungen

Die Selektion erfolgt schrittweise, wie es in Abschnitt 3.2 festgelegt ist, wozu in Abbildung 3.10 gleichfalls ein Klassendiagramm skizziert ist. Hierin befindet sich die Klasse „Selektion“, welche

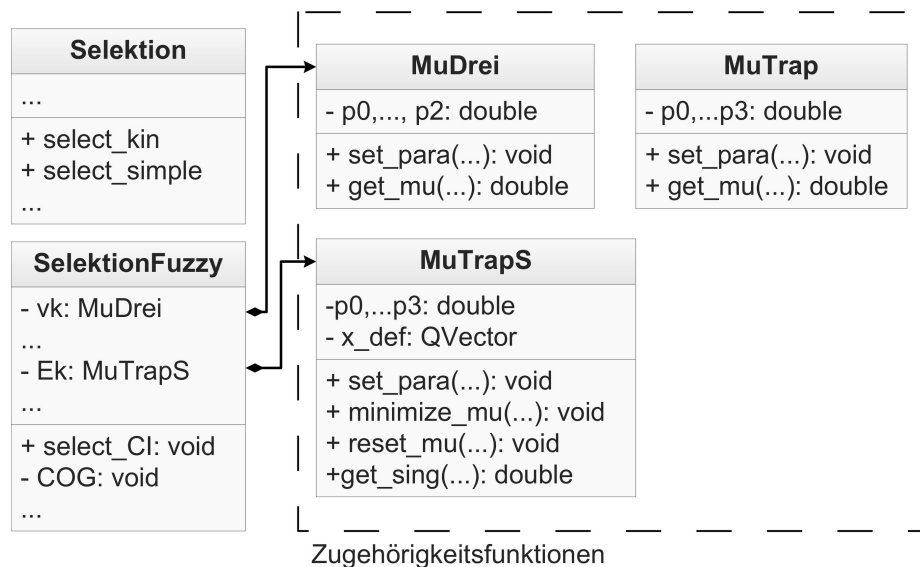


Abbildung 3.10: Klassendiagramm zur Selektion

zwei zentrale Methoden besitzt:

- „select_kin“ zur optionalen kinematischen Selektion, deren Algorithmus als Ablaufdiagramm in Abbildung 3.11 aufgezeichnet ist.
- „select_simple“ zur Selektion einer Kurve aus einer Schar nach den Gütekriterien (3.24) bis (3.26).

Wie aus dem Klassendiagramm hervorgeht, gibt es eine weitere Klasse für die CI-Selektion. Sie beinhaltet als Variablen die Zugehörigkeitsfunktionen, die zur Implementierung der in Abschnitt 3.2.3 erläuterten Fuzzifizierung und Regelauswertung notwendig sind. Zudem ist die Funktion „COG“ von Bedeutung, welche die gleichnamige Möglichkeit zur Defuzzifizierung darstellt. Mit der Funktion „select_CI“ ist es schließlich möglich, aus einer Schar von Funktionen zu selektieren. Der Ablauf des implementierten Algorithmus ist im Wesentlichen analog zur bereits theoretisch beschriebenen Methode und wird daher nicht gesondert als Flussdiagramm dargestellt.

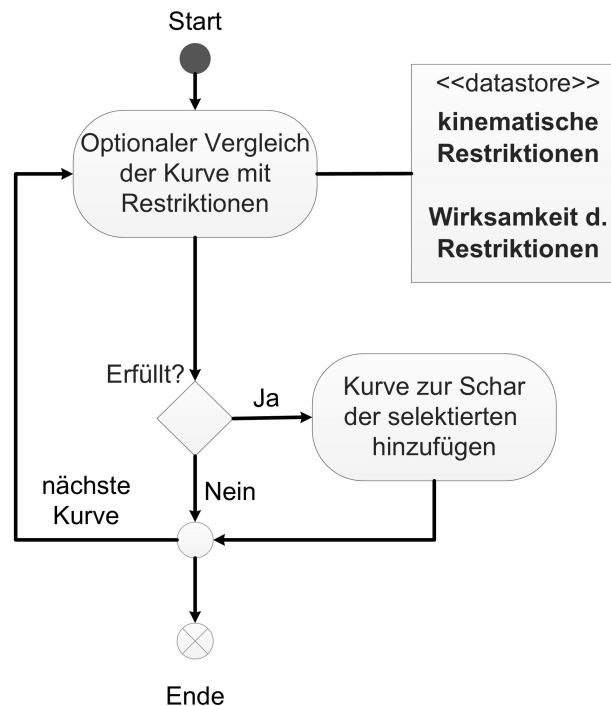


Abbildung 3.11: Algorithmus zur kinematischen Selektion

3.3.3 Teilprogramm zum Bewegungsentwurf

Die Implementierung des gesamten Bewegungsentwurfs mit eingehend erläuterten Eigenschaften und mathematischen Vorgehensweisen ist ein Teil eines gesamten Programms zum ganzheitlichen Engineering, das automatisch als Teil einer Software agiert. Dennoch ist eine Benutzeroberfläche erstellt, die es ermöglicht, gezielt in den Entwurf einer Teilbewegung einzugreifen. Zur Verdeutlichung des Bewegungsentwurfs ist die Oberfläche der Teillösung in Abbildung 3.12 dargestellt und wird im Folgenden erläutert.

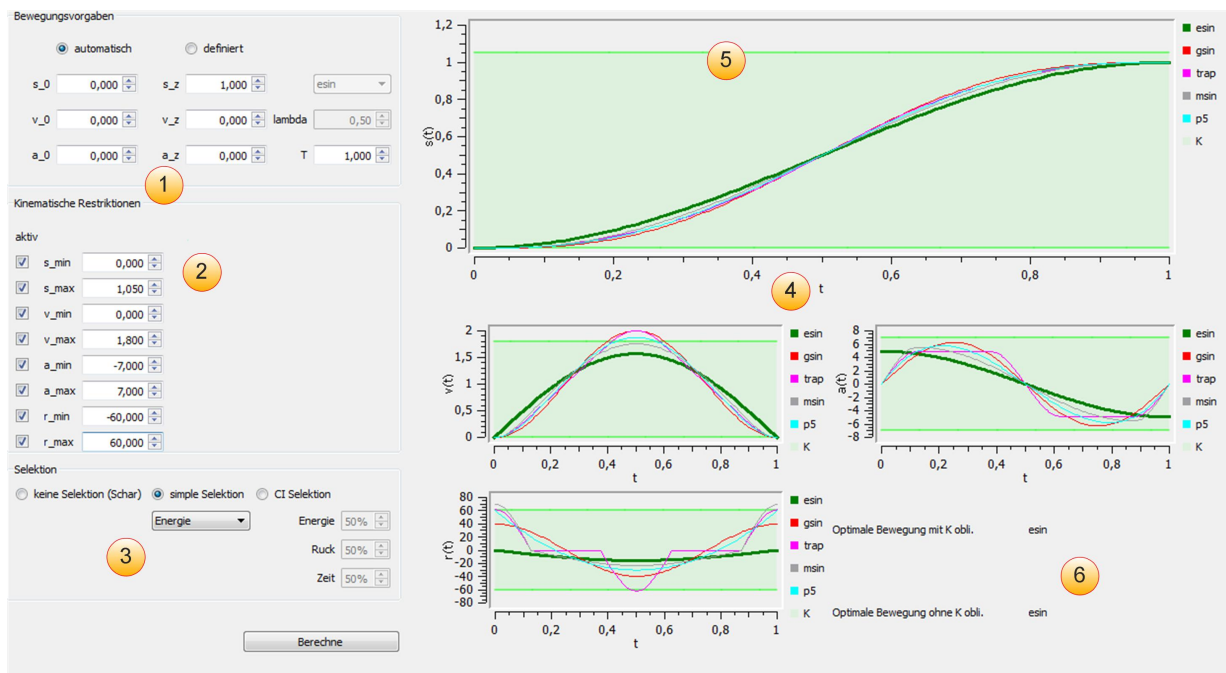


Abbildung 3.12: Programmauszug aus Bewegungsentwurf

Nach Eingabe der Randbedingungen einer Bewegung unter „1“ und der eventuellen kinematischen Einschränkungen unter „2“ kann eine Interpolation der gewünschten Bewegung erfolgen. Das Ergebnis, zunächst eine Schar, wird in „4“ mit den drei Ableitungen dargestellt. Ebenso sind die kinematischen Restriktionen in Form der grünen Bereiche verdeutlicht („5“). Wird wie im gezeigten Beispiel eine Selektion durchgeführt, dann wird das Ergebnis in den Diagrammen als hervorgehobener grüner Graph der selektierten Funktion markiert. Unter „6“ sind die Ergebnisse der Selektionen mit und ohne kinematischer Vorselektion als Bezeichnung zu finden.

3.4 Zusammenfassung der Neuerungen im Bewegungsentwurf

In den vorausgehenden Abschnitten des Kapitels ist das neu entwickelte Verfahren zum Bewegungsentwurf beschrieben. Hierzu ist zunächst die Feststellung der Eignung von Bewegungsgesetzen formalisiert worden und zudem die dritte Ableitung der Bewegungsgesetze aus der VDI-Richtlinie 2143 gebildet worden. Damit sind zwei Erweiterung zur genannten Richtlinie entstanden, die das neue Verfahren zum Bewegungsentwurf ermöglichen.

Zur Selektion von Bewegungen, wie sie ebenfalls im Rahmen der neuen Vorgehensweise erforderlich ist, sind ebenfalls zwei neue Methoden entstanden. Die Einfache Selektion basiert auf den drei neuen Verlustfunktionen, siehe Gleichungen (3.24) bis (3.26). Die CI-Selektion stützt

sich auf eine Fuzzy-Logik. Hierfür ist eine Formel (3.27) zur Berechnung von Kennwerten herausgearbeitet und sind zwei Regelsätze, einer zur Analyse und einer zur Eignungsfeststellung für eine bestimmte Anforderung, neu entwickelt worden. Die vorgestellten neuen und modifizierten Methoden sind vollständig in C++ implementiert.

4 Neu- und Weiterentwicklungen zum Hardwareentwurf

Der Hardwareentwurf stellt den zweiten Schwerpunkt der Arbeit dar. Anhand der Ausgangssignale des Bewegungsentwurfs, insbesondere der Bewegung einer Achse, kann mithilfe eines Modells der Mechanik eine Auslegungsrechnung und die Auswahl von einem Motor erfolgen. Das Kapitel beginnt mit einem Abschnitt zur Formulierung eines Modells, welches für das neue Gesamtkonzept geeignet ist. Dazu werden einige aus der Literatur bekannte Zusammenhänge so angepasst, dass sie algorithmisch verwendbar sind. Im Abschnitt 4.2 wird auf eine Auslegungsrechnung und im Abschnitt 4.3 auf die Auswahl von Antriebskomponenten eingegangen. Zuvor wird ein Überblick über die Vorgehensweise innerhalb des Hardwareentwurfs gegeben. In Abbildung 4.1 ist eine Übersicht über den Hardwareentwurf, eines der drei Teilsysteme des Gesamtkonzepts, dargestellt. Bei einer gegebenen Bewegung B einer Achse, die auch hypo-

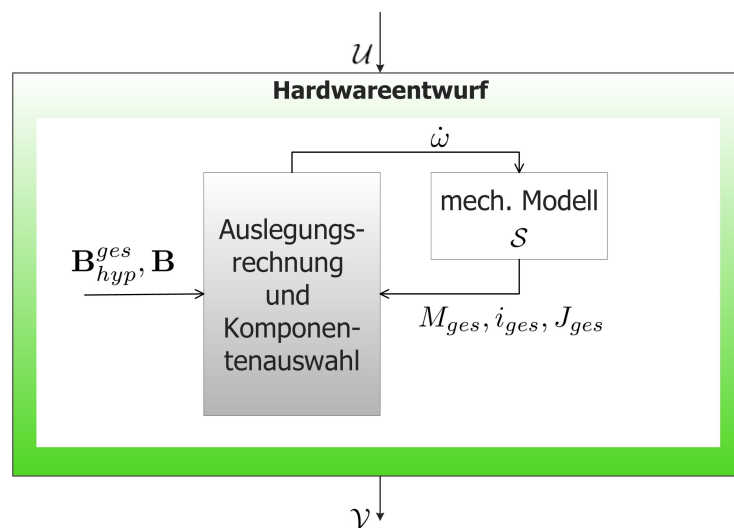


Abbildung 4.1: Übersicht zum Hardwareentwurf

tisch B_{hyp}^{ges} aus einem unoptimierten System stammen kann, ist es mit einem mechanischen Modell S möglich, die wesentlichen Auslegungsgrößen zu berechnen. Mit den Größen zur Komponentenauswahl, dem gesamten Drehmoment M_{ges} , der Übersetzung i_{ges} und der Massenträgheit

J_{ges} kann die Hardware bestimmt werden. Deren Parameter, beispielsweise Restriktionen, gehen mit in weitere Iterationen der Systemoptimierung ein.

Die Auslegung muss im Hinblick auf die Optimierung des gesamten Systems, wie auch der Bewegungsentwurf, automatisch erfolgen. Dazu muss nicht nur ein genormter Vorrat an möglicher einsetzbarer Hardware (Motoren, Getriebe, Stromrichter, Regler etc.) zur Verfügung stehen. Eine Auslegungsrechnung fordert zudem ein hinreichend genaues Modell über die Mechanik, die jeder Antrieb zu bewegen hat. Nur bei Kenntnis über deren Massenträgheit, Dämpfung (Reibung) und Lastverläufe kann mithilfe der geforderten Bewegungsvorgänge eine Auslegung erfolgen oder es können die kinematischen Restriktionen für den Bewegungsentwurf bestimmt werden. Im Folgenden wird dargestellt, wie für betrachtete Antriebssysteme relevante Mechaniken hinreichend genau modellierbar sind.

4.1 Angepasste Modellierung der Mechanik

Um eine Bewegungsachse dimensionieren zu können, müssen die mechanischen Eigenschaften einer Achse hinreichend genau abgebildet werden. Die vom Servomotor angetriebene Mechanik kann nahezu beliebiger Zusammensetzung sein. Ein Modell einer solchen Mechanik, das der Planung einer Applikation und der Auslegungsrechnung genügt, kann jedoch sehr verallgemeinernd aufgestellt werden. Im Wesentlichen wird die gewünschte Bewegung des Endpunktes des Antriebsstranges auf die dazu erforderliche Bewegung des elektromechanischen Wandlers umgerechnet. Der Entwurf eines Antriebssystems erfolgt naturgemäß intuitiver über die Gestaltung von Bewegungen in Bezug auf die gewünschte Zielbewegung als über die Bewegung des Motors.

4.1.1 Formulierung eines modularisierten Gesamtmodells

Um ein Modell der Mechanik ausreichend variabel gestalten zu können, ist es sinnvoll, es aus immer wiederkehrenden Teilmodellen modularisiert aufzubauen [113]. Im vorliegenden Fall wird daher die gesamte Mechanik, die zu einer Bewegungsachse gehört, als eine Kette von nacheinander geschalteten typischen Standardmechaniken betrachtet. Die Mechaniken können wiederum aus kleineren Einheiten zusammengesetzt werden, aus einfachen bewegten Körpern, beispielsweise Zylinder oder Hohlwellen. Physikalische Gleichungen zur Bestimmung der kinematischen Eigenschaften derartiger Körper sind in Grundlagenwerken vielfach dokumentiert, stellvertretend sei auf [37] verwiesen.

Ein Modul sei im vorliegenden Kontext eine Standardmechanik mit An- und Abtriebsseite, welche als physikalische Eigenschaften eine Gesamtmassträgheit J , ein Übersetzungsverhältnis i und einen Wirkungsgrad η besitzt. Hinzu kommt ein eventuell statisches Drehmoment M_{stat} aus

dem Antriebsstrang. Somit ist ein Modul \mathcal{M} definiert durch die Menge:

$$\mathcal{M} := \{J; i; \eta; M_{stat}\} \quad (4.1)$$

Durch Reihenschaltung von Modulen kann schließlich das Modell erstellt werden, das die gesamte Mechanik einer Achse beschreibt (Abbildung 4.2). Prinzipiell ist auch eine Parallelschaltung von Modulen möglich, was im wesentlichen zu einer Addition von Drehmomenten und Massenträgheiten am Knotenpunkt führt. Im weiteren werden die Ausführung auf die Reihenschaltung beschränkt. Ausgehend von einer starren Kopplung resultieren die Regeln für zwei verknüpfte

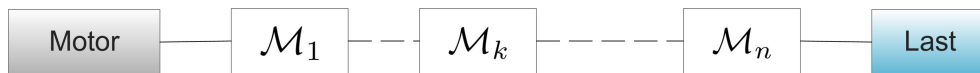


Abbildung 4.2: Reihenschaltung von Modulen mit Motor und Last

Module \mathcal{M}_k und \mathcal{M}_{k-1} :

$$\begin{aligned} \dot{\omega}_k &= \dot{\omega}_{an,k} = \dot{\omega}_{ab,k-1} = i_k^{-1} \cdot \dot{\omega}_{k+1} \\ M_{stat,k} &= M_{ab,(k-1)} = (M_{stat,k+1} i_k^{-1}) \\ M_{d,k} &= M_{t/b,k} = M_{ab,(k-1)} = \tilde{\eta} (M_{d,k+1} i_k^{-1} + J_k \dot{\omega}_k) \end{aligned} \quad (4.2)$$

Zur Aufstellung der Gleichungen sind die allgemein bekannten Zusammenhänge für Übersetzungen in mechanischen Vorrichtungen verwendet [63]. Die Winkelbeschleunigung $\dot{\omega}$ eines Moduls entspricht der Antriebswinkelbeschleunigung $\dot{\omega}_{an,k}$ des Moduls, welche gleich der Abtriebswinkelbeschleunigung $\dot{\omega}_{ab,k-1}$ des vorgeschalteten Moduls sein muss. Oder bezogen auf das nachgeschaltete Modul dessen Winkelbeschleunigung dividiert durch die Übersetzung i_k des betrachteten Moduls. Der Zusammenhang gilt in Analogie für die statischen Drehmomente M_{stat} , die Abtriebsmomente M_{ab} sowie die dynamischen Drehmomente M_d und die treibenden/bremsenden Drehmomente $M_{t/b}$ der Module.

Bei der Aufschaltung von n Modulen resultieren Gesamtwerte für den gesamten mechanischen Antriebsstrang $\mathcal{S} = \{\mathcal{M}_1, \dots, \mathcal{M}_k, \dots, \mathcal{M}_n\}$ mit Last von:

$$\begin{aligned} i_{ges} &= \prod_{k=1}^n i_k \quad \text{und} \quad \tilde{\eta}_{ges} = \prod_{k=1}^n \tilde{\eta}_k \\ M_{stat,ges} &= M_{stat,1} + \sum_{l=1}^{n-1} \left(\prod_{k=1}^l [i_k^{-1}] M_{stat,l+1} \right) + i_{ges}^{-1} M_{stat,L} \\ J_{ges} &= J_1 + \sum_{l=1}^{n-1} \left(\prod_{k=1}^l [i_k^{-2}] J_{l+1} \right) + i_{ges}^{-2} J_L \end{aligned} \quad (4.3)$$

Die ersten beiden Gleichungen entstammen aus [63]. Die beiden letzten Gleichungen wurden als Erweiterung hergeleitet, um eine direkte Transformation zwischen Motorwelle und beim Entwurf des Systems angewendeten Endeffektor einer Achse zu erhalten. Die Gesamtgrößen

beziehen sich auf die Motorachse. Für die Berechnung der Transformation ist die Kenntnis der Größen aller aufgeschalteten Module erforderlich.

Im anschließenden Abschnitt sind exemplarisch einige komplexere Standardmechaniken entsprechend der Modularisierung aufgeführt. Richtungweisende Zusammenstellungen von solchen Mechaniken sind bekannt aus der Auslegungsrechnung, beispielsweise [29, 98], jedoch unpassend oder zum vorliegenden Modellierungsanspruch ungeeignet dokumentiert. So können beispielsweise für einen Riementrieb mit Spannrolle diverse Modelle herangezogen werden, die je nach Abstraktionsgrad beispielsweise die Steifigkeit des Riemens, Polygoneffekte oder die Riemenvorspannung berücksichtigen. Im vorliegenden Fall bedarf es eines möglichst allgemeingültigen Modells, welches im Rahmen der entworfenen Modularisierung eine hinreichend genaue Abschätzung von Größen für die Antriebsauslegung bereitstellen kann.

4.1.2 Zweckmäßige Notation von Standardmodulen

Im Folgenden werden die wichtigsten Standardmodule zusammengestellt. Dabei handelt es sich um:

- Stirnradgetriebe
- Riementrieb mit Spannrolle
- Spindeltrieb
- Zahnstangenantrieb
- Transportwalzen.

Die jeweiligen Modellparameter wie Radien r , Zähnezahlen z , Massen m oder lastende Kräfte F_L sind den dazugehörigen Abbildungen zu entnehmen.

Stirnradgetriebe, nach [15]

Abbildung 4.3 zeigt ein einstufiges Getriebe. Das Modell kann auch vereinfachend für baulich abweichende Getriebe, z.B. Planetengetriebe, benutzt werden, wenn deren Parameter ausreichend bekannt sind.

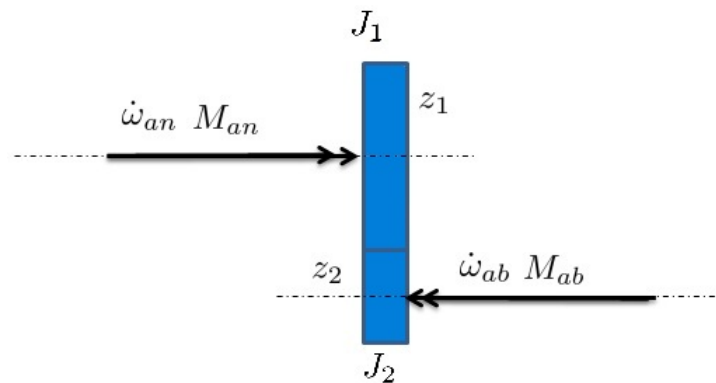


Abbildung 4.3: Stirnradgetriebe, nach [15]

Übersetzungsverhältnis und Massenträgheit sind ermittelbar zu [15]:

$$i = \frac{z_2}{z_1} \quad (4.4)$$

$$J = J_1 + \frac{J_2}{i^2}$$

$$M_{stat} = 0$$

Riementrieb mit Spannrolle, nach [114]

Das Modell des in Abbildung 4.4 dargestellten Riementriebs ist bezeichnend für alle mechanischen Konstruktionen, in denen eine Rotation (Antrieb) durch ein umlaufendes Element, beispielsweise Riemen oder Kette, auf eine andere Rotation (Abtrieb) übersetzt.

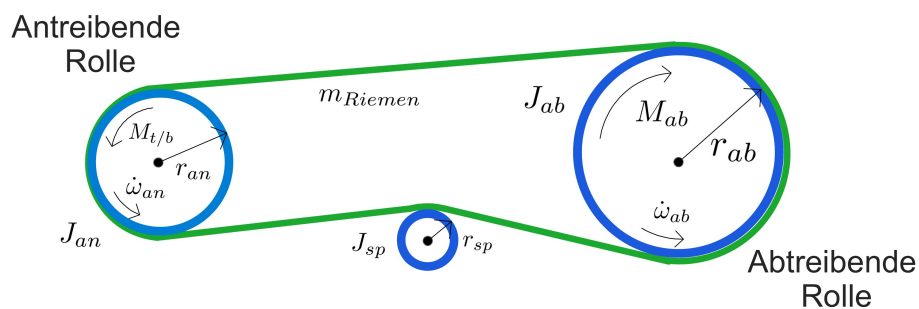


Abbildung 4.4: Riementrieb, nach [114]

Die Parameter berechnen sich in Anlehnung an [114] zu:

$$i = \frac{r_{ab}}{r_{an}} \quad (4.5)$$

$$J = J_{ab} i^{-2} + J_{sp} \left(\frac{r_{an}}{r_{sp}} \right)^2 + J_{an} + m_R r_{an}^2$$

$$M_{stat} = 0$$

Förderband mit Last und Gegengewicht, in Anlehnung an [114]

Das Schema aus Abbildung 4.5 lässt sich für alle Arten von Transportbändern anwenden, auf denen Lasten m_L mit beispielsweise Reibung F_L transportiert werden.

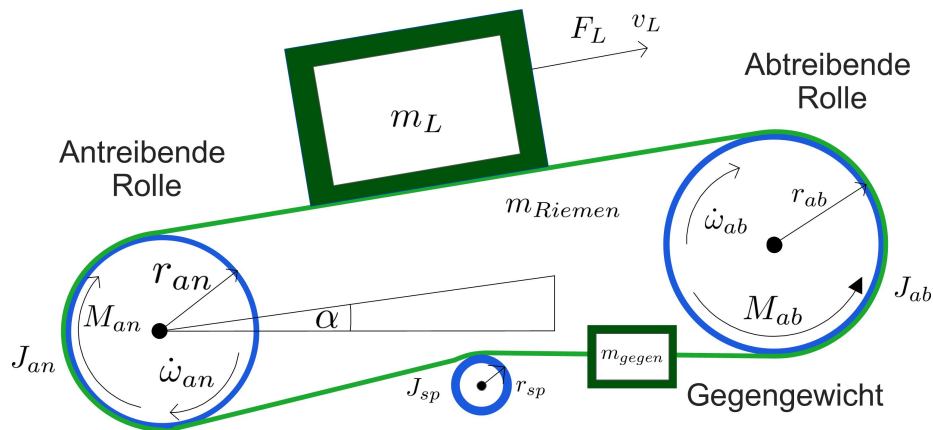


Abbildung 4.5: Förderband mit Last und Gegengewicht

Die Modulparameter ergeben sich in Anlehnung an [114] zu:

$$i = \frac{r_{ab}}{r_{an}} \quad (4.6)$$

$$J = (m_{Riemen} + m_{gegen} + m_L) r_{an}^2 + J_{sp} \left(\frac{r_{an}}{r_{ab}} \right)^2 + J_{an}$$

$$M_{stat} = ((m_L - m_{gegen}) g \cdot \sin \alpha + F) r_{an}$$

außerdem: $\omega_{an} = v_L r_{an}$

Spindelantrieb [115]

Das Modul für Spindelantriebe kann am Ende einer kinematischen Kette eingesetzt werden und ist verwendbar für diverse Strukturen von Spindelantrieben, siehe Abbildung 4.6.

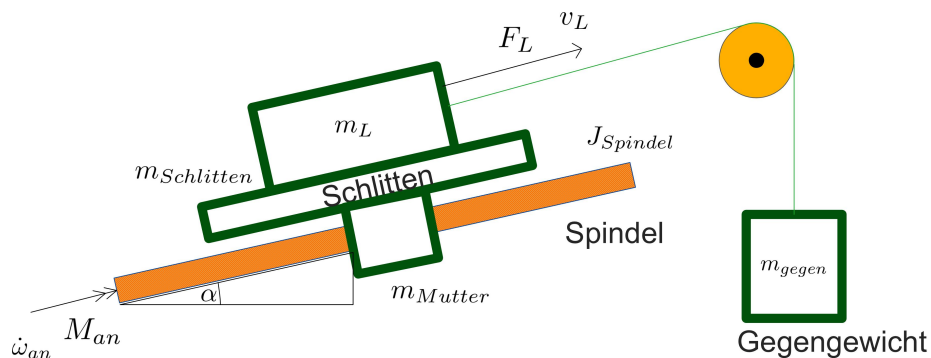


Abbildung 4.6: Spindeltrieb

Bei einer Spindelsteigung P gelten die Beziehungen [115]:

$$i = 1 \quad (4.7)$$

$$J = J_{Schnecke} + \frac{m_{Mutter} + m_{Schlitten} + m_L + m_{gegen}}{(2\pi P)^2}$$

$$M_{stat} = \frac{(m_{Mutter} + m_{Schlitten} + m_L) g \sin \alpha - m_{gegen} g + F_L}{2\pi P}$$

außerdem: $\omega_a n = 2\pi P v_L$

Zahnstangenantrieb [116]

Abbildung 4.7 zeigt einen verallgemeinerten Aufbau einer Bewegungsumsetzung über eine Zahnstange.

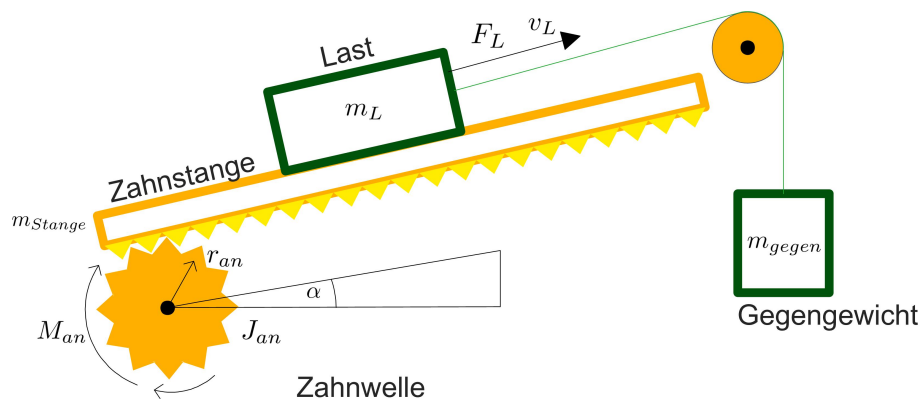


Abbildung 4.7: Zahnstangenantrieb

Hierbei ergeben sich aus [116] die Formeln:

$$\begin{aligned}
 i &= 1 & (4.8) \\
 J &= (m_{Stange} + m_L + m_{gegen}) r_{an}^2 + J_a n \\
 M_{stat} &= (m_{Stange} + m_L) g \sin \alpha - m_{gegen} g \\
 \text{außerdem: } \omega_{an} &= v_L r_{an}
 \end{aligned}$$

Transportwalzen [116, 117]

Beim Materialtransport über $2n$ Walzen kann das Modell aus Abbildung 4.8 verwendet werden.

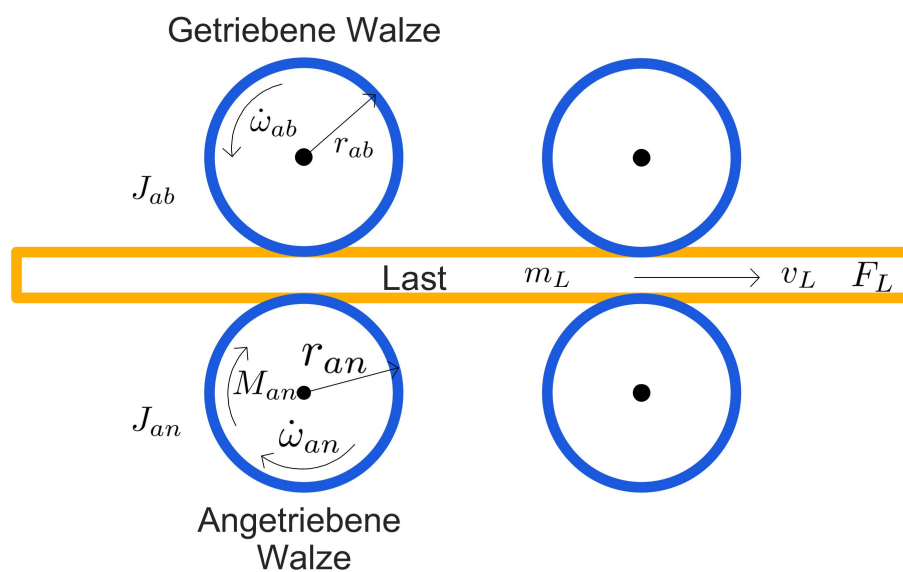


Abbildung 4.8: Transportwalzen, abgeleitet aus [116]

Die Formeln hierfür lauten, abgeleitet aus [116, 117]:

$$\begin{aligned}
 i &= \left(\frac{r_{ab}}{r_{an}} \right) & (4.9) \\
 J &= n J_{ab} \left(\frac{r_{ab}}{r_{an}} \right)^2 + n J_{an} + m_L r_{an}^2 \\
 M_{stat} &= F_L r_{an} \\
 \text{außerdem: } \omega_{an} &= v_L r_{an}
 \end{aligned}$$

Zusätzlich, aber nicht explizit aufgeführt sind:

- Welle

- drehsteife Kupplung
- Rotationskörper mit Unwucht
- Riemengetriebe.

Hierfür sind entsprechende Beschreibungen in [37] zu finden.

Durch die Modelle zur Abschätzung von Massenträgheit und Drehmomenten kann zum Einen von der Zielbewegung auf die Motorbewegung geschlossen werden, zum Anderen kann schließlich durch die vorgegebene Bewegung, die Massenträgheit und den statischen Momenten auf den Drehmomentverlauf an der Motorwelle geschlossen werden. Die Kenntnis der genannten Größen ist unerlässlich, um eine Auslegungsrechnung durchführen zu können, die im folgenden Abschnitt erläutert wird.

4.2 Integration der Auslegungsrechnung

Für die Auslegungsrechnung eines Servomotors müssen zumindest die maximal auftretende Drehzahl n_{max} , das maximal auftretende Drehmoment M_{max} nach (1.5), das mittlere Drehmoment \bar{M} aus (1.9) sowie die gesamte Massenträgheit J_{ges} an der Motorwelle bekannt sein. Während der Wert für die gesamte Massenträgheit allein aus dem mechanischen Modell \mathcal{S} bekannt ist, können die Drehmomente nur durch hinzunehmen der Dynamik der entsprechenden Achse ermittelt werden. Die Information über die Bewegung der Achse sind aus der Ausgangsmenge des Bewegungsentwurfs \mathcal{U} bekannt. Sie entspricht jedoch den Größen der Zielbewegung und muss mithilfe des mechanischen Modells \mathcal{S} auf die Größen an der Motorwelle umgerechnet werden. Demnach ergibt sich im Kontext des neuen Konzepts für die maximale Drehzahl an der Motorwelle:

$$n_{max} = \max \left\{ i_{ges} \left| \left[0 \ 1 \ 0 \ 0 \right] \mathbf{K}_B \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right| ; i_{ges} \left| \left[0 \ 1 \ 0 \ 0 \right] \mathbf{K}_B \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right| \right\} \quad (4.10)$$

Das Übersetzungsverhältnis des gesamten Antriebsstranges i_{ges} ist aus der mechanischen Modellbildung bekannt. Aus dem Bewegungsentwurf folgen zudem die kinematischen Extremwerte \mathbf{K}_B der entworfenen Bewegung bezogen auf den Endeffektor.

Für die Algorithmen des neuartigen Hardwareentwurfs lässt sich mithilfe der mechanischen Modellbildung und der Gleichungen (1.6) und (1.7) aus [63] der Verlauf vom dynamischen Moment \vec{m}_d herleiten zu:

$$\vec{m}_d = (J_{ges} + J_{Motor}) i_{ges} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{B}^T \quad (4.11)$$

Darin geht auch die Massenträgheit des Motors J_{Motor} ein.

Das wirkungsgradbehaftete richtungsabhängige statische Moment $\tilde{M}_{stat,ges}$ an der Motorwelle ist ebenfalls aus der Modellbildung bekannt und bildet, wie aus der zitierten Gleichung (1.5)

erkenntlich, zusammen mit einem additiven richtungsabhängigen Reibmoment \tilde{M}_R und dem dynamischen Drehmoment den Verlauf des gesamten Drehmoments \vec{m}_{ges} .

$$\vec{m}_{ges} = \tilde{\eta}_{ges} \vec{m}_d + \left(\tilde{M}_{stat,ges} + \tilde{M}_R \right) [1 \quad \dots \quad 1]^T \quad (4.12)$$

Bei der Addition des dynamischen Drehmoments ist der Wirkungsgrad $\tilde{\eta}$ der gesamten modellierten Mechanik zu berücksichtigen. Mithilfe des Verlaufs des gesamten Drehmoments kann auch das mittlere Moment \bar{M} (siehe Gleichung 1.9) im vorliegenden Kontext bestimmt werden:

$$\bar{M} = \frac{1}{n} \sum_{i=0}^{(n-1)} |m_i| \quad \text{mit: } \vec{m}_{ges} = \begin{bmatrix} m_0 \\ \vdots \\ m_i \\ \vdots \\ m_{n-1} \end{bmatrix} \quad (4.13)$$

Die horizontale Lage des Betriebspunkts, dessen Ordinate durch das mittlere Drehmoment ($M_B = \bar{M}$) gegeben ist, kann ebenfalls aus der Matrix der Bewegung einer Achse bestimmt werden (basierend auf Grundlagen aus [63]):

$$n_B = \frac{i_{ges}}{n} \sum_{i=0}^{n-1} |v_i| \quad \text{mit: } \vec{v} = \begin{bmatrix} m_0 \\ \vdots \\ v_i \\ \vdots \\ m_{n-1} \end{bmatrix} = [0 \quad 0 \quad 1 \quad 0 \quad 0] \mathbf{B}^T \quad (4.14)$$

Die Bestimmung der Größe entspricht ebenfalls einer an das Konzept angepassten Mittelwertbildung.

Mit den obigen Gleichungen (4.10)-(4.14) kann für vorgegebene Parameter die Auslegungsrechnung durchgeführt werden. Mit deren Ergebnis ist die Auswahl von Komponenten zum Zusammensetzen des Antriebssystems möglich.

4.3 Automatisierte Auswahl von Antriebskomponenten

Anhand der Kennwerte der Auslegungsrechnung kann aus einem Vorrat an Motoren ein geeigneter ausgewählt werden, der den Bedingungen (1.11) genügt. Der Vorrat an Motoren ist als Menge \mathcal{V}_{Motor} bestehend aus Untermengen \mathcal{E} hinterlegt, die die notwendigen Motorkennwerte beinhalten:

$$\mathcal{E} := \{ \bar{M}_{Motor}; n_{Motor}; J_{Motor}; \mathbf{L} \} \quad (4.15)$$

Es handelt sich um die in Abschnitt 1.2.4 angegebenen Kennwerte. Die Kennlinie des maximalen Drehmoments ist in Form von Stützpunkten $S_i(n_i|M_i)$ in der Matrix \mathbf{L} enthalten:

$$\mathbf{L} = \begin{bmatrix} n_1 & \dots & n_i & \dots \\ M_1 & \dots & M_i & \dots \end{bmatrix} \quad (4.16)$$

Die Restriktionen für Drehzahl und mittleres Moment lassen sich einfach überprüfen, für die Überprüfung des maximalen Moments ist eine Umrechnung ausgehend von der parametrischen Kennlinie erforderlich. Der k -te Wert des zulässigen maximalen Drehmoments $M(n_k)$ wird durch lineare Interpolation [118] bestimmt:

$$M(n_k) = M_i + \frac{M_{i+1} - M_i}{n_{i+1} - n_i} n_k \quad \text{mit: } n_i \leq n_k < n_{i+1} \quad (4.17)$$

Mit dem erhaltenen diskreten Verlauf der zulässigen Momente \vec{m}_{zul} lautet die an die Aufgabenstellung angepasste Restriktion:

$$\vec{m}_{zul} \geq \vec{m}_{ges} \quad (4.18)$$

Auch die Einhaltung des erlaubten Wertebereichs für den Koeffizienten k ist vergleichsweise einfach zu überprüfen. Eine weitere Problematik liegt darin, dass zur Massenträgheit aus dem mechanischen Modell (J_{ges} aus \mathcal{S}) die des Motors J_{Motor} addiert werden muss, welche erst nach Auswahl eines Motors bekannt ist. Zur Umgehung des Problems empfiehlt sich folgende Vorgehensweise (Abbildung 4.9). Das zu erwartende Massenträgheitsmoment \hat{J}_{Motor} wird zunächst entsprechend Gleichung (1.11) abgeschätzt:

$$\hat{J}_{Motor} = \frac{J_{ges}}{5} \quad (4.19)$$

Es wird ein wünschenswertes Verhältnis zwischen Motor- und Gesamtmassenträgheit entsprechend Gleichung (1.11) zugrunde gelegt.

Begonnen wird mit einer Auslegungsrechnung mit geschätzter Massenträgheit und dann wird vorab der kleinste Motor, sprich der mit kleinster Leistung, ausgewählt, der alle Restriktionen erfüllt. Danach wird mithilfe der Trägheit des gewählten Motors eine erneute Auslegungsrechnung durchgeführt. Werden danach immer noch alle Restriktionen erfüllt, dann ist ein passender Motor gefunden worden, andernfalls sind die Restriktionen zur Motorauswahl zu überarbeiten. Es handelt sich um die Kennwerte aus der Berechnung zur Kontrolle, die für eine erneute Motorauswahl herangezogen werden.

Nach einer erfolgreichen Motorauswahl erfolgt die Auswahl eines Stromrichters nach den Gleichungen (1.13) bis (1.14) aus einem Vorrat an Stellgeräten $\mathcal{V}_{Stellgeraete}$. In der Menge sind die Stellgeräte mit den Eigenschaften der Stromrichter, dem Strom I_S und dem maximalen Spitzenstrom $I_{S,max}$ als \mathcal{R} zusammengefasst enthalten.

$$\mathcal{R} := \{I_S; I_{S,max}\} \quad (4.20)$$

Nach erfolgtem Hardwareentwurf können kinematische Restriktionen K_H ermittelt werden, die sich aus der ausgewählten Hardware ergeben und an den Bewegungsentwurf weitergegeben werden können. Restriktionen ergeben sich vor allem aus der maximalen Geschwindigkeit bzw. Drehzahl und der maximalen Beschleunigung resultierend aus dem maximalen Moment. Da die Bewegung vom Bewegungsentwurf geändert wird, ist die maximale Beschleunigung nur ein grober Richtwert. Die Berechnung der kinematischen Restriktionen erfolgt im Rahmen des neuen

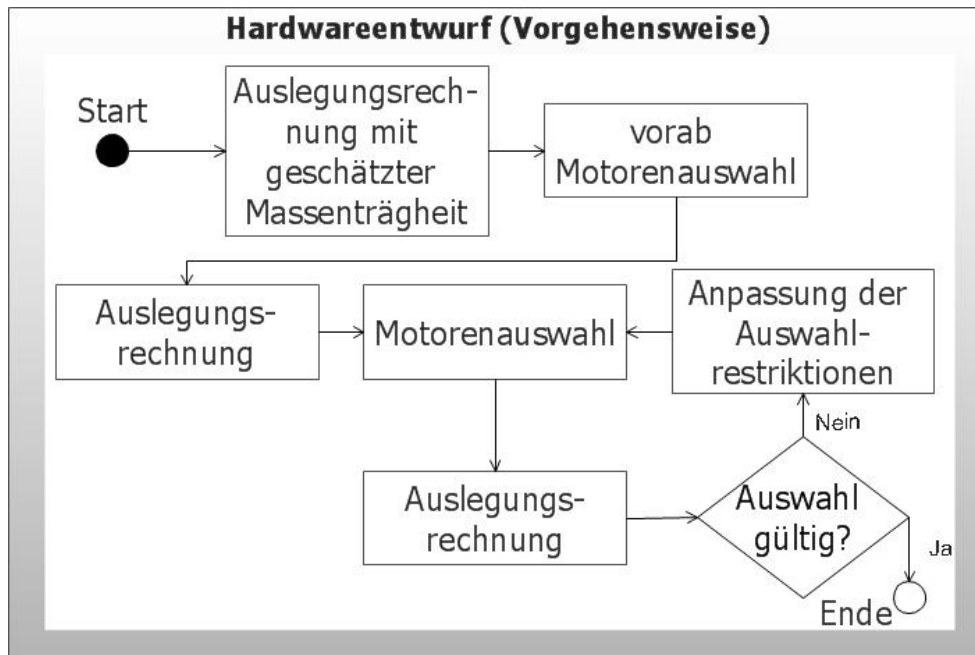


Abbildung 4.9: Vorgehensweise bei Auslegung und Auswahl

ganzheitlichen Konzepts zu:

$$K_H = \begin{bmatrix} s_{min,ultimo} & s_{max,ultimo} \\ -2\pi i_{ges}^{-1} n_{Motor} & 2\pi i_{ges}^{-1} n_{Motor} \\ -i_{ges}^{-1} \frac{\max\{\bar{m}_{zul}\}}{J_{ges} + J_{Motor}} & i_{ges}^{-1} \frac{\max\{\bar{m}_{zul}\}}{J_{ges} + J_{Motor}} \\ r_{min,ultimo} & r_{max,ultimo} \end{bmatrix}, \quad (4.21)$$

wobei K_{ultimo} aus dem Systementwurf bekannt ist. Da aus Parametern der Hardware gewöhnlich keine Restriktionen für die Strecke¹ und den Ruck erfolgen, sind die Restriktionen $s_{min,ultimo}$, $s_{max,ultimo}$, $r_{min,ultimo}$ und $r_{max,ultimo}$ gleich denen aus den ultimativen Restriktionen anzunehmen. Die Grenzwerte für die Geschwindigkeit werden von der maximalen Motordrehzahl auf die Geschwindigkeit am Endeffektor umgerechnet. Die maximale Beschleunigung wird mit dem maximalen zulässigen Drehmoment sowie der Massenträgheit berechnet und ebenfalls von der Motorwelle, durch Division mit dem gesamten Übersetzungsverhältnisses, auf den Endpunkt des Antriebsstrangs bezogen. Ein unerlaubter Bezug ergibt sich nicht, weil die ultimativen kinematischen Restriktion algorithmisch gesehen aus einem Initialsystem zunächst ohne Restriktionen aus der Hardware bestimmt werden.

¹Eine Ausnahme bilden Linearmotoren.

4.4 Implementierung des Hardwareentwurfs

Für den Hardwareentwurf sind zwei grundsätzliche Aufgaben zu implementieren, nämlich die Modellbildung für die Mechanik und die Auslegungsrechnung mit Auswahl von Komponenten.

4.4.1 Implementierung der Modellbildung

Der Programmteil zur Modellbildung muss dem Hardwareentwurf vor allem die Daten der Menge \mathcal{S} , sprich das Übersetzungsverhältnis, das statische Moment und die gesamte Massenträgheit zur Verfügung stellen.

Hierzu sind die in Abbildung 4.10 dargestellten Klassen essentiell. Eine Mutterklasse enthält die vorgeschriebenen Transformationsgrößen, über die jedes Standardmodul verfügen muss und schreibt mit der virtuellen Methode „calc_modul“ dessen Implementierung in den abgeleiteten Klassen vor. Hierbei werden aus den modulspezifischen Parametern die vorgeschriebenen Größen ermittelt. Die Datenhaltung der Parameter und die unterschiedlichen Rechenvorschriften werden in den Unterklassen, beispielhaft dargestellt als „Stirradgetriebe“ und „Riementrieb“, umgesetzt.

Ein geeigneter Vorschlag für die Umsetzung einer graphischen Oberfläche für die Interaktion

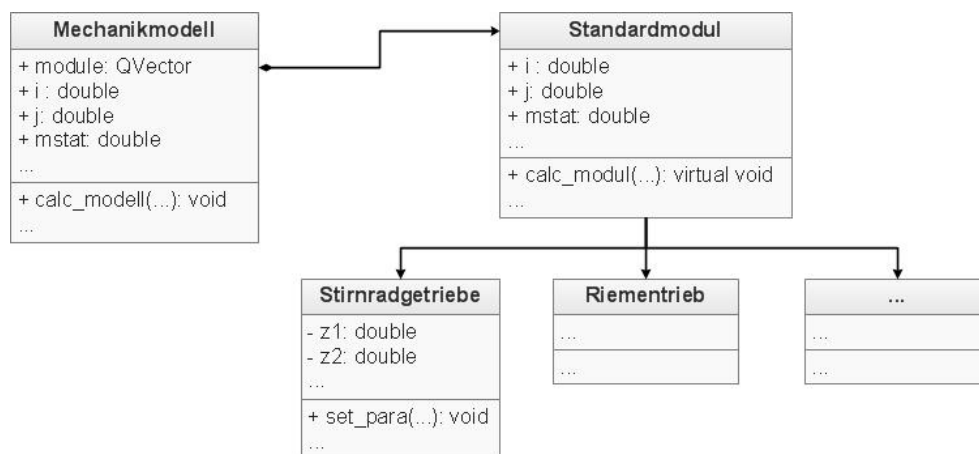


Abbildung 4.10: Klassendiagramm des mechanischen Modells

mit dem Anwender ist in [119] beschrieben, die Abbildung 4.11 stellt Teile der Oberfläche dar. Im linken Programmausschnitt der Abbildung können die mechanischen Module einer Antriebsmechanik konfiguriert werden. Der rechte exemplarische Programmausschnitt stellt die Eingabefelder für die Modellierung eines Riementriebs zur Verfügung. Die Implementierung ist mit der Zielstellung entworfen, eine möglichst hohe Integrität der Modellbildung in die Algorithmen aus der vorliegenden Arbeit zu erhalten.

Die Berechnung der Kennwerte des gesamten mechanischen Antriebsstrangs erfolgt innerhalb

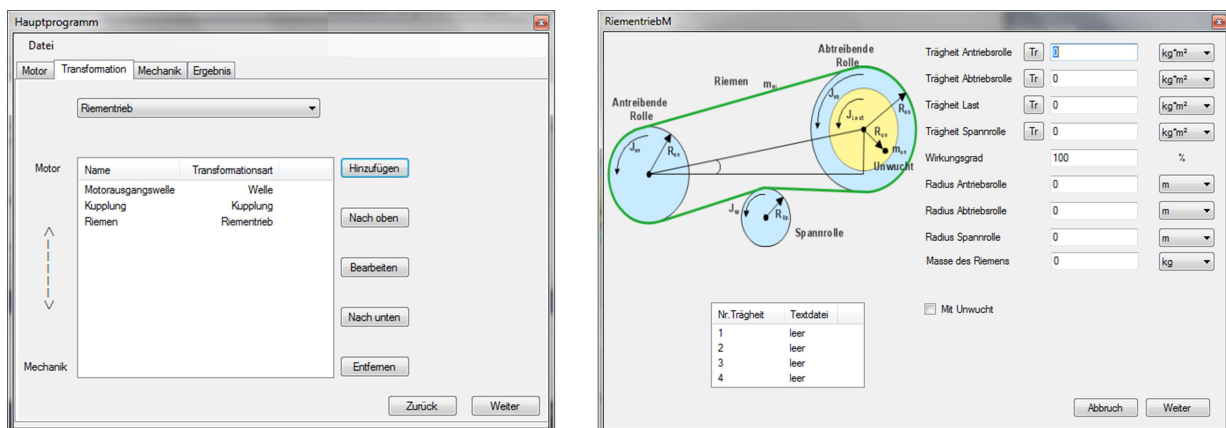


Abbildung 4.11: Programmauszug aus der mechanischen Modellbildung [119]

der Klasse „Mechanikmodell“, in der sämtliche Module in einem Vektor „module“ gespeichert sind. Die Methode „calc_modell“ bestimmt schließlich ein Gesamtmodell \mathcal{S} nach (4.3).

4.4.2 Implementierung der Auslegungsrechnung und Komponentenauswahl

Der beschriebene Vorrat an Komponenten (siehe (4.15) und (4.20)) liegt als Objekte der Klassen „Motor“ und „Stromrichter“ vor. Zudem ist eine Klasse zur Auslegungsrechnung implementiert, die mit den zuvor genannten beiden Komponenten im Klassendiagramm nach Abbildung 4.12 dargestellt ist. An die Methode „calc_auslegung“ werden die für die Auslegungsrechnung notwendigen Daten nach den Abschnitten 4.2 und 4.3 mittels Parameter übergeben. Sie wählt aus dem Vorrat Komponenten aus und bestimmt die kinematischen Restriktionen, die der Hardwareentwurf ausgibt.

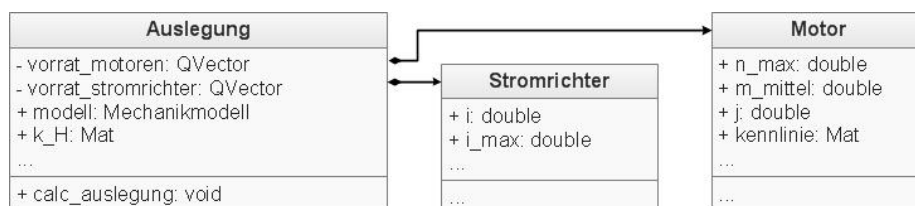


Abbildung 4.12: Klassendiagramm zur Auslegungsrechnung und Komponentenauswahl

Das Initialisieren der Objekte für Motoren und Stromrichter sowie das Belegen mit Werten geschieht über Schnittstellen zu Datenbanken. Über exklusive Datenbanken verfügen die Antriebshersteller und -anbieter, weshalb im Rahmen der vorliegenden Implementierung auf eine exemplarische Untersuchung [120] zurückgegriffen wird.

4.5 Zusammenfassung der Neuerung im Hardwareentwurf

Die bestehenden Methoden zur Modellbildung, Auslegungsrechnung und Komponentenauswahl müssen so modifiziert und erweitert werden, dass sie den Anforderungen an das neue Konzept genügen und algorithmisch integriert werden können.

Hierzu wurde zunächst eine Modularisierung von mechanische Antriebskomponenten entworfen. Bestehende Modelle von derartigen Komponenten wurden dann so angepasst, dass sie zum Einen hinreichend genaue Abbildungen der Mechanik zur Auslegungsrechnung liefern und zum Anderen formell als Elemente obiger Modularisierung verwendbar sind. Insgesamt genügen sieben entwickelte Standardmodule um den größten Teil existierender Antriebsmechanik für die Auslegungsrechnung zu beschreiben.

Neu ist die Vorgehensweise zur gezielten automatischen Auswahl von Motoren und Leistungselektronik, die durch eine Modifikation der Mathematik zur Auslegungsrechnung ermöglicht wird, siehe Gleichungen (4.10) bis (4.14). Eine weitere Neuentwicklung stellt die Implementierung der Auslegungsrechnung und der Komponentenauswahl, kompatibel im Kontext des neuen Gesamtkonzepts und dessen Umsetzung, dar.

5 Neu- und Weiterentwicklungen zum Systementwurf

Der dritte wesentliche Schwerpunkt der vorliegenden Arbeit beschäftigt sich mit dem Systementwurf.

Abbildung 5.1 zeigt die entwickelte Vorgehensweise beim Systementwurf. Zu Beginn des Kapitels soll die im Systementwurf zu lösende Aufgabenstellung analysiert und mathematisch definiert werden, bevor der Lösungsansatz mittels zeitbehafteter Petrinetze erläutert wird. Danach wird eine Optimierungsstrategie des Netzes dargestellt, die auf der Anpassung eines Initialsystems basiert (Abschnitt 5.2). Anschließend wird die Implementierung beschreiben (Abschnitt 5.3). Die

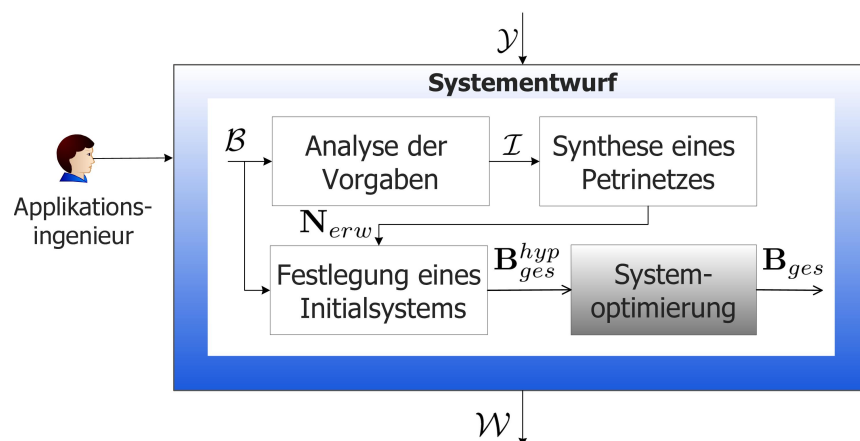


Abbildung 5.1: Übersicht über den Systementwurf

Vorgaben des Applikationsingenieurs sind in der Menge \mathcal{B} enthalten. Sie bestehen aus den gewünschten Zwischenzuständen, Restriktionen und Informationen über die Zusammenhänge im System. Daraus werden die Modellparameter, formuliert in der Menge \mathcal{I} für die Synthese eines Petrinetzes mit der erweiterten Struktur \mathcal{N}_{erw} abgeleitet. Hier wird ein Initialsystem festgelegt und deren Bewegungen \mathcal{B}_{ges}^{hyp} berechnet. Im letzten Schritt wird das System optimiert, das heißt, die optimierten Bewegungen \mathcal{B}_{ges} werden bestimmt. Zur Berechnung von Bewegungen und zur Anpassung der Hardware werden die bereits beschriebenen Signale \mathcal{W} an den Bewegungsent-

wurf gesendet und vom Hardwareentwurf \mathcal{Y} erhalten (siehe Abbildung 2.1).

5.1 Mathematische Modellierung der Anforderungen

Das gesamte Antriebssystem setzt sich aus n Achsen zusammen, wobei eine Achse i insgesamt m_i Teilbewegungen nacheinander ausführt. Die Teilbewegungen werden durch den Bewegungsentwurf ermittelt, und lassen sich deshalb im Rahmen des Systementwurfs rein durch Parameter beschreiben.

5.1.1 Beschreibung einer Bewegung

Um Anforderungen an Bewegungen so beschreiben zu können, dass sie durch die Algorithmen des ganzheitlichen Engineerings umgesetzt und schließlich das System optimiert werden kann, muss eine Konvention zur Modellierung einer Bewegung definiert werden. Die folgenden Gleichungen formulieren die im Rahmen der Arbeit entwickelte Darstellungsweise.

Die Gesamtbewegung einer Achse i lässt sich durch die in der Menge \mathcal{B}_i befindlichen Parameter beschreiben:

$$\mathcal{B}_i = \{\mathbf{X}_i; \vec{t}_i; \mathcal{B}_{konf,i}\} \text{ mit } i = 0, \dots, (n - 1) \quad (5.1)$$

Die Matrix \mathbf{X}_i

$$\mathbf{X}_i = \begin{bmatrix} \vec{x}_{i,0} & \dots & \vec{x}_{i,j} & \dots & \vec{x}_{i,m_i-1} \end{bmatrix}$$

enthält die Randwerte der Bewegungsabschnitte i, j , mit

$$j = 0, \dots, (m_i - 1),$$

wobei jeder Anfangspunkt einer Teilbewegung (i, j) auch der Endpunkt der vorausgegangenen Teilbewegung $(i, j - 1)$ ist. Handelt es sich insgesamt um eine periodische Bewegung, dann ist der letzte Randwert \vec{x}_{m_i} gleich dem ersten Randwert $\vec{x}_{i,0}$. In der Menge sind außerdem die Zeitpunkte \vec{t}_i hinterlegt, zu denen die Randwerte gültig sind.

Die Bewegung einer Achse i und deren Parameter werden durch Abbildung 5.2 verdeutlicht. Die Bewegung beginnt mit den Zeit $t_{i,0} = 0$ und endet mit der Taktzeit der Maschine $t_{i,m_i+1} = T_{Takt}$. Aus den absoluten Zeiten $t_{i,j}$ der Zwischenzustände folgen die Zeitdifferenzen der Bewegungsabschnitte. Exemplarisch ist der Verlauf der Strecke s_i aufgetragen.

Um eine Bewegung durch den Bewegungsentwurf und im Einklang mit dem Hardwareentwurf berechnen zu lassen, sind zu jedem Bewegungsabschnitt weitere Informationen in der Menge $\mathcal{B}_{konf,i}$ hinterlegt. Es handelt sich um spezifische Optimierungskriterien \mathbf{O}_i und kinematische Restriktionen \mathbf{K}_i . Je nach Anwenderwunsch können für eine gesamte Achse übergeordnete Optimierungskriterien $\vec{o}_{A,i}$ und kinematische Restriktionen $\mathbf{K}_{A,i}$ vorhanden sein. Die Kennzeichnung zur Bevorzugung wird in der Implementierung logisch festgelegt. Der Applikationsingenieur

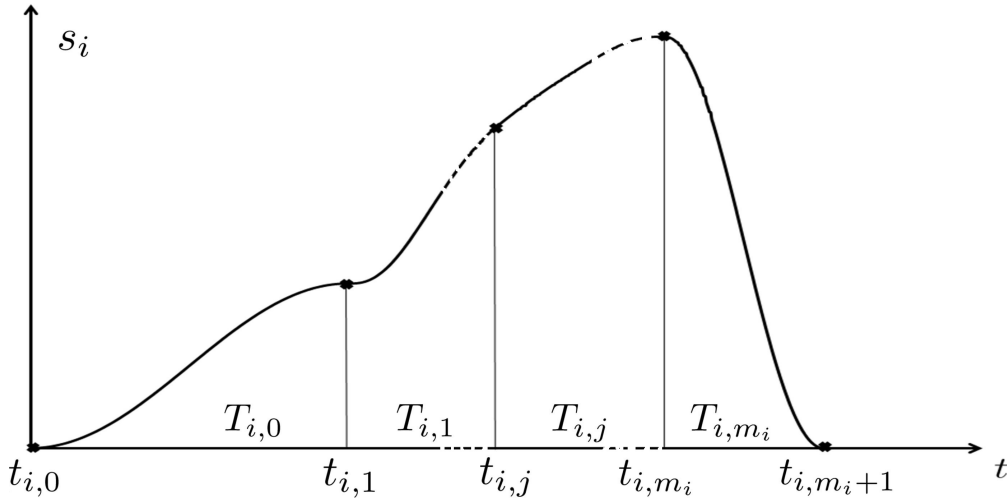


Abbildung 5.2: Exemplarische Strecke einer Achsenbewegung i

kann so festlegen, ob bei widersprüchlichen Vorgaben die der Achse oder die eines bestimmten Bewegungsabschnitts gültig sind.

$$\mathcal{B}_{konf,i} = \{\vec{o}_{A,i}; \mathbf{K}_{A,i}; \mathbf{O}_i; \mathbf{K}_i; \mathbf{K}_{ultimo,i}\} \quad (5.2)$$

Die Matrix \mathbf{O}_i enthält die Optimierungskriterien als Spalten für jeden Bewegungsabschnitt j :

$$\mathbf{O}_i = [\vec{o}_{i,0} \dots \vec{o}_{i,j} \dots \vec{o}_{i,m_i}] \quad \text{mit } \vec{o}_{i,j} = (o_{Energie} \ o_{Ruck} \ o_{Zeit})^T \quad (5.3)$$

Werte für die Relevanz von Energieoptimierung $o_{Energie}$, die Ruckoptimierung o_{Ruck} und Zeitoptimierung o_{Zeit} sind im Bereich $[0; 1]$ zugelassen und werden, falls nur einer der drei 1 und die anderen zwei 0 sind, der Einfachen Selektion, andernfalls der CI-Selektion zugeordnet (siehe Abschnitt 3.2).

Kinematische Restriktionen der einzelnen Abschnitte sind folgendermaßen in der Matrix \mathbf{K}_i angeordnet:

$$\mathbf{K}_i = [\mathbf{K}_{i,0} \ \dots \ \mathbf{K}_{i,j} \ \dots \ \mathbf{K}_{i,m_i}] \quad (5.4)$$

Die Matrix mit den ultimativen kinematischen Restriktionen nach Gleichung (5.6) ist analog aufgebaut:

$$\mathbf{K}_{ultimo,i} = [\mathbf{K}_{ultimo,i,0} \ \dots \ \mathbf{K}_{ultimo,i,j} \ \dots \ \mathbf{K}_{ultimo,i,m_i}] \quad (5.5)$$

Kinematische Restriktionen resultieren aus diversen Quellen und gehorchen nicht zwangsläufig einer Hierarchie. Soll eine Teilbewegung jedoch allen Restriktionen genügen, sind die Extrema zu ermitteln. Bezüglich der eingeführten Beschreibungen von Restriktionen leiten sich die ultimativen, für einen Bewegungsabschnitt gültigen kinematischen Restriktionen $\mathbf{K}_{ultimo,i,j}$ her zu:

$$\mathbf{K}_{ultimo,i,j} = \begin{bmatrix} \max \left(\mathbf{K}_{A,i} \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \mathbf{K}_{H,i} \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \mathbf{K}_{i,j} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \\ \min \left(\mathbf{K}_{A,i} \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \mathbf{K}_{H,i} \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \mathbf{K}_{i,j} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \end{bmatrix}^T \quad (5.6)$$

Hierbei werden die Maxima der Minima und die Minima der Maxima in $\mathbf{K}_{ultimo,i,j}$ eingetragen. Die Auswahl des größte Minimum aus allen Restriktionen schließt alle anderen Minima ein und umgekehrt. In der Formel sind die kinematischen Restriktionen der gesamten Achse $\mathbf{K}_{A,i}$ und des Teilabschnittes $\mathbf{K}_{i,j}$ durch den Applikationsingenieur vorgegeben, der den Prozess oder Vorgang an der Maschine abbildet. Einzig die durch den Hardwareentwurf vorgegebenen Restriktionen $\mathbf{K}_{H,i}$ können im Sinne einer Optimierung des Systems übergangen werden, was einer größeren Dimensionierung der Komponenten entsprechen kann. Um beispielsweise eine schnellere Bewegung zu realisieren, kann diese durch Missachtung der Restriktionen des momentan gewählten Motors berechnet werden. Anschließend wird für die schnellere Bewegung ein neuer Motor gewählt, dessen Restriktionen auch von der schnelleren Bewegung wieder eingehalten werden. Anhand der Parameter in \mathcal{B} kann bereits allein durch den Bewegungsentwurf, zusammengesetzt aus hypothetischen Teilbewegungen \mathbf{B}_{hyp} , eine hypothetische Gesamtbewegung \mathbf{B}_{ges}^{hyp} erstellt werden. Zur Berechnung eines Initialsystems werden kinematische Restriktionen ignoriert.

5.1.2 Innere Zusammenhänge des Antriebssystems

Die Parameter der Bewegungen sind nicht allein durch den Prozess beziehungsweise durch die expliziten Vorgaben des Applikationsingenieurs fest vorgeschrieben. Vielmehr geht es darum, jene Bewegungsparameter, die unter Restriktionen frei wählbar sind, so einzustellen, dass ein optimiertes Gesamtsystem erhalten wird. Dazu müssen jedoch die inneren Zusammenhänge des Systems, also die Beziehungen zwischen den Achsen, beschrieben und mathematisch abgebildet werden. Im Rahmen der Arbeit wurden mögliche Beziehungen analysiert, typisiert und schließlich eine Beschreibungsform entwickelt, die auf den im letzten Abschnitt entwickelten Konventionen basiert. Im Folgenden werden die Regeln zur Beschreibung dargestellt.

Es existieren vor allem zeitliche Zusammenhänge zwischen den Achsen, die in der Praxis aus einer konkurrierenden Raumnutzung oder einer erforderlichen Prozessreihenfolge resultieren. Zeitpunkte $t_{i,j}$, zu denen Randbedingungen gelten, können divers vorgegeben sein. Abbildung 5.3 gibt einen Überblick darüber, wie Zeitwerte zugewiesen werden können. Die entwickelte zugehörige mathematische Notation ist in Tabelle 5.1 notiert. Werte können explizit durch den Anwender bekannt sein. Eine Vorgabe kann aber auch lauten, die Zeit zu minimieren oder automatisch zu einem ausgeglichenen Wert in Bezug auf die benachbarten Bewegungsabschnitte zu gelangen. Eine weitere Möglichkeit ist die relative Wertzuweisung. Hierbei werden die Verknüpfungen zu anderen Achsen und deren Teilbewegungen hergestellt.

Auch für die relative Wertzuweisung existieren diverse Möglichkeiten. So kann der Zeitpunkt $t_{i,j}$ mit dem Zeitwert $t_{i,\tilde{j}}$ eines Zustandes einer anderen Achse gleichgesetzt werden. Er kann aber auch restriktiv von einer anderen Teilbewegung abhängen, zum Beispiel mit einem statischen zeitlichen Abstand. Unter einer dynamischen Restriktion ist ein durch den Bezug variabler zeitlicher Abstand zu verstehen. Dynamische Restriktionen können auch vielfach auf einen Randwert angewendet werden. Sie unterteilen sich in diverse Situationen, die in Tabelle 5.1 aufgelistet sind.

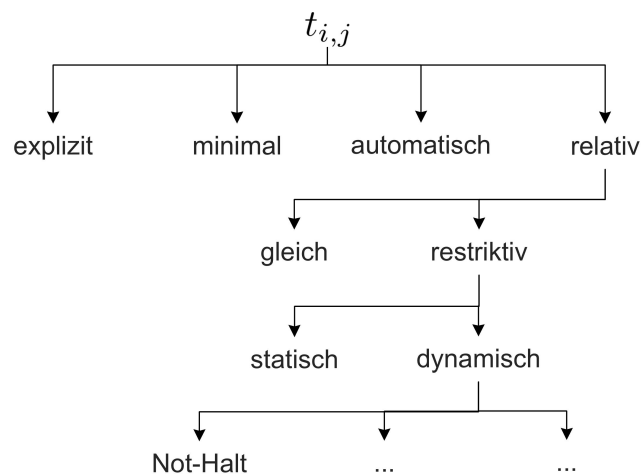


Abbildung 5.3: Arten innerer Zeitabhängigkeiten

Matrix	Bedeutung	Definitionsmenge	Dimension
C	Typen der Zeitzuweisungen der Randwerte	$c_{i,j} \in \{0 \dots 4\}$: 0 - nicht definiert 1 - explizit 2 - automatisch 3 - minimal 4 - relativ (siehe D)	$(n) \times (\max(m_i))$
D	Art der zeitlichen Relation eines Randwertes i, j zu jeweils anderen Randwerten \tilde{i}, \tilde{j}	$d_{\tilde{i}, \tilde{j}}^{i,j} \in \{0; 1; 2\}$: 0 - gleich 1 - restriktiv	$(n^2) \times (\max(m_i)^2)$
E	Typ eines zusätzlichen additiven Anteils einer Relation	$e_{\tilde{i}, \tilde{j}}^{i,j} \in \{0; 1\}$: 0 - statisch 1 - dynamisch	$(n^2) \times (\max(m_i)^2)$
F	Werte für statische Addition	$f_{\tilde{i}, \tilde{j}}^{i,j} \in \mathcal{R}$	$(n^2) \times (\max(m_i)^2)$
G	Fallunterscheidung für dynamische Addition	$f_{\tilde{i}, \tilde{j}}^{i,j} \in \{0 \dots 9\}$: 0 - Not-Stopp $< s_k$ 1 - Not-Stopp $> s_k$ 2 - $s < s_k$ von rechts 3 - $s < s_k$ von links 4 - $s > s_k$ von rechts 5 - $s > s_k$ von links 6 ... 9 - analog zu 2 ... 5 für v und v_k	$(n^2) \times (\max(m_i)^2)$
H	Werte der Relationen aus G	$h_{\tilde{i}, \tilde{j}}^{i,j} \in \mathcal{R}$	$(n^2) \times (\max(m_i)^2)$

Tabelle 5.1: Notation der inneren Zusammenhänge eines Antriebssystems

Die Menge \mathcal{I} besteht aus 6 Matrizen, die notwendig sind, um die beschriebenen Arten von Abhängigkeiten abzubilden:

$$\mathcal{I} = \{C; D; E; F; G; H\} \quad (5.7)$$

Grundsätzlich wird der Bewegungsabschnitt vor dem zugewiesenen Zwischenzustand einer anderen Achse verknüpft.

Die Matrix **C** enthält eine Typisierung der Zeiten aller Zustände. Da nicht alle Achsen eine gleiche Anzahl von vorgegebenen Zwischenzuständen haben, werden in der Matrix Komponenten ohne zugehörigen Zuständen mit „0“ gekennzeichnet. Vorhandene Zustände werden zwischen „1“ und „4“ bewertet. Zeiten können explizit als Wert vorgegeben sein, oder sie müssen durch Optimierungsalgorithmen minimiert oder automatisch an andere Zustände angepasst werden. Sie können jedoch auch relativ zu anderen Zuständen und Bewegungsabschnitten des Systems vorgegeben werden, was zu weiteren Fallunterscheidungen führt. So legt die Matrix **D** fest, ob die Zeit eines anderen Zustandes äquivalent übernommen wird oder ob restriktive Bedingungen vorliegen. Im Fall restriktiver Bedingungen legt die Matrix **E** fest, ob es sich um eine statische Addition oder Subtraktion eines Wertes (siehe Matrix **F**) handelt, oder ob der zu addierende / subtrahierende Wert dynamisch vom Verlauf der Bewegung vor dem verknüpften Zustand abhängt. Die dynamische Restriktion wird wiederum in zehn Fälle unterteilt (siehe Matrix **G**). Grundsätzlich ist zu unterscheiden, ob der Zeitpunkt so festgelegt wird, dass ein kinematischer Wert nicht über oder unterschritten wird. Die Fälle „2“ bis „9“ legen das für diverse Kriterien fest. Die aufgestellten Restriktionen lassen sich durch numerische Untersuchungen der bezogenen Bewegungen von rechts oder links erfüllen. Es ist jeweils der erste Zeitschritt gesucht, der die Restriktion erfüllt.

Die Fälle „0“ und „1“ sind, im Hinblick auf die Kollisionsvermeidung zweier Antriebsachsen

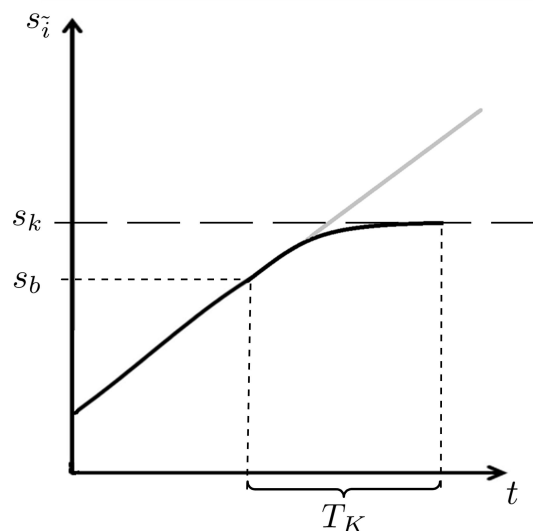


Abbildung 5.4: Bestimmung der kritischen Zeit für den rechtzeitigen Not-Stopp

in einem definierten Bereich der kritischen gemeinsamen Raumnutzung, als zeitlicher Puffer zu

verstehen. Der soll ein Stillsetzen des bezogenen Antriebs vor einer kritischen Position s_k ermöglichen, falls es bei der betrachteten Achse zum betrachteten Zeitpunkt zu einem Fehlerzustand kommt. Der entsprechende Zeitpuffer T_k ergibt sich durch eine Abschätzung mittels Bewegungsgleichung [37] mit konstanter Beschleunigung zu:

$$s = s_b + \frac{v_b^2}{2a_k} < s_k \quad \text{mit } s_b = f(T_k); \quad v_b = f(T_k) \quad (5.8)$$

Da der Bremspunkt s_b, v_b von der bezogenen Bewegung abhängt, wird er im vorliegenden Fall numerisch ermittelt, so dass die vorausgegangene Bedingung erfüllt ist:

$$\begin{aligned} T_k^u &= 0 \\ T_k^{u+1} &= T_k^u - T_A \end{aligned} \quad (5.9)$$

Abbildung 5.4 stellt die relevanten Größen in einem Diagramm dar. Die Achse, auf die sich bezogen wird, wird mit maximaler entgegen der Bewegung gesetzten Beschleunigung a_k gestoppt, wozu die Zeit T_K benötigt wird.

Die Struktur des Antriebssystems sowie dessen Parameter entstehen aus Informationen, die der Applikationsingenieur über eine Benutzerschnittstelle eingeben kann. Um algorithmisch analysieren und optimieren zu können, muss ein Modell abgeleitet werden, für das entsprechende mathematische Methoden zur Verfügung stehen. Im folgenden Abschnitt wird erläutert, wie mithilfe eines Petrinetzes das System abgebildet werden kann.

5.1.3 Synthese eines Petrinetzes

Die Zusammenhänge des Antriebssystems, insbesondere die Bewegungen der Achsen und die Abhängigkeiten der Zwischenzustände, werden im Folgenden als Petrinetz formuliert. Ziel ist die Analyse der Systemstruktur und die automatische Festlegung einer Optimierungsstrategie für das Antriebssystem. Das Petrinetz ist dazu so zu formalisieren, dass es dem entwickelten Gesamtkonzept und den Algorithmen zur Umsetzung gerecht wird. Zunächst wird ein Petrinetz für die Abbildung eines einzelnen Bewegungsabschnitts definiert, um es anschließend als Subnetz für ein Petrinetz des Gesamtsystems zu nutzen, was in einem weiteren Abschnitt erläutert wird.

Bewegungsabschnitt als hybrides Petrinetz

Ein hybrides Petrinetz berücksichtigt nicht nur die Abfolge von Zuständen hinsichtlich diskreter Schaltbedingungen, sondern kann auch kontinuierliche Aspekte berücksichtigen. Im vorliegenden Fall wird die Bewegung einer Achse in einem bestimmten Abschnitt als kontinuierlicher

Vorgang in ein Petrinetz einbezogen, indem Schaltvorgänge im Netz vom zeitlichen Bewegungsverlauf abhängen. Nachstehend wird ein geeignetes hybrides Netz zur Darstellung eines Bewegungsabschnittes hergeleitet.

In allen Fällen gilt die Kausalität der Bewegung einer Achse $t_{i,j-1} < t_{i,j} < t_{i,j+1}$. Abbildung 5.5 zeigt das hybride Petrinetz eines einzelnen Bewegungsabschnitts. Der obere Teil des Netzes stellt einen ereignisdiskreten Systemteil dar. Hat der vorherige Bewegungsabschnitt seinen Zielzustand erreicht (zum Zeitpunkt $t_{i,j-1}$) und ist der Eingang E ebenfalls markiert, kann die Eingangstransition feuern und der Abschnitt geht in einen Zwischenzustand über (Markierung von a und c). Im Zwischenzustand findet praktisch die Bewegung statt. Beim Erreichen des

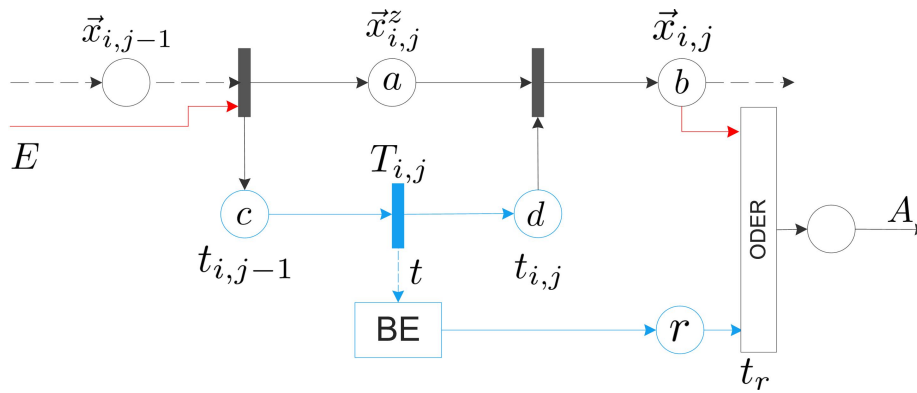


Abbildung 5.5: Petrinetzausschnitt eines Bewegungsabschnitts

Zielzustandes (zum Zeitpunkt $t_{i,j}$) wird die Stelle d markiert. Im diskreten Teil folgt die Markierung von b , die Teilbewegung ist abgeschlossen. Da die Zeiten und Bewegungen optimiert werden sollen, ist das Netz um einen kontinuierlichen Bereich erweitert worden. Die blau gefärbte Transition liefert einen Zeitfluss, mit dem, unter Zuhilfenahme des Bewegungsentwurfs (BE), die dynamischen Restriktionen behandelt werden können. Die Stellen r zum Ausgang A des Bewegungsabschnittes werden nach dem Erfüllen der jeweiligen Restriktion oder nach dem Durchlaufen des Abschnitts markiert.

Der diskrete Systemteil gibt Aufschluss über den grundsätzlichen Ablauf im System, während der kontinuierliche die zeitlichen Abläufe bestimmt. So kann entsprechend der formulierten inneren Zusammenhänge ein Ausgang auch vor Abschluss einer Bewegung aktiviert werden, wenn die Restriktion entsprechendes voraussetzt.

Der diskrete Zustand eines Bewegungsabschnitts zum Schritt k ist durch das Notieren der Markierungen in $\mathbf{Z}_{i,j}$ möglich:

$$\mathbf{Z}_{i,j}[k] = [a_{i,j} \quad b_{i,j} \quad c_{i,j} \quad d_{i,j}]^T \quad (5.10)$$

Durch Verschalten von mehreren Bewegungsabschnitten lässt sich das Antriebssystem darstellen. Dabei wird die Stelle b mit der linken Transition des folgenden Abschnitts verknüpft und bei

lateralen Verbindungen werden Ausgänge A mit Eingängen E verknüpft. Wie sich das im Rahmen der Arbeit entwickelte Petrinetz eines Bewegungsabschnitts als Subnetz für die Abbildung des gesamten Systems verwenden lässt, ist im folgenden Abschnitt erläutert.

Antriebssystem als hybrides Petrinetz

Das im letzten Abschnitt entwickelte Petrinetz wird im Folgenden als abstraktes Subsystem verwendet. Abbildung 5.6 stellt dar, wie ein Antriebssystem aus derartigen Subsystemen aufzubauen ist. Das Subsystem eines Bewegungsabschnitts i, j ist definiert als Menge $\mathcal{A}_{i,j}$. Die rot markierten lateralen Verbindungen stellen Abhängigkeiten dar. Ein Petrinetz kann, wie in der Einleitung

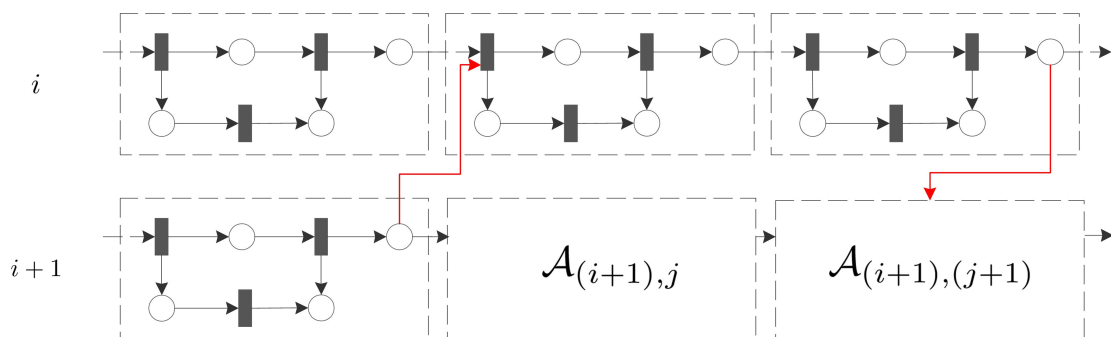


Abbildung 5.6: Ausschnitt des Petrinetzes eines Antriebssystems

(Abschnitt 1.2.5) beschrieben, durch eine Matrix dargestellt werden [74]. Für die vorliegende Aufgabenstellung wird zur algorithmischen Umsetzung ebenfalls eine Beschreibung des Netzes als Matrix hergeleitet. Der diskrete Zustand des gesamten Systems \mathbf{Z} zu einem diskreten Schritt k ist in der Matrix $\mathbf{Z}[k]$ gespeichert:

$$\mathbf{Z}[k] = \begin{bmatrix} \mathbf{Z}_{0,0}[k] & \dots & \dots & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ & \dots & \mathbf{Z}_{i,j}[k] & \dots & \dots & \mathbf{Z}_{i,m_{max}}[k] \\ \vdots & & & & & \vdots \\ \mathbf{Z}_{n,0}[k] & \dots & \mathbf{Z}_{n,m_n}[k] & 0 & \dots & 0 \end{bmatrix} \quad (5.11)$$

Die Laufvariable k kennzeichnet somit eine Änderung der Markierung im gesamten Petrinetz, was einer diskreten Zustandsänderung oder der Erfüllung einer Bedingung aus dem kontinuierlichen Teil eines Bewegungsabschnitts entspricht.

Im Folgenden wird die Herleitung der Struktur des Netzes erläutert, welches obige Zustandsnotation besitzt.

Ein Aufbauen der Netzstruktur beginnt mit einem Netz aus Bewegungsabschnitten, beschrieben

anhand der Matrix \mathbf{N} . Hierin wird zunächst festgehalten, welche Bewegungsabschnitte es zu jeder Achse gibt. Die Elemente $n_{i,j}$ tragen den Wert 1 für einen vorhandenen Bewegungsabschnitt und sonst den Wert 0.

Bei Betrachtung der diskreten Zusammenhänge zwischen den Bewegungsabschnitten, wird eine erweiterte Matrix \mathbf{N}_{erw} abgeleitet, welche die Verknüpfungen enthält.

$$\mathbf{N}_{erw} = \begin{bmatrix} \mathbf{N}_{0,0} & \dots & \dots & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ & \dots & \mathbf{N}_{i,j} & \dots & \dots & \mathbf{N}_{i,m_{max}} \\ \vdots & & & & & \vdots \\ \mathbf{N}_{n,0} & \dots & \mathbf{N}_{n,m_n} & 0 & \dots & 0 \end{bmatrix} \quad (5.12)$$

Zu jedem Bewegungsabschnitt des Systems ist eine Untermatrix $\mathbf{N}_{i,j}$ vorhanden, in der die Präkanten zu anderen Abschnitten (\tilde{i}, \tilde{j}) mit einer 1 gekennzeichnet sind.

$$\mathbf{N}_{i,j} = \begin{bmatrix} n_{i,j}^{0,0} & \dots & \dots & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ & \dots & n_{i,j}^{\tilde{i},\tilde{j}} & \dots & \dots & n_{i,j}^{\tilde{i},m_{max}} \\ \vdots & & & & & \vdots \\ n_{i,j}^{0,0} & \dots & n_{i,j}^{n,m_n} & 0 & \dots & 0 \end{bmatrix} \quad (5.13)$$

Die Art einer jeden Verknüpfung ist in der bereits unter Abschnitt 5.1.2 definierten Menge \mathcal{I} vorgeschrieben. Ausgehend von der bekannten Netzstruktur und der geforderten inneren Zusammenhänge können die zu optimierenden Parameter und schließlich die Bewegungsverläufe über die in den Abschnitten 3.1 und 3.2 hergeleiteten Gleichungen bestimmt werden.

Nachdem die Berechnung der Struktur des Antriebssystems formuliert ist, müssen geeignete Vorgehensweisen und Gleichungen zur Bestimmung eines zu optimierenden Initialsystems entworfen werden.

5.2 Entwürfe zur Optimierung des Gesamtsystems

Zur Optimierung des Antriebssystems muss zunächst ein System mit konkreten Werten für alle Parameter bestimmt werden, die dann numerisch überarbeitet werden. Durch den Anwender sind jedoch nicht zwingend alle Parameter bestimmt, sondern vielmehr deren Restriktionen. Im folgenden Abschnitt wird darum gezeigt, wie ein initiales System anhand gegebener Anforderungen aufgebaut werden kann. Danach wird erläutert, wie zwei Optimierungsverfahren für zeitliche Zusammenhänge aufgebaut sind. Die Optimierung des Gesamtsystems erfolgt entsprechend dem Ablaufdiagramm aus Abbildung 5.7. Ausgehend vom Initialsystem ist zu unterscheiden, ob die

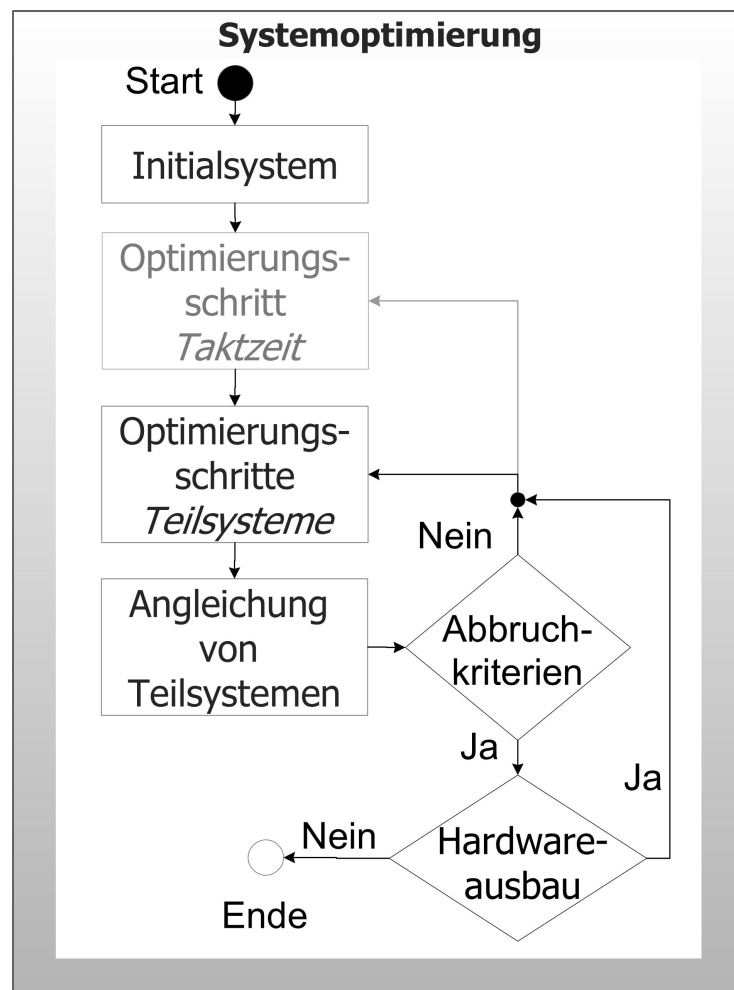


Abbildung 5.7: Systemoptimierung

Taktzeit der Maschine minimiert werden soll oder nur Teilbewegungen des Antriebssystems. Eine Anpassung der Taktzeit erfordert die Anpassung von zu minimierenden Teilsystemen. Nach jedem Anpassungsschritt wird eine Angleichung von Teilsystemen durchgeführt. Nach erfolgter Optimierung, die durch Abbruchkriterien beendet wird, ist zu prüfen, ob eine Anpassung der Hardware und damit der zugehörigen Restriktionen erfolgt, oder ob sich keine weitere Optimierung des Systems anschließt.

5.2.1 Ermittlung eines Initialsystems

Um ein Initialsystem zu ermitteln wird eine Erreichbarkeitsanalyse des zugehörigen Petrinetzes durchgeführt. Dabei ergibt sich unter Betrachtung des diskreten Systems eine Markierungsfolge Z_i . Der erste und letzte Randwert einer Bewegung besitzen immer eine explizite zeitliche

Zuordnung, nämlich die Null sowie die Taktzeit T_{Takt} . Nicht explizit vorgegebene Zeiten von Randwerten müssen für ein initiales System erst bestimmt werden.

Die Vorgabe der Zeiten erfolgt sukzessive in Orientierung an der Markierungsfolge entsprechend der vorliegenden Fälle:

- Ist ein expliziter Zeitwert mit dem Zielzustand verbunden, dann wird der folgende Zielzustand untersucht.
- Ist nur ein minimal oder automatisch zu belegender Zeitwert gefordert, wird solange weiter untersucht, bis ein expliziter oder relativer Zeitwert vorliegt. Die zeitlich nicht explizit festgelegten Randwerte werden schließlich äquidistant belegt.
- Wird ein relativ zu belegender Zeitwert angetroffen, dann ist entweder der bedingende Zeitwert schon gesetzt und der Wert kann zunächst übernommen werden, oder es muss die folgende Achse untersucht werden.

Anhand der entwickelten Vorgehensweise können alle Parameter des Initialsystems festgelegt werden und schließlich mit Hilfe des Bewegungsentwurfs hypothetische Bewegungen B_{ges}^{hyp} berechnet werden.

Liegen die Bewegungsverläufe aller Achsen vor, so können die relativ dynamisch gesetzten Zeitwerte entsprechend obiger Reihenfolge präziser gesetzt werden. In den noch anzupassenden Zeitwerten liegt schließlich das zeitliche Optimierungspotenzial der Maschine. Sowohl zu minimierende Zeiten, als auch die Zeiten, die entsprechend innerer Zusammenhänge anzupassen sind, müssen angepasst werden.

Hinzu kommen Zeiten, die „automatisch“ an die Randbedingungen anzupassen sind. Der Fall tritt ein, wenn ein Zwischenzustand zeitlich nicht explizit festgelegt ist und somit eine Angleichung von Bewegungsabschnitten an deren Kosten entsprechend relevanter Verlustfunktionen notwendig ist. Dabei muss es sich nicht um Abschnitte einer Achse handeln, sondern es kann auch ein Pfad von Bewegungen durch das Systemnetz über diverse Achsen sein.

Die Angleichung von mehreren Bewegungsabschnitten untereinander wird im anschließenden Abschnitt erläutert.

5.2.2 Ausgewogener Zeitverlust in einem zusammengesetzten Bewegungsabschnitt

Sollen mehrere Bewegungsabschnitte zeitlich entsprechend deren Anforderungen angeglichen werden, muss eine geschickte Verteilung der verfügbaren und unveränderlichen Gesamtzeit T_{ges} erfolgen. Das Problem bei der Angleichung der Bewegungen liegt darin, dass kein gegebener Zusammenhang zwischen der Zeit einer Teilbewegung und der Kosten der Bewegung verfügbar ist.

Ein hilfreiches Vergleichssystem zur Lösung des Problems kann mit der Gestaltung eines physikalisch fluidischen Modells dargestellt werden. Es handelt sich um einen Aufbau aus wassergefüllten Zylindern, wie exemplarisch in Abbildung 5.8 dargestellt.

Sind verschieden hoch gefüllte Zylinder unterschiedlicher Radien über ein Leitungssystem von

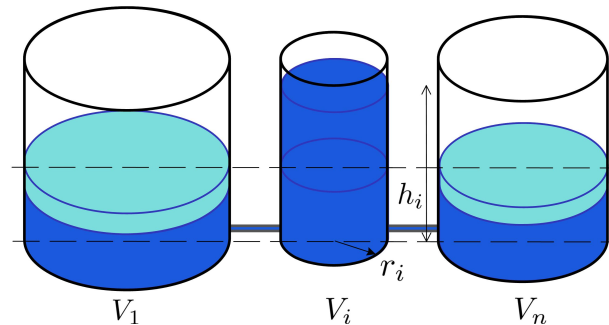


Abbildung 5.8: Wassersäulenmodell

einem zum anderen Zylinder und vom ersten zum letzten verbunden, so stellt sich durch verlustbehaftete Strömungen eine stabile Ruhelage ein, in der alle Zylinder die gleiche Füllhöhe aufweisen. Zur Bestimmung des Flüssigkeitsdrucks auf die Zu-/Ableitung eines Gefäßes, ist nur die Füllhöhe h_i , nicht aber das enthaltene Volumen V_i oder der Radius r_i eines Zylinders relevant ($p = \rho gh$, mit Dichte ρ und Gravitationskonstante g , siehe [121]). Für die verlustbehaftete Strömungsstärke I_v zwischen den Zylindern gilt ein proportionaler Zusammenhang mit der Druckdifferenz Δp über die Konstante R_v ($\Delta p = R_v I_v$, [122]). Bei Vernachlässigung der Trägheit des Fluids ist das System nicht schwingungsfähig.

Das Modell stellt ein Analogon zur Optimierungsstrategie dar, in dem die Volumina V_i den Zeiten T_i für die Bewegungsabschnitte und die invertierten Füllhöhen $1 - h_i$ den Kosten K_p entsprechen. Der Flüssigkeitsaustausch zwischen den Zylindern entspricht somit einem Zeitaustausch zwischen den Bewegungsabschnitten. Gleiche Füllhöhen entsprechen gleichen Kosten für alle Teilbewegungen. Der Zusammenhang zwischen Kosten und Zeitbedarf eines Bewegungsabschnitts, sprich der Radius im Modell, ist nicht bekannt. Dennoch kann in rekursiver Weise ein Gleichgewicht bestimmt werden, denn allein durch die Kenntnis der Füllhöhen kann der Volumenstrom zwischen den Behältern bestimmt werden, also auch der „Zeitfluss“ zwischen den Bewegungsabschnitten allein durch die Kosten. Die geflossene Zeit von einem Abschnitt zu den benachbarten in einem Berechnungsschritt s ist folglich in Analogie zum fluidischen Vergleichssystem ($\Delta V = I_v \cdot \Delta t = \frac{\Delta p \Delta t}{R_v}$) definiert zu:

$$\Delta T_p = \frac{T_{ges}}{Rl} (K_{p-1}^s + K_{p+1}^s - 2K_p^s) \quad (5.14)$$

Dabei stellt der Faktor R die Konstanten für Dichte, Verlustfaktor und Gravitation des Vorbildsystems dar. Anstelle des Drucks auf Leitungshöhe müssen die Kosten K_p für jeden Bewegungsabschnitt p definiert werden. Als Kostenfunktion ist das Produkt der Betragsmaxima von Geschwindigkeit v_p und Beschleunigung a_p des Abschnitts bezogen auf die globalen Betragsmaxi-

ma aller betrachteten Bewegungsabschnitte $p = 1 \dots l$ aufgestellt:

$$K_p = \frac{\max |v_p(t)|_t}{\max |v_p(t)|_{p,t}} \cdot \frac{\max |a_p(t)|_t}{\max |a_p(t)|_{p,t}} \quad (5.15)$$

Wegen der im Initialsystem vorhandenen gleichen Zeiten für alle Bewegungsabschnitte $T_p = T_{ges} l^{-1}$, kann über die notierte Rekursion das angestrebte Gleichgewicht schrittweise bestimmt werden, siehe Abbildung 5.9.

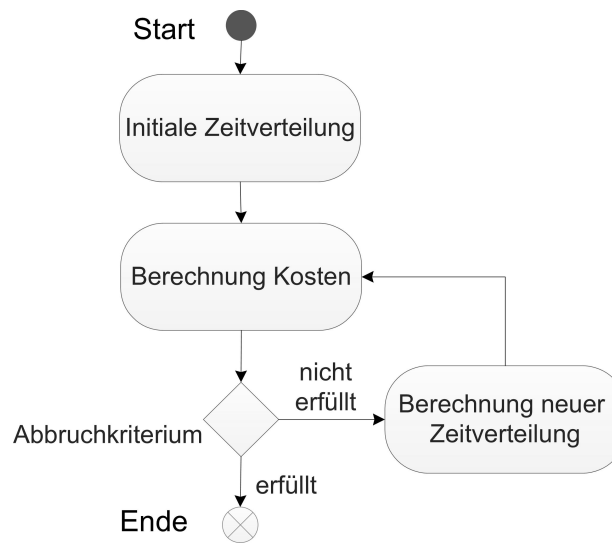


Abbildung 5.9: Verteilungsalgorithmus

Dabei wird für den Berechnungsschritt $s + 1$ aus den Gleichung (5.14) und (5.15) hergeleitet:

$$T_p^{s+1} = T_p^s + \frac{T_{ges}}{Rl} (K_{p-1}^s + K_{p+1}^s - 2K_p^s), \text{ für } p = 2, \dots, l-1 \quad (5.16)$$

$$T_1^{s+1} = T_1^s + \frac{T_{ges}}{Rl} (K_l^s + K_2^s - 2K_1^s)$$

$$T_n^{s+1} = T_l^s + \frac{T_{ges}}{Rl} (K_{l-1}^s + K_1^s - 2K_p^s)$$

Als Abbruchkriterium σ wird der Durchschnitt der Summe der quadratischen Abweichungen der Einzelkosten von den durchschnittlichen Kosten E verwendet:

$$\sigma = \frac{1}{l} \sum_{p=1}^l (K_p - E)^2 \text{ mit } E = \frac{1}{l} \sum_{p=1}^l K_p \quad (5.17)$$

$$\sigma < 5\%$$

Der Algorithmus bricht ab, sobald der Wert der entwickelten Formel die experimentell festgelegte Grenze von 5% unterschreitet.

Die für die Optimierungsstrategie entworfene Methode zur Angleichung der Zeit für Bewegungsabschnitte kann im Rahmen der zeitlichen Anpassung für das ganze System benutzt werden, die im folgenden Abschnitt beschrieben wird.

5.2.3 Zeitminimierung

Die Minimierung eines Zeitabschnitts ($t_{i,j} \rightarrow Min.$) oder der Taktzeit des gesamten Systems erfolgt über ein Intervallhalbierungsverfahren (Bisektion) [123]. Vor dem Beginn des Algorithmus müssen Startwerte festgelegt werden. Den kleineren Zeitwert liefert dabei der letzte vorangegangene Randwert mit definierter Zeit. Die obere Grenze besteht in der zu minimierenden Zeit selbst. Ebenfalls vor Beginn der Optimierung ist zu prüfen, ob der obere Grenzwert eine umsetzbare Lösung darstellt. Andernfalls ist sie solange um 20% der Intervallbreite zu erhöhen, bis eine akzeptable Lösung vorliegt. Der Vergrößerungsfaktor ist im Rahmen der Implementierung durch empirische Versuche festgelegt worden.

Von einem Optimierungsschritt s zum nächsten $s + 1$ läuft das Verfahren für die Anpassung der Taktzeit entsprechend folgender Notation, basierend auf [123]:

$$t_{\frac{1}{2}}[s] = t_{min}[s] + \frac{1}{2} (t_{max}[s] - t_{min}[s]) \quad (5.18)$$

$$\begin{aligned} \text{falls } t_{\frac{1}{2}}[s] \text{ eine Lösung liefert: } & t_{min}[s + 1] = t_{\frac{1}{2}}[s] \\ & t_{max}[s + 1] = t_{max}[s] \\ \text{sonst: } & t_{min}[s + 1] = t_{min}[s] \\ & t_{max}[s + 1] = t_{\frac{1}{2}}[s] \end{aligned}$$

Liefert die Zeit $t_{\frac{1}{2}}$ eine umsetzbare Lösung, dann ist sie für den nächsten Rechnungsschritt die obere Intervallgrenze t_{max} andernfalls die untere t_{min} . Das Verfahren ist fortzusetzen, bis die Intervallbreite eine Genauigkeit von t_e unterschreitet. Die Genauigkeit kann vom Applikationsingenieur vorgegeben werden, sie darf aber die zeitliche Auflösung der Bewegung nicht unterschreiten.

Eine brauchbare Lösung liegt genau dann vor, wenn für alle betroffenen Teilbewegungen unter vorangehender kinematischer Selektion eine Bewegung selektiert werden konnte. Bei Abschnitten mit Optimierungskriterien, die nicht eindeutig ein Zeitoptimum fordern (Einfache Selektion), dürfen die selektierten Bewegungen mit und ohne kinematischer Selektion nicht abweichen. Hiermit wird erreicht, dass eine hinsichtlich bestimmter Kriterien selektierte Bewegung nicht durch die Taktzeitminimierung eliminiert werden kann.

5.3 Implementierung des Systementwurfs

Abbildung 5.10 zeigt das Klassendiagramm des Systementwurfes. Das Diagramm gibt die Klassen an, in denen die beschriebenen Algorithmen umgesetzt sind.

Die übergeordnete Klasse „Maschine“ enthält globale Daten, wie die Taktzeit der Maschine,

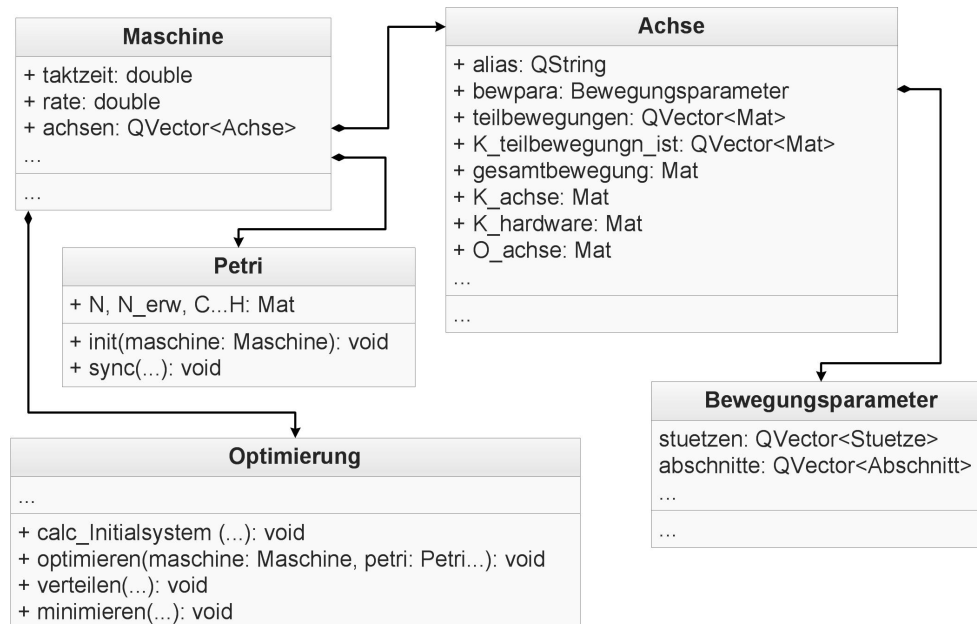


Abbildung 5.10: Klassendiagramm zum Systementwurf

aber auch die Abtaste „rate“, in der die Bewegungen numerisch bestimmt werden. Sie enthält jedoch auch Objekte des Typs „Achse“, in denen die Achsen des Antriebssystems hinterlegt sind. Die genannten Objekte beinhalten die Teilbewegungen B , die vom Bewegungsentwurf bestimmt wurden (Gleichungen (3.16) bis (3.1.2)). Aber auch für die Achse bestimmte Restriktionen und Optimierungsbedingungen werden hierin festgehalten.

Die Bewegungsparameter B und die zugehörigen Bedingungen werden wiederum in Objekten der Klasse „Bewegungsparameter“ innerhalb eines Objektes von „Achse“ festgehalten. Für die Definition von Randbedingungen und Restriktionen sowie weiterer Vorgaben sind die zur Bewahrung der Übersicht nicht dargestellten Klassen „Stuetze“ und „Abschnitt“ entworfen. Sie beinhalten neben der Bewegungsparameter der Achse auch die Informationen über innere Zusammenhänge, die der Anwender vorgibt, siehe Abbildung 5.11.

Weitere bedeutsame Klassen sind die Klassen „Petri“ und „Optimierung“. Sie verfügen über die Methoden zur Synthese und Speicherung eines Petrinetzes entsprechend der Abschnitte 5.1.2 und 5.1.3. Außerdem stellen sie die Optimierungsalgorithmen zur Berechnung eines Initialsystems „calc_Initialsystem“ und zur Gesamtoptimierung „optimieren“ bereit. Die Methoden „verteilen“ und „minimieren“ enthalten die Berechnungen nach den vorausgegangenen Abschnitten.

Abbildung 5.12 zeigt einen Programmausschnitt aus dem Systementwurf. Die Darstellung aus

der experimentellen Implementierung der Software stellt das Hauptfenster dar, welches der Applikationsingenieur anwendet.

Im Hauptfenster werden die Bewegungsverläufe von zwei Achsen „1“ gleichzeitig dargestellt. In den jeweiligen Tabellen „2“ können die Randwerte „3“, die spezifischen Restriktionen und Optimierungskriterien „4“ sowie die Abhängigkeiten manipuliert werden „5“. Relative Abhängigkeiten werden gesondert behandelt.



Abbildung 5.11: Programmausschnitt für innere Abhängigkeiten

Ebenfalls dargestellt sind die Taktzeit und Abtastezeit der Maschine „6“ und Schaltflächen, mit denen die Bearbeitung der Achsen „7“ und deren Hardware „8“ sowie dem mechanischen Modell „9“ erfolgen kann.

5.4 Zusammenfassung der Neuerungen im Systementwurf

Der Systementwurf stellt den Kern der implementierten Benutzeranwendung dar und verfügt über eine neue Strategie zum Entwurf und zur Optimierung eines Antriebssystems. Er enthält Neuentwicklungen um die Vorgaben des Applikationsingenieurs mathematisch abzubilden. Hierzu gehört das mathematische Modell einer Bewegung mit den in den Gleichungen (5.1) bis (5.6) festgelegten Parametern.

Eine wesentliche Neuheit bildet die mathematische Beschreibung von inneren Zusammenhängen des Antriebssystems. Hierzu wurden zunächst Typen von Abhängigkeiten definiert (siehe Abbildung 5.3 und Tabelle 5.1) und dann sechs Matrizen definiert (siehe Gleichung (5.7)), welche die Zusammenhänge mathematisch abbilden. Die Modifikation einer Bewegungsgleichung ermöglicht dabei zusätzlich eine Kollisionsbetrachtung, es entstanden die Gleichungen (5.8) bis (5.9). Um nun den gesamten Aufbau und die zeitlichen Abläufe des Antriebssystems darzustellen wurde ein hybrides Petrinetz entwickelt. Dieses Netz besteht aus miteinander verknüpften Subnetzen, wie in den Abbildung 5.5 und 5.6 dargestellt. Die neue Optimierungsstrategie für Antriebssysteme wurde mithilfe zweier neu entworfener Algorithmen realisiert. Die zeitliche Anpassung einzelner Bewegungsabschnitte geschieht über ein, aus der Analogie zu Wassersäulen hergeleitetes, Verfahren (siehe Abschnitt 5.2.2). Die Minimierung der Takzeit wird über eine modifizierte Bisektion vorgenommen, wie sie durch die Gleichung (5.18) beschrieben ist.



Abbildung 5.12: Programmausschnitt der Benutzeroberfläche des Systementwurfs

6 Ergebnisse

In diesem Kapitel werden die im Rahmen der vorliegenden Arbeit erhaltenen simulativen und experimentellen Ergebnisse vorgestellt und diskutiert. In den ersten beiden Abschnitten des Kapitels sind Antriebssysteme für einen exemplarischen Anwendungsfall dokumentiert. Sie stellen die Ergebnisse für herkömmliche Entwurfsmethoden dar und dienen als Referenz für einen Entwurf nach den vorgestellten neuen Methoden. Im Weiteren werden die erstellten Algorithmen hinsichtlich der Anforderungen (siehe Abschnitt 2.1) bewertet. Schließlich werden die Qualität der Algorithmen zum Entwurf und zur Optimierung eines Antriebssystems auf Basis der ganzheitlichen Methode experimentell an einer neu konzipierten Modellmaschine nachgewiesen.

6.1 Referenzdaten

Als industriell angewendetes Beispiel dient ein Längstaktmontageautomat (LTMA), wie er bereits in Abschnitt 1.2.2 erläutert ist. Nach der Beschreibung der Aufgabenstellung werden zwei Lösungen aus einem manuellen Entwurf und einer Entwicklung mittels gängiger CAE-Software vorgestellt.

Zum Vergleich der Ergebnisse werden sowohl für die Referenzdaten als auch für die nach der neuen Methoden berechneten Lösungen Kennwerte ermittelt. Sie werden jeweils über äquivalente Berechnungen gewonnen. Zum Vergleich der Berechnungsergebnisse dienen die Taktzeit zur Bewertung der Zeitoptimalität, der maximal auftretende Ruck zur Ruckoptimalität sowie ein Kennwert zur Untersuchung der Energieoptimalität. Während die Taktzeit T_{Takt} des entworfenen Systems explizit als Entwurfparameter zur Verfügung steht, wird der maximale Ruck $|r|_{max}$ einer jeden bewegten Achse aus dem jeweiligen Bewegungsverlauf entnommen. Eine Kennzahl für den energetischen Aufwand k_E einer Bewegung lässt sich in Anlehnung an die bereits hergeleitete Verlustfunktion (3.24) gewinnen. Über den gesamten Bewegungsverlauf mit n numerischen Werten einer Achse i werden die Produkte von den durchschnittlichen Beschleunigungen a_i und den relativen Streckenabschnitten aus s_i summiert.

$$k_E = \sum_{s=2}^n \left| \frac{1}{2} (a_i(s) + a_i(s-1)) (s_i(s) - s_i(s-1)) \right| \quad (6.1)$$

Die Kennzahl ist proportional zur geleisteten mechanischen Arbeit bei der untersuchten Bewegung an einer Achse. Verschiedene Bewegungsentwürfe sind darüber vergleichbar, wenn sie mit der selben Last, sprich bewegter Masse und Trägheitsmoment, belastet sind.

Das Maschinenkonzept und die Antriebsaufgaben des LTMA entsprechen der in [124] behandelten Aufgabenstellung mit Bezug zu einem real eingesetzten System. Für die beabsichtigten Berechnungen sollen der Vorschub und der Werkzeugantrieb des LTMA betrachtet werden, wobei der LTMA eine Taktzahl von ungefähr 90 Takten pro Minute erreichen soll. Folgende Größen sind für alle hierzu untersuchten Berechnungsverfahren äquivalent vorgegeben.

Werkzeugantrieb:

Es handelt sich verallgemeinernd um einen rotatorischen Antrieb, der im Bereich „offener Winkel“ (zwischen $\varphi_{O,W}$ und $\varphi_{G,W}$) eine Neupositionierung des Vorschubs zulässt. Der Werkzeugantrieb muss still gesetzt werden können, falls der Vorschub nicht korrekt stoppt. Das bedeutet, wenn nicht garantiert ist, dass der Vorschub korrekt und rechtzeitig positioniert, muss die Werkzeugachse noch vor Erreichen des kritischen Winkels $\varphi_{G,W}$ zum Stillstand kommen. Das Massenträgheitsmoment J_W beinhaltet bereits das vorab entsprechend Gleichung (4.19) geschätzte Massenträgheitsmoment des Motors.

Gesamtes Massenträgheitsmoment	$J_W = 16,38 \text{ kgdm}^2$
Offener Winkel (1/3)	$\varphi_{O,W} \dots \varphi_{G,W} = 1,047 \dots 2,094 \text{ rad}$
Statisches Lastmoment	$M_{S,W} = \pm 192 \text{ Nm}$
Reibmoment	$M_{R,W} = 20 \text{ Nm}$
Übersetzung Planetengetriebe	$i_W = 25$

Vorschub:

Beim Vorschub handelt es sich verallgemeinernd um einen Riemenantrieb, der über zwei direkt angetriebene Riemenscheiben bewegt wird. Die Last verteilt sich somit auf zwei Antriebe, die Berechnungen beziehen sich auf einen Motor mit den halben Belastungen des LTMA. Der Riemen muss so positioniert werden, dass er die Bedingungen der Werkzeugachse hinsichtlich des Kollisionsschutzes einhält. Das Massenträgheitsmoment J_V setzt sich zusammen aus dem der Riemenscheiben und dem der bewegten Masse inklusive Riemen sowie einem vorab bestimmten Trägheitsmoment des Motors.

Gesamtes Massenträgheitsmoment	$J_V = 135 \text{ kgdm}^2$
Vorschubstrecke	$\Delta s_V = 150 \text{ mm}$
Wirkkreisdurchmesser	$d_V = 305,8 \text{ mm}$
Reibmoment (aus 40 N)	$M_{R,V} = 12 \text{ Nm}$
Übersetzung Planetengetriebe	$i_V = 25$

Vereinfachend ist anzunehmen, dass der Wirkungsgrad der Antriebsstränge Eins beträgt, weil der tatsächliche Wert keine Rolle für einen Vergleich zwischen den Entwurfsmethoden spielt. Ebenso werden Auslegungsregeln bezüglich der Antriebselektronik, Baugrößen und Umwelteinflüssen nicht berücksichtigt. Als weitere Einschränkung wird nur die Maschine im normalen Betriebszustand und im Fehlerfall betrachtet. Anfahr- und Stillsetzvorgänge werden außen vor gelassen, um den Umfang der folgenden Berechnungen überschaubar zu gestalten.

6.1.1 Manueller Systementwurf für einen Längstaktmontageautomat

Im Folgenden sind die Ergebnisse des manuellen Entwurfs eines Antriebssystems für einen LT-MA zusammengetragen.

Die Auslegung unterteilt sich in die für Normalbetrieb und die für das notfallmäßige Stillsetzen. Aus der geforderten Drehzahl von 90 min^{-1} der Werkzeugachse folgt eine Taktzeit T_{Takt} von

$$T_{Takt} = 0,67 \text{ s}$$

und daraus eine Winkelgeschwindigkeit ω_W von

$$\omega_W = 9,38 \frac{\text{rad}}{\text{s}}$$

für den gesamten LTMA. Das notwendige Drehmoment im Normalbetrieb entspricht dem statischen Drehmoment aus der Anforderung, da eine Bewegung mit konstanter Geschwindigkeit vorliegt. Nach empirischen Rechnungen wird die Notbremskinematik mit konstanter Beschleunigung ausgehend von der normalen Betriebsgeschwindigkeit der Werkzeugachse auf folgende Werte festgelegt (siehe auch Gleichung 5.8). Hierbei ergeben sich die Werte für die kritische Zeit $T_{Stopp,W}$ zum rechtzeitigen Stoppen der Achse, der zugehörige Bremswinkel $\varphi_{Stopp,W}$ und die dabei angesetzte maximale Verzögerung $\dot{\omega}_{Stopp,W}$:

$$T_{Stopp,W} = 0,08 \text{ s} \quad \varphi_{Stopp,W} = 0,34 \text{ rad} \quad \dot{\omega}_{Stopp,W} = -140 \frac{\text{rad}}{\text{s}^2}$$

Damit ergibt sich ein dynamisches Moment $M_{dyn,Stopp,W}$ von:

$$M_{dyn,Stopp,W} = J_W \dot{\omega}_{Stopp,W} = -23 \text{ Nm} \quad (6.2)$$

und ein betragsmäßig maximales Gesamtdrehmoment $M_{max,W}$ von:

$$\begin{aligned} M_{max,W} &= |M_{S,W}| + |M_{dyn,Stopp,W}| - M_{R,W} \\ &= 192 \text{ Nm} + 23 \text{ Nm} - 20 \text{ Nm} = 195 \text{ Nm} \end{aligned} \quad (6.3)$$

Da es sich bei der berechneten Kinematik nicht um ruckarme Bewegungen mit stetigen Beschleunigungsverläufen handelt, die aber an der Maschine schlussendlich eingestellt werden, werden das Drehmoment sowie die maximale Drehzahl mit dem Faktor 1,75 beaufschlagt. Der Faktor

wird von Applikationsingenieuren gewählt, weil er der Steigerung der maximalen Beschleunigung bei Wandlung einer gleichmäßig beschleunigten Bewegung, wie in der Auslegung verwendet, in einem sinusförmigen Bewegungsverlauf entspricht. Für das über das Getriebe gewandelte maximale Drehmoment $M_{max,Motor,W}$ und das effektive Drehmoment $M_{eff,Motor,W}$ am Motor sowie die maximale Motordrehzahl der Werkzeugachse ergeben sich:

$$\begin{aligned} M_{max,Motor,W} &= 13,64 \text{ Nm} \\ n_{max,Motor,W} &= 2250 \text{ min}^{-1} \\ M_{eff,Motor,W} &= 13,3 \text{ Nm} \end{aligned}$$

Bei der zitierten Vorlage für die Rechnung erfolgte die Auswahl des Motors aus der SMH-Reihe der Servomotoren von der Fa. Parker. Die Drehmomentkennlinien des gewählten Motors sind in Abbildung 6.1 dargestellt.

Damit lässt sich auch der Antrieb für den Vorschub gestalten. Der Vorschub befindet sich im

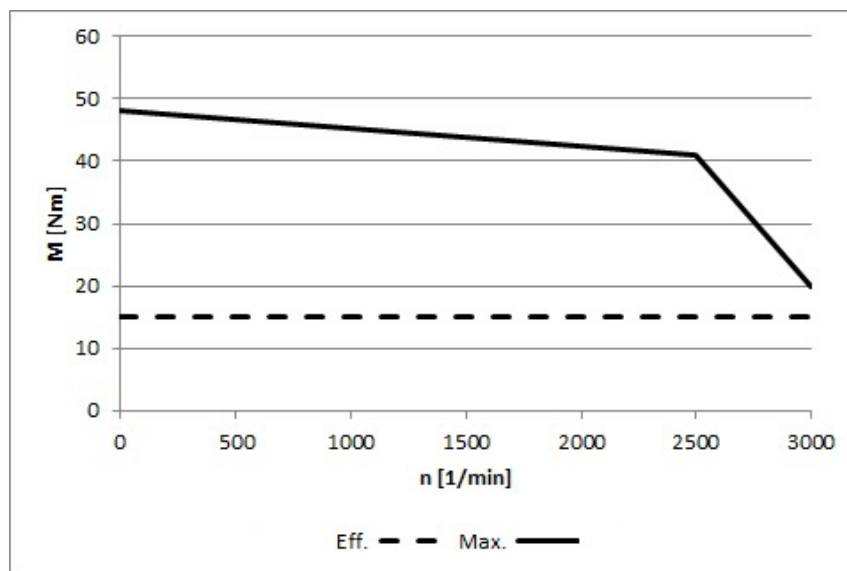


Abbildung 6.1: Kennlinie des gewählten Motors (Werkzeugantrieb) [33]

ersten und letzten Drittel des Taktes grundsätzlich in Rast. Um ein rechtzeitiges Notstoppen der Werkzeugachse zu ermöglichen, muss die Positionierung im zweiten Drittel des Taktes bereits $0,08 \text{ s}$ früher abgeschlossen sein. Es ergibt sich eine Vorschubzeit von

$$T_V = 0,15 \text{ s}$$

in der die Positionierung stattfinden muss. Die notwendige konstant angenommene Winkelbeschleunigung des Motors für den beschleunigenden Teil der Bewegung ergibt sich zu folglich zu:

$$\dot{\omega}_V = 85 \frac{\text{rad}}{\text{s}^2}$$

Hieraus folgt mit dem Massenträgheitsmoment für den Vorschub ein dynamisches Drehmoment von

$$M_{dyn,V} = J_V \dot{\omega}_V = 115 \text{ Nm} \quad (6.4)$$

Zuzüglich des Reibmoments, durch das Planetengetriebe übersetzt und mit dem Faktor für den flacheren Beschleunigungsverlauf, ergibt sich ein maximales Motormoment von

$$M_{max,Motor,V} = 8,52 \text{ Nm}$$

bei einer ebenfalls beaufschlagten Drehzahl von

$$n_{max,Motor,V} = 2724 \text{ min}^{-1}$$

und mit einem effektiven Moment von

$$M_{eff,Motor,V} = 4,6 \text{ Nm}$$

womit der Motor für den Vorschub kleiner als beim Werkzeugantrieb zu wählen ist (SMH 100). Abbildung 6.2 zeigt die zugehörige Motorkurve.

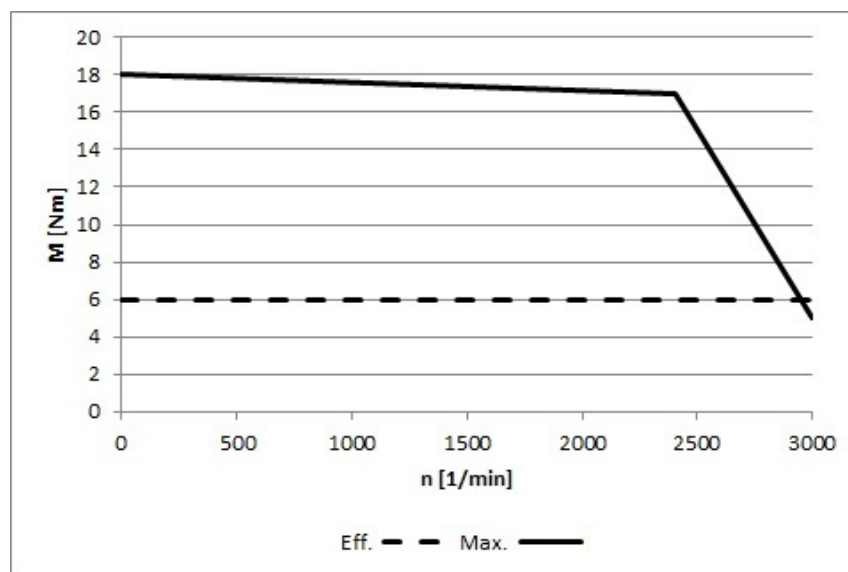


Abbildung 6.2: Kennlinie des gewählten Motors (Vorschubantrieb)

Der Entwurf der tatsächlichen Bewegung erfolgt nun quasi unabhängig von der Antriebsauslegung. Im Rahmen der Annahmen über den zeitlichen Verlauf der Bewegungsgrößen können ruckarme Bewegungen entworfen werden, weil die dynamischen Drehmomente mit dem Vergrößerungsfaktor abgeschätzt wurden. Geeignete Bewegungsgesetze hierfür sind in der zitierten VDI 2143 zu finden. Die Bewegung der Werkzeugachse ist trivial, da sie sich nur mit konstanter Geschwindigkeit bewegt. Die Bewegung des Vorschubs (siehe Abbildung 6.3) zwischen dem

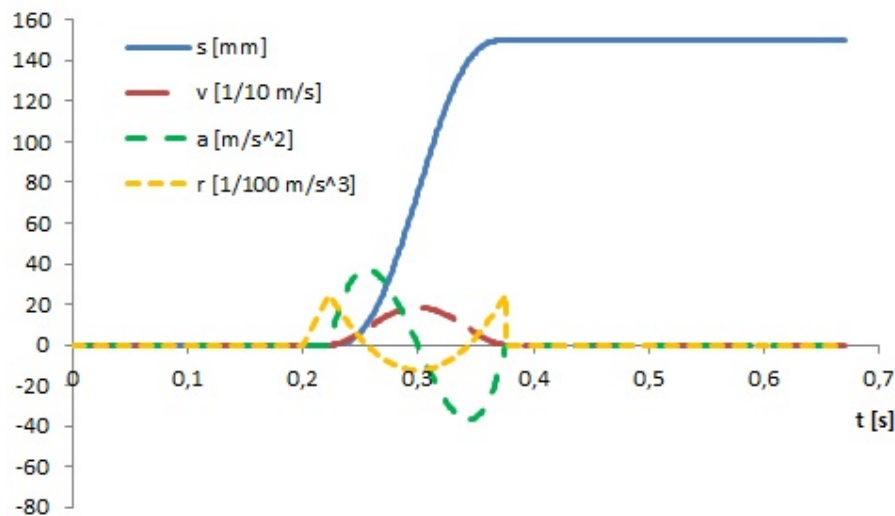


Abbildung 6.3: Bewegung für manuellen Systementwurf (Vorschub)

Rastbereichen ist mit einem Polynom 5. Ordnung interpoliert. Die Auswahl des Bewegungsgesetzes ermöglicht Nullstellen bis zur zweiten Ableitung der Strecke in beiden Randzuständen bei einer Bewegung von Rast in Rast.

Die Kennwerte des manuellen Systementwurfs sind in Tabelle 6.1 zusammengefasst.

	T_{Takt} [s]	$ r _{max}$ [$\frac{m}{s^3}$]	k_E [1]
Königswelle	0,670	0	0
Vorschub	0,670	2513	3,378

Tabelle 6.1: Kennwerte des manuellen Systementwurfs

Die manuelle klassische Methode zum Entwurf eines Antriebssystem beinhaltet zahlreiche Nachteile, die bereits in den Abschnitten 1.3 und 1.4 erläutert wurden. Der Bewegungsentwurf findet nicht integriert statt, da das ohne geeignete Algorithmen zur rechnergestützten Bearbeitung nicht umsetzbar ist. Eine vollständige Optimierung hin zur vollen Auslastung der gewählten Hardware ist nicht möglich. Zur ausreichenden Auslegung für ruckarme Bewegungen wird lediglich ein Vergrößerungsfaktor angesetzt, der auf der Beschleunigungsvergrößerung in einer sinusförmigen Bewegung im Gegensatz zu einer mit konstanter an den Randwerten nicht stetigen Beschleunigung basiert. Ebenfalls ist der Puffer für die Kollisionssicherheit nicht automatisch optimierbar. Zwar ist eine Optimierung grundsätzlich möglich und auch im Rechenbeispiel durchgeführt, allerdings beschränkt sie sich auf eine vergleichsweise kleine Anzahl empirischer Berechnungen. Im Beispiel sind etliche Iterationsschritte notwendig gewesen, die mehr als drei Stunden erforderten.

6.1.2 Entwurf für einen Längstaktmontageautomat mittels gängiger CAE-Software

Um letztlich vergleichbare Werte zu erhalten, werden der CAE-Entwurf und weitere Entwürfe mit denselben Motoren, Übersetzungen und Mechanik durchgeführt. Durch empirische Verbesserung der Bewegungsvorgänge kann die Taktzahl der Maschine unter Einhaltung aller Restriktionen auf 93 min^{-1} gegenüber dem des manuellen Entwurfs erhöht werden. Der Vorteil liegt vor allem darin, dass mittels der CAE Software ruckarme Bewegungen direkt entworfen und deren kinematischen Eigenschaften abgelesen werden können. Der Umweg über den Vergrößerungsfaktor ist nicht notwendig. Eine Optimierung hinsichtlich der Taktzeit oder eine Verknüpfung zwischen der Kollisionsvermeidung durch Nothalt der Werkzeugachse und der Vorschubbewegung existiert jedoch nicht. Der Applikationsingenieur muss aufgrund seiner Erfahrung geeignete Bewegungsgesetze auswählen und die Maschine mithilfe eines Taschenrechners empirisch verbessern.

Im Folgenden sind die Auslegungsgrößen resultierend aus der Berechnung mittels der Software [98] aufgelistet. Wesentliche Unterschiede zeigen sich im maximalen Drehmoment des Werkzeugantriebs und im effektiven Drehmoment des Werkzeugantriebsmotors im Vergleich zum manuellen Entwurf ($M_{max,W} = 195 \text{ Nm}$ und $M_{eff,Motor,W} = 13,3 \text{ Nm}$). Diese erklären sich durch einen schnelleren und genauer berechneten Not-Stopp sowie durch eine präzisere Modellierung des Lastverhaltens durch die CAE-Software.

$$\begin{aligned}
 M_{max,W} &= 350 \text{ Nm} \\
 M_{max,Motor,W} &= 14,03 \text{ Nm} \\
 n_{max,Motor,W} &= 2204 \text{ min}^{-1} \\
 M_{eff,Motor,W} &= 8,48 \text{ Nm} \\
 \\
 T_V &= 0,145 \text{ s} \\
 M_{max,Motor,V} &= 14,06 \text{ Nm} \\
 n_{max,Motor,V} &= 3027 \text{ min}^{-1} \\
 M_{eff,Motor,V} &= 4,47 \text{ Nm}
 \end{aligned}$$

Tabelle 6.2 zeigt die Kennwerte und in der Abbildung 6.4 ist der Bewegungsverlauf des Vorschubs aufgetragen.

	$T_{Takt} [s]$	$ r _{max} [\frac{m}{s^3}]$	$k_E [1]$
Königswelle	0,645	0	0
Vorschub	0,645	2952	3,761

Tabelle 6.2: Kennwerte des CAE Systementwurfs

Die Vorteile des Entwurfes mittels CAE-Software gegenüber der rein manuellen Methode liegen besonders in der Verbesserung des Entwurfsprozesses. Durch die Rechnerunterstützung gelingen

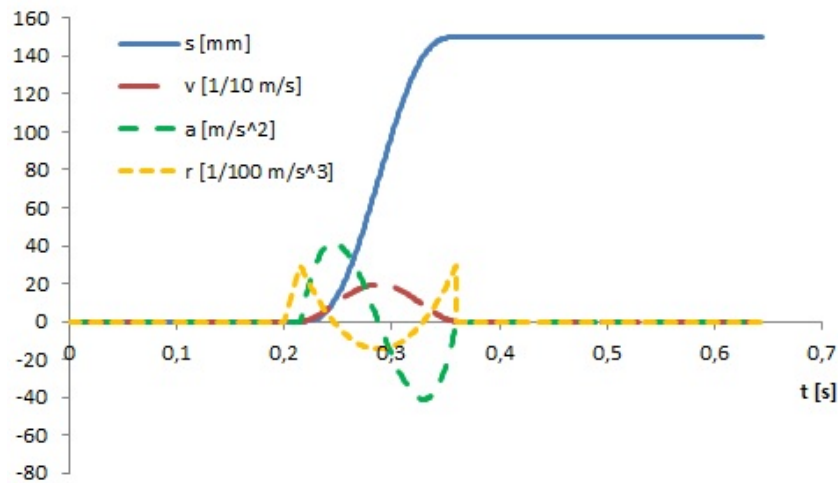


Abbildung 6.4: Bewegung für den CAE Systementwurf (Vorschub)

dem Applikationsingenieur mehr Iterationen in kürzerer Zeit. Zudem kann die Wirkung der Auswahl von Bewegungsgesetzen getestet werden. Hinzu kommen Hilfestellungen zur Auswahl von Antriebskomponenten, wie Datenbanken mit Suchfunktionen. Es stehen auch Algorithmen zur Verfügung, die einen Entwurf auf die Einhaltung von Auslegungsgrenzen bezüglich der modellierten Anforderungen untersuchen. CAE-Software für den Entwurf von Antriebssystemen bietet viele Möglichkeiten der Visualisierung von Größen und Kurvenverläufen.

Der nachfolgend untersuchte neuartige Ansatz für den Entwurf des LTMA bietet über die genannten Eigenschaften hinaus weitere, für einen qualitativ hochwertigen Entwurf essentielle, Fähigkeiten.

6.2 Neuartiger Entwurf für einen Längstaktmontageautomat

Die Untersuchung des neuartigen Entwurfes unterteilt sich in die Beschreibung der Berechnungen hinsichtlich drei unterschiedlicher Optimierungsziele und die Bewertung im Vergleich zu den bereits dargestellten Methoden.

6.2.1 Berechnung der Werte für den energiesparenden, ruckarmen und zeitoptimalen Entwurf

Als Grundlage zum hier vorgestellten Systementwurf dienen die in den Kapiteln 2 bis 5 dokumentierten neuen Vorgehensweisen und Methoden zum ganzheitlichen Engineering von Servoantriebssystemen. Die Berechnungen der im Weiteren vorgestellten Werte erfolgte mit der

ebenfalls in den jeweiligen Kapiteln beschriebenen Implementierung der neuen Algorithmen.

Formulierung der Anforderungen

Zum repräsentativen Vergleich mit den herkömmlichen Entwürfen sind die Konfiguration der Mechanik sowie die Auswahl der Antriebshardware (Motoren und Getriebe) äquivalent festgelegt.

Eine Berechnung auf Basis von den in Abschnitt 4.1 vorgestellten Methoden liefert die kinematischen Eigenschaften der Mechanik.

Für den Werkzeugantrieb werden auf Grundlage der Gleichungen (4.2) bis (4.4) bei einer Getriebeumsetzung von $i_W = 25$ und der additiven Last von $M_{stat} = \pm 192 Nm$ die folgenden Werte erhalten:

$$\begin{aligned} i_{ges,W} &= 25 \\ J_{ges,W} &= 26,208 \text{ kgmm}^2 \\ M_{stat,ges,W} &= \pm 7,68 \text{ Nm} \end{aligned}$$

Die Werte entsprechen bezogen auf die Motorachse denen aus den Referenzdaten, die am Anfang des Kapitels unter Abschnitt 6.1 aufgeführt sind. Die Referenzdaten wurden manuell berechnet und bestätigen somit die aus der Implementierung der mechanischen Modellierung stammenden Werte.

Die Auslegungsgrößen der Referenzdaten können für die Mechanik des Vorschubs mittels der Gleichungen 4.6 sowie 4.2 und 4.3 bei einer Getriebeumsetzung $i = 25$ und einem Wirkkreisdurchmesser $r_{an} = 0,1529 \text{ m}$ ebenfalls bestätigt werden:

$$\begin{aligned} i_{ges,V} &= 25 \\ J_{ges,V} &= 216 \text{ kgmm}^2 \\ M_{stat,ges,V} &= \pm 0,48 \text{ Nm} \end{aligned}$$

Durch die Übernahme der Motorkennlinien ergeben sich insgesamt zu den Referenzdaten äquivalente Größen in der Anforderung. Methodenspezifisch können weitere Anforderungen formuliert und in die programmierte Software eingepflegt werden.

Die Taktzeit vor der Optimierung wird auf $0,6 \text{ s}$ festgelegt und eine Abtastzeit von $0,03 \text{ s}$ eingestellt.

Die Werkzeugachse erhält nur einen Bewegungsabschnitt mit der Anforderung einer vollen Umdrehung pro Taktzeit mit konstanter Geschwindigkeit.

Der Vorschub erhält drei Bewegungsabschnitte (Rast - Vorschub - Rast) mit folglich 4 Stützwerten:

1. Zustand:

$s = 0; v = 0; a = 0$; Der Zeitpunkt ist absolut auf $t = 0$ festgelegt.

2. Zustand:

$s = 0; v = 0; a = 0$; Der Zeitpunkt ist restriktiv dynamisch verknüpft mit der Position der Werkzeugachse auf dem offenen Winkel von $\varphi_W = 1,047 \text{ rad}$.

3. Zustand:

$s = 150 \text{ mm}; v = 0; a = 0$; Der Zeitpunkt ist restriktiv dynamisch verknüpft mit der Werkzeugachse, sodass die Kollisionsfreiheit mit der kritischen Position $\varphi_{K,W} = 2,094 \text{ rad}$ bedingt ist.

4. Zustand:

$s = 150 \text{ mm}; v = 0; a = 0$; Der Zeitpunkt entspricht der Taktzeit.

Die ultimativen kinematischen Restriktionen aller Abschnitte entsprechen denen aus der Hardware, wodurch Begrenzungen für Beschleunigung und Drehzahl beziehungsweise Geschwindigkeit resultieren. Bezogen auf den Endeffektor, also die Rotation der Werkzeugachse und die Translation des Vorschubs, errechnet das neue Programm zum ganzheitlichen Systementwurf folgende Restriktionen für Werkzeugachse $K_{ultimo,W}$ und Vorschub $K_{ultimo,V}$, bei denen sämtliche Bewegungsabschnitte identisch sind:

$$K_{ultimo,W} = \begin{pmatrix} \# & \# \\ \# & 12,57 \frac{\text{rad}}{\text{s}} \\ -245 \frac{\text{rad}}{\text{s}^2} & 245 \frac{\text{rad}}{\text{s}^2} \\ \# & \# \end{pmatrix} \quad K_{ultimo,V} = \begin{pmatrix} \# & \# \\ -1,92 \frac{\text{m}}{\text{s}} & 1,92 \frac{\text{m}}{\text{s}} \\ -45,3 \frac{\text{m}}{\text{s}^2} & 45,3 \frac{\text{m}}{\text{s}^2} \\ \# & \# \end{pmatrix}$$

Die Berechnung erfolgt nach Gleichung (5.6) und die Platzhalter (#) geben an, dass keine Restriktion für die jeweilige Größe vorgegeben ist.

Abbildung 6.5 zeigt die Bewegungen von Werkzeugachse und Vorschub des Initialsystems (siehe Abschnitt 5.2.1).

Ausgehend von der Formulierung der Anforderungen wird das System hinsichtlich der Kriterien Energieminimierung, Ruckminimierung und Zeitminimierung eindeutig optimiert. Die gesamte Anwendung lässt sich insbesondere über die Anpassung der Vorschubbewegung optimieren, die wiederum über die Kollisionsvermeidung von der Rotationsgeschwindigkeit der Werkzeugachse abhängt.

Energiesparender Entwurf

Die Auswahl einer energieoptimalen Lösung erfordert bei der Anpassung eine Funktionenschar, aus der eine geeignete Bewegung ausgewählt werden kann. Die entwickelte Strategie ist im Geschwindigkeits-Zeit-Diagramm aus Abbildung 6.6 erkennbar.

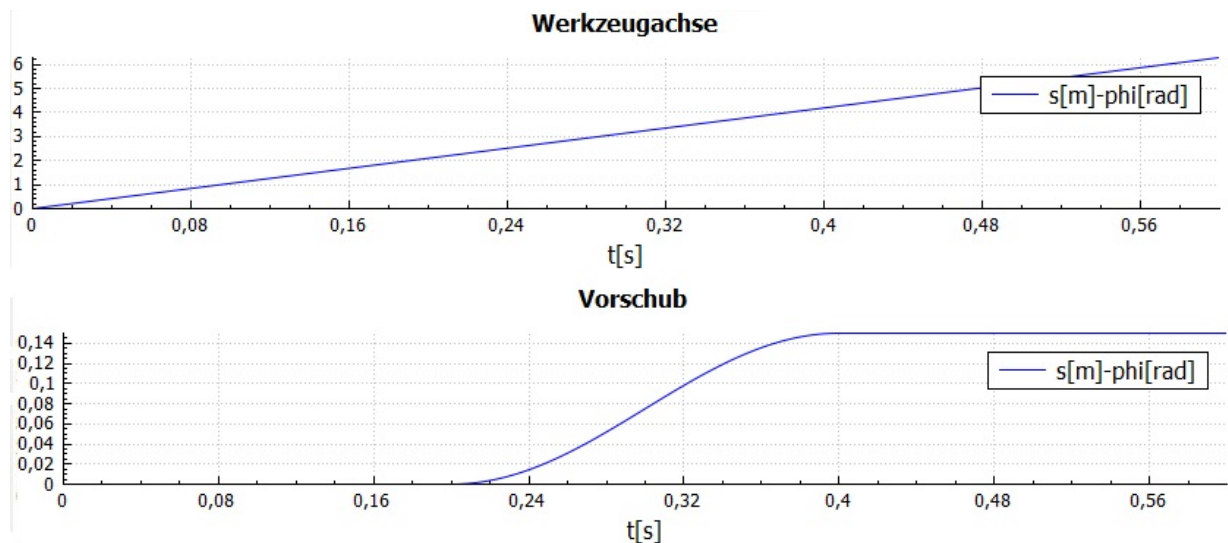


Abbildung 6.5: Bewegungen des Initialsystems (oben: $\varphi[\text{rad}]$; unten: $s[\text{m}]$)

Die Taktzeit ist nicht weiter reduziert, um die Mannigfaltigkeit zu erhalten. Das Diagramm zeigt den Bereich, in dem der Vorschub stattfindet. Wie bereits erläutert, enthält der Bewegungsabschnitt die wesentlichen optimierbaren Anteile am Antriebssystem des LTMA. Aufgetragen ist die Funktionenschar aller geeigneter Bewegungsgesetze sowie die kinematischen Restriktionen (horizontale gestrichelte Linien) soweit aktiviert. Die Abbildung unterteilt sich in Diagramme für den zeitlichen Verlauf von Weg, Geschwindigkeit, Beschleunigung und Ruck. Die Unterschiede im Verlauf der verfügbaren Bewegungen sind besonders in den Ableitungen zu erkennen. Die einschränkende kinematische Restriktion stellt die maximale Beschleunigung dar.

In den Kurvenverläufen aus Abbildung 6.6 ist entsprechend der für energieoptimale Bewegungen gültigen Verlustfunktion (siehe Gleichung (3.24)) der Verlauf von Geschwindigkeit und Beschleunigung zu begutachten. Den geringsten Verlust weist der Modifizierte Sinus ($m\sin$) auf. Im Geschwindigkeitsdiagramm ist erkennbar, dass die zugehörige grüne Linie durchschnittlich unter den anderen verläuft. Noch markanter ist der Verlauf im Beschleunigungsdiagramm. Die selektierte Bewegung steigt zwar mit der Beschleunigung schnell an, weist dann jedoch über einen weiten Zeitbereich kleinere Werte auf als die anderen Bewegungsformen.

Die entwickelten Algorithmen selektieren demnach die korrekte Bewegung.

In Tabelle 6.3 sind die Kennwerte für die gesamte Bewegung von Werkzeugachse und Vorschub festgehalten.

	$T_{Takt} [s]$	$ r _{max} [\frac{m}{s^3}]$	$k_E [1]$
Königswelle	0,737	0	0
Vorschub	0,737	2689	2,826

Tabelle 6.3: Kennwerte des ganzheitlichen energieoptimierten Entwurfs

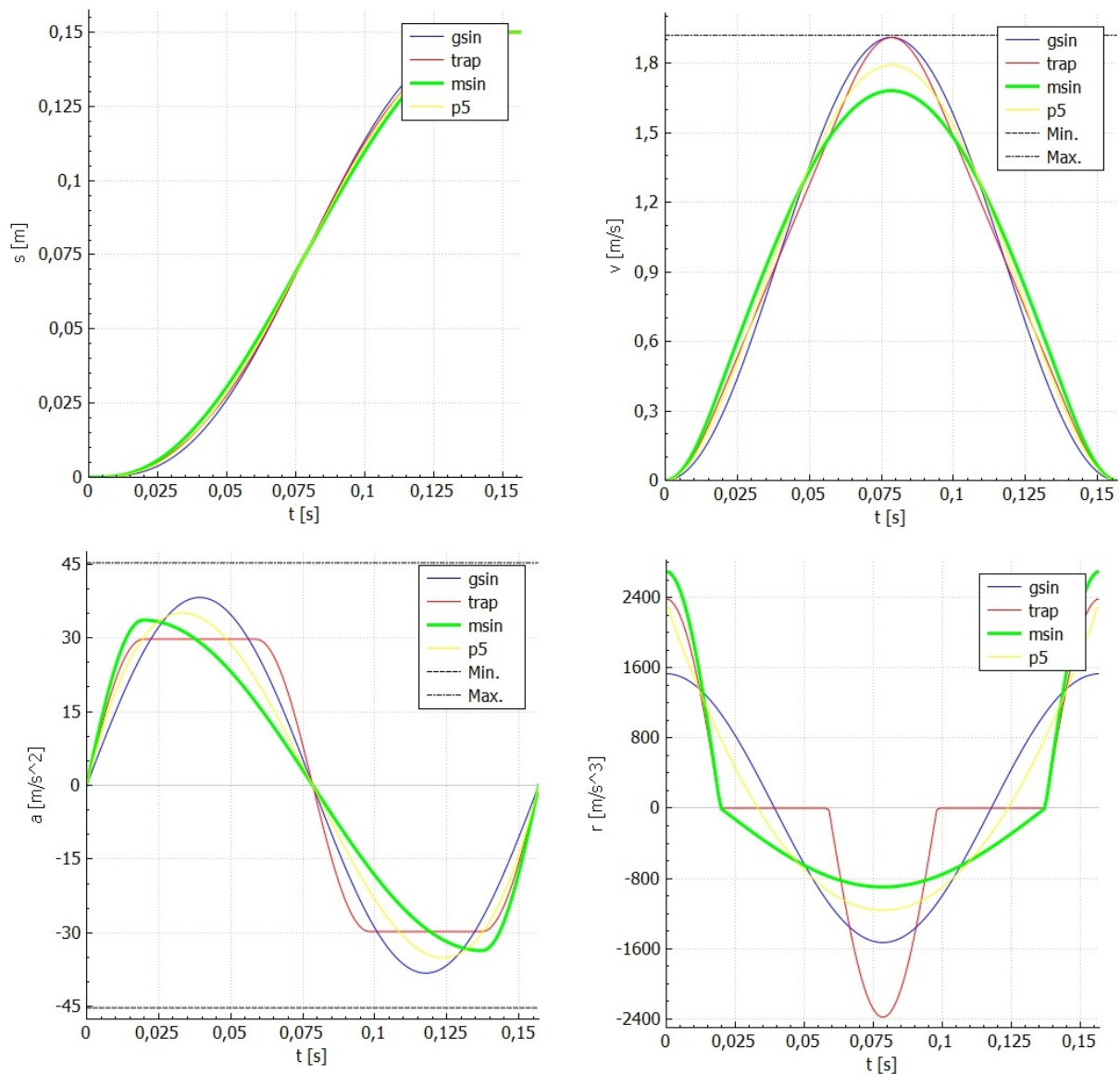


Abbildung 6.6: Energieoptimaler Vorschub

Ruckarmer Entwurf

Abbildung 6.7 zeigt die Diagramme für den ruckarmen Verlauf. Ebenso wie für den energiesparenden Verlauf wird eine Mannigfaltigkeit von Bewegungsgesetzen bei der zeitlichen Anpassung der Teilbewegung berücksichtigt. Als ruckärmste Bewegung ist die nach der Geneigten Sinuslinie (gsin) geformten ausgewählt. Die blaue hervorgehobene Bewegung gibt im Ruckdiagramm leicht Aufschluss über die Richtigkeit der Auswahl. Die verfügbaren Bewegungen sind die glei-

chen wie für den energiesparenden Entwurf, jedoch wird die Bewegung mit dem kleinsten maximal auftretenden Ruck ausgewählt. Im Geschwindigkeits- und Beschleunigungsdiagramm ist zu erkennen, dass die selektierte Geneigte Sinuslinie die Bewegung mit den jeweils größten Beträgen darstellt. Das ist dadurch zu erklären, dass eine vergleichsweise moderate Änderung der Beschleunigung mit höheren Maximalwerten ausgeglichen werden muss, um die Bewegung in der gleichen Zeit vollziehen zu können.

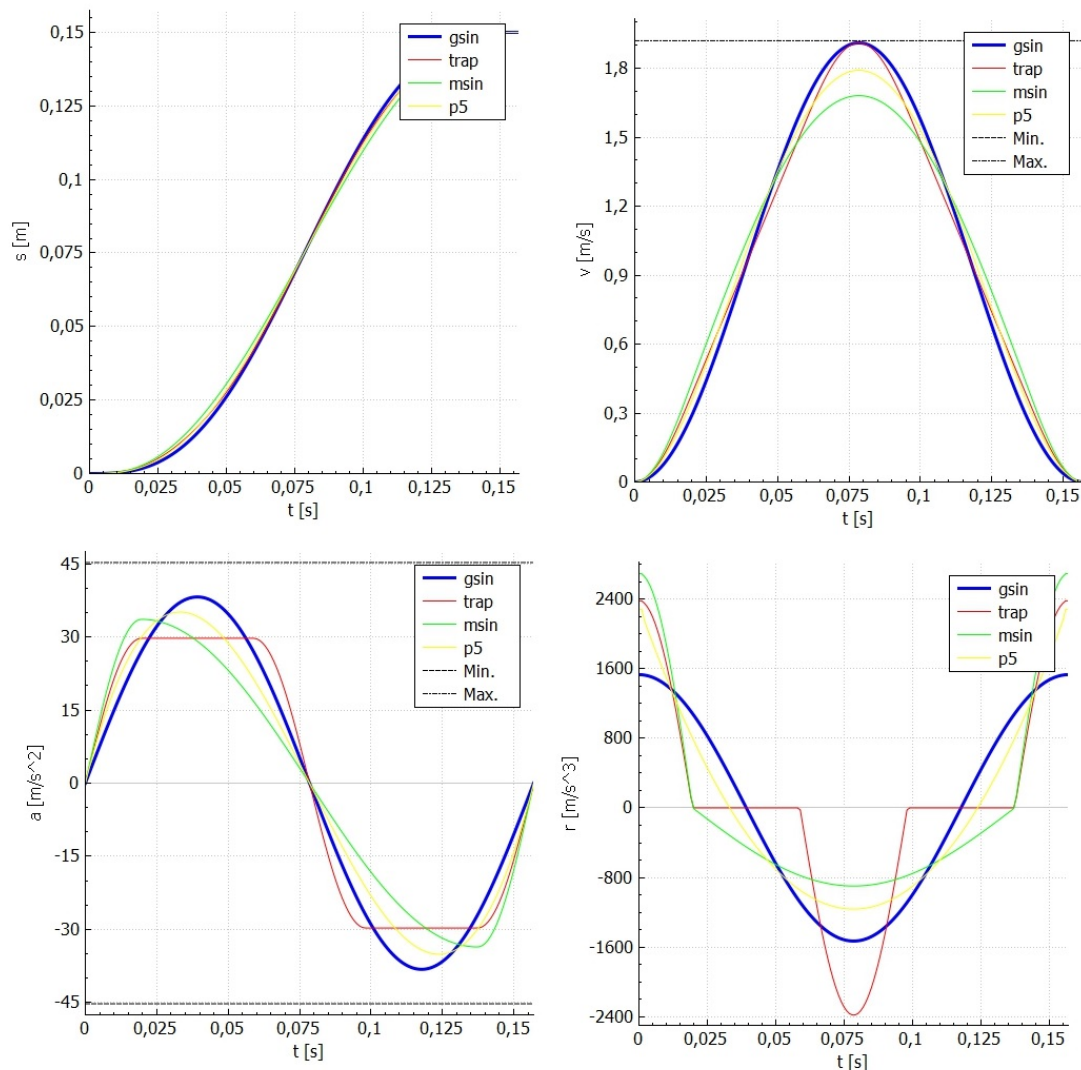


Abbildung 6.7: Ruckoptimaler Vorschub

Tabelle 6.4 enthält die Kennwerte des LTMA nach Ruckoptimierung. Ein Vergleich der Methoden und Optimierungskriterien erfolgt im anschließenden Abschnitt 6.2.2.

	T_{Takt} [s]	$ r _{max}$ [$\frac{m}{s^3}$]	k_E [1]
Königswelle	0,737	0	0
Vorschub	0,737	1530	3,650

Tabelle 6.4: Kennwerte des ganzheitlichen ruckoptimierten Entwurfs

Zeitoptimaler Entwurf

Der zeitoptimale Entwurf ist nicht auf eine Mannigfaltigkeit angewiesen. Die Zeit einer Teilbewegung kann so lange minimiert werden, bis nach kinematischer Selektion nur eine Bewegung übrig bleibt. Abbildung 6.8 zeigt die Vorschubbewegung mit zeitoptimaler Ausprägung, in der das Bewegungsgesetz Einfacher Sinus (esin) selektiert ist. Die eingrenzende Restriktion sind im zeitoptimalen Fall die minimale und maximale Beschleunigung, dargestellt durch die gestrichelten horizontalen Linien im Beschleunigungsdiagramm. Die Bewegung des Modifizierten Trapezes (trap) liegt im Beschleunigungsdiagramm noch im Rahmen der Restriktionen scheidet jedoch durch zu hohe Geschwindigkeit aus, wie das Geschwindigkeitsdiagramm zeigt.

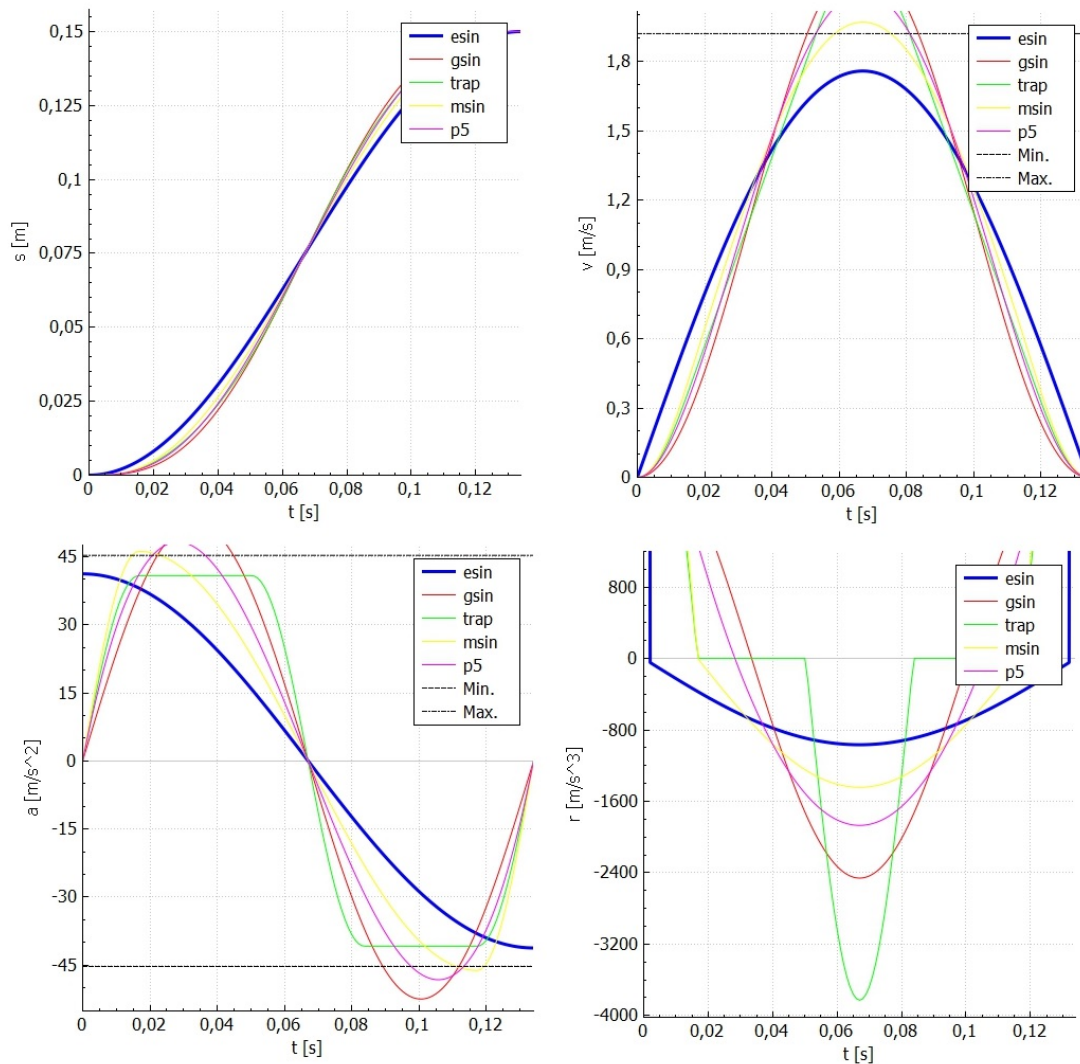


Abbildung 6.8: Zeitoptimaler Vorschub

Die Kennwerte in Tabelle 6.5 zeigen die minimale Taktzeit auf Kosten eines sehr großen maximalen Rucks (ungefähr das zwanzigfache im Vergleich zu den beiden anderen Optimierungen), der sich durch einen abgeschnittenen Impuls im Ruckdiagramm äußert. Dieses Verhalten resultiert aus dem nicht stetigen Verlauf des einfachen Sinus im Geschwindigkeitsdiagramm, er springt zum Beginn und zum Ende der dargestellten Bewegung.

	$T_{Takt} [s]$	$ r _{max} [\frac{m}{s^3}]$	$k_E [1]$
Königswelle	0,501	0	0
Vorschub	0,501	41212	3,091

Tabelle 6.5: Kennwerte des ganzheitlichen zeitoptimierten Entwurfs

6.2.2 Vergleich der erstellten Entwürfe

Im Folgenden werden alle beschriebenen Entwürfe für den Längstaktmontageautomat miteinander verglichen und die Vorteile der neuartigen Vorgehensweise unter Beweis gestellt.

In Tabelle 6.6 sind die Ergebnisse der klassischen Entwurfsmethoden, dem manuellen und dem CAE-Entwurf sowie die Ergebnisse der neuen Vorgehensweise für die drei untersuchten Optimierungsvorgaben für den Vorschub gegenübergestellt. Die in der Tabelle ergänzten prozentualen relativen Werte beziehen sich jeweils auf den maximal auftretenden Wert unter allen Berechnungen. In Abbildung 6.9 werden die prozentualen Anteile visuell veranschaulicht.

	T_{Takt} [s]	vom Max.	$ r _{max}$ [$\frac{m}{s^3}$]	vom Max.	k_E [1]	vom Max.
Manueller Entwurf	0,67	90,9%	2.513	6,1%	3,38	50,8%
CAE	0,66	87,5%	2.952	7,2%	3,76	56,6%
Energieoptimal	0,74	100,0%	2.689	6,5%	2,83	42,5%
Ruckoptimal	0,74	100,0%	1.529	3,7%	6,65	100,0%
Zeitoptimal	0,50	68,0%	41.212	100,0%	3,10	46,5%

Tabelle 6.6: Vergleich der Entwurfsmethoden und Optimierungskriterien

Aus dem Vergleich der Daten wird ersichtlich, dass eine gezielte Optimierung eines Antriebssystems mit der neuartigen Vorgehensweise umgesetzt wurde. Die hinsichtlich Energieverbrauch, Ruck und Zeit optimierten Entwürfe liefern jeweils die besten relevanten Werte im Vergleich zu allen verwendeten Methoden. So wird die kleinste Taktzeit deutlich beim zeitoptimalen Entwurf erreicht, was aber den Ruck mehr ungefähr verzwanzigfacht. Der ruckoptimale Entwurf zeigt seinen markanten Vorteil beim maximal auftretendem Ruck, erfordert jedoch den größten Energie und Zeitaufwand. Die energiesparendste Bewegung wird ebenfalls durch den entsprechenden Entwurf nach dem neu entwickelten Konzept erzielt.

Neben den durch Kennwerte manifestierten Vorteilen der neuen Methode besitzt sie weitere entscheidenden Verbesserungen. So liegt im manuellen Entwurf ein großes Maß an Unsicherheit durch die pauschale Beaufschlagung der Auslegungsgrößen für den erst in der Inbetriebnahme implementierten ruckarmen Bewegungsentwurf. Das kann sowohl eine Überdimensionierung als auch im schlimmsten Fall eine Unterdimensionierung zur Folge haben. In jedem Fall werden jedoch die Potenziale hinsichtlich optimalem Bewegungsentwurf nicht erschöpft. Eine multikriterielle Optimierung oder gar der Vergleich von diversen Anforderungen ist mit dem notwendigen Entwicklungsaufwand nicht vereinbar. Der Einbezug von Zusatzbedingungen, wie Kollisionsfreiheit ist ebenfalls nur überschlägig umsetzbar.

Beim Entwurf mittels CAE-Software wird dem Entwickler vieles erleichtert, eine Möglichkeit der Modellierung von Mechanik und Lasten gehört dazu. Auch Bestandteile zum Entwurf von Bewegungen und deren Bezug zur Auslegung der Motoren und Antriebsregler sind enthalten. Durch den Einsatz der CAE-Software wird der iterative Entwicklungsprozess deutlich beschleunigt und das Rechnen mit verschiedenen Bewegungsgesetzen ermöglicht einen realistischeren Entwurf. Jedoch existieren praktisch keine Lösungen, um innere Zusammenhänge der Achsen

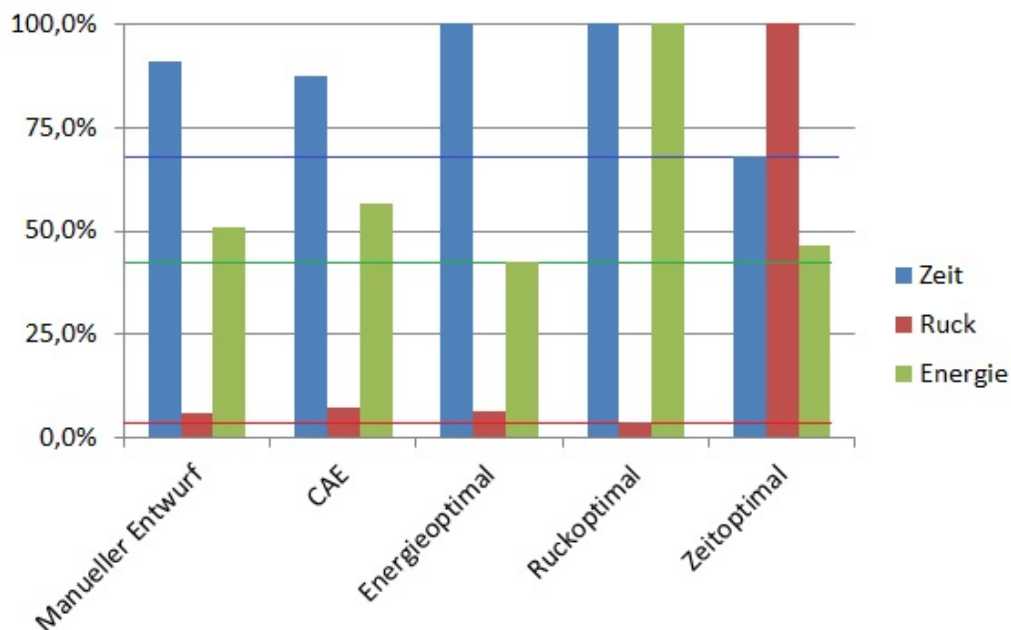


Abbildung 6.9: Vergleich der Methoden

untereinander abzubilden, automatisch Bewegungsgesetze auszuwählen oder eine gesamte Maschine zu optimieren. Spezielle Lösungen für die in der vorliegenden Arbeit behandelten Maschinen mit Leitachse und periodischem Verhalten sind ebenfalls nicht umgesetzt.

Der Entwurf mittels der Algorithmen nach der neuartigen Methode ermöglicht nicht nur die dargestellten optimierten Systementwürfe, sondern führt auch zu einer deutlichen Verkürzung der Entwurfszeit. Nach der Definition beziehungsweise Eingabe der relevanten Zusammenhänge wird ein System automatisch entworfen. Die numerische Anpassung bringt den Entwurf insgesamt an die kinematischen Grenzen, und zwar unter automatischer Berücksichtigung aller Bedingungen, wie die Optimierungskriterien und Kollisionsvermeidung. Hierbei werden Bremspunkte numerisch exakt bestimmt und in jedem Iterationsschritt automatisch angepasst statt nur abgeschätzt, siehe Gleichungen (5.8) und (5.9).

Damit ergibt sich ein nennenswerter Mehrwert der neuen Methode nicht nur aus der höheren Qualität des entwickelten Antriebssystems, sondern auch aus der Vereinfachung und Optimierung des Entwicklungsprozesses. Es wird eine höhere Effizienz beim Engineering erreicht. Wo beim manuellen und sogar beim CAE-Entwurf zeitaufwendige Iterationen vom Ingenieur durchgearbeitet werden müssen, um zu einem vergleichsweise suboptimalen Ergebnis zu gelangen, liefern die Algorithmen der neuen Vorgehensweise innerhalb von Minuten optimale Ergebnisse. Im untersuchten, vergleichsweise kleinen Beispiel dauert die Optimierungsrechnung weniger als zehn Sekunden. Die gesamte Entwicklungszeit, die der Applikationsingenieur aufbringen muss, liegt ebenfalls deutlich unter der des manuellen und auch CAE-Entwurfes. Für das genannte Beispiel wird mittels neuartiger Software für die Eingabe der Daten und Anforderungen sowie die Ausarbeitung weniger als eine halbe Stunde benötigt.

Anmerkung:

Die aufgeführten Berechnungen dienen einer Einschätzung der Qualität der Algorithmen und der Bestimmung von Größen zum Vergleich von Entwurfsmethoden. Zur realen Auslegung des LT-MA sind weitere Anpassungen besonders in den herkömmlichen Berechnungsverfahren, auf die sich die neuartigen Berechnungen beziehen, notwendig. So müssen beispielsweise die tatsächlichen Massenträgheitsmomente der schließlich eingesetzten Motoren mit einbezogen werden. Auf die Beschreibung der vollständigen Auslegung ist verzichtet, da sie im vorliegenden Kontext unerheblich ist.

Im folgenden Abschnitt werden die erstellten Algorithmen im Allgemeinen bewertet.

6.3 Bewertung der entwickelten Algorithmen und deren Implementierung

Im Folgenden wird Bezug auf die in Abschnitt 2.1 gestellten Anforderungen genommen. Die Erfüllung der Anforderungen ist anhand der entwickelten Methoden evident oder kann exemplarisch an der Implementierung der entwickelten Algorithmen erkannt werden.

Abbildung 6.10 stellt einen Überblick der im Rahmen der Arbeit entwickelten Software dar. In dem Diagramm sind die Gleichungen notiert, die in den jeweiligen Softwaremodulen programmiert sind. Zudem wird auf die Abbildung verwiesen, welche den Aufbau der zugehörigen Vorgehensweise darstellt. Die wichtigsten Klassen, das neuartige Konzept betreffenden, sind ebenfalls notiert. Hinzu kommen zahlreiche Klassen und Methoden zur Visualisierung, Datenverarbeitung für Benutzerein- und ausgaben oder Datenexporte sowie Kontrollmechanismen und Umsetzungen grundlegender mathematischer Operationen. Der genannte Ausbau der Software ist nicht im Diagramm erhalten, weil darauf nicht weiter eingegangen wird. Die Implementierung ist wie das Konzept in drei wesentliche Teile untergliedert.

Softwareentwicklung zum ganzheitlichen Engineering von elektrischen Servoantriebssystemen

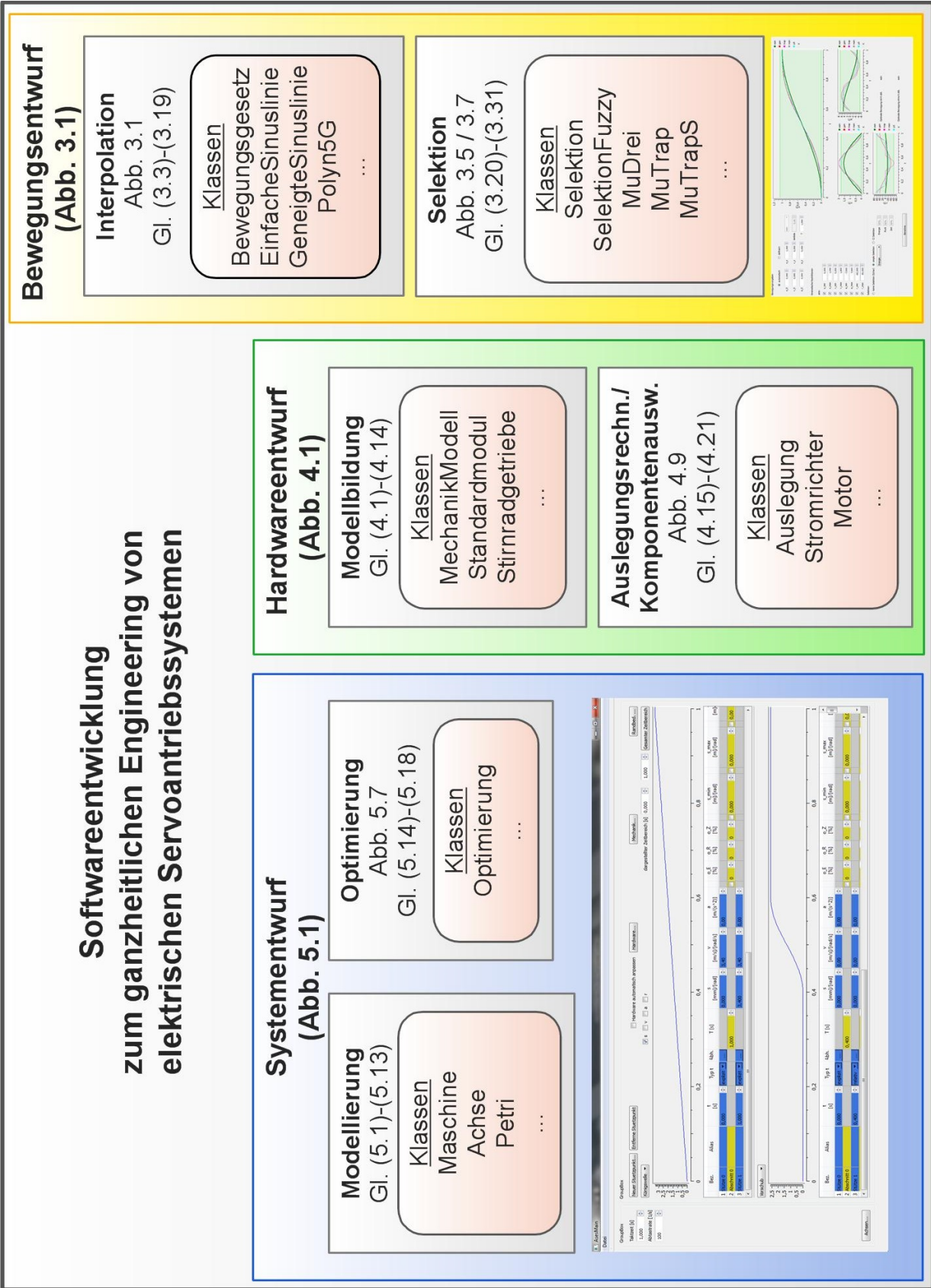


Abbildung 6.10: Softwareimplementierung

Das Hauptprogramm stützt sich auf den ersten Teil, den Systementwurf, dessen Implementierung in Abschnitt 5.3 erläutert wird. Die Implementierung besteht aus zwei Modulen, der Modellierung und der Optimierung. Zudem ist die Benutzeroberfläche des Hauptfensters dargestellt, siehe Abbildung 5.12 für Details.

In dem zweiten Programmteil, zum Hardwareentwurf, sind die Module Modellbildung und Komponentenauswahl mit Auslegungsrechnung enthalten, deren Implementierung in den Abschnitten 4.4.1 und 4.4.2 zu finden ist.

Der dritte Programmteil, der für die Umsetzbarkeit des Konzepts wichtigste, ist der Bewegungsentwurf. Die Beschreibung der Implementierung ist in Abschnitt 3.3 zu finden. Auch der Bewegungsentwurf ist in zwei Modulen entwickelt. Er enthält die Algorithmen zur Berechnung von Bewegungen, bzw. Scharen von Bewegungen (siehe auch Abschnitt 3.1) und ein Modul zur Selektion (Abschnitt 3.2).

In Orientierung an die formulierten Anforderungen enthält auch die Bewertung folgende Punkte:

- Allgemeine Anforderungen
- Anforderungen an den Bewegungsentwurf
- Anforderungen an den Hardwareentwurf
- Anforderungen an den Systementwurf
- Anforderungen an die Algorithmen

Um der allgemeinen Anwendbarkeit von Servoantriebssystemen auch in deren Entwicklung gerecht zu werden, sind die Algorithmen so gestaltet, dass sie eine variable Struktur zulassen. Die Allgemeinheit des Konzepts verdeutlicht sich in der freien Wahl von der Achsenzahl und deren Zusammensetzung von Bewegungsverläufen und deren kinematischen Eigenschaften, Restriktionen und Optimierungsvorgaben (siehe Gleichungen 5.1 bis 5.5).

Durch die Berücksichtigung bestehender Normen, wie der Integration der Bewegungsgesetze nach VDI 2143, gelingt es auf etablierte Methoden und Erfahrungen auch im Rahmen der neuen Methoden zurückzugreifen. Die erwähnten Bewegungsgesetze sind Hauptbestandteil der in Gleichung (3.3) definierten Menge, aus der im Bewegungsentwurf Funktionen zur Interpolation entnommen werden. Mit der automatischen Interpolation und auch Optimierung des Bewegungsentwurfs, wie er in den Abschnitten 3.1 und 3.2 dokumentiert ist, werden weitere Anforderungen an das neuartige Konzept befriedigt. Das automatische Berechnen von Bewegungen anhand minimaler gegebener Größen vereinfacht das Engineering des Antriebssystems und macht die Optimierung der gesamten Anwendung erst möglich. Auch die obligatorische Bedingung physikalisch und regelungstechnisch umsetzbar Bewegungen zu entwerfen, also den

in Abschnitt 1.2.3 beschriebenen kinematischen Gesetzen gerecht zu werden, ist erfüllt. Die Bewegungsgesetze sind entsprechend der relevanten Eignungen klassifiziert und der Endzustand einer Teilbewegung ist als Anfangszustand der folgenden Teilbewegung definiert, wodurch praktisch nicht realisierbare Vorgaben, wie Orts- oder Geschwindigkeitssprünge, analytisch ausgeschlossen sind.

In Abschnitt 4.1 ist die Notation der Eigenschaften von Mechanismen festgelegt. Durch die modulare und standardisierte Formulierung kann ein breites Spektrum an Mechanik beschrieben werden, die von Servoantrieben bewegt wird. Durch die anwendungsbezogene Aufstellung ist eine hinreichende Modellbildung zur Auslegungsrechnung und damit Auswahl von Komponenten gegeben. Auch die Vielfalt einsetzbarer Motoren und anderer Hardware ist garantiert, da sich die Algorithmen auf den minimalen Bedarf von bekannten Eigenschaften und Kennwerten beschränkt (siehe Gleichungen 4.15 und 4.16). Die universellen Größen sind unabhängig von Herstellern und Lieferanten verfügbar.

Die Definition eines diskret-kontinuierlichen Petrinetzes (siehe Abschnitt 5.1) liefert die erforderliche mathematische Beschreibung der Anforderungen an das Antriebssystem mit inneren Zusammenhängen und Kollisionsschutz. Wie in Tabelle 5.1 zusammengefasst, sind Zusammenhänge zwischen den Achsen ebenfalls klassifiziert, um zu konditionierten Anforderungsbeschreibungen zu gelangen. Auf Basis des abgeleiteten Initialsystems (siehe Abschnitt 5.2.1) kann dann auch die wichtige Anforderung nach zielgerichteter Optimierung erfüllt werden. Die Eignung der Optimierungsalgorithmen für das mittels Petrinetzen beschriebene Antriebssystem ist im Abschnitt 6.2 dokumentiert.

Für die Anforderungen an die Algorithmen im Allgemeinen gilt es unter anderem den Berechnungsaufwand im Rahmen zu halten. Mit dem in den Anforderungen genannten beispielhaften PC (Windows 7 64-bit; 4 GB RAM) liegen die Berechnungszeiten, für die Optimierung des Minimalbeispiels LTMA je nach Startwerten und Kriterien, immer im einstelligen Sekundenbereich. Zudem besteht die Möglichkeit durch Anpassen der Abtastzeit im Entwicklungsprozess die Rechenzeiten markant zu senken. Für die Berechnung der präzisen Steuerungsgrößen der Maschinen sind Entwicklungszeiten von mehreren Stunden hinnehmbar, da dem ein großer Gewinn an Prozessqualität gegenüber steht. Durch die Implementierung in einer Hochsprache und der plattformmultiplen Bibliothek Qt [125, 126] ist das entwickelte Programm auch auf anderen Plattformen und damit in diverse Entwicklungs- und CAE-Umgebungen integrierbar. Auch die Forderung nach der Erweiterbarkeit der implementierten Algorithmen ist umgesetzt. So ist zum Beispiel ein Standard festgelegt, nach dem Bewegungsgesetze integriert werden können. Über die in Abbildung 6.11 ausschnittsweise dargestellte virtuelle Klasse ist klar definiert, über welche Methoden die Implementierung eines neuen Gesetzes verfügen muss. Damit werden kompatible, gezielt anwendbare Berechnungsergebnisse für weitere Programmteile verfügbar. Wie die Abbildung ebenfalls verdeutlicht, ist die Hinzunahme neuer Bewegungsgesetze in die

Algorithmen durch die definierten Standards schließlich einfach zu implementieren.

Die Genauigkeit der Algorithmen zur Berechnung von Zeitwerten ist über die Vorgabe der

```

class Bewegungsgesetz
{
public:
    Bewegungsgesetz();
    QString gesetz_bez;
    MMatrix<int> xD_verf;
    int check_xD(QVector<int>* xD);
    void calc_BN(MMatrix<double>* BN, QVector<d

protected:
    virtual double Formelf(double z, double kED|
    virtual double Formelfp1(double z, double ki
    virtual double Formelfp2(double z, double ki
    virtual double Formelfp3(double z, double ki

bewegungsgesetze << new EinfacheSinuslinie();
bewegungsgesetze << new GeneigteSinuslinie();
bewegungsgesetze << new HarmonischeKomb();
bewegungsgesetze << new MBeschleunigungstrapez();
bewegungsgesetze << new MSinuslinie();
bewegungsgesetze << new Polyn5G();
bewegungsgesetze << new SingKomb();
bewegungsgesetze << new Rast();
bewegungsgesetze << new KonstGeschw();

```

Abbildung 6.11: Code-Ausschnitt (virtuelle Methoden, integrierte Bewegungsgesetze)

Abtastzeit definierbar, zudem stellt die Verwendung von 64-bit Datentypen eine Genauigkeit über der der meisten Antriebsregler bereit. Zudem sind die Algorithmen zur Interpolation und Optimierung von Bewegungen so aufgebaut, dass sie explizit Werte zu nach der Abtastung vorhandenen Zeitpunkten berechnen. Es existiert dazu ein übergeordneter Zeitvektor für den gesamten Maschinentakt.

Durch die Implementierung einer fortschrittlichen Model/View-Architektur [126] ist es möglich, flexibel Benutzerschnittstellen zu implementieren, ohne Änderungen an den eigentlichen Berechnungsalgorithmen vorzunehmen und dennoch ein kompaktes Programm ohne komplexe Modulschnittstellen zu entwerfen. Damit ist der Weg für benutzerfreundliche Anwendungen bereit.

6.4 Analyse und Bewertung der Methoden mittels Modellmaschine

Im Folgenden wird die Modellmaschine vorgestellt, die zur Verifikation und experimentellen Analyse der aufgestellten Methoden und Algorithmen entworfen ist. Anschließend werden die Ergebnisse aus Versuchen mit Bezug zum LTMA dargestellt und bewertet.

6.4.1 Aufbau der Modellmaschine

Bei der Modellmaschine handelt es sich um ein Portalsystem mit vier voneinander unabhängig ansteuerbaren Bewegungsachsen. Abbildung 6.12 zeigt eine Konstruktionszeichnung und Fotos der Maschine mit Linearachsen, von denen drei in der gleichen Raumdimension (y -Richtung) parallel angeordnet von der vierten senkrecht dazu (x -Richtung) getragen werden. Zur Verdeutlichung des mechanischen Aufbaus folgt ein zusätzlicher Absatz. Die Modellmaschine ist konzipiert, um strukturell unterschiedliche Anforderungen an ein Antriebssystem abbilden zu können. Dazu können Strukturen variabel gesteckt werden, um unterschiedliche räumliche Anforderungen an ein Antriebssystem nachstellen zu können. Abbildung 6.13 zeigt, aus zwei Kamerabildern zusammengesetzt, die entsprechende Struktur für den LTMA. Dabei sind in grün die Vorgaben für die Werkzeugachse, also der Lauf mit konstanter Geschwindigkeit, und in den Farben blau der Vorschub sowie in rot die Fixierung¹ modelliert. Wenn die vorgegebenen Strukturen nun mit jeweilig gefärbten Laserpointern synchron abzufahren sind, dann ergibt sich für die Modellmaschine eine analoge Aufgabe zu der des realen LTMA. Die Pointer bewegen sich stets auf gleicher Strecke in x -Richtung (auf dem Photo von links nach rechts) innerhalb der jeweiligen oberen und unteren Markierung. Bei fehlender Begrenzung sind die Bewegungen frei interpolierbar, hierin liegt das Optimierungspotenzial. Eine konstante x -Bewegung stellt hierbei den Zeitverlauf bzw. das synchronisierende Mastersignal nach.

Konstruktion

Die Konstruktion besteht aus einem quaderförmigen Gestell, in dem ein fahrbarer Schlitten aufgehängt ist. Der Schlitten wird über eine Kugelumlaufspindel (Steigung $10mm$) bewegt, die von einem Schrittmotor (Sanyo 103-H7126-6640 / 1, $65Nm$ Haltemoment [127]) angetrieben wird.

¹Im vorausgegangenen Vergleich der Entwurfsmethoden wurde die Fixierung wegen des höheren Betrachtungsaufwandes im manuellen Entwurf nicht berücksichtigt.



Abbildung 6.12: Mechanischer Aufbau der Modellmaschine

Die Achse ist für Translationen des Schlittens von mindestens $20 \frac{cm}{s}$ und Beschleunigung von bis zu $35 \frac{cm}{s^2}$ ausgelegt.

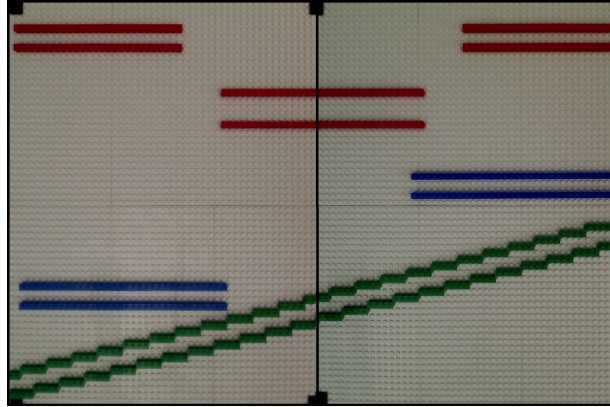


Abbildung 6.13: Anforderungen des Längstaktmontageautomats in der Modellumgebung

Die erreichbare mechanische Genauigkeit einer Positionierung liegt unter $500 \mu m$ und damit weit über der Präzision des Modells mit einer Auflösung von einem Element pro $8 mm$.

Die Riemen, welche die Laserpointer tragen, werden über Scheiben mit $34 mm$ Durchmesser mittels Gleichstromservomotoren (EMG30 / $0,5 Nm$ Nenndrehmoment [128]) mit zwischengeschaltetem Getriebe (Untersetzung 30 : 1) angetrieben.

Abbildung 6.14 zeigt den Schlitten ohne Riemen im Detail. Die Pointer können so ausgerichtet

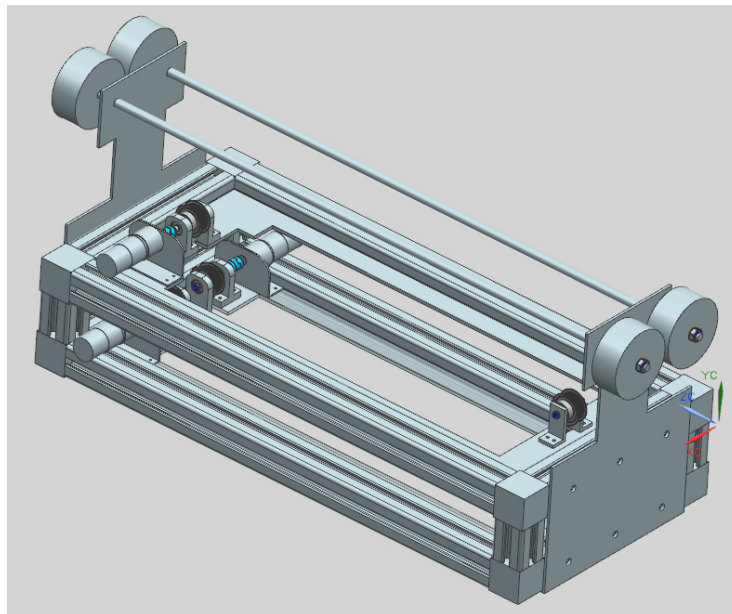


Abbildung 6.14: Schlitten der Modellmaschine mit drei parallelen Querantrieben [129]

werden, dass sie stets auf der selben Linie in y-Richtung verfahren werden können. Das Gesamt-

gewicht des Schlittens liegt bei 5kg . Sowohl das Gewicht des Schlittens, als auch das durch die einzelnen Riemen bewegten, kann erhöht werden, um geänderte kinematische Randbedingungen zu simulieren.

Elektronik und Software

Die Elektronik des Antriebssystems ist in Abbildung 6.15 dargestellt. Da der Schrittmotor über eine komplette Abtastzeit gesteuert werden muss, kommen zwei parallel arbeitende Controller zum Einsatz. Die Hauptsteuerung ist im RN-Controller² implementiert. Die Steuerung läuft mit einer Abtastzeit von 25ms und pro Abtastschritt gibt der Hauptcontroller Geschwindigkeiten bzw. Schrittzahlen für alle vier Motoren vor [131]. Die Gleichstrommotoren werden über zusätzliche Controller, die MD25³, mit standardmäßig implementierten Algorithmen geregelt. Die Controller regeln bis zu zwei Motoren und bereiten die Gebersignale als Reglerückführung auf. Der Controller II für den Schrittmotor steuert je nach Bedarf die Phasen des Schrittmotors im Voll- oder Halbschrittverfahren an. Die Signale werden durch den SM-Controller⁴ auf Phasenströme bis zu 5A verstärkt.

Zum Einspielen von Testszenarien, sprich der Bewegungsvorgaben der Motoren und zum Implementieren der für den Aufbau entwickelten Mikrocontroller-Programme stehen außerdem zwei Programer und ein PC (Host-System) mit dem AVR-Studio⁵ als Entwicklungsumgebung zur Verfügung.

Bei der Datenverarbeitung kommen diverse Schnittstellen und Bus-Systeme zum Einsatz. So werden die Programer über USB an den PC angeschlossen und die Controller von dort aus über die ISP-Schnittstelle programmiert. Die Controller untereinander kommunizieren über den I2C Bus.

Zur Analyse der Parameter des Antriebssystem stehen Messeinrichtungen und Anzeigen zur Verfügung. Über die Motor-Controller ist es möglich, zu jeder Abtastzeit den gemessenen Strom des Motors auszulesen. Mithilfe der ebenfalls bekannten Spannung kann somit zu jeder Abtastzeit die Motorleistung und damit durch Aufsummieren der elektrische Energieverbrauch der einzelnen Motoren ermittelt werden.

Weitere eingebaute Peripherien sind, eine Spannungsversorgung der Baugruppen und Laserpointer, eine Kamera zur Analyse der Modellstruktur [134] sowie Endschalter zur Verhinderung einer Kollision der Modellmaschine. Die genannten Komponenten sind Standardelemente und für die durchgeführten Versuche nicht von wesentlicher Bedeutung.

²RN-Control V1.4: basierend auf ATMega32 [130]

³MD 25: Motorregler mit vordefiniertem Befehlssatz über I2C ansteuerbar [132]

⁴SchrittMotor-Controller: Konfigurierbarer Controller mit Leistungsstufe [133]

⁵Entwicklungsumgebung (IDE) der Firma ATMEL für den Entwurf und Test von Programmen für Mikrocontroller

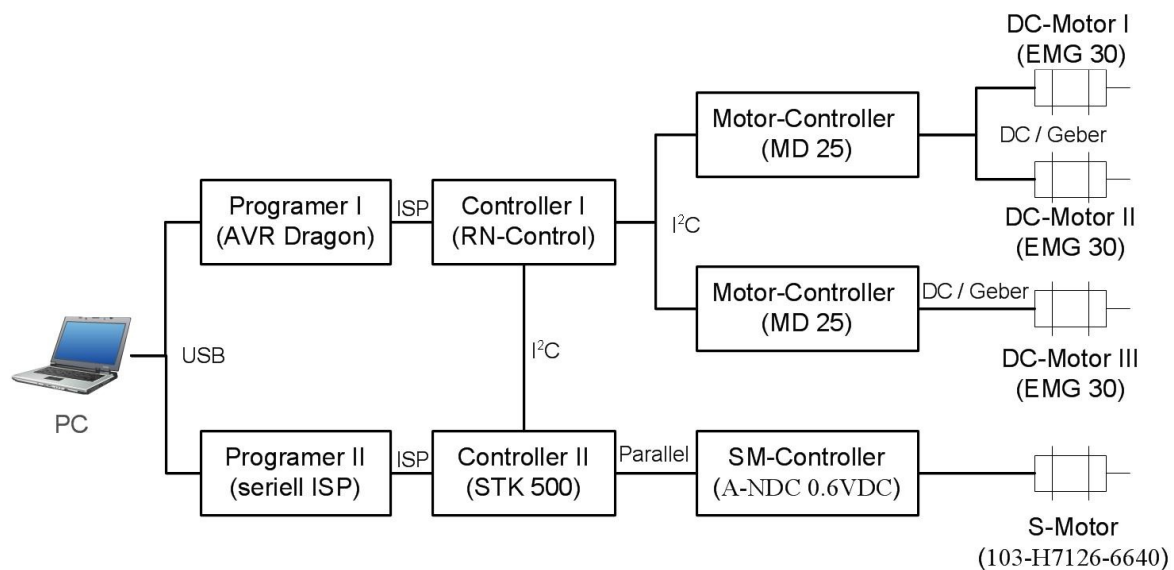


Abbildung 6.15: Elektronik des Versuchsaufbaus

6.4.2 Versuchsergebnisse

Zur Darstellung der Versuchsergebnisse werden zuerst die Anforderungen, sprich die Benutzervorgaben an die Software, festgelegt. Nach einer exemplarischen Untersuchung der Konvergenz der Optimierungsalgorithmen gegen ein Abbruchkriterium werden die Ergebnisse für unterschiedliche Optimierungsvorgaben im Einzelnen beschrieben.

Bezogen auf die Geometrie der Modellmaschine lassen sich Anforderungen an den LTMA wie in Tabelle 6.7 festlegen. Hervorzuheben ist der Bewegungsabschnitt zum „Schließen“ der Fixierung, welche als kollisionsicher gegenüber der Werkzeugachse bedingt ist. Im vorliegenden Beispiel ist der offene Bereich der Werkzeugachse in das letzte Drittel der Taktzeit gelegt. Sind die Abschnitte „Anfahren“ und „Stillsetzen“ beim realen LTMA zwar nur beim Ein- und Ausschalten der Maschine von Relevanz, müssen sie aber im Modell berücksichtigt werden, um auf die Anfangsgeschwindigkeiten eines Taktes von Master und Werkzeugachse zu gelangen.

Im Weiteren folgen aus der mechanischen Modellierung nach Abschnitt 4.1 die kinematischen Restriktionen $\mathbf{K}_{H,i}$, welche für alle Querachsen ($i = 0, 1, 2$) gleich sind, was aus dem gleichen konstruktiven Aufbau folgt. Die Längsachse $i = 3$, als Träger der Querachsen und als Master-signal ist ebenfalls modelliert, ist aber im weiteren nicht mehr Betrachtungsgegenstand, da sie nicht Bestandteil des LTMA ist. Die Restriktionen ergeben sich aus der aufgebauten Mechanik und den Kennwerten der verwendeten Motoren [128, 127]. Entsprechend der in Abschnitt 4.3

Achsen	Abschnitte		Randzustände					
			s [cm]	v [cm/s]	a [cm/s ²]	t [s]	Abhängigkeit	
Master (Schrittmotor)	Anfahren		Start	5	0	0	0	absolut
			Ziel	10	-	0	2	absolut
	Takt	v=konst.	Start	110	-	0	-	statisch relativ zu Master Stillsetzen (-2s)
			Ziel	115	0	0	-	Taktzeit
	Stillsetzen		Start					
			Ziel					
Werkzeugachse (DC-Motor I)	Anfahren		Start	4	0	0	0	absolut
			Ziel	5	-	0	2	absolut
	Takt	v=konst.	Start	40	-	0	-	statisch relativ zu Master Stillsetzen (-2s)
			Ziel	41	0	0	-	Taktzeit
	Stillsetzen		Start					
			Ziel					
Fixierung (DC-Motor II)	Rast 1		Start	5	0	0	0	absolut
			Ziel	5	0	0	-	dynamisch relativ zu Werkzeug Takt (s>17cm v.l.)
	Öffnen		Start	10	0	0	-	automatisch
			Ziel	10	0	0	-	automatisch
	Rast 2		Start	10	0	0	-	automatisch
			Ziel	10	0	0	-	automatisch
	Schließen		Start	5	0	0	-	Kollisionssicher zu Werkzeugachse Takt (s=40)
			Ziel	5	0	0	-	Taktzeit
	Rast 3		Start	5	0	0	0	absolut
			Ziel	10	0	0	-	statisch relativ zu Fixierung Öffnen
Vorschub (DC-Motor III)	Rast		Start	20	0	0	-	statisch relativ zu Fixierung Rast 2
			Ziel	25	0	0	-	Taktzeit

Tabelle 6.7: Anforderungen an das Modell des Längstaktmontageautomats

entwickelten Notation lauten die Matrizen mit den Restriktionen:

$$\mathbf{K}_{H,0} = \mathbf{K}_{H,1} = \mathbf{K}_{H,2} = \begin{pmatrix} 0 \text{ cm} & 50 \text{ cm} \\ -30 \frac{\text{cm}}{\text{s}} & 30 \frac{\text{cm}}{\text{s}} \\ -21 \frac{\text{cm}}{\text{s}^2} & 21 \frac{\text{cm}}{\text{s}^2} \\ -100 \frac{\text{km}}{\text{s}^3} & 100 \frac{\text{km}}{\text{s}^3} \end{pmatrix} \quad (6.5)$$

$$\mathbf{K}_{H,3} = \begin{pmatrix} 0 \text{ cm} & 160 \text{ cm} \\ -15 \frac{\text{cm}}{\text{s}} & 15 \frac{\text{cm}}{\text{s}} \\ -34 \frac{\text{cm}}{\text{s}^2} & 34 \frac{\text{cm}}{\text{s}^2} \\ -100 \frac{\text{km}}{\text{s}^3} & 100 \frac{\text{km}}{\text{s}^3} \end{pmatrix}$$

Die Optimierungskriterien werden entsprechend der im Weiteren vorgestellten Fälle gesetzt. Zunächst werden einige Optimierungsschritte betrachtet, um die Eignung der entwickelten Algorithmen zu zeigen.

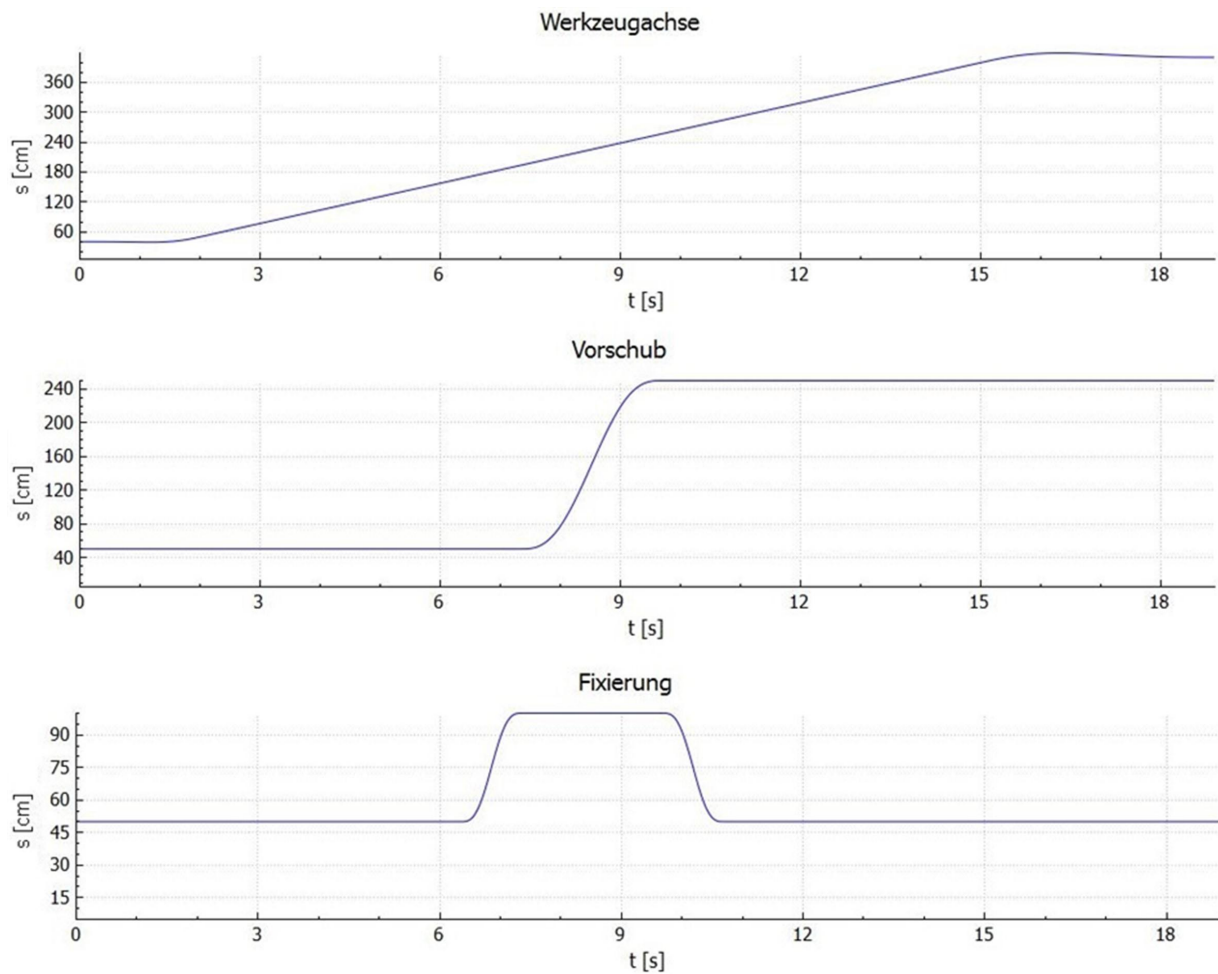


Abbildung 6.16: Positionsverläufe im Initialsystem

Aus den aufgeführten Anforderungen wird, nach den in Abschnitt 5.2.1 beschriebenen Verfahren ein Initialsystem berechnet. Abbildung 6.16 zeigt die zeitlichen Verläufe der Strecken in dem System. Der prinzipielle Verlauf der Bewegungen ist bereits festgelegt, jedoch stimmen die zeitlichen Verläufe nur in der grundsätzlichen Abfolge. Die Anpassung der Bewegung an die festgelegten Bedingungen, wie die Kollisionsvermeidung zwischen Werkzeugachse und Fixierung sind Teil der Optimierung, die im Folgenden detaillierter betrachtet wird.

Konvergenz der Optimierungsalgorithmen

Zunächst wird exemplarisch die zeitliche Angleichung zwischen Teilbewegungen analysiert. Die Entwicklung des Algorithmus ist in Abschnitt 5.2.2 beschreiben. Im vorliegenden Beispiel sind die Abschnitte „Öffnen“ und „Schließen“ der Fixierung und der Abschnitt „Schub“ des

Vorschubs zeitlich gegeneinander auszugleichen. Abbildung 6.17 zeigt, wie sich die jeweiligen Zeiten $T_{\text{Öffnen}}$, $T_{\text{Schliessen}}$ und T_{Schub} über die Schritte s der Rekursion entwickeln. Zudem ist der Verlauf des zehnfachen Wertes des Abbruchkriteriums σ dargestellt. Der Algorithmus

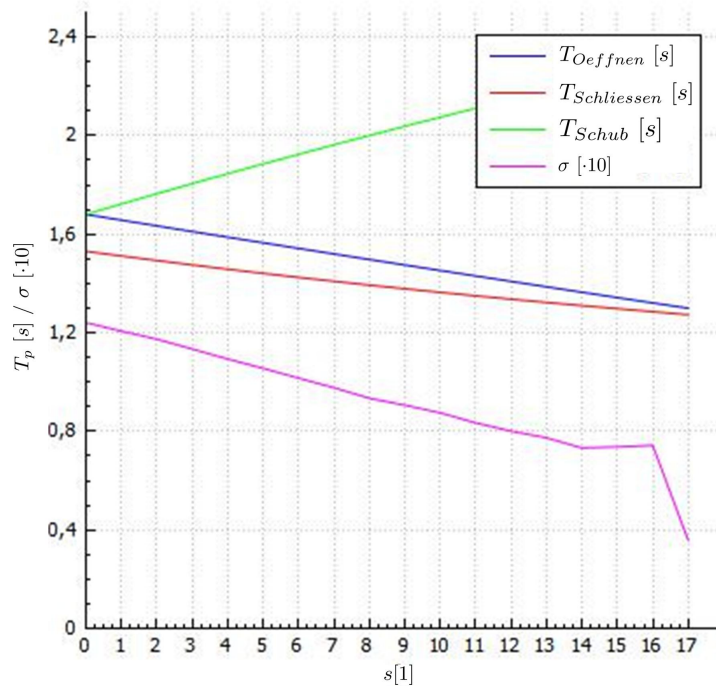


Abbildung 6.17: Konvergenz des Algorithmus zum zeitlichen Ausgleich

bricht, wie in Gleichung 5.17 festgelegt, bei einem Wert kleiner 5% ab. Die Teilbewegungen für das Öffnen und Schließen der Fixierung geben Zeit an den Vorschub ab. Es ist ersichtlich, dass der Algorithmus gegen den Abbruchwert für σ konvergiert.

Ebenfalls exemplarisch kann die gewünschte Konvergenz des Algorithmus zur Taktzeitminimierung gezeigt werden. Die Entwicklung des Zeitintervalls $[t_{\text{min}}; t_{\text{max}}]$ über die Rechenschritte s ist in Abbildung 6.18 aufgetragen. Aus dem Kurvenverlauf wird deutlich, dass die Kurven gegeneinander konvergieren. Der Algorithmus bricht nach sechs Berechnungsschritten ab und bestimmt die minimale Taktzeit zu einem Wert von 17,4 s. Damit ist auch die Eignung des Optimierungsalgorithmus, der in Kapitel 5.2.3 dokumentiert ist, gezeigt.

Optimierung des modellierten LTMA

Im Folgenden werden die Ergebnisse für die Optimierung des Modells entsprechend der drei Optimierungsvorgaben aufgeführt. Zunächst werden die berechneten Sollgrößen für die Modellmaschine beschrieben. Die Abtastzeit des Modells beträgt 25ms. In Abbildung 6.19 sind die

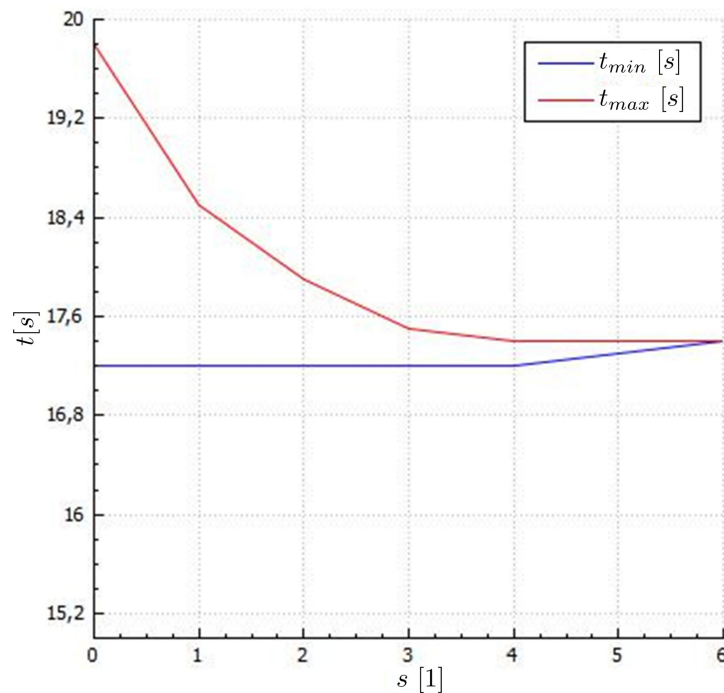


Abbildung 6.18: Konvergenz des Intervallhalbierungsverfahrens

Positionsverläufe für die Werkzeugachse, den Vorschub und die Fixierung für je drei Optimierungsfälle, nämlich für den energiesparenden, ruckoptimalen und zeitminimierenden Entwurf dargestellt. Alle Bewegungen gehen aus den in Tabelle 6.7 notierten Anforderungen hervor. Es ist darauf zu achten, dass für den Weg des Vorschubs und der Fixierung nur der interessierende Zeitbereich ab der zehnten Sekunde aufgetragen ist. Davor befinden sich die Achsen in ruhen-dem Zustand.

Die Optimierung der Bewegungen erfolgt insbesondere in den Abschnitten „Öffnen“ und „Schließen“ der Fixierung sowie „Schub“ des Vorschubs. Die Abschnitte können nicht nur zeitlich gegeneinander ausgewogen, sondern auch in ihrer Gesamtzeit angepasst werden. Im vorliegenden Berechnungsbeispiel beträgt die Rechengenauigkeit $t_\epsilon = 50 \text{ ms}$.

Weil den Bewegungen von Vorschub und Fixierung besondere Bedeutung zukommt, ist von der Werkzeugachse nur das Streckendiagramm abgedruckt. Die Werkzeugachse wird angefahren und läuft dann 350 mm mit konstanter Geschwindigkeit. Beim zeitoptimalen Entwurf (blau) ist diese Geschwindigkeit größer. Der Vorschub wird durch eine Verfäherung um 200 mm von Rast in Rast simuliert, wozu die Fixierung öffnet und schließt, was im untersten Diagramm aufgetragen ist. An den Kurven ist zu erkennen, dass die Taktzeit des zeitoptimalen Entwurfes kleiner ist, die zugehörige blaue Linie endet bereits nach 17,4 Sekunden. Der Vorschub und das Öffnen und Schließen der Fixierung erfolgen entsprechend früher als beim ruckoptimalen (gelb) und energieoptimalen Entwurf (grün), deren Positionsverläufe sich kaum unterscheiden.

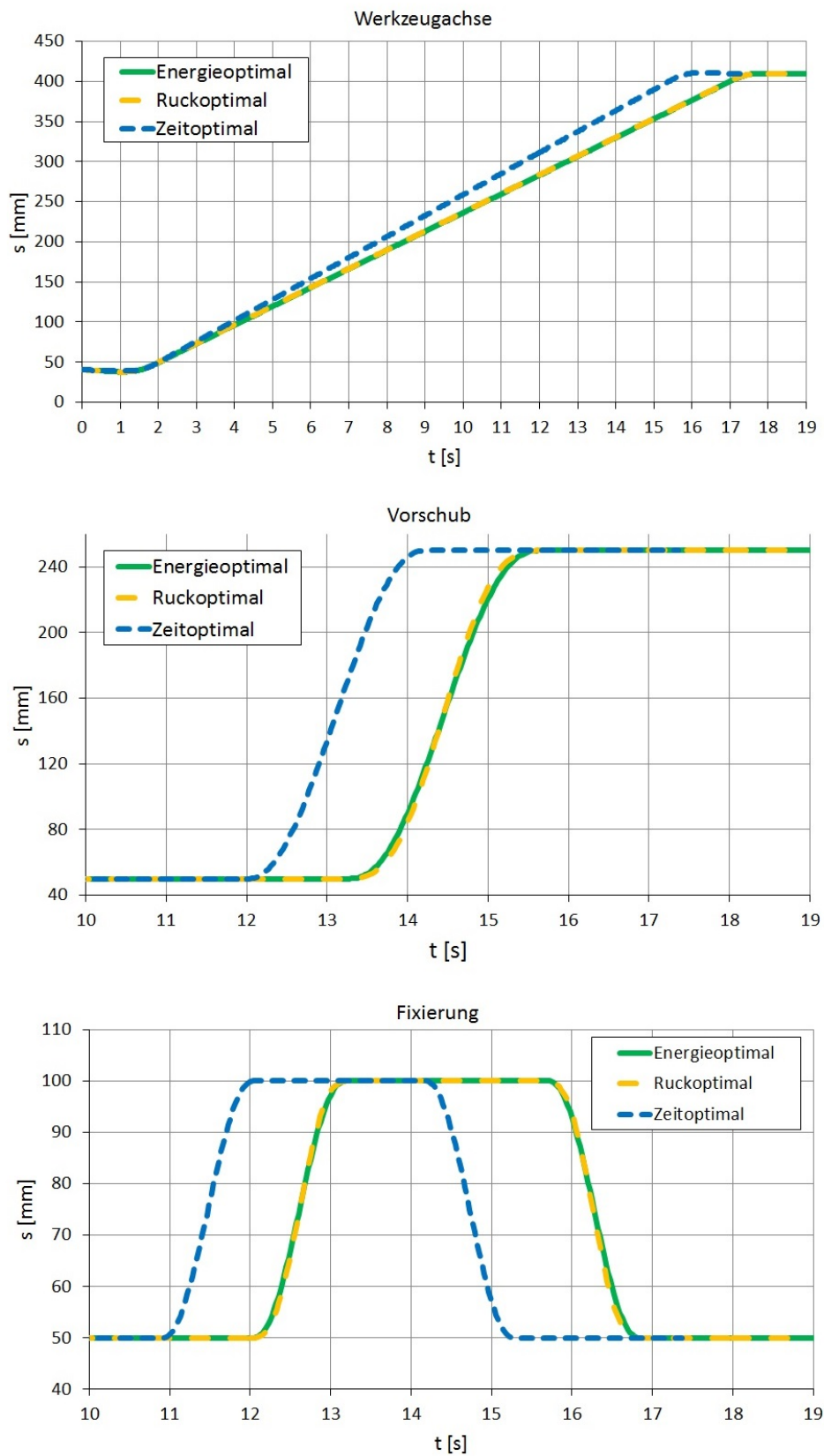


Abbildung 6.19: Wege des Versuchsmodells nach Achsen und Optimierungsvorgaben

Die energieoptimale Bewegung schließt kaum erkennbar nach der ruckoptimalen ab. Die Unterschiede zwischen diesen Bewegungen zeigen sich im Geschwindigkeits- und Beschleunigungsdiagramm. Die Geschwindigkeitsdiagramme hierfür sind in Abbildung 6.20 dargestellt.

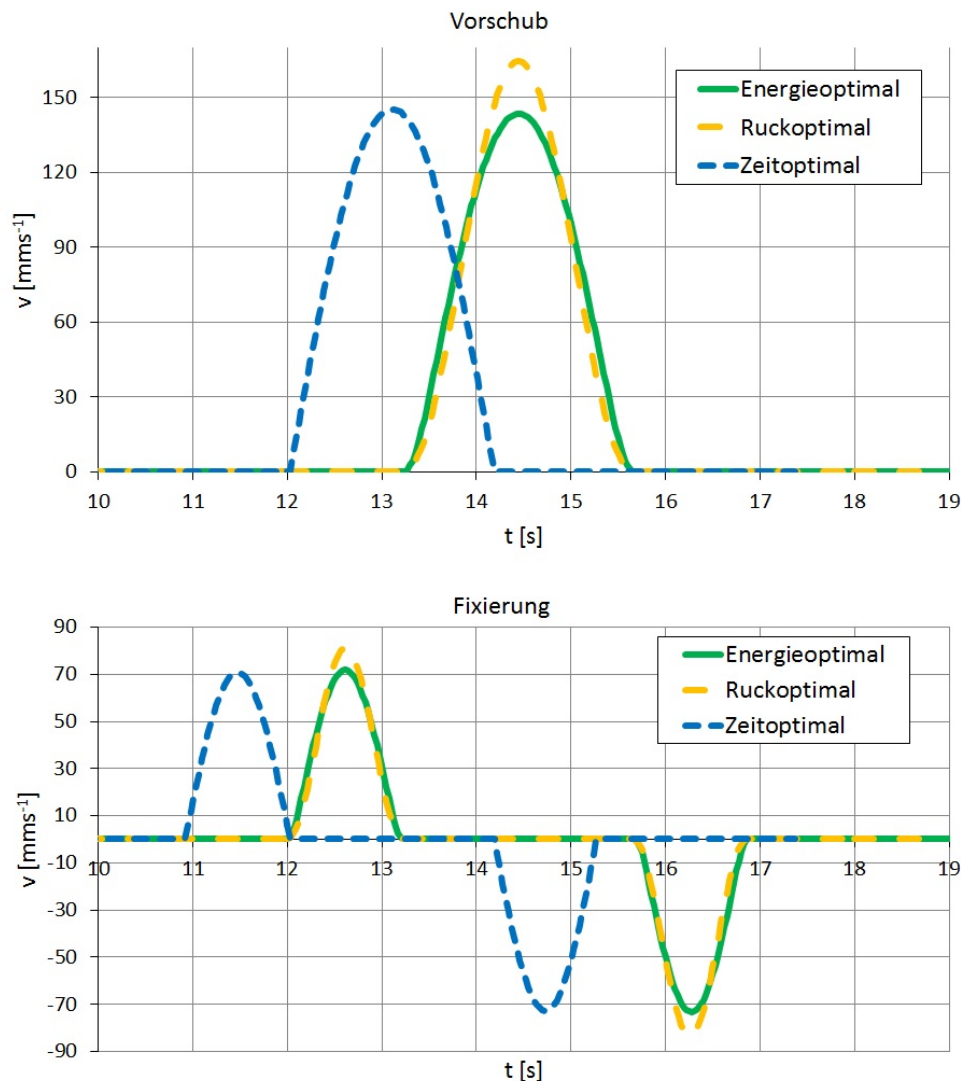


Abbildung 6.20: Geschwindigkeiten des Versuchsmodells nach Achsen und Optimierungsvorgaben

Die Bewegung für den ruckoptimalen Entwurf besitzt in beiden Bewegungen die höchsten maximalen Geschwindigkeiten, aber auch schwächere Steigungsänderungen. Das resultiert zuletzt auch aus einem gewünschten Verlauf mit möglichst kleinen Beschleunigungsänderungen (siehe Abbildung 6.21), wodurch der Ruck betragsmäßig klein gehalten wird. Die Geschwindigkeit der ruckoptimalen Bewegung steigt langsamer an, muss dann aber für das Zurücklegen der gleichen Strecke in vergleichbarer Zeit zu einem höheren Maximalwert ansteigen.

Am Beschleunigungsdiagramm ist zu erkennen, dass die zeitoptimale Bewegung durch das sprunghafte ansteigen und abfallen der Beschleunigung realisiert ist. Auch der Unterschied zwischen energieoptimaler und ruckoptimaler Bewegung ist in dem Diagramm deutlich erkennbar. Beim energieoptimalen Entwurf steigt die Beschleunigung vergleichsweise kurz an und flacht dann schneller wieder ab. Die Kurve hat damit einen kleineren betragsmäßigen Mittelwert. Der Ruckverlauf ist in Abbildung 6.22 ebenfalls für alle drei Optimierungsvorgaben und den Vorschub sowie die Fixierung aufgetragen. Es ist markant, dass der Ruck für die zeitoptimale Bewegung an den Stellen der Beschleunigungssprünge völlig aus dem Rahmen tritt, während Kurven im Ruckdiagramm für den ruckoptimalen Entwurf bei beiden Achsen die $10 \frac{dm}{s^3}$ nicht überschreiten (s. Ruckdiagramm).

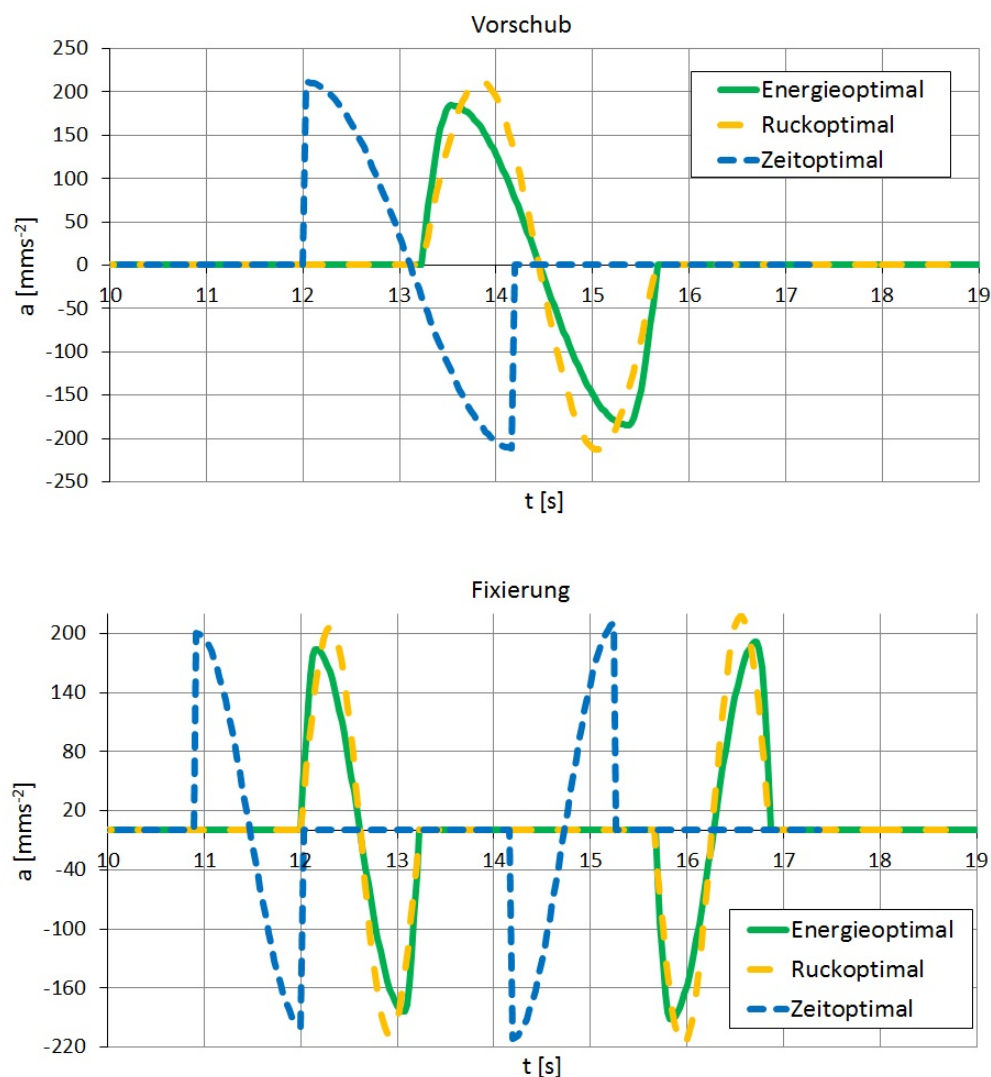


Abbildung 6.21: Beschleunigungen des Versuchsmodells nach Achsen und Optimierungsvorgaben

Neben der Taktzeit und dem Ruck kann durch die Diagramme auch der Energieverbrauch interpretiert werden. Je kleiner die Werte im Geschwindigkeits- und Beschleunigungsdiagramm sind, desto geringer ist der zu erwartende Energieverbrauch der Achse. Der beschriebene Zusammenhang kann aus der Verlustfunktion für den Energieverbrauch (Gleichung (3.24)) abgeleitet werden. Der Faktor aus der Wegdifferenz zweier benachbarter Werte kann als Geschwindigkeit gesehen werden. Der Faktor aus der mittleren Beschleunigung zweier benachbarter Werte ist direkt als interpolierter Plot im Beschleunigungsdiagramm zu sehen.

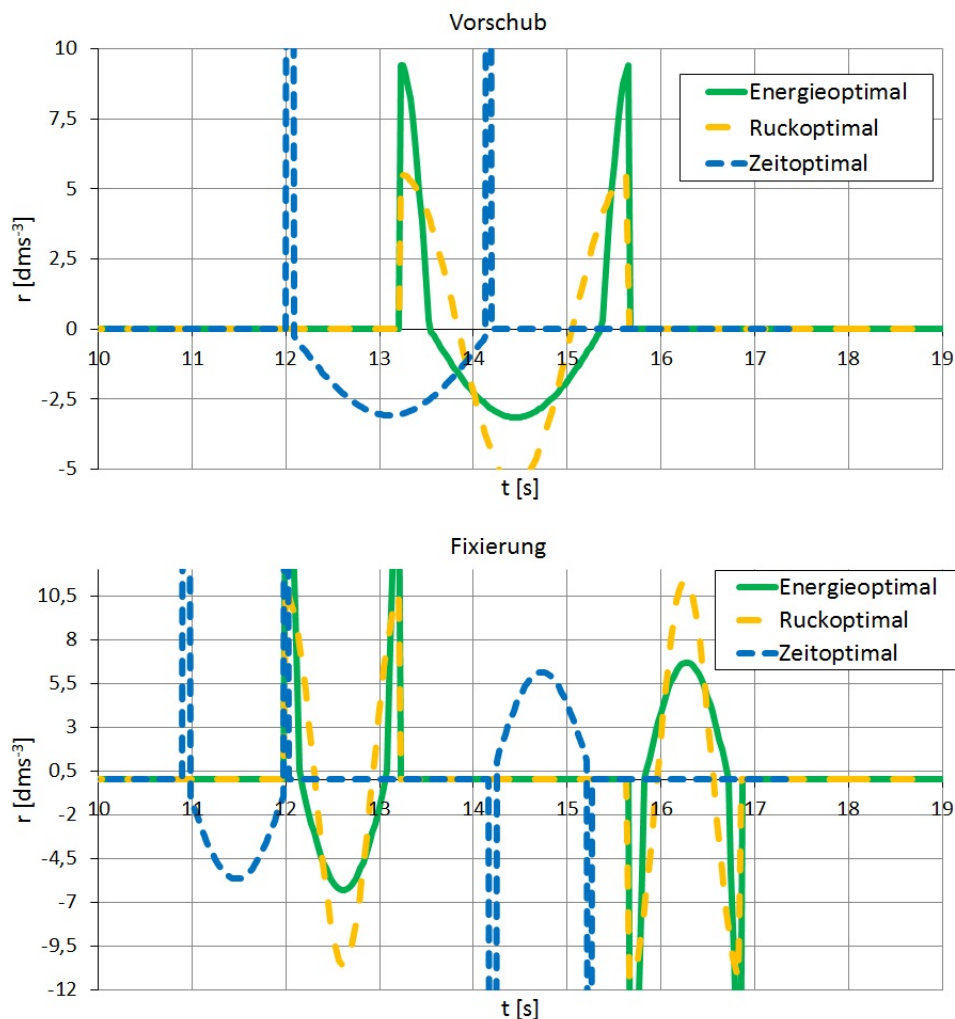


Abbildung 6.22: Ruckverläufe des Versuchsmodells nach Achsen und Optimierungsvorgaben

Die exakten Werte der Kennwerte der Bewegungen von Fixierung und Vorschub sind in Tabelle 6.8 für die drei Optimierungsvorgaben zusammengestellt. Die hervorgehobenen Werte bilden die jeweiligen Minima der Kennwerte für die jeweilige Achse. Es ist ersichtlich, dass die optimierten Entwürfe in allen Fällen die gewünschten Optima bilden.

Das Versuchsmodell enthält gegenüber der theoretischen Modellierung des LTMA auch die Bewegungsvorgänge zum Anfahren und Stillsetzen der Antriebe, was insbesondere die, real mit konstanter Geschwindigkeit bewegten, Master- und Werkzeugachse betrifft. Das ist notwendig, weil das Modell nur über einen begrenzten Arbeitsraum verfügt, denn Rotationen der echten Maschine werden als Translationen der Laserpointer dargestellt. Somit bildet der Entwurf für das physikalische Modell eine komplexere Aufgabenstellung als der in Abschnitt 6.2 beschriebene theoretische Entwurf.

Damit ist auch das physikalische Modell durch die neu entwickelten Algorithmen optimierbar

	T_{Takt} [s]	rel.	$ r _{max}$ [$\frac{cm}{s^3}$]	rel.	k_E [1]	rel.	E_{Mess} [mWs]	rel.
E (F)	19,000	100%	1.987	28,3%	10.275	75,8%	574	61,5%
R (F)	18,950	99,8%	1.124	16,0%	13.546	100%	933	100%
Z (F)	17,375	91,4%	7.025	100%	10.458	77,2%	796	85,3%
E (V)	19,000	100%	942	13,4%	20.625	76,1%	1.568	82,8%
R (V)	18,950	99,8%	550	7,8%	27.103	100%	1.893	100%
Z (V)	17,375	91,4%	7.045	100%	21.140	78,0%	1.590	84,0%

Tabelle 6.8: Optimierung des Versuchsmodells
E-energieoptimal, R-ruckoptimal, Z-zeitoptimal
(F)-Fixierung, (V)-Vorschub

und die Leistungsfähigkeit des neuen Konzepts zum ganzheitlichen Engineering unter Beweis gestellt.

Die praktische Beweisführung durch die Modellmaschine erfolgt durch die Messung des elektrischen Energieverbrauchs E_{Mess} eines Gleichstrommotors beim Abfahren der jeweiligen Bewegung. Die Ergebnisse sind in ebenfalls Tabelle 6.8 aufgetragen und bestätigen qualitativ die Werte aus der Verlustfunktion für den Energieverbrauch.

Die Taktzeit der Bewegungen kann durch den Controller über eine RS232-Schnittstelle ausgegeben werden. Die Ergebnisse sind nicht explizit notiert, weil sie genau den berechneten in der Tabelle entsprechen. Auf den Aufbau einer Messeinrichtung für den Ruck bzw. die Beschleunigung ist verzichtet, weil die Richtigkeit der gefahrenen Bewegung bezogen auf die, in den Diagrammen 6.19 und 6.20 dargestellten, Sollkurven methodisch sicher gestellt werden kann. Die Mikrocontroller überwachen in jedem Abtastschritt die Abweichung der Sollwerte von den über Drehgeber beobachteten Istwerten der Achsen. Bei einer nicht hinnehmbaren Abweichung werden alle Antriebe stillgesetzt und ein Fehlersignal wird ausgegeben. Da das Versuchsmodell die komplette Bewegung aller Achsen fehlerfrei ausführt, kann folglich von der Korrektheit der maximalen Ruckwerte als Vergleichswert ausgegangen werden.

Die Optimierung der besonders interessierenden Teilbewegungen zum Öffnen und Schließen der Fixierung sowie der Vorschubbewegung ist mit detaillierten Zeitwerten und kinematischen Eigenschaften detailliert im Anhang 9.3 beschrieben.

7 Zusammenfassung

Im modernen Maschinenbau kommen, beim Entwurf von komplexen Bewegungsmechanismen, zunehmend elektrische Servoantriebe anstelle von aufwendigen mechanischen Konstruktionen zum Einsatz. Bei einer Applikation mit vielen Einzelbewegungen, die in der Regel räumliche und zeitliche Abhängigkeiten aufweisen, entsteht ein komplexes System, für das Bewegungen im Detail entworfen, Antriebskomponenten ausgewählt und Optimierungen durchgeführt werden müssen. Für die vielfältigen und komplizierten Zusammenhänge innerhalb und zwischen den drei Aufgaben des Engineerings genügen die existierenden Methoden nicht mehr, um in effizienter Arbeitsweise zu einem hinreichend optimierten Antriebssystem zu gelangen. Unter der Ausnutzung stetig wachsender Leistungen in der Computertechnik, ist es jedoch möglich, neue Vorgehensweisen zu entwickeln und algorithmisch umzusetzen sowie zu implementieren.

Die vorliegende Arbeit ist ein Beitrag zur Weiterentwicklung des Engineerings von elektrischen Antriebssystemen. Sie beschreibt eine neue Vorgehensweise zur ganzheitlichen Betrachtung aller wesentlichen Entwurfsschritte, die beim Erarbeiten eines Antriebssystems zum Einsatz kommen. Die übliche sequenzielle Vorgehensweise wird durch ein neues integrierendes Konzept ersetzt, welches

- den Entwurf von Bewegungsvorgängen (Sollgrößengenerierung) für alle Antriebe,
- die Auswahl von Antriebshardware nach erfolgter Auslegungsrechnung, sowie
- die Optimierung der gesamten Maschine hinsichtlich (multi-)kriterieller Vorgaben

als Einheit behandelt. Die aufgezählten Aufgaben werden bislang größtenteils manuell von Entwicklern bearbeitet. Das vorgestellte neuartige Konzept automatisiert weitgehend notwendige Entwurfsschritte.

Bei den Optimierungskriterien handelt es sich um die Minimierung der Zeit, des Ruckes oder des Energieverbrauchs. Zur Entwicklung des neuen Konzepts werden als Basis die Anforderungen an ein neues Konzept herausgearbeitet, die dem Stand der Technik von heutigen Antriebssystemen und deren Einsatzklassen genügen. Die Formulierung von Methoden und Algorithmen nimmt zudem Rücksicht auf die Zielstellung, daraus eine Software zur Umsetzung der neuen

Methode implementieren zu können.

Die drei zentralen Aufgaben finden sich als Grundbestandteile des neuen Konzepts wieder. Es handelt sich um ineinander integrierte Gruppen von Algorithmen, dem Bewegungsentwurf, dem Hardwareentwurf und dem Systementwurf.

Um mittels numerischer Algorithmen schließlich ein Antriebssystem optimieren zu können, ist der Bewegungsentwurf automatisiert. Bewegungsvorgänge werden klassifiziert und können somit Typen von Aufgabenstellungen zugeordnet werden. Hierdurch wird eine automatische Interpolation von Bewegungen anhand definierter mathematischer Bedingungen möglich, womit eine Mannigfaltigkeit von Bewegungsvorgängen ermittelbar ist.

Um hieraus eine Bewegung zu wählen, die den Optimierungsvorgaben Rechnung trägt, bedarf es Algorithmen zur Selektion. Dazu sind Verlustfunktionen formuliert, die eine eindeutige Selektion ermöglichen und als Alternative ein auf künstlicher Intelligenz basierender Auswahlalgorithmus wiederum auf Grundlage der Fuzzy-Logik entwickelt worden.

Der Auslegungsrechnung und Auswahl von Komponenten kommt bereits hohe Aufmerksamkeit zu, wodurch praktisch alle notwendigen mathematischen Zusammenhänge gegeben sind. Die Integration in ein ganzheitliches Konzept mit hohem Automatisierungsgrad erfordert jedoch eine Anpassung der bekannten Notationen. Dazu wird eine modularisierte und standardisierte Modellierung von angetriebener Mechanik und die Formulierung einer Auslegungsrechnung mit geeigneter Notation sowie eine Methodik zur automatischen Auswahl und Anpassung von Komponenten entworfen.

Durch die im Bewegungsentwurf manifestierten Selektionsverfahren ist eine Möglichkeit zur multikriteriellen Anpassung einer Teilbewegung gegeben. Um eine Optimierung des gesamten Systems zu erreichen, muss aber auch die Zeit eines gesamten Vorgangs im Antriebssystem angepasst werden. Hierfür ist es notwendig,

- kausale Zusammenhänge zwischen den Achsen abbilden zu können,
- konkurrierende Raumnutzung und Bedingungen zur Kollisionsvermeidung zu beschreiben und
- zeitliche Abläufe im System sowie die Taktzeit optimieren zu können.

Zur Abbildung der Zusammenhänge im System und zu deren Analyse ist ein diskretkontinuierliches Petrinetz konzipiert worden. Hiermit lassen sich ein Initialsystem entwerfen und hinsichtlich der Optimierungsaufgabe kritische Pfade ermitteln.

Diverse Typen von Abhängigkeiten zwischen Achsen sind klassifiziert und deren Darstellung mathematisch formalisiert worden, dazu gehört auch die Kollisionsvermeidung im Fehlerfall.

Zur Optimierung der zeitlichen Abläufe im Antriebssystem sind zwei Optimierungsalgorithmen entwickelt worden, einer zur Anpassung der Taktzeit des Systems und ein weiterer zur Anpassung der Zeit von Teilbewegungen untereinander.

Im Einklang mit dem wissenschaftlichen Fortgang der vorliegenden Arbeit entstand eine sich stets weiterentwickelnde Basis an implementierten Algorithmen mit einer zum Endstand der Arbeit erreichten Komplexität von über 15.000 Anweisungen in Hochsprache. Auf den grundsätzlichen Aufbau der Software wird mehrfach Einblick gegeben. Anhand der Implementierung werden die neuen Vorgehensweisen und Methoden verifiziert und die Leistungsfähigkeit des neuen ganzheitlichen Konzepts unter Beweis gestellt. Hierfür wird, als Beispiel aus der Fertigungstechnik, das Antriebssystem eines Längstaktmontageautomats (LTMA) optimiert. Zudem wurde eine physikalische Modellmaschine entwickelt, mit der es möglich ist, diverse Strukturen von Antriebssystemen mit bis zu vier Achsen abzubilden. Im Rahmen der Arbeit wurde mit dem Versuchsmodell ebenfalls der LTMA untersucht.

Das neue Konzept und die dazu neu entwickelten Algorithmen und Vorgehensweisen liefern folgende wesentliche Vorteile:

- Durch automatisierte Vorgänge ist das Engineering an sich wirtschaftlicher und einfacher.
- Neue Eigenschaften entstehen, wie die methodische Betrachtung der Kollisionsvermeidung auch im Fehlerfall.
- Ein Maschinenentwurf mit besseren Leistungsdaten hinsichtlich Energieverbrauch, Schwingungspotenzial und Taktzeit ist erreichbar.
- Eine Auswahl ressourcenschonenderer Komponenten ist realisierbar.
- Die Programme liefern nicht nur Bewegungen zur Verwendung im Systementwurf sondern direkt als Sollgrößen einsetzbare Kurven.

Im Folgenden werden die wichtigsten Ergebnisse der Arbeit zusammengestellt:

1. Ein neues Grundkonzept zum ganzheitlichen Engineering von elektrischen Servoantriebssystemen wird hergeleitet.
2. Eine Entwicklungsvorgehensweise für die Ermittlung einer zeit-, energie- oder schwingungsoptimierten Gesamtlösung von komplexen verteilten Bewegungsaufgaben unter Berücksichtigung der Auslegungsregeln für Antriebssysteme wird festgelegt.
3. Mathematische Methoden und Algorithmen zur automatischen Bewegungssynthese, Hardwareauswahl und Systemoptimierung werden entwickelt.

4. Auf Grundlage der VDI-2143 und weiterer Bewegungsgesetze wird eine vollständige Klassifizierung von Bewegungsvorgängen in mögliche kinematische Strukturen von Randbedingungen festgelegt.
5. Zu den in der VDI-Richtlinie 2143 enthaltenen Bewegungsgesetzen wird als zusätzliche Formel die dritte Ableitung, der Ruck, berechnet.
6. Drei Verlustfunktionen werden hergeleitet, mit denen eine Bewegung hinsichtlich Energieverbrauch, maximalem Ruck und nötiger Zeit bewertet wird.
7. Selektionsmethoden zur Auswahl von Bewegungsvorgängen nach kompromissorientierten widersprüchlichen Optimierungskriterien werden auf Grundlage der Fuzzy-Logik hergeleitet.
8. Zwei Regelsätze zur multikriteriellen Selektion werden formuliert und implementiert.
9. Modelle angetriebener Mechanik zur Ermittlung notwendiger Größen hinsichtlich angestrebter Auslegungsrechnungen innerhalb des neuen Konzepts werden entwickelt.
10. Mathematische Modelle, basierend auf Petrinetzen, sind zur Abbildung eines Antriebssystems mit inneren Zusammenhängen entworfen.
11. Die Berücksichtigung der Kollisionsfreiheit im Fehlerfall ist formalisiert worden.
12. Algorithmen zur Optimierung der Taktzeit und zur Angleichung der Zeiten von Teilbewegungen werden konstruiert.
13. Sämtliche Algorithmen sind zur Verifikation und weiteren Verwendung in der Hoschsprache C++ implementiert.
14. Eine Benutzeranwendung für den Einsatz des neuen Konzepts ist mitentwickelt.
15. Eine physikalische Modellmaschine zur Analyse von Funktionalität und Mehrwert des Konzepts ist konzipiert, aufgebaut und auf bekannte Problemstellungen angewendet.

Die Weiterentwicklung zur praktischen Anwendbarkeit der entwickelten Software im Engineering von Antriebssystemen und eine Integration der neuartigen Methode in die angewendeten Entwicklungsverfahren ist die zukünftige Zielstellung. Daraus ergeben sich zahlreiche weitere Entwicklungsschritte und zu lösende Aufgaben für die Zukunft, die den Rahmen der vorliegenden Arbeit gesprengt hätten. Im Folgenden sind einige wünschenswerte Verbesserungen aufgezählt:

- Die Software ist zu einem Produkt mit marktgerechter Qualität weiterzuentwickeln.
- Der Energieaustausch zwischen den Achsen über eventuell gekoppelte Zwischenkreise ist zu modellieren. Hierbei empfiehlt sich eine Integration in das entwickelte Petrinetz.

-
- Die unterschiedliche Priorität der Optimierung für verschiedene Achsen muss abgebildet werden. Hierfür ist eine Bewertung mittels Fuzzy-Logik geeignet.
 - Der Algorithmus zum Angleichen der Zeit von Teilbewegungen muss eine Adaption zur dynamischen Anpassung der Komplexität erhalten, um die Rechenzeit zu reduzieren. Eine Erweiterung der Methode um einen Computational Intelligence-Anteil ist möglich.
 - Schnittstellen zu Mehrkörpersimulationen, CAD-Systemen und zu Software zur Inbetriebnahme und Überwachung von Antriebsreglern sind wünschenswert.

8 Literaturverzeichnis

- [1] ZVEI. *Elektroindustrie in Zahlen*. Zentralverband Elektrotechnik und Elektroindustrie, 2012.
- [2] R. Isermann. *Mechatronische Systeme : Grundlagen*. Springer, Berlin, Heidelberg, 2008.
- [3] H. Bernstein. *Grundlagen der Mechatronik*. VDE Verl., Berlin, 2004.
- [4] W. Roddeck. *Einführung in die Mechatronik*. Vieweg+Teubner Verlag, Wiesbaden, 2012.
- [5] B. Heimann; W. Gerth; K. Popp. *Mechatronik: Komponenten, Methoden, Beispiele*. Carl-Hanser-Verl., München, 2007.
- [6] C. Melchiorri L. Biagiotti. *Trajectory planning for automatic machines and robots*. Springer, Berlin, 2008.
- [7] M. Husty. *Kinematik und Robotik*. Springer, Berlin, 1997.
- [8] J. Fleischer; J.P. Schmidt-Ewig. Simulationsgestützte Konzeption von Handhabungskinematiken. *wt - Werkstattstechnik, Jahrgang 98 (2008)*, 2008.
- [9] A. Heim. *Modellierung, Simulation und optimale Bahnplanung bei Industrierobotern*. Dissertation, Herbert Utz Verlag, München, 1998.
- [10] H. Choset; K. Lynch; S. Hutchinson; G. Kantor; W. Burgard; L. Kavraki; and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.
- [11] P. Falco; C. Natale. On the Stability of Closed-Loop Inverse Kinematics Algorithms for Redundant Robots. *Robotics, IEEE Transaction on (99)*, 2011.
- [12] D. Wloka. *Robotersimulation*. Springer, Berlin, 1991.
- [13] G. Agricola; G. Fraustadt [Bearb.]. *Ausgewählte Werke = (Bergbau und Hüttenkunde)*, volume 8: De re metallica : libri XII. Dt. Verl. der Wiss., Berlin, 1974.
- [14] P. Brosch. *Praxis der Drehstromantriebe : mit fester und variabler Drehzahl; Maschinen, Leistungselektronik, Einsatz*. Kamprath-Reihe. Vogel, Würzburg, 2002.

- [15] W. Steinhilper [Hrsg.]; A. Albers; L. Deters; J. Feldhusen; u.a. *Konstruktionselemente des Maschinenbaus 2 : Grundlagen von Maschinenelementen für Antriebsaufgaben*. Springer, Berlin, Heidelberg, 2006.
- [16] H. Kerle. *Getriebetechnik : Grundlagen, Entwicklung und Anwendung ungleichmäßig übersetzender Getriebe*. Vieweg+Teubner Verlag, Wiesbaden, 2012.
- [17] M. Kaufmann. *Ein Beitrag zur Informationsverarbeitung in mechatronischen Systemen*. Habil.-Schrift, Karlsruhe, 2010.
- [18] B. Bertsche; P. Göhner; U. Jensen; W. Schinköthe; H. J. Wunderlich. *Zuverlässigkeit mechatronischer Systeme : Grundlagen und Bewertung in frühen Entwicklungsphasen*. Berlin, Heidelberg, 2009.
- [19] P. Brosch. *Motion Control : Konzept, Systemaufbau und Anwendung*. Verl. Moderne Industrie, Landsberg/Lech, 2002.
- [20] Firma Heidenhain. *Drehgeber für hochdynamische Servoantriebe*. <http://www.heidenhain.de>, 2008.
- [21] R. Fischer. *Elektrische Maschinen*. Hanser, München, 2009.
- [22] U. Riefenstahl. *Elektrische Antriebssysteme : Grundlagen, Komponenten, Regelverfahren, Bewegungssteuerung*. Teubner, Wiesbaden, 2006.
- [23] A. Kremser. *Elektrische Maschinen und Antriebe : Grundlagen, Motoren und Anwendungen; mit 10 Tab. und 14 Beispielaufgaben mit Lösungen*. Teubner, Stuttgart, 2004.
- [24] K. Fuest. *Elektrische Maschinen und Antriebe : Lehr- u. Arbeitsbuch; mit zahlr. durchgerechneten Beisp. u. Uebungen*. Vieweg, Braunschweig, 1983.
- [25] R. Schönfeld; W. Hofmann. *Elektrische Antriebe und Bewegungssteuerungen : von der Aufgabenstellung zur praktischen Realisierung*. VDE-Verl., Berlin, 2005.
- [26] Firma Parker Hannifin. *Motion Control Produkte*. <http://www.parker.com>, 2011.
- [27] C. Legler. *Modellbildung einer Servoachse*. Diplomarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2012.
- [28] J. Weidauer. *Elektrische Antriebstechnik : Grundlagen - Auslegung - Anwendungen - Lösungen*. Publicis Corporate Publ., Erlangen, 2008.
- [29] E. Kiel; T. Loy. Der digitale antriebsregler auf dem weg vom einfachen aktor zur dezentralen intelligenten bewegungssteuerung. *Tagungsband SPS/IPC/Drives*, pages 476–487, 1998.
- [30] Firma Müller-Kurtz. 2006.

- [31] M. Lorch; M. Kaufmann; G. Bretthauer. Sicherstellung der kollisionsfreiheit einer hochdynamischen fertigungsmaschine mit dezentralen servoantrieben. *Automation*, 2008.
- [32] R. Langmann [Hrsg.]. *Taschenbuch der Automatisierung*. Fachbuchverl. Leipzig, München, 2010.
- [33] Firma Parker Hannifin. *Intelligenter Servoantrieb Compax3*. <http://www.parker.com>, 2011.
- [34] C. C. Hsieh; P. L. Hsu. The CAN-Based Synchronized Structure for Multi-Axis Motion Control Systems. *Systems, Man and Cybernetics, 2005 IEEE International Conference*, S. 1314-1319, 2005.
- [35] M. Franke. *Intelligente Antriebe im Systemverbund : ein Beitrag zur Untersuchung des Betriebsverhaltens von dezentralen feldbusgekoppelten Antrieben*. Fortschritt-Berichte VDI : Reihe 21, Elektrotechnik ; 277. VDI-Verl., Düsseldorf, 1999.
- [36] D. Gross; Hauger; Schröder; Wall. *Technische Mechanik : Band 1: Statik*. Springer-Verlag, Berlin, Heidelberg, 2008.
- [37] H. Dubbel [Begr.]; K.-H. Grote [Hrsg.]. *Taschenbuch für den Maschinenbau*. Springer, Berlin, 2005.
- [38] R. C. Hibbeler. *Technische Mechanik*, volume 3: Dynamik. Pearson Studium, München [u.a.], 2006.
- [39] R. Busch. *Elektrotechnik und Elektronik : für Maschinenbauer und Verfahrenstechniker*. Vieweg+Teubner, Wiesbaden, 2011.
- [40] J. Lunze. *Regelungstechnik I : Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [41] Verband Deutscher Ingenieure (VDI). *VDI 3547 - Gütebewertung für Bewegungssysteme und gesteuerte Bewegungsabläufe*. Beuth-Verlag, Berlin, 2003.
- [42] H. Grzonka. Energieeffiziente Antriebslösungen für Verpackungsmaschinen. *SPS IPC DRIVES Nürnberg*, S. 638-650, 2001.
- [43] K. Ehrlenspiel. *Integrierte Produktentwicklung : Denkabläufe, Methodeneinsatz, Zusammenarbeit*. Hanser, München, 2009.
- [44] H. Grabowski [Hrsg.]. *Neue Wege zur Produktentwicklung*. Raabe, Stuttgart, 1997.
- [45] U. Lindemann; R. Reichwald; M. Zäh. *Individualisierte Produkte - Komplexität beherrschen in Entwicklung und Produktion*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

- [46] J. C. Ponn; U. Lindemann. *Konzeptentwicklung und Gestaltung technischer Produkte : optimierte Produkte - systematisch von Anforderungen zu Konzepten*. Springer, Berlin, 2008.
- [47] A. Albers; M. Meboldt. *SPALTEN Matrix - Product Development Process on the Basis of Systems Engineering and Systematic Problem Solving, aus: The Future of Product Development*. Springer, Heidelberg, 2007.
- [48] A. Albers; N. Burkardt; M. Meboldt. SPALTEN problem solving methodology in the product development. *Conference on Engineering Design - ICED*, 2005.
- [49] A. Yoji. *QFD-Quality Function Deployment*. Moderne Industrie, Landsberg/Lech, 1992.
- [50] S. Pugh. *Total design : integrated methods for successful product engineering*. Prentice-Hall, Harlow, Engl. [u.a.], 2010.
- [51] D. Rainey. *Product innovation : leading change through integrated product development*. Cambridge Univ. Press, New York, 2005.
- [52] A. Albers; S. Matthiesen. Konstruktionsmethodisches Grundmodell zum Zusammenhang von Gestalt und Funktion technischer Systeme - Das Elementmodell Wirkflächenpaare und Leitstützstrukturen zur Analyse und Synthese technischer Systeme. *Konstruktion, Zeitschrift für Produktentwicklung*, 2002.
- [53] E. Kiel [Hrsg.]. *Antriebslösungen : Mechatronik für Produktion und Logistik; mit 51 Tabellen*. VDI. Springer, Berlin, 2007.
- [54] Verband Deutscher Ingenieure (VDI). *VDI 3684 - Herstellerneutrale Konfiguration von Antriebssystemen - Beschreibung ereignisgesteuerter Bewegungsabläufe mit Funktionsplänen*. Beuth-Verlag, Berlin, 1997.
- [55] R. Blümel. Entwurf der Bewegungssteuerung dezentral angetriebener Verarbeitungsmaschinen. *SPS IPC DRIVES Nürnberg*, S. 629-637, 2001.
- [56] G.H. Geitner. Die Anwendung der SIMULINK Blockbibliothek Funktionsplan 3.0 (FUP) zum Entwurf dezentraler Bewegungssteuerungen. *Engineering komplexer Automatisierungssysteme - EKA*, S. 381-389, 2001.
- [57] VDI-Fachbereich Produktentwicklung und Mechatronik. *VDI 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Beuth-Verlag, Berlin, 1993.
- [58] VDI-Fachbereich Produktentwicklung und Mechatronik. *VDI 2206 - Entwicklungsmethodik für mechatronische Systeme*. Beuth-Verlag, Berlin, 2004.
- [59] Firma SEW-Eurodrive. *Praxis der Antriebstechnik: Antriebe Projektieren*. 2001.
- [60] VDI-Fachbereich Getriebe und Maschinenelemente. *VDI 2143 - Bewegungsgesetze für Kurvengetriebe*. Beuth-Verlag, Berlin, 1980.

- [61] R. Nolte. Bewegungsdesign mit Polynomsplines. *Bewegungstechnik*, S. 77-86, 2010.
- [62] J. Matthes; A. Heine; M. Berger. Methode zur Fertigung von Zylinderkurvenkörpern räumlicher Kurvengetriebe mit Fräserradienkorrektur. *Bewegungstechnik*, S. 97-110, 2010.
- [63] M. Schulze. *Elektrische Servoantriebe : Baugruppen mechatronischer Systeme; mit Beispielen mit Lösungen*. Fachbuchverl. Leipzig im Carl-Hanser-Verl., München, 2008.
- [64] F.-W. Garbrecht. *Auswahl von Elektromotoren - leicht gemacht : der Weg von der Anwendungsanalyse zum richtig dimensionierten Elektromotor*. VDE Verl., Berlin, 2008.
- [65] C. A. Petri. *Kommunikation mit Automaten*. Dissertation. VDI-Verl., Institut für Instrumentelle Mathematik der Universität Bonn, 1962.
- [66] D. Abel. *Modellbildung und Analyse ereignisorientierter Systeme mit Petri-Netzen*. PhD thesis, Düsseldorf, 1987.
- [67] R. Kruse. *Computational Intelligence : Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Vieweg+Teubner Verlag, Wiesbaden, 2012.
- [68] H. Kiendl. *Fuzzy Control methodenorientiert*. Oldenbourg, München [u.a.], 1997.
- [69] R. König; L. Quäck. *Petri-Netze in der Steuerungs- und Digitaltechnik*. Oldenbourg, München, 1988.
- [70] D. Abel. *Petri-Netze für Ingenieure : Modellbildung und Analyse diskret gesteuerter Systeme*. Springer, Berlin, 1990.
- [71] B. Baumgarten. *Petri-Netze : Grundlagen und Anwendungen*. Spektrum, Akad. Verl., Heidelberg, 1996.
- [72] C. Girault; R. Valk. *Petri nets for systems engineering : a guide to modeling, verification, and applications; with 9 tables*. Springer, Berlin, 2003.
- [73] U. Kiencke; H. Jäkel. *Signale und Systeme*. Oldenbourg-Verl., München, 2008.
- [74] J. Lunze. *Automatisierungstechnik : Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*. Oldenbourg, München, 2008.
- [75] L. Priese; H. Wimmel. *Petri-Netze*. Springer, Berlin, 2008.
- [76] L. Litz. *Grundlagen der Automatisierungstechnik : Regelungssysteme, Steuerungssysteme, Hybride Systeme*. Oldenbourg, München, 2005.
- [77] H. Bandemer; S. Gottwald. *Einführung in Fuzzy-Methoden : Theorie und Anwendungen unscharfer Mengen*. Akademie-Verlag, Berlin, 1992.

- [78] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:498–516, 1965.
- [79] G. Bretthauer; R. Mikut; H.-P. Opitz. *Stabilität von Fuzzy-Regelungen - Eine Übersicht*. VDI, Düsseldorf, 1994.
- [80] J. Jantzen. *Foundations of fuzzy control*. Wiley, Chichester, 2007.
- [81] M. Koch; T. Kuhn; J. Wernstedt. *Fuzzy control : optimale Nachbildung und Entwurf optimaler Entscheidungen*. Oldenbourg, München, 1996.
- [82] J. Adamy. *Fuzzy-Logik, neuronale Netze und evolutionäre Algorithmen*. Berichte aus der Steuerungs- und Regelungstechnik. Shaker, Aachen, 2011.
- [83] R. Mikut. *Modellgestützte on-line Stabilitätsüberwachung komplexer Systeme auf der Basis unscharfer Ljapunov-Funktionen*. VDI-Verl., Düsseldorf, 1999.
- [84] G. Böhme. *Fuzzy-Logik : Einführung in die algebraischen und logischen Grundlagen*. Springer, Berlin, 1993.
- [85] H.-H. Bothe. *Fuzzy logic : Einführung in Theorie und Anwendungen*. Springer, Berlin, 1993.
- [86] D. H. Traeger. *Einführung in die Fuzzy-Logik*. Teubner, Stuttgart, 1994.
- [87] B. Stroustrup. *The design and evolution of C++*. Addison-Wesley, Boston [u.a.], 2007.
- [88] J. Rada-Vilela. *fuzzylite: A fuzzy logic control library written in C++*. <http://www.fuzzylite.com>, 2013.
- [89] P. Cingolani; J. Alcala-Fdez. jfuzzylogic: A robust and flexible fuzzy-logic inference system language implementation. *IEEE World Congress on Computational Intelligence - Brisbane Australia*, 2012.
- [90] Firma Mathworks. *Fuzzy Logic Toolbox*. <http://www.mathworks.com>, 2012.
- [91] C. Blume; W. Jakob. *GLEAM - General Learning Evolutionary Algorithm and Method : ein Evolutionärer Algorithmus und seine Anwendungen*. Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik am Karlsruher Institut für Technologie, 32, KIT Scientific Publishing, Karlsruhe, 2009.
- [92] R. Mikut; F. Hendrich. Produktionsreihenfolgeplanung in Ringwalzwerken mit wissensbasierten und evolutionären Methoden. *Automatisierungstechnische Praxis (atp) 46, Heft 1/98*, 1998.
- [93] J. Sprave. *Evolutionäre Algorithmen zur Parameteroptimierung*. 43, 1995.
- [94] C. Thomalla; O. Eschbach. *Maschinenbelegungsplan mit einem genetischen Algorithmus*. 43, 1995.

- [95] H. Pohlheim. *Evolutionäre Algorithmen : Verfahren, Operatoren und Hinweise für die Praxis*. Springer, Berlin, 2000.
- [96] Firma Lenze. *CamDesigner*. <http://www.lenze.com/produkte/vorgaengerprodukte/engineering-tools/camdesigner>, 2011.
- [97] Firma Bosch. *IndraSize*. <http://www.boschrexroth.com>, 2011.
- [98] Firma Baumüller. *SizeMaxx*. <http://www.baumueller.de/sizemaxx-antriebskonfigurator.htm>, 2011.
- [99] Firma ControlEng. *SERVOsoft*. <http://www.controleng.ca/servosoft/>, 2013.
- [100] U. Simon. Optimierung normierter Bewegungsgesetze. *SPS-Magazin 11*, 2011.
- [101] M. Lorch; G. Bretthauer. Ganzheitliches Engineering von elektrischen Servoantriebssystemen. *SPS/IPC/DRIVES Nürnberg*, 2012.
- [102] M. Lorch; G. Bretthauer. Ein Konzept zum einheitlichen softwarebasierten Engineering komplexer Applikationen verteilter Motion Control Systeme. *Bewegungstechnik*, S. 65-76, 2010.
- [103] F. Pratiwi. *Machbarkeitsanalyse und Umsetzung eines neuartigen Bewegungsgesetzes basierend auf Fuzzy-Logik*. Diplomarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2011.
- [104] D. Dickinson. *Klassifizierung und rechnergestützte Umsetzung von Bewegungsgesetzen nach VDI 2143 für eine neue Engineering Software*. Diplomarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2010.
- [105] M. Ning. *Selektion optimaler Bewegungsvorgaben*. Diplomarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2011.
- [106] J. M. Schmitt. *Entwicklung eines Optimierungs-/Auswahlalgorithmus mithilfe von Fuzzy Logic und/oder Künstlichen Neuronalen Netzen*. Studienarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2011.
- [107] A. Ezust; P. Ezust. *An introduction to design patterns in C++ with Qt*. Prentice Hall open source software development series. Prentice Hall, Upper Saddle River, 2012.
- [108] D. Mol Kentin. *Qt 4 : Einführung in die Applikationsentwicklung*. Open Source Press, München, 2006.
- [109] J. Wolf. *Qt 4, GUI-Entwicklung mit C++*. Galileo Computing. Galileo Press, Bonn, 2009.
- [110] M. Brzyk. *Planung und Entwicklung eines spezifischen Benutzerinterfaces zur Visualisierung und Manipulation von kinematischen Kurvenverläufen*. Studienarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2011.

- [111] B. Rumpe. *Modellierung mit UML : Sprache, Konzepte und Methodik*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [112] B. Oestereich. *Die UML-Kurzreferenz 2.3 für die Praxis*. Oldenbourg-Verl., München, 2009.
- [113] J. Fleischer; J. Wieser; M. Schopp; T. Oesterreicher; M. Zölfl; W. Hahn. Modularisierung komplexer Produktionsanlagen - Analyse von Störgroßen und Verfügbarkeitsbestimmung am Beispiel von Getränkeabfüllanlagen. *wt - Werkstattstechnik, Jahrgang 97 (2007)*, 2007.
- [114] T. Nagel. *Zahnriemengetriebe : Eigenschaften, Normung, Berechnung, Gestaltung*. Hanser, München, 2008.
- [115] R. Neugebauer. *Werkzeugmaschinen : Aufbau, Funktion und Anwendung von spanenden und abtragenden Werkzeugmaschinen*. Springer Vieweg, Berlin, Heidelberg, 2012.
- [116] U. Fischer. *Tabellenbuch Metall*. Verl. Europa-Lehrmittel, Haan-Gruiten, 2008.
- [117] H. Bräutigam; S. Becker. *Richten mit Walzenrichtmaschinen*. Vogel Business Media, 2009.
- [118] N. Bronstejn; K.A. Semendjaev; G. Musiol; H. Mühlig. *Taschenbuch der Mathematik*. Deutsch, Frankfurt am Main, 2008.
- [119] C. K. Apostolides. *Bereitstellung eines Tools zur Berechnung, Visualisierung und Speicherung einfacher Standard-Mechaniken*. Studienarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2012.
- [120] H. Krause. *Umsetzung einer Auslegungsrechnung für elektrische Servoantriebe im Einklang mit einer neuartigen Engineering Software zur Optimierung von Antriebssystemen*. Studienarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2011.
- [121] H. Oertel. *Strömungsmechanik : Grundlagen, Grundgleichungen, Lösungsmethoden, Softwarebeispiele*. Friedr. Vieweg u. Sohn Verlag, GWV Fachverlage GmbH, Wiesbaden, Wiesbaden, 2006.
- [122] G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge mathematical library. Cambridge Univ. Press, Cambridge, 2007.
- [123] M. Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Vieweg+Teubner, Wiesbaden, 2009.
- [124] M. Lorch. *Machbarkeitsanalyse eines neuen Antriebskonzepts für einen Längstaktmontageautomaten mit hoher Taktzahl*. Diplomarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2007.

- [125] Mark Summerfield Jasmin Blanchette. *C++ GUI-Programmierung mit Qt 4 : die offizielle Einführung*. Addison Wesley, München [u.a.], 2007.
- [126] M. Summerfield. *Advanced Qt programming : creating great software with C++ and Qt 4*. Prentice Hall open source software development series. Addison-Wesley, Upper Saddle River, NJ, 2011.
- [127] Firma Sanyo. *Stepping Motor Catalogue*. Sanyo, 2012.
- [128] Firma Robotikhardware. *EMG 30*. <http://www.shop.robotikhardware.de>, 2011.
- [129] A. Kalk. *Entwurf und Aufbau der Mechanik eines vierachsigen Portalsystems als kostengünstiger Versuchsstand*. Studienarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2013.
- [130] Firma Robotikhardware. *RN-Control V1.4*. <http://www.robotikhardware.de/download/rncontrol14.pdf>, 2011.
- [131] D. Tchokonthe. *Aufbau eines mikrocontrollerbasierten mehrachsigen Antriebssystems als Versuchsplattform mit Entwurf von Systemsteuerung und -regelung*. Diplomarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2012.
- [132] Firma Robotikhardware. *MD25 - Dual 12Volt 2.8Amp H Bridge Motor Drive*. <http://www.robot-electronics.co.uk/htm/md25tech.htm>, 2011.
- [133] Firma RTA. *A-NDC Series Drive*. RTA, 2011.
- [134] T. D. Nguyen. *Entwicklung einer Software zur kamerabasierten Analyse einer Labyrinthstruktur*. Studienarbeit, Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2011.

9 Anhang

9.1 Zur Erweiterung der VDI 2143 um die dritte Ableitung

Die folgenden Gleichungen beschreiben den normierten Ruck $r(t)$ in Abhängigkeit der normierten Zeit $t \in [0; 1]$ und teilweise abhängig von λ . Es handelt sich um die dritte Ableitung der Funktionen für den Weg.

Die notierten Funktionen bilden eine Erweiterung zur VDI-Richtlinie 2143 [60] und sind im Rahmen der Entwicklung des neuen Konzepts hergeleitet worden. In der Richtlinie sind nur Ableitungen bis zum zweiten Grade notiert, was den gegebenen Ansprüchen der neuen Methoden nicht genügt. So muss der Ruckverlauf für einen ruckoptimalen Entwurf bekannt sein. Die Ableitungen sind in die Algorithmen zum ganzheitlichen Engineering von Servoantriebssystemen integriert und in der Implementierung verwendet sowie durch vielfache Anwendung auf Korrektheit geprüft worden.

Die Auflistung der Gleichungen erfolgt nach den in der VDI-Richtlinie enthaltenen Funktionstypen und nach den Kombinationen aus diskreten Randbedingungen, auf welche sie anwendbar sind (siehe auch Abschnitt 3.1.1).

Einfache Sinuslinie (esin)

Für alle festgelegten Kombinationen von Randbedingungen:

$$r(t) = -\frac{1}{2}\pi^3 \sin(\pi t) \quad (9.1)$$

Bemerkung: Der Ruckverlauf beim Einsatz des Einfachen Sinus beginnt und endet mit einem Dirac-Impuls. Der Sachverhalt ist nicht analytisch abgebildet und muss gegebenenfalls, wie in der Implementierung zur vorliegenden Arbeit, numerisch umgesetzt werden.

Geneigte Sinuslinie (gsin)

Für alle festgelegten Kombinationen von Randbedingungen:

$$r(t) = 4\pi^2 \cos(2\pi t) \quad (9.2)$$

Harmonische Kombination (hkom)

Für Rast in Umkehr bzw. Umkehr in Rast:

$$r(t) = \begin{cases} \frac{2c_{af}\pi}{\lambda} \cos\left(\frac{2\pi t}{\lambda}\right) & \text{für } t \leq \frac{\lambda}{4} \\ -\frac{2c_{af}\pi}{3\lambda} \sin\left(\frac{2\pi}{3\lambda}(t - \frac{1}{4}\lambda)\right) & \text{für } \frac{\lambda}{4} < t \leq \lambda \\ \frac{c_{aSf}\pi}{2-2\lambda} \cos\left(\frac{\pi}{2-2\lambda}(t - \lambda)\right) & \text{für } \lambda < t \leq 1 \end{cases} \quad (9.3)$$

mit:

$$c_{aSf} = \frac{1}{-0.40528 + 0.44877\lambda - 0.04379\lambda^2} \quad (9.4)$$

$$c_{af} = -\frac{c_{aSf}(1 - \lambda)}{\lambda}$$

Für Umkehr in Rast bzw. Rast in Umkehr:

$$r(t) = \begin{cases} -\frac{2c_{af}\pi}{\lambda} \cos\left(\frac{2\pi}{\lambda}(1 - t)\right) & \text{für } t \leq \frac{\lambda}{4} \\ \frac{2c_{af}\pi}{3\lambda} \sin\left(\frac{2\pi}{3\lambda}(1 - t - \frac{1}{4}\lambda)\right) & \text{für } \frac{\lambda}{4} < t \leq \lambda \\ -\frac{c_{aSf}\pi}{2-2\lambda} \cos\left(\frac{\pi}{2-2\lambda}(1 - t - \lambda)\right) & \text{für } \lambda < t \leq 1 \end{cases} \quad (9.5)$$

mit:

$$c_{aSf} = \frac{1}{-0.40528 + 0.44877\lambda - 0.04379\lambda^2} \quad (9.6)$$

$$c_{af} = -\frac{c_{aSf}(1 - \lambda)}{\lambda}$$

Für Geschwindigkeit in Umkehr bzw. Geschwindigkeit in Umkehr:

$$r(t) = \begin{cases} \frac{2c_{af}\pi}{\lambda} \cos\left(\frac{2\pi t}{\lambda}\right) & \text{für } t \leq \frac{\lambda}{4} \\ -\frac{2c_{af}\pi}{3\lambda} \sin\left(\frac{2\pi}{3\lambda}(t - \frac{1}{4}\lambda)\right) & \text{für } \frac{\lambda}{4} < t \leq \lambda \\ \frac{c_{aSf}\pi}{2-2\lambda} \cos\left(\frac{\pi}{2-2\lambda}(t - \lambda)\right) & \text{für } \lambda < t \leq 1 \end{cases} \quad (9.7)$$

mit:

$$c_{aSf} = \frac{-2\pi^2 \left(1 - v_0\lambda \left(1 - \frac{\pi+4}{4\pi}\right)\right)}{(1-\lambda)(8+\lambda(\pi-4))} \quad (9.8)$$

$$c_{af} = \frac{1}{\lambda} \left(-c_{aSf}(1-\lambda) - \frac{1}{2}v_0\pi\right)$$

Für Umkehr in Geschwindigkeit bzw. Geschwindigkeit in Umkehr:

$$r(t) = \begin{cases} -\frac{2c_{aSf}\pi}{\lambda} \cos\left(\frac{2\pi}{\lambda}(1-t)\right) & \text{für } t \leq \frac{\lambda}{4} \\ -\frac{2c_{aSf}\pi}{3\lambda} \sin\left(\frac{2M_{PI}}{3\lambda}(t - \frac{1}{4}\lambda)\right) & \text{für } \frac{\lambda}{4} < t \leq \lambda \\ \frac{c_{aSf}\pi}{2-2\lambda} \cos\left(\frac{\pi}{2-2\lambda}(t-\lambda)\right) & \text{für } \lambda < t \leq 1 \end{cases} \quad (9.9)$$

mit:

$$c_{aSf} = \left| \frac{-2\pi^2 \left(1 - v_z\lambda \left(1 - \frac{\pi+4}{4\pi}\right)\right)}{(1-\lambda)(8+\lambda(\pi-4))} \right| \quad (9.10)$$

$$c_{af} = \frac{-c_{aSf}(1-\lambda) - \frac{1}{2}v_z\pi}{\lambda}$$

Modifiziertes Beschleunigungstrapez (trap)

Für Rast in Rast:

$$r(t) = 32 \frac{\pi^2}{\pi+2} \cdot \begin{cases} \cos(4\pi t) & \text{für } t \leq \frac{1}{8} \\ 0 & \text{für } \frac{1}{8} < t \leq \frac{3}{8} \\ \cos\left(4\pi\left(t - \frac{1}{4}\right)\right) & \text{für } \frac{3}{8} < t \leq \frac{1}{2} \\ \cos\left(4\pi\left(\frac{3}{4} - t\right)\right) & \text{für } \frac{1}{2} < t \leq \frac{5}{8} \\ 0 & \text{für } \frac{5}{8} < t \leq \frac{7}{8} \\ \cos(4\pi(1-t)) & \text{für } \frac{7}{8} < t \leq 1 \end{cases} \quad (9.11)$$

Für Umkehr in Rast bzw. Rast in Umkehr:

$$r(t) = 2c_{af} \cdot \begin{cases} -\frac{\pi}{\lambda} \cos\left(\frac{2\pi}{\lambda}(1-t)\right) & \text{für } t \leq \frac{1}{4}\lambda \\ 0 & \text{für } \frac{1}{4}\lambda < t \leq \frac{3}{4}\lambda \\ \frac{\pi}{\lambda} \sin\left(\frac{2\pi}{\lambda}\left(1-t-\frac{3}{4}\lambda\right)\right) & \text{für } \frac{3}{4}\lambda < t \leq \lambda \\ -\frac{\pi}{1-\lambda} \cos\left(\frac{2\pi}{1-\lambda}(1-t-\lambda)\right) & \text{für } \lambda < t \leq \frac{1}{4}(3\lambda+1) \\ 0 & \text{für } \frac{1}{4}(3\lambda+1) < t \leq 1 \end{cases} \quad (9.12)$$

mit:

$$c_{aSf} = \left| \frac{1}{-0.49408 + 0.53358\lambda - 0.03950\lambda^2} \right| \quad (9.13)$$

$$c_{af} = \frac{-c_{aSf}(-1.11102)(1-\lambda)}{\lambda}$$

Modifizierte Sinuslinie (msin)

Für Umkehr in Rast bzw. Rast in Umkehr:

$$r(t) = \begin{cases} \frac{16\pi^3}{4+\pi} \cos(4\pi t) & \text{für } t \leq \frac{1}{8}\lambda \\ \frac{16\pi^3}{3(4+\pi)} \cos\left(\frac{1}{3}\pi(1+4t)\right) & \text{für } \frac{1}{8}\lambda < t \leq \frac{7}{8}\lambda \\ \frac{16\pi^3}{4+\pi} \cos(4\pi t) & \text{für } \frac{1}{4}(3\lambda+1) < t \leq 1 \end{cases} \quad (9.14)$$

Für alle sonstigen Kombinationen von Randbedingungen:

$$r(t) = \begin{cases} \frac{2c_{af}\pi}{\lambda} \cos(2\pi\lambda t) & \text{für } t \leq \frac{1}{4}\lambda \\ \frac{2c_{af}\pi}{\lambda} \cos\left(\frac{2\pi}{3\lambda}\left(t+\frac{1}{2}\lambda\right)\right) & \text{für } \frac{1}{4}\lambda < t \leq \lambda \\ -\frac{2c_{aSf}\pi}{3-3\lambda} \cos\left(\frac{2\pi}{3-3\lambda}\left(t+\frac{3}{2}-\frac{5}{2}\lambda\right)\right) & \text{für } \lambda < t \leq \frac{1}{4}(3+\lambda) \\ -\frac{2c_{aSf}\pi}{1-\lambda} \cos\left(\frac{2\pi}{1-\lambda}(t-\lambda)\right) & \text{für } \frac{1}{4}(3+\lambda) < t \leq 1 \end{cases} \quad (9.15)$$

mit:

$$c_{aSf} = -\frac{2\pi^2 \left(1 - v_z - ((v_0 - v_z) \left(\lambda \left(\frac{3}{4} - \frac{1}{\pi}\right)\right))\right)}{(1 - \lambda)(\pi + 4)} \quad (9.16)$$

$$c_{af} = (v_z - v_0) \frac{\pi}{2\lambda} - c_{aSf} \frac{1 - \lambda}{\lambda}$$

Polynom 5. Ordnung (p5)

Für Umkehr in Rast bzw. Rast in Umkehr:

$$r(t) = 60 - 360t + 360t^2 \quad (9.17)$$

Für alle sonstigen Kombinationen von Randbedingungen:

$$r(t) = 6A_3 + 24A_4t + 60A_5t^2 \quad (9.18)$$

mit:

$$\begin{aligned} K_1 &= 1 - v_0 - 0,5a_0 & (9.19) \\ K_2 &= v_z - v_0 - a_0 \\ K_3 &= a_z - a_0 \\ A_3 &= 10 * K_1 - 4K_2 + 0,5K_3 \\ A_4 &= -15K_1 + 7K_2 - K_3 \\ A_5 &= 6K_1 - 3K_2 + 0,5K_3 \end{aligned}$$

Sinus-Gerade-Kombination (sing)

Für alle festgelegten Kombinationen von Randbedingungen:

$$r(t) = \begin{cases} -\frac{c_{af}\pi}{4\lambda} \sin\left(\frac{\pi}{4\lambda}t\right) & \text{für } t \leq \frac{1}{2}\lambda \\ 0 & \text{für } \frac{1}{2}\lambda < t \leq \frac{1}{2}(1 + \lambda) \\ -\frac{a_{af}\lambda}{2(1-\lambda)^2\frac{\pi}{2}} \cos\left(\frac{\pi}{4(1-\lambda)}\left(t - 1 + \frac{1}{2}(1 - \lambda)\right)\right) & \text{für } \frac{1}{2}(1 + \lambda) < t \leq 1 \end{cases} \quad (9.20)$$

9.2 Zur Ausarbeitung der Fuzzy-Regelsätze

Im Folgenden werden die kompletten Regelsätze, die zur CI-Selektion entwickelt sind aufgelistet.

Im Abschnitt 1.2.5 wird erläutert, wie linguistische Terme unscharfes Vorwissen abbilden können. Es wird auch gezeigt, wie daraus mittels Fuzzy-Logic Algorithmen zur Auswertung von Daten erzeugbar sind.

In der vorliegenden Arbeit ist über zwei Regelsätze eine multikriterielle Selektion einer Bewegung aus einer Mannigfaltigkeit realisiert. Das neu entwickelte Verfahren ist in Abschnitt 3.2.3 beschrieben. Die dargestellten Regeln sind in die neue Software zum Engineering von Antriebssystemen implementiert.

9.2.1 Erster Regelsatz (Bewertung einer Bewegung)

Der erste Regelsatz beschreibt welchen Ausprägungen von Kennwerten einer Bewegung, sprich Geschwindigkeit, Beschleunigung und Ruck, bestimmte Ausprägungen von Eigenschaften zugeordnet werden können. Bei den Eigenschaften handelt es sich um drei Optimierungskriterien. Im Fall der Bewertung, wie zeitoptimal eine Bewegung ist, geht es darum, das Potenzial für eine Verkürzung der Taktzeit festzustellen. Daher ist beispielsweise die erste Regel so zu interpretieren, dass wenn die Bewegung eine vergleichsweise kleine Geschwindigkeit und eine mittelmäßige Beschleunigung besitzt, dann eignet sie sich besonders für Zeitverkürzungen, da sie einen größeren Puffer zu den kinematischen Restriktionen besitzt.

Bei den Regeln zum Energieverbrauch und Ruck wird wie in der siebten Regel argumentiert. Wenn die Geschwindigkeit groß ist aber die Beschleunigung klein, dann besitzt die Bewegung eine hohe Grundgeschwindigkeit und es wird wenig Energie für Beschleunigungen und Bremsvorgänge aufgewendet. Die Bewegung ist demnach eher energieoptimal. Für die folgenden Regeln kann in ähnlicher Weise argumentiert werden.

Für den ersten Regelsatz wurden insgesamt 17 solcher Regeln aufgestellt, die nachfolgend zusammengestellt sind und das vorhanden Vorwissen abbilden:

1. Regel¹: WENN Geschwindigkeit ↓ UND Beschleunigung ↕ DANN
Zeitoptimal ↑ ↑
2. Regel: WENN Geschwindigkeit ↓ UND Beschleunigung ↓ DANN
Zeitoptimal ↑
3. Regel: WENN Geschwindigkeit ↕ UND Beschleunigung ↕ DANN
Zeitoptimal ↑

¹↓=klein; ↕=neutral; ↑=groß; ↑↑=sehr groß

4. Regel: WENN Geschwindigkeit \downarrow UND Beschleunigung \downarrow DANN Zeitoptimal
5. Regel: WENN Geschwindigkeit \uparrow DANN Zeitoptimal \downarrow
6. Regel: WENN Geschwindigkeit \downarrow UND Beschleunigung \downarrow DANN Energieoptimal \uparrow
7. Regel: WENN Geschwindigkeit \uparrow UND Beschleunigung \downarrow DANN Energieoptimal \uparrow
8. Regel: WENN Geschwindigkeit \downarrow UND Beschleunigung \downarrow DANN Energieoptimal \uparrow
9. Regel: WENN Geschwindigkeit \uparrow UND Beschleunigung \downarrow DANN Energieoptimal \uparrow
10. Regel: WENN Geschwindigkeit \downarrow UND Beschleunigung \uparrow DANN Energieoptimal \downarrow
11. Regel: WENN Geschwindigkeit \downarrow UND Beschleunigung \uparrow DANN Energieoptimal \downarrow
12. Regel: WENN Geschwindigkeit \downarrow UND Beschleunigung neutral DANN Energieoptimal \downarrow
13. Regel: WENN Ruck \downarrow UND Beschleunigung \uparrow DANN Ruckoptimal $\uparrow \uparrow$
14. Regel: WENN Ruck \downarrow UND Beschleunigung \downarrow DANN Ruckoptimal \uparrow
15. Regel: WENN Ruck \downarrow UND Beschleunigung neutral DANN Ruckoptimal \uparrow
16. Regel: WENN Beschleunigung \downarrow UND Geschwindigkeit \uparrow DANN Ruckoptimal \uparrow
17. Regel: WENN Ruck \uparrow DANN Ruckoptimal \downarrow

9.2.2 Zweiter Regelsatz (Affinität einer Bewegung)

Die im ersten Regelsatz festgestellten Eigenschaften werden im zweiten Regelsatz mit den Optimierungsvorgaben, die vom Benutzer der Algorithmen gestellt werden, in Verbindung gebracht. Wenn beispielsweise, wie in der ersten Regel, im ersten Regelsatz das Kriterium für energieoptimale Bewegungen als hoch erfüllt bestimmt ist und der Benutzer die Vorgaben zu einem hoch energieoptimalen Entwurf gesetzt hat, dann besitzt die Bewegung eine hohe Affinität zum

Anwenderwunsch und ist somit hoch geeignet. Die folgenden Regeln sind in gleicher Weise zu interpretieren.

Hier wurden 12 Regeln entworfen um alle Zusammenhänge zu beschreiben:

1. **Regel:** WENN Kriterium Energieoptimal \uparrow UND Energieoptimal \uparrow
DANN Affinität \uparrow
2. **Regel:** WENN Kriterium Energieoptimal \uparrow UND Energieoptimal \downarrow
DANN Affinität \downarrow
3. **Regel:** WENN Kriterium Energieoptimal \updownarrow UND (Energieoptimal \updownarrow
ODER Energieoptimal \uparrow) DANN Affinität \uparrow
4. **Regel:** WENN Kriterium Energieoptimal \uparrow UND Energieoptimal \updownarrow
DANN Affinität \updownarrow
5. **Regel:** WENN Kriterium Ruckoptimal \uparrow UND Ruckoptimal \uparrow DANN
Affinität \uparrow
6. **Regel:** WENN Kriterium Ruckoptimal \uparrow UND Ruckoptimal \downarrow DANN
Affinität \downarrow
7. **Regel:** WENN Kriterium Ruckoptimal \updownarrow UND (Ruckoptimal \updownarrow ODER
Ruckoptimal \uparrow) DANN Affinität \uparrow
8. **Regel:** WENN Kriterium Ruckoptimal \uparrow UND Ruckoptimal \updownarrow DANN
Affinität \updownarrow
9. **Regel:** WENN Kriterium Zeitoptimal \uparrow UND Zeitoptimal \uparrow DANN
Affinität \uparrow
10. **Regel:** WENN Kriterium Zeitoptimal \uparrow UND Zeitoptimal \downarrow DANN
Affinität \downarrow
11. **Regel:** WENN Kriterium Zeitoptimal \updownarrow UND (Zeitoptimal \updownarrow ODER
Zeitoptimal \uparrow) DANN Affinität \uparrow
12. **Regel:** WENN Kriterium Zeitoptimal \uparrow UND Zeitoptimal \updownarrow DANN
Affinität \updownarrow

9.3 Zur Optimierung der Modellmaschine

Im Folgenden werden die Zahlenwerte für die Optimierung der Modellmaschine aus Abschnitt 6.4.2 aufgeführt. Es handelt sich um die drei für die Optimierung des Systems wesentlichen Teilbewegungen, die Vorschubbewegung sowie das Öffnen und Schließen der Fixierung, die abschnittsweise vorgestellt werden. Tabellarisch dargestellt sind die Anfangs- und Endzeit t des Abschnitts, Anfangs- und Zielzustand \vec{x} , die Zeitdifferenz T und die kinematischen Werte der optimierten Bewegung für drei Optimierungsvorgaben. Dabei handelt es sich um den energiesparenden „E“, den ruckoptimalen „R“ und den zeitoptimalen „Z“ Entwurf, zu denen in der Tabelle die selektierte Bewegung und die zugehörigen Diagramme notiert sind. Die Diagramme enthalten jeweils die Scharen von Bewegungsvorgängen, aus denen im letzten Optimierungsschritt eine Bewegung ausgewählt worden ist. Die Diagramme enthalten zudem, falls im Darstellungsbereich vorhanden, die kinematischen Restriktion in Form von horizontalen Linien („Min.“/„Max.“). Die kompletten resultierenden Kurvenverläufe sind im Haupttext unter Abschnitt 6.4.2 zu finden. Die verwendeten neuen Methoden zur Optimierung sind in Abschnitt 5.2 und die benutzte Einfache Selektion wird in Abschnitt 3.2.2 hergeleitet.

Optimierungen der Vorschubbewegung

Folgende Tabelle und Abbildungen enthalten die detaillierte Darstellung der Vorschubbewegung. Die Diagramme zeigen, dass die ruckoptimale Bewegung besonders weiche Verläufe im Gegensatz zur energieoptimalen Bewegung und besonders zur zeitoptimalen Bewegung besitzt, welche auf Kosten hohen Rucks eine sehr kurze Zeitdifferenz ermöglicht.

	E	R	Z
$t_1 [s]$	13,225	13,200	12,025
$\vec{x}_1 \begin{bmatrix} mm \\ mm s^{-1} \\ mm s^{-2} \end{bmatrix}$	$\begin{pmatrix} 50 \\ 0 \\ 0 \end{pmatrix}$		
$t_2 [s]$	15,675	15,630	14,200
$\vec{x}_2 \begin{bmatrix} mm \\ mm s^{-1} \\ mm s^{-2} \end{bmatrix}$	$\begin{pmatrix} 250 \\ 0 \\ 0 \end{pmatrix}$		
$T_1 [s]$	2,450	2,430	2,175
$K_B \begin{bmatrix} mm & mm \\ mm s^{-1} & mm s^{-1} \\ mm s^{-2} & mm s^{-2} \\ mm s^{-3} & mm s^{-3} \end{bmatrix}$	$\begin{pmatrix} 50 & 250 \\ 0 & 144 \\ -184 & 184 \\ -314 & 942 \end{pmatrix}$	$\begin{pmatrix} 50 & 250 \\ 0 & 165 \\ -212 & 212 \\ -550 & 550 \end{pmatrix}$	$\begin{pmatrix} 50 & 250 \\ 0 & 145 \\ -211 & 211 \\ -308 & 7045 \end{pmatrix}$
Selektion	msin	gsin	esin
Abb.	9.1	9.2	9.3

Tabelle 9.1: Optimierung der Vorschubbewegung

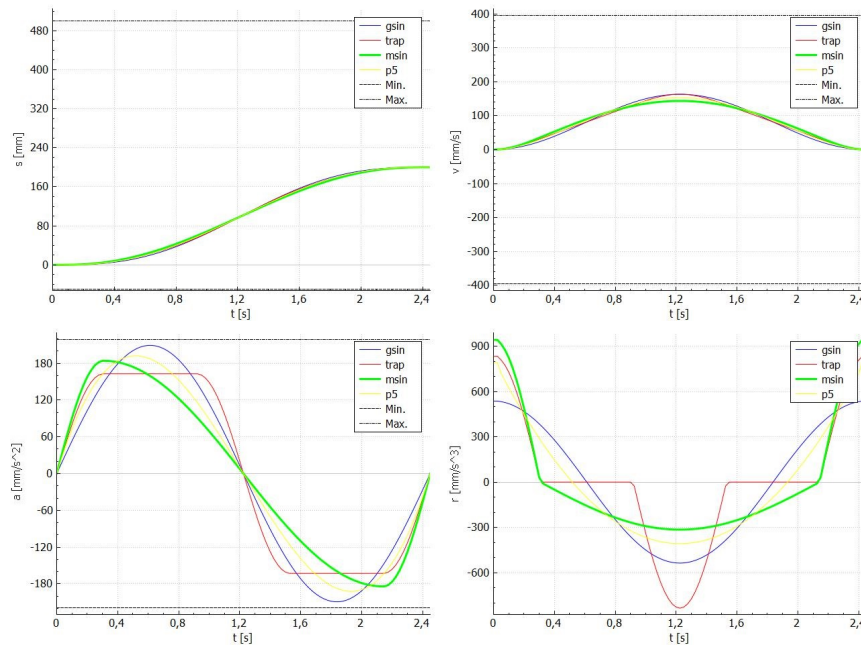


Abbildung 9.1: Energieoptimale Vorschubbewegung

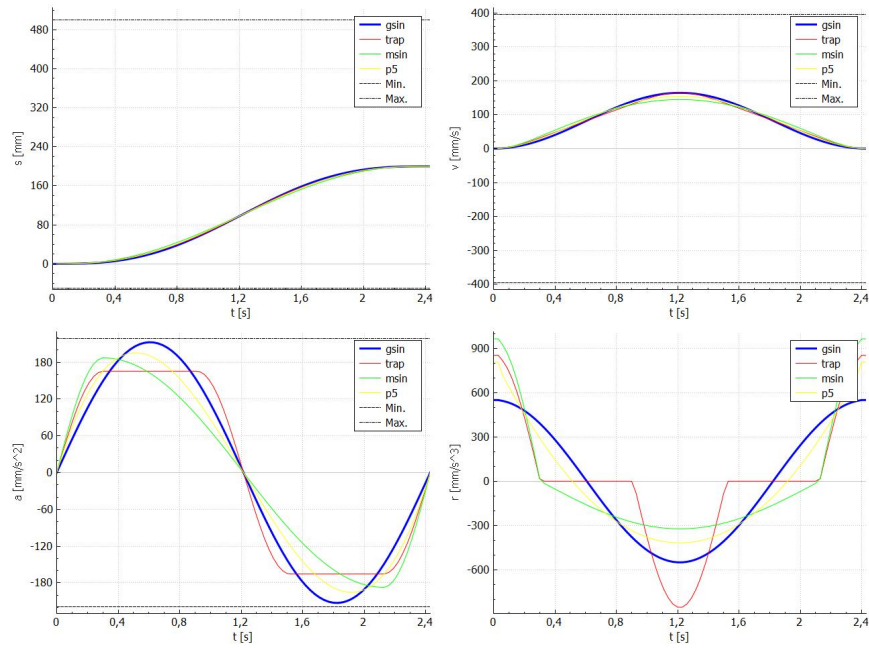


Abbildung 9.2: Ruckoptimale Vorschubbewegung

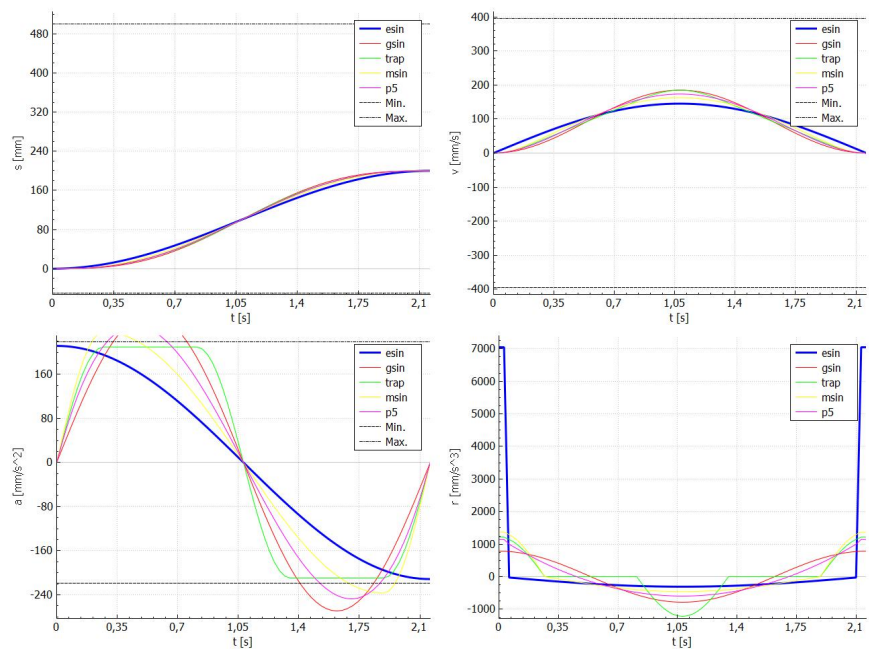


Abbildung 9.3: Zeitoptimale Vorschubbewegung

Optimierungen der Fixierung (Öffnen)

Die Bewegung der Optimierung zeigt einen qualitativ gleichen Verlauf wie die Vorschubbewegung, weil es sich um die gleichen diskreten Randbedingungen handelt.

	E	R	Z
$t_1[s]$	12,000	11,975	10,925
$\vec{x}_1 \begin{bmatrix} mm \\ mms^{-1} \\ mms^{-2} \end{bmatrix}$	$\begin{pmatrix} 50 \\ 0 \\ 0 \end{pmatrix}$		
$t_2[s]$	13,225	13,200	12,025
$\vec{x}_2 \begin{bmatrix} mm \\ mms^{-1} \\ mms^{-2} \end{bmatrix}$	$\begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix}$		
$T_1[s]$	1,225	1,225	1,100
$\mathbf{K}_B \begin{bmatrix} mm & mm \\ mms^{-1} & mms^{-1} \\ mms^{-2} & mms^{-2} \\ mms^{-3} & mms^{-3} \end{bmatrix}$	$\begin{pmatrix} 50 & 100 \\ 0 & 72 \\ -184 & 184 \\ -630 & 1869 \end{pmatrix}$	$\begin{pmatrix} 50 & 100 \\ 0 & 81 \\ -208 & 208 \\ -1058 & 1057 \end{pmatrix}$	$\begin{pmatrix} 50 & 100 \\ 0 & 71 \\ -200 & 200 \\ -566 & 6651 \end{pmatrix}$
Selektion	msin	gsin	esin
Abb.	9.4	9.5	9.6

Tabelle 9.2: Optimierung der Fixierung (Öffnen)

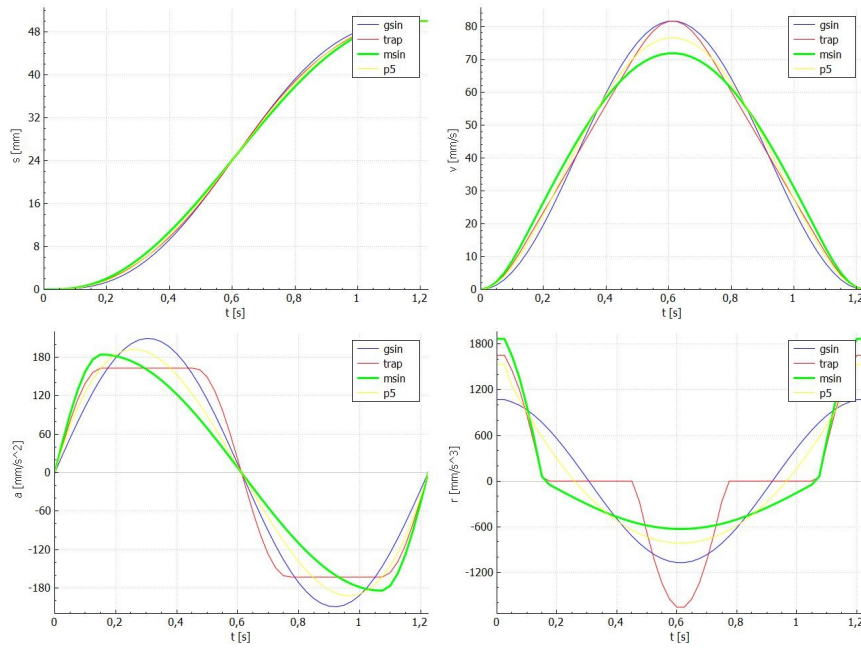


Abbildung 9.4: Energieoptimale Öffnung der Fixierung

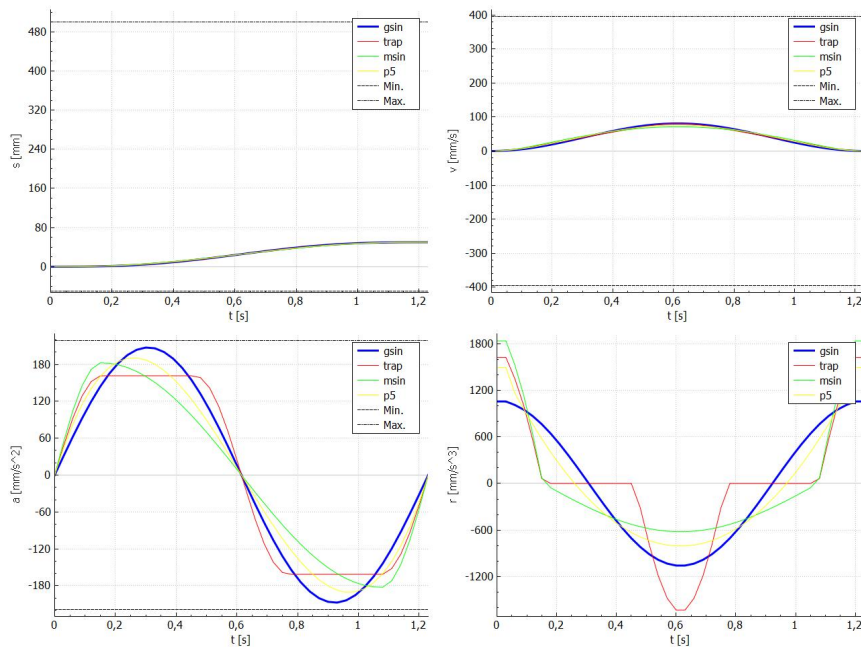


Abbildung 9.5: Ruckoptimale Öffnung der Fixierung

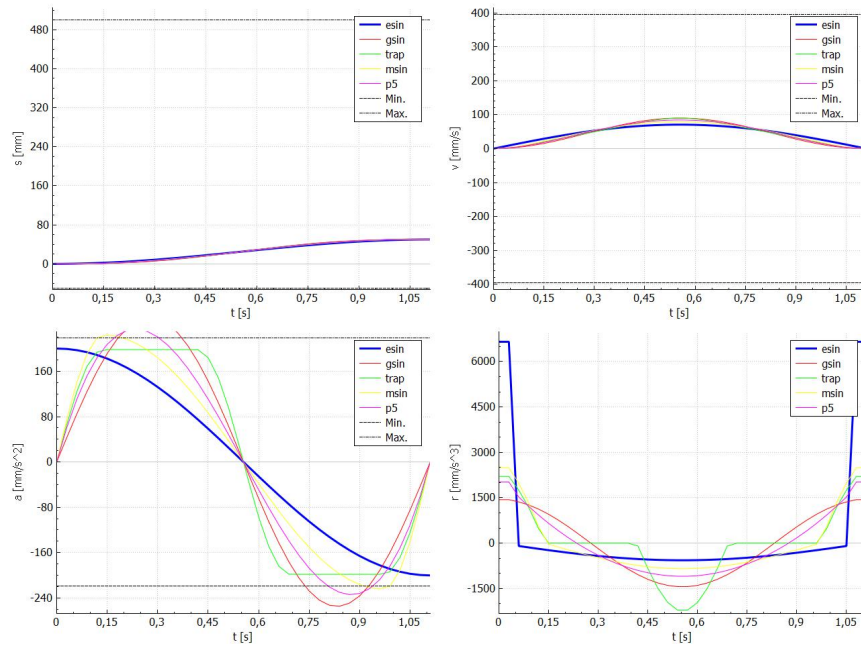


Abbildung 9.6: Zeitoptimale Öffnung der Fixierung

Optimierungen der Fixierung (Schließen)

Die Schließung der Fixierung zeigt einen prinzipiell zur Öffnung gespiegelten Verlauf. Die genauen Zahlen und Kurvenverläufe können ebenfalls der Tabelle und den Abbildungen entnommen werden.

	E	R	Z
$t_3[s]$	15,675	15,625	14,200
$\vec{x}_3 \begin{bmatrix} mm \\ mms^{-1} \\ mms^{-2} \end{bmatrix}$	$\begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix}$		
$t_4[s]$	16,875	16,825	15,275
$\vec{x}_4 \begin{bmatrix} mm \\ mms^{-1} \\ mms^{-2} \end{bmatrix}$	$\begin{pmatrix} 50 \\ 0 \\ 0 \end{pmatrix}$		
$T_3[s]$	1,200	1,200	1,075
$\mathbf{K}_B \begin{bmatrix} mm & mm \\ mms^{-1} & mms^{-1} \\ mms^{-2} & mms^{-2} \\ mms^{-3} & mms^{-3} \end{bmatrix}$	$\begin{pmatrix} 50 & 100 \\ -73 & 0 \\ -192 & 192 \\ -1987 & 670 \end{pmatrix}$	$\begin{pmatrix} 50 & 100 \\ -83 & 0 \\ -218 & 218 \\ -1138 & 1138 \end{pmatrix}$	$\begin{pmatrix} 50 & 100 \\ -73 & 0 \\ -212 & 212 \\ -7025 & 615 \end{pmatrix}$
Selektion	msin	gsin	esin
Abb.	9.7	9.8	9.9

Tabelle 9.3: Optimierung der Fixierung (Schließen)

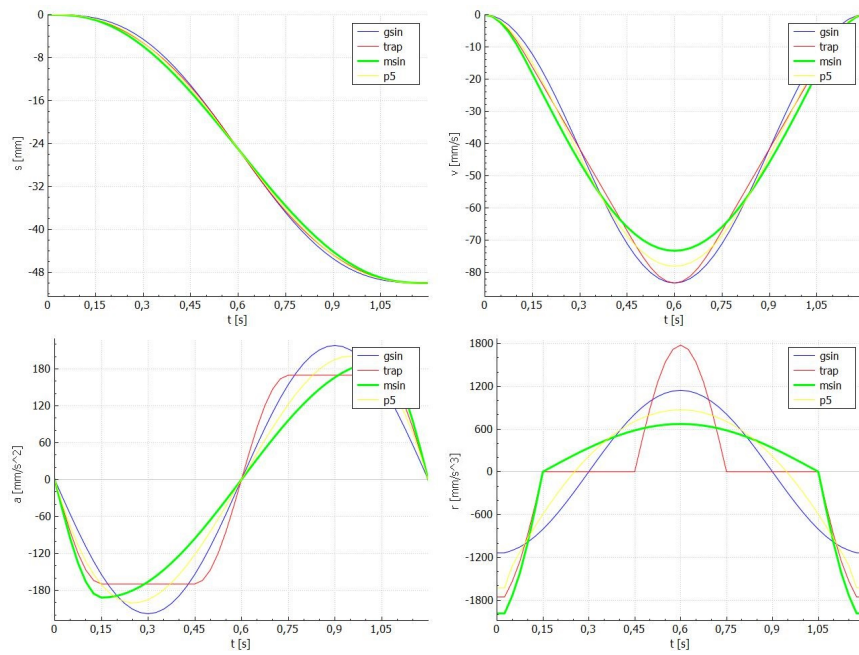


Abbildung 9.7: Energieoptimale Schließung der Fixierung

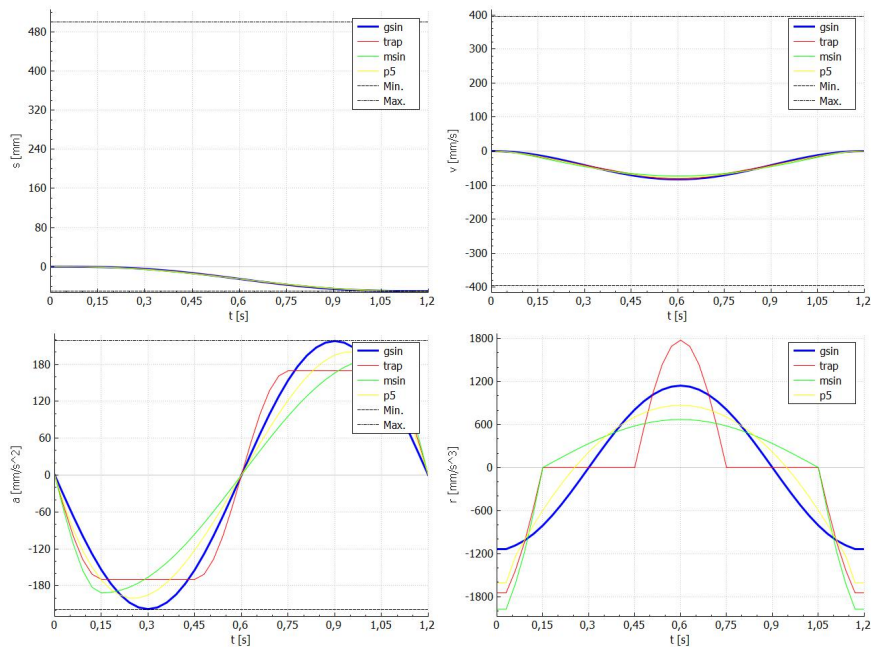


Abbildung 9.8: Ruckoptimale Schließung der Fixierung

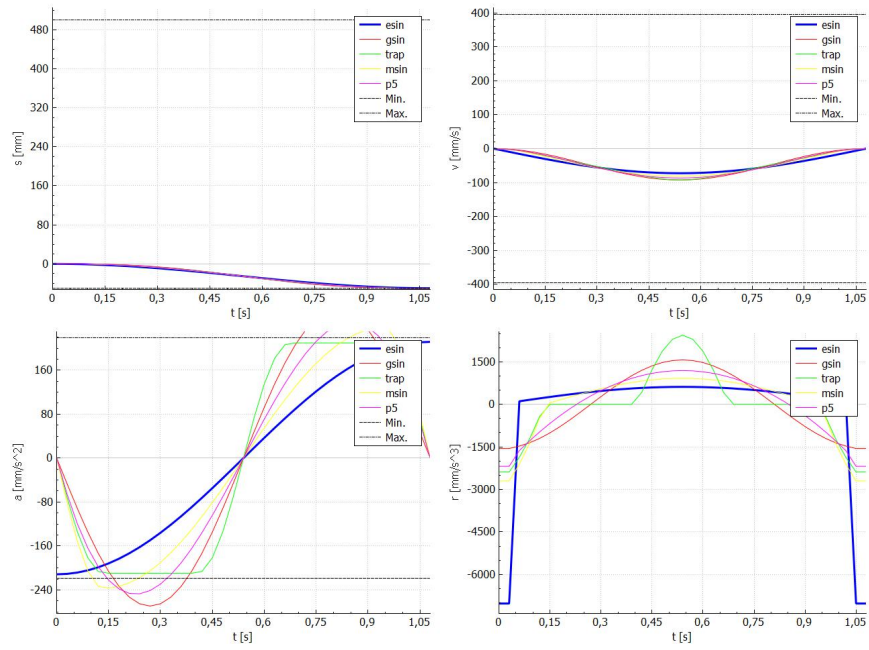


Abbildung 9.9: Zeitoptimale Schließung der Fixierung

Symbolverzeichnis

Das Symbolverzeichnis enthält die wichtigsten Bezeichnungen nach Art des bezeichneten Wertes und dem Auftreten im Text geordnet. Es ist zu beachten, dass Bezeichnungen je nach Kapitel und Kontext diverse Bedeutungen zukommen können.

zu Kapitel 1

Symbol	Bezeichnung
s	Strecke
v	Geschwindigkeit
a	Beschleunigung
r	Ruck
F	Kraft
m	Masse
t	Zeit
R	Rast
U	Umkehr
G	konstante Geschwindigkeit
B	beschleunigte Bewegung
\bar{M}_{Motor}	Nenndauerdrehmoment
$M_{Motor}(n)$	maximales Motordrehmoment
n_{Motor}	Nennzahl
J_{Motor}	Massenträgheit des Motors
n_{max}	maximale Drehzahl
M_{max}	maximales Drehmoment
M	Drehmomentverlauf
M_s	statisches Drehmoment
M_d	dynamisches Drehmoment
\tilde{M}_R	verallgemeinertes Reibmoment
ω	Winkelgeschwindigkeit
M_t	treibendes Moment
M_b	bremsendes Moment
η	Wirkungsgrad
$\tilde{\eta}$	abhängiger Wirkungsgrad
\bar{M}	effektives Drehmoment
\bar{n}	mittlere Drehzahl
I_S	Bemessungsstrom
$I_{S,max}$	Maximalstrom

Symbol	Bezeichnung
$I_{S,G}$	Grundlaststrom
I_{eff}	effektiver Motorstrom
\mathcal{N}	Petrinetz
\mathcal{P}	Stellen
\mathcal{T}	Transitionen
\mathcal{F}	Präkanten
\mathcal{W}	Postkanten
$\vec{t}_S(k)$	Schaltvektor
$\vec{p}(k)$	markierte Stellen
\mathbf{N}	Netzmatrix
μ	Zugehörigkeitsfunktion

zu Kapitel 2

Symbol	Bezeichnung
\mathcal{W}	Ausgangssignale Systementwurf entspricht Eingangssignale Bewegungsentwurf
\mathcal{U}	Ausgangssignale Bewegungsentwurf entspricht Eingangssignale Hardwareentwurf
\mathcal{Y}	Ausgangssignale Hardwareentwurf entspricht Eingangssignale Systementwurf
\mathbf{B}_{hyp}	hypothetische Bewegung
\mathbf{B}_{hyp}^{ges}	hypothetische Gesamtbewegung (numerisch)

zu Kapitel 3

Symbol	Bezeichnung
\mathcal{F}	Bewegungsgesetze
\mathcal{F}_A	geeignete Bewegungsgesetze
\mathcal{A}	Fähigkeiten eines Bewegungsgesetzes
\mathbf{A}	mögliche diskrete Randbedingungen
esin	Einfache Sinuslinie (nach VDI)
gsin	Geneigte Sinuslinie (nach VDI)
hkom	Harmonische Sinuslinie (nach VDI)
trap	Modifiziertes Beschleunigungstrapez (nach VDI)
msin	Modifizierte Sinuslinie (nach VDI)
p5	Polynom 5. Grad (nach VDI)
sing	Sinus Gerade Kombination (nach VDI)

Symbol	Bezeichnung
fuz	Fuzzy Bewegungsgesetz (neu)
Δs	relativer Verfahrensweg
s_0	Startposition
s_z	Zielposition
v_0	Startgeschwindigkeit
v_z	Zielgeschwindigkeit
a_0	Startbeschleunigung
a_z	Zielbeschleunigung
\vec{x}_0	Startbedingungen
\vec{x}_z	Zielbedingungen
\vec{x}_{RN}	normierte Randbedingungen
\vec{x}_D	diskrete Randbedingungen
ϵ_S	zulässiger Schleppfehler
λ	Kurvenparameter
B	Bewegung (numerisch)
\mathbf{B}_N	normierte Bewegung (numerisch)
\mathbf{B}_{Schar}	Schar von Bewegungen (numerisch)
$\mathbf{B}_{Schar,K}$	kinematisch selektierte Schar von Bewegungen (numerisch)
K	kinematische Restriktionen
\mathbf{K}_{ultimo}	ultimative kinematische Restriktionen
V_E	Verlustfunktion Energie
V_R	Verlustfunktion Ruck
V_t	Verlustfunktion Zeit
\bar{v}_{max}	Geschwindigkeitskennwert
\bar{a}_{max}	Beschleunigungskennwert
\bar{r}_{max}	Ruckkennwert
\vec{o}	Optimierungsfaktoren
μ	Zugehörigkeitsfunktion
g	Güte

zu Kapitel 4

Symbol	Bezeichnung
\mathcal{M}	Modul Standardmechanik
\mathcal{S}	Antriebsstrang
$\tilde{\eta}$	abhängiger Wirkungsgrad
\bar{M}_{stat}	statisches Moment
\bar{M}_d	dynamisches Moment
J	Massenträgheit
ω	Winkelgeschwindigkeit

Symbol	Bezeichnung
η	Wirkungsgrad
$\tilde{\eta}$	abhängiger Wirkungsgrad
i	Übersetzungsverhältnis
\mathcal{V}_{Motor}	Motorenvorrat
\mathcal{E}	Motorkennwerte
\vec{m}_{zul}	zulässige Motormomente
\vec{m}	numerischer Drehmomentverlauf
$\mathcal{V}_{Stellgerate}$	Stellgerätevorrat
\mathcal{R}	Stromrichter kennwerte
\mathbf{K}_H	kinematische Restriktionen aus Hardware

zu Kapitel 5

Symbol	Bezeichnung
t	kontinuierlicher Zeitverlauf
k	Laufvariable für diskrete Zustandsfolge
s	Schritt einer numerischen Berechnung
i	Nummer einer Achse
n	Achsenanzahl
j	Nummer eines Abschnitts
m_i	Anzahl der Bewegungsabschnitte einer Achse
\mathcal{B}	Parameter eines Bewegungsverlaufs
\mathcal{B}_{konf}	Bedingungen eines Bewegungsabschnitts
\mathcal{I}	innere Zusammenhänge
\mathbf{N}	Bewegungsabschnitte
\mathbf{N}_{erw}	Petrinetz mit inneren Zusammenhängen
\mathbf{K}_A	achsenspezifische kinematische Restriktionen
\mathbf{K}	abschnittsspezifische kinematische Restriktionen
\vec{o}	achsenspezifische Optimierungskriterien
\mathbf{O}	abschnittsspezifische Optimierungskriterien
$t_{i,j}$	Zeitpunkt eines Randwertes
$\tilde{t}_{i,j}$	Zeit eines bezogenen Randwertes
\mathbf{C}	Siehe Tabelle 5.1
\mathbf{D}	Siehe Tabelle 5.1
\mathbf{E}	Siehe Tabelle 5.1
\mathbf{F}	Siehe Tabelle 5.1
\mathbf{G}	Siehe Tabelle 5.1
\mathbf{H}	Siehe Tabelle 5.1
s_b	Bremsposition

Symbol	Bezeichnung
T_k	Zeitpuffer
s_k	kritische Position
v_k	kritische Geschwindigkeit
$Z_{i,j}$	Zustand eines Bewegungsabschnitts
Z	Zustand des Antriebssystems
K_p	Kosten eines Bewegungsabschnitts
T	Zeit eines Bewegungsabschnitts

zu Kapitel 6

Symbol	Bezeichnung
k_E	Energiekennwert einer Achsenbewegung
T_{Takt}	Taktzeit der Achse / Maschine

Abbildungsverzeichnis

1.1	„Kerrad“ in der „De re metallica“ [13]	9
1.2	Aufbau eines Servoantriebs	11
1.3	Längstaktmontageautomat [30]	11
1.4	Antriebssystem	14
1.5	Ablauf des Engineerings von Antriebssystemen, entsprechend [53]	17
1.6	Exemplarische Motorkennlinie (entsprechend [63])	18
1.7	Grundeinheit eines Petrinetzes	22
1.8	Vergleich zwischen alter und neuer Vorgehensweise	27
2.1	Gesamtübersicht	35
3.1	Bewegungsentwurf	39
3.2	Vergleich von Bewegungsgesetzen aus [60, 103]	40
3.3	Schar von Strecken (Umkehr-Umkehr)	45
3.4	Graph einer Bewegung mit Ableitungen	46
3.5	Übersicht über das Selektionsverfahren	47
3.6	Selektion nach kinematischen Restriktionen	48
3.7	Größen zur Bestimmung der Kennwerte für die Klassifizierung	51

3.8	Selektion über zwei Fuzzy Regelsätze	53
3.9	Klassendiagramm zur Interpolation von Bewegungen	56
3.10	Klassendiagramm zur Selektion	57
3.11	Algorithmus zur kinematischen Selektion	58
3.12	Programmauszug aus Bewegungsentwurf	59
4.1	Übersicht zum Hardwareentwurf	61
4.2	Reihenschaltung von Modulen mit Motor und Last	63
4.3	Stirnradgetriebe, nach [15]	65
4.4	Riementrieb, nach [114]	65
4.5	Förderband mit Last und Gegengewicht	66
4.6	Spindelantrieb	67
4.7	Zahnstangenantrieb	67
4.8	Transportwalzen, abgeleitet aus [116]	68
4.9	Vorgehensweise bei Auslegung und Auswahl	72
4.10	Klassendiagramm des mechanischen Modells	73
4.11	Programmauszug aus der mechanischen Modellbildung [119]	74
4.12	Klassendiagramm zur Auslegungsrechnung und Komponentenauswahl	74
5.1	Übersicht über den Systementwurf	76
5.2	Exemplarische Achsenbewegung	78
5.3	Arten innerer Zeitabhängigkeiten	80
5.4	Bestimmung der kritischen Zeit für den rechtzeitigen Not-Stopp	81
5.5	Petrinetausschnitt eines Bewegungsabschnitts	83
5.6	Ausschnitt des Petrinetzes eines Antriebssystems	84

5.7	Systemoptimierung	86
5.8	Wassersäulenmodell	88
5.9	Verteilungsalgorithmus	89
5.10	Klassendiagramm zum Systementwurf	91
5.11	Programmausschnitt für innere Abhängigkeiten	92
5.12	Programmausschnitt der Benutzeroberfläche des Systementwurfs	93
6.1	Kennlinie des gewählten Motors (Werkzeugantrieb) [33]	97
6.2	Kennlinie des gewählten Motors (Vorschubantrieb)	98
6.3	Bewegung für manuellen Systementwurf (Vorschub)	99
6.4	Bewegung für den CAE Systementwurf (Vorschub)	101
6.5	Bewegungen des Initialsystems (oben: $\varphi[rad]$; unten: $s[m]$)	104
6.6	Energieoptimaler Vorschub	105
6.7	Ruckoptimaler Vorschub	106
6.8	Zeitoptimaler Vorschub	108
6.9	Vergleich der Methoden	110
6.10	Softwareimplementierung	112
6.11	Code-Ausschnitt (virtuelle Methoden, integrierte Bewegungsgesetze)	115
6.12	Mechanischer Aufbau der Modellmaschine	117
6.13	Anforderungen des Längstaktmontageautomats in der Modellumgebung	118
6.14	Schlitten der Modellmaschine mit drei parallelen Querantrieben [129]	118
6.15	Elektronik des Versuchsaufbaus	120
6.16	Positionsverläufe im Initialsystem	122
6.17	Konvergenz des Algorithmus zum zeitlichen Ausgleich	123

6.18	Konvergenz des Intervallhalbierungsverfahrens	124
6.19	Wege des Versuchsmodells nach Achsen und Optimierungsvorgaben	125
6.20	Geschwindigkeiten des Versuchsmodells nach Achsen und Optimierungsvorgaben	126
6.21	Beschleunigungen des Versuchsmodells nach Achsen und Optimierungsvorgaben	127
6.22	Ruckverläufe des Versuchsmodells nach Achsen und Optimierungsvorgaben . . .	128
9.1	Energieoptimale Vorschubbewegung	153
9.2	Ruckoptimale Vorschubbewegung	154
9.3	Zeitoptimale Vorschubbewegung	154
9.4	Energieoptimale Öffnung der Fixierung	156
9.5	Ruckoptimale Öffnung der Fixierung	156
9.6	Zeitoptimale Öffnung der Fixierung	157
9.7	Energieoptimale Schließung der Fixierung	159
9.8	Ruckoptimale Schließung der Fixierung	159
9.9	Zeitoptimale Schließung der Fixierung	160

Tabellenverzeichnis

1.1	Kinematische Eigenschaften von Randwerten (entsprechend [60])	17
1.2	Berechnung von Stromwerten nach Motorbauart [28]	21
1.3	Unschärfe Operatoren	24
3.1	Fähigkeiten der Bewegungsgesetze, durch Analyse von [60]	42
5.1	Notation der inneren Zusammenhänge eines Antriebssystems	80
6.1	Kennwerte des manuellen Systementwurfs	99
6.2	Kennwerte des CAE Systementwurfs	100
6.3	Kennwerte des ganzheitlichen energieoptimierten Entwurfs	104
6.4	Kennwerte des ganzheitlichen ruckoptimierten Entwurfs	107
6.5	Kennwerte des ganzheitlichen zeitoptimierten Entwurfs	108
6.6	Vergleich der Entwurfsmethoden und Optimierungskriterien	109
6.7	Anforderungen an das Modell des Längstaktmontageautomats	121
6.8	Optimierung des Versuchsmodells E-energieoptimal, R-ruckoptimal, Z-zeitoptimal (F)-Fixierung, (V)-Vorschub	129
9.1	Optimierung der Vorschubbewegung	153
9.2	Optimierung der Fixierung (Öffnen)	155

9.3 Optimierung der Fixierung (Schließen) 158