

M. Günther

Computersimulationen mit
biomechanischen
Mehrkörpermodellen und Fuzzy-Regelung
zur Untersuchung der
Bewegungskoordination
am Beispiel Standhochsprung

DISSERTATION
zur Erlangung des Grades eines Doktors
der Naturwissenschaften
der Fakultät für Physik
der Eberhard-Karls-Universität zu Tübingen

vorgelegt von
Karin Widmayer
aus
Sindelfingen

1995

Tag der mündlichen Prüfung: 21. Februar 1995
Dekan: Prof. Dr. A. Fäßler
1. Berichterstatter: Prof. Dr. H. Ruder
2. Berichterstatter: Prof. Dr. K. Schneider

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit Computersimulationen von Mehrkörpersystemen für biomechanische Fragestellungen. Für die Simulation des Standhochsprungs wurde ein Modell des Menschen verwendet, das aus vier ebenen starren Körpern besteht, die durch drei Scharniergelenke verbunden sind, welche Hüftgelenk, Kniegelenk und Sprunggelenk in der Sagittalebene repräsentieren sollen.

Meine Motivation war, die Berechnung der Gelenkdrehmomente, die die Bewegung des Modells erzeugen, auf Soft-Computing-Methoden wie Fuzzy Logik, künstliche Neuronale Netze und Genetische Algorithmen aufzubauen. Zur Steuerung des Mehrkörpersystems habe ich einen Fuzzy-Regler sowie ein Kohonen-Netz als Vertreter für Neuronale Netze mit unüberwachtem Lernen implementiert.

Für die Simulation von Standhochsprüngen als Beispiel für eine Bewegungsaufgabe, welche die Koordination der drei Gelenke erfordert, habe ich Fuzzy-Regelung verwendet. Da keine Meßdaten als Simulationsinput eingehen sollten, arbeitete ich mit Bewertungsfunktionen, die die spezielle Bewegung auszeichnen. Aufgrund dieser Bewertung konnte ich mit Genetischen Algorithmen die Regelbasis des Fuzzy-Systems für die gewählte Bewegungsaufgabe optimieren. Auf diese Weise konnten wesentliche Eigenschaften des maximalen Standhochsprungs durch die Simulation reproduziert werden. Die Simulationsergebnisse stimmen gut mit experimentellen Daten überein, die eine biomechanische Arbeitsgruppe in Amsterdam aus Messungen an maximalen Standhochsprüngen von Sportlern gewonnen hat. Weiterhin wurde die Animation der Bewegung und die Visualisierung der Fuzzy-Regeln als wichtige Analyseinstrumente zur Beurteilung der simulierten Bewegung eingesetzt.

Der Simulationsansatz basiert auf starken Vereinfachungen der komplexen Realität der menschlichen Bewegung. Spezielle Annahmen, auf deren Grundlage die Berechnung der Drehmomente basiert, verhindern die Verallgemeinerung der Steuerung auf andere Fragestellungen und Bewegungen.

Das Interesse an Fuzzy-Regelung ist groß bei Arbeitsgruppen, die Computersimulation in der Biomechanik einsetzen, dennoch gab es bisher noch keine Anwendung in diesem Bereich. Anhand von Simulationen von Standhochsprüngen unterschiedlicher Sprunghöhe, Simulationen unter variierenden Startbedingungen und verschiedenen Optimierungskriterien habe ich Vorteile und Grenzen der Fuzzy-Regelung in diesen Fällen und meinem Simulationsansatz dargestellt. Darüber hinaus werden zukünftige Anwendungsmöglichkeiten der Soft-Computing-Methoden in der Biomechanik aufgezeigt.

Inhaltsverzeichnis

I	Einleitung	7
I.1	Problemstellung	7
I.2	Stand der Forschung	8
I.3	Motivation und Inhalt	10
II	Biomechanisches Modell und Vorwärtssimulation	14
II.1	Bewegungsgleichungen	14
II.2	Vorwärtssimulation	15
II.3	Das dreigelenkige Modell	15
II.3.1	Gelenkwinkelbegrenzungen	18
II.4	Numerische Integration der Bewegungsgleichungen	19
III	Fuzzy Logik, Neuronale Netze und Genetische Algorithmen	21
III.1	Fuzzy-Systeme und Neuronale Netze als Funktionenapproximierer	21
III.2	Fuzzy-Regelung	24
III.2.1	Fuzzy-Mengen	24
III.2.2	Basisverknüpfungen von Fuzzy-Mengen und Fuzzy-Logik	25
III.2.3	Fuzzy Control	26
III.2.4	Wie funktioniert Fuzzy Control?	26
III.2.5	Beispiel: Balancieren auf drei Gelenken	28
III.2.6	Vergleich mit PID-Regler	36
III.2.7	Vergleich mit Kohonen-Netz	37
III.2.8	Adaptive Fuzzy-Systeme	42
III.3	Optimieren mit Genetischen Algorithmen	45
III.3.1	Funktionsweise Genetischer Algorithmen	45
III.3.2	Empirische Analyse	48
III.3.3	Parallelisieren auf einem Workstation-Cluster	50

IV Simulation von Standhochsprüngen	54
IV.1 Simulation des maximalen Standhochsprungs	54
IV.1.1 Experimentelle Ergebnisse	55
IV.1.2 Simulation	57
IV.1.3 Diskussion der Simulationsergebnisse	62
IV.1.4 Mehrzielige Optimierung	78
IV.1.5 Maximalkraft und Schnellkraft	82
IV.2 Submaximale Standhochsprünge	84
IV.2.1 Vergleich von Sprunghöhe 151 cm, 140 cm und 133 cm	84
IV.3 Standhochsprünge unter variierenden Startbedingungen	93
IV.4 Hypothetische Prinzipien für Bewegungskoordination	97
IV.4.1 Jerk	97
IV.5 Animation und Performance	98
IV.5.1 Programmstruktur	100
IV.6 Zusammenfassung	103
IV.7 Ausblick	105
A Anhang	108
A.1 Anthropometrische Daten	108
A.2 Eingabedatei	110
Literaturverzeichnis	111

I

Einleitung

I.1 Problemstellung

Computersimulationen von Bewegungen des Menschen, die die Koordination mehrerer Körpersegmente und mehrerer Gelenke beinhalten, sind aus unterschiedlichen Gründen ein aktuelles Forschungsgebiet in der funktionellen Biomechanik des Menschen. Zum einen hofft man, daß die Computersimulation helfen könnte, die Mechanismen hinter der komplexen menschlichen Bewegungssteuerung zu verstehen. Zum anderen will man Rückschlüsse auf bestimmte Größen im menschlichen Körper ziehen, die der direkten Messung nicht zugänglich sind. Dabei handelt es sich typischerweise um Belastungen des Körpers wie innere Kräfte, Druck oder Drehmomente.

Weit verbreitet ist der Einsatz von Computersimulationen in der Sportwissenschaft, der medizinischen Vorsorge und Rehabilitation sowie in der Unfallforschung, wo kinematische Größen zur Beschreibung einer Bewegung dienen und mit inverser Dynamik die dabei auftretenden Kräfte und Momente analysiert werden können. Weit weniger Nutzen wurde bis jetzt aus der direkten Dynamik oder Vorwärtssimulation gezogen, mit deren Hilfe eine Bewegung im Computer synthetisiert wird. Dabei werden keine Trajektorien vorgegeben und Messdaten werden nur zur Verifizierung der Simulation und des Modells herangezogen. Die direkte Dynamik ist attraktiv, denn abgesehen von den starken Vereinfachungen, beschreitet sie im Unterschied zur inversen Dynamik denselben Weg wie ein belebter Körper.

Felix Zajac schreibt über die Vorwärtssimulation in der Biomechanik in einem Übersichtsartikel [Zajac, 1993]:

The effort necessary to develop forward dynamical models has been so high, however, that model-generated data of jumping or any other task are meager.

Zajac spielt darauf an, daß die Anforderungen für die Vorwärtssimulation hoch sind: Es muß für jede Bewegungsaufgabe ein sinnvolles Modell gefunden, dann müssen die Bewegungsgleichungen des Modells erstellt und die zugehörigen biomechanischen Parameter erhoben werden. Weiter müssen die Kontrollalgorithmen für die gewünschte Bewegung gefunden, numerische Methoden zur Lösung der Gleichungen implementiert, die Simulationsergebnisse ausgewertet und visualisiert werden. Schließlich muß das Modell durch das Experiment verifiziert werden.

Jede dieser Teilaufgaben überlappt mit Fragestellungen aus anderen Disziplinen. Daraus kann man teilweise profitieren. Als Beispiel seien Differentialgleichungserzeuger und Integratoren genannt, die für Mehrkörpersysteme im Fahrzeugbau entwickelt wurden.

Zum gegenwärtigen Stand der Forschung liegt die eigentliche Herausforderung für die Computersimulation von Bewegungen in der Entwicklung von Algorithmen, die die Bewegungssteuerung modellieren, denn die Steuerung und Koordination einer ganzen Bewegung sind kaum verstanden.

I.2 Stand der Forschung

Sehr viele Forschungsarbeiten in der Biomechanik beschäftigen sich mit periodischen Bewegungen wie Gehen, Fahrradfahren etc. Ein Vorteil von periodischen Bewegungen ist, daß sie im Vergleich zu einmaligen Bewegungen leichter experimentell zu untersuchen sind, denn man kann über einen längeren Zeitraum messen (Kräfte, Beschleunigungen, Elektromyogramme, optische Aufzeichnungen) und dann über viele Zyklen mitteln. Bei explosiven Bewegungen, die durch kurze Bewegungsdauer und vergleichsweise hohe Geschwindigkeiten ausgezeichnet sind, ergeben sich dagegen meßtechnische Schwierigkeiten, zum Beispiel beim Befestigen von Markern oder Beschleunigungsmessern auf der sich bewegenden Haut. Filmaufnahmen müssen mit hoher Bildfrequenz gemacht werden, damit nach dem Differenzieren von Positionen noch aussagekräftige Daten für Geschwindigkeiten und Beschleunigungen resultieren.

Für die Vorwärtssimulation von periodischen Bewegungen seien hier zwei bemerkenswerte Simulationsmodelle erwähnt.

Die Simulationen von McGeer (1990) zum passiven Gehen (ballistic walking) zeigen, daß die Struktur des Bewegungsapparates so gestaltet ist, daß ausschließlich

die durch Gravitation und Trägheit bedingten Kräfte die Bewegung bewirken. Dieses passive Gehen ohne Muskelkraft ist allerdings nicht tolerant gegen Störungen und nur für einen eng begrenzten Bereich von Anfangsbedingungen möglich.

Ein anderes Beispiel, in dem wie beim passiven Laufen die Eigendynamik eines Systems bei bestimmten Anfangsbedingungen zur Kontrolle ausreicht, sind die zyklischen, rhythmischen Bewegungen beim Jonglieren von Bällen. Shannon hat das durch den Bau einer Jongliermaschine gezeigt. Ein weiterer Vorteil von rhythmischen Bewegungen ist, daß sie mathematisch mit Fourieranalyse, Poincaré-Schnitten und anderen Werkzeugen der nichtlinearen Dynamik analysiert werden können. Beek (1989) hat die Hände beim Jonglieren als gekoppelte Oszillatoren betrachtet und mit diesen Untersuchungsmethoden Phasenkopplung, periodische und quasi-periodische sowie chaotische Bereiche ausmachen können. Damit kann der Unterschied zwischen Könnern und Anfängern beim Jonglieren quantifiziert werden.

Einen wesentlichen Schritt weiter gekommen ist Taga (1991), der ein gut funktionierendes Modell mit Steuerung für aktives Gehen und Laufen entwickelt hat. Durch Oszillatormodelle (rhythm pattern generator), die den spinalen Lokomotionsgenerator modellieren und durch sensorischen Feedback erzielt er stabile Bewegungszyklen. Seine zweibeinigen Modelle können sogar Störungen und unebenes Gelände in gewissen Grenzen verkraften. Durch eine zentrale Steuerungseinheit kann er den Übergang vom Gehen zum Laufen und wieder zurück sehr überzeugend simulieren.

Interessante Experimente und Simulationen zu nichtperiodischen Bewegungen gibt es unter anderem von Arbeitsgruppen in Amsterdam [Bobbert, 1988], [Soest, 1993 a], [Soest, 1993 b] und Stanford [Zajac, 1993], [Pandy, 1990], [Pandy, 1991]. Ihre Modelle vom Bewegungsapparat bestehen aus Skelett sowie Muskelgruppen, deren dynamisches Verhalten durch das Hill'sche Muskelmodell beschrieben wird. Die Kontrolle der Muskelkräfte geschieht dabei auf der Ebene der Muskelaktivierung. Da das Implementieren von Muskeln ein komplexes Unterfangen ist, ist es bis jetzt nur möglich, solche Modelle für Teile des Körpers und für ganz bestimmte Bewegungsaufgaben zu erstellen.

Ein Beispiel ist die Modellierung der Beinmuskulatur, die für die Beschleunigung beim Standhochsprung von Bedeutung ist. Dazu wurden 6 Muskelgruppen isoliert. Entscheidend für die Modellierung ist die Frage, wie die Muskeln an das Skelett angekoppelt sind, d. h. wie die Muskeln in der Simulation Drehmomente in den Gelenken erzeugen. In diesem Punkt unterscheiden sich die Modelle von [Soest, 1993 a] und [Pandy, 1990]. In jedem Fall werden anatomische Größen wie Ansatz, Ursprung und Umlenkung der Muskeln und Sehnen beziehungsweise Hebelarme bezüglich der Gelenke benötigt. Diese stammen überwiegend aus Leichen-

untersuchungen. Die neuromuskuläre Innervierung wird durch eine Stimulation im kontraktile Element des Muskelmodells realisiert. Van Soest macht dazu die vereinfachende Annahme, ein Muskel sei zuerst nicht aktiviert und dann aktiviert, und setzt dazu die Aktivierung zu einem bestimmten Zeitpunkt von 0 auf 1. Der Zeitpunkt der Aktivierung eines Muskels in dieser sogenannten *bang-bang control* Methode wird durch Optimieren bestimmt.

Aufbauend auf den Daten und der Erfahrung solcher Standhochsprung-Simulationen mit einem Muskel-Skelett-Modell und Optimierung hat Kuo (1993) ein viergliedriges Modell mit drei Freiheitsgraden entwickelt. 14 Muskeln stehen zur Kontrolle der Körperhaltung zur Verfügung. In dieser Arbeit wird davon ausgegangen, daß es sich bei der Stützmotorik um kleine quasi-statische Bewegungen handelt. Die geschwindigkeitsabhängige Terme der Bewegungsgleichungen und der geschwindigkeitsabhängige Teil der Muskelfunktion werden vernachlässigt. Die Aktivierung der Muskeln kann jeden Wert zwischen 0 und 1 annehmen. Die Hypothese ist, daß Zwangsbedingungen die Strategie der Bewegung bestimmen. Einerseits müssen die Füße flach auf dem Boden bleiben, der Körperschwerpunkt über den Füßen sein, etc. Andererseits werden die Beschleunigungen in den Gelenken durch die Stärke der Muskeln und durch ihre Aktivierung beschränkt. Am Ende der Berechnung mittels *Linear Programming* bleibt im dreidimensionalen Raum, der von den Beschleunigungen der drei Gelenke aufgespannt wird, aufgrund dieser Nebenbedingungen nur ein schmaler Bereich für mögliche Lösungen übrig.

I.3 Motivation und Inhalt

Die Anfangsphase dieser Arbeit war geprägt von den Erfahrungen mit *trial and error* Steuerungsversuchen von mehrgliedrigen biomechanischen Computermodellen. Die Ergebnisse zeigten eindrucksvoll, zu welcher vielfältigen Bewegungen ein schlecht gesteuertes Mehrkörpersystem fähig ist und die Animationen davon sorgten für Heiterkeit.

Andererseits zeigen die oben genannten Arbeiten, daß schon die Eigendynamik eines Systems für manche Bewegungen (fast) ausreicht. Allgemein gilt für Bewegungen von Lebewesen, daß trotz der vielen Freiheitsgrade, die für die Bewegung zur Verfügung stehen, nur sehr wenige bei der Ausführung wirklich genutzt werden.

Für die Simulation von koordinierten, dynamischen Bewegungen sind die Möglichkeiten der Steuerung durch einen physikalischen Ansatz auf der Ebene von Gesamtgelenkmomenten noch nicht erschöpfend untersucht worden. Allerdings sind gängige und wirkungsvolle Methoden wie Lineares Optimieren für dynamische Probleme nicht geeignet.

Das Ziel ist eine spezielle Bewegung im Computer zu synthetisieren, so daß sie wesentliche Charakteristika der realen Bewegung aufweist. Die Schwierigkeit liegt darin, einen Steuerungsansatz zu finden, der nicht zu komplex ist, denn ein kompliziertes System hat viele unbekannte Parameter, die gefunden werden müssen. Die Gefahr ist groß, daß jedes gewünschte, vielleicht durch Messungen suggerierte Ergebnis reproduziert werden kann, ohne daß dabei ein Erkenntnis gewonnen wird. Andererseits darf der Ansatz nicht zu simpel sein, damit als Ergebnis nicht nur vorhersehbare Lösungen resultieren.

Seit einiger Zeit versprechen neue Methoden der intelligenten Datenverarbeitung Lösungsansätze für Probleme mit vielen Freiheitsgraden. Die Computerzeitschriften sind voll mit Artikel über *Neuronale Revolution, Fuzzy Boom, Evolution im Computer* . . . Nähere Beschäftigung mit den Theorien, die sich hinter den Schlagworten Neuronale Netze, Fuzzy Logik und Genetische Algorithmen verbergen, zeigen tatsächlich interessante Ansätze zur Behandlung komplexer, nichtlinearer Systeme. Besonders angesprochen haben mich die Ideen L. Zadehs, dem Erfinder der Fuzzy Logik, zum Wesen komplexer Systeme, wie sie zum Beispiel in folgendem Zitat geäußert wurden [Zadeh, 1973]:

The closer one looks at a 'real world' problem, the fuzzier becomes its solution. Stated informally, the essence of this principle is that as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive.

Zadeh hat seine Überlegungen mathematisch formuliert und als *fuzzy set theory* etabliert. Daraus entstand wiederum eine Methode, *Fuzzy Control*, die in der Regelungstechnik umgesetzt wurde.

Eine Fülle von Arbeiten zeigt, daß in den letzten Jahren sehr viele Forscherinnen und Forscher in unterschiedlichen Disziplinen das Anwendungspotential hinter den von Zadeh und anderen gelegten Grundsteinen entdeckten. Da in der Zwischenzeit mehrere der genannten Methoden kombiniert verwendet werden und sich dadurch laufend neue Aspekte entwickeln, hat sich dafür in jüngster Zeit der Begriff *Soft-Computing* eingebürgert.¹

Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty and partial truth. In effect, the role model for soft computing is the human

¹L.A. Zadeh als Direktor der *Berkeley Initiative in Soft Computing* in einem elektronischen Report zum dritten Geburtstag der Initiative, 1994.

mind. The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost. The principal constituents of soft computing (SC) are fuzzy logic (FL), neural network theory (NN) and probabilistic reasoning (PR), with the latter subsuming belief networks, genetic algorithms, chaos theory and parts of learning theory. What is important to note is that SC is not a melange of FL, NN and PR. Rather, it is a partnership in which each of the partners contributes a distinct methodology for addressing problems in its domain. In this perspective, the principal contributions of FL, NN and PR are complementary rather than competitive.

Im Gegensatz zur Robotik ist Soft-Computing in der Biomechanik noch kaum zur Anwendung gekommen.

Diese Arbeit ist so aufgebaut, daß nach dieser Einleitung in Kapitel II ein Computermodell vorgestellt wird, das verwendet wird, um Standhochsprünge zu simulieren. An diesem Beispiel wird auch erklärt, welche Arbeitsschritte nötig sind, um ein mechanisches Modell des Menschen im Computer zu erstellen und wie man mit Mehrkörpermodellen dynamische Vorgänge simulieren kann.

Indem ich eigene Programme für Fuzzy-Regler, Kohonen-Netz und Genetische Algorithmen erstellt habe, konnte ich diese Soft-Computing-Techniken zur Steuerung des Mehrkörpersystems einsetzen. Die Erfahrung beim Simulieren mit den genannten Methoden sind in Kapitel III an einem Beispiel dokumentiert. Das Balancieren des dreigelenkigen Modells ist geeignet, um die Vor- und Nachteile, sowie die Kombination der Techniken aufzuzeigen.

In Kapitel IV ist dann das eigentliche Ziel, die Simulation von Standhochsprüngen als Beispiel für eine koordinierte Bewegungsaufgabe, beschrieben. Da ich keine Meßdaten als Input verwende, muß ich mit Kriterien arbeiten, die die spezielle Bewegung auszeichnen. Mit einem Steuerungsansatz auf der Ebene von Gesamtgelenkmomenten mit Hilfe von Fuzzy Control und Genetische Algorithmen konnte ich Standhochsprünge simulieren. Zum Vergleich und Diskussion meiner Simulationsergebnisse habe ich sowohl experimentelle Ergebnisse als auch simulierte Ergebnisse anderer Arbeitsgruppen von Standhochsprüngen mit dem Ziel maximale Sprunghöhe herangezogen.

An Simulationen von Standhochsprüngen geringerer Sprunghöhe, an Simulationen unter variierenden Startbedingungen und verschiedenen Optimierungskriterien werden noch einmal die Stärken und Grenzen des gewählten Steuerungsansatzes deutlich. Dabei stellt die Animation der simulierten Bewegung auf Graphik-Workstations ein wichtiges Analyseinstrument dar. Der Ausblick zeigt, wie Fuzzy-

Systeme in Verbindung mit anderen Modellen und Methoden weitere Verwendung in der Biomechanik finden könnten.

II

Biomechanisches Modell und Vorwärtssimulation

II.1 Bewegungsgleichungen

Die Bewegungsgleichungen für ein biomechanisches Mehrkörpersystem (MKS), die für die Vorwärtssimulation benötigt werden, können symbolisch oder numerisch rechnerunterstützt in verschiedenen Formulierungen erzeugt werden [Hospach, 1993], [Krieg, 1992], [Krebs, 1991], [Widmayer, 1990].

Die Bewegungsgleichungen für das im folgenden beschriebene Modell wurden mit dem symbolischen Differentialgleichungserzeuger für Mehrkörpersysteme NEWEUL erzeugt [NEWEUL, 1988].

Ein MKS bestehe aus gekoppelten starren Körpern. Die Zwangsbedingungen seien holonom. Für ein System mit f Freiheitsgraden wird durch das Programm NEWEUL aus Impulssatz und Drallsatz durch Anwendung des d'Alembertschen Prinzips eine f -dimensionale Differentialgleichung 2. Ordnung erzeugt. Die symmetrische Massenmatrix, die dabei entsteht, enthält nur die verallgemeinerten Koordinaten $\mathbf{y} \in \mathbb{R}^f$ als Unbekannte. Dieses System aus *Newton-Euler-Gleichungen* kann dargestellt werden als:

$$\mathbf{M}(\mathbf{y}, t) \ddot{\mathbf{y}} + \mathbf{K}(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{QE}(\mathbf{y}, \dot{\mathbf{y}}, t) \quad (\text{II.1})$$

mit

\mathbf{M} : $f \times f$ symmetrische, positiv definite Massenmatrix $\mathbf{M} = \mathbf{M}^T > 0$,
 \mathbf{K} : $f \times 1$ Vektor der verallg. Kreisel-, Zentrifugal- und Corioliskräfte,
 \mathbf{QE} : $f \times 1$ Vektor der verallg. eingepprägten Kräfte.

NEWEUL gibt dieses Gleichungssystem in symbolischer Form in FORTRAN 77 kompatibler Schreibweise aus.

II.2 Vorwärtssimulation

Unter Vorwärtssimulation oder direkter Dynamik versteht man, wenn die auf das Mehrkörpersystem wirkenden Kräfte und Momente vorgegeben werden und die Beschleunigungen durch das Gleichungssystem (II.1) berechnet werden. Die Bewegung des Mehrkörpersystems erhält man dann durch Integration der Beschleunigungen. Voraussetzung für die Vorwärtssimulation ist die Kenntnis der inneren und äußeren Kräfte und Momente während der ganzen Bewegung. Deshalb müssen sowohl die biomechanischen Zwangsbedingungen, denen der menschliche Körper unterliegt, berücksichtigt werden, als auch die mechanischen Eigenschaften von Kontaktstellen bei der Wechselwirkung des Körpers mit der Umgebung. Im nächsten Abschnitt werden deshalb die analytischen Zusammenhänge für die Berechnung der Bodenreaktionskräfte und Gelenkwinkelbegrenzungsmomente erklärt. Die Parameter dieser Gleichungen können an verschiedene Modelle und Simulationsbedingungen angepaßt werden.

Bei der inversen Dynamik sind in Folge von Messungen an einer realen Bewegung Koordinaten, Geschwindigkeiten und Beschleunigungen bekannt. Aus den Bewegungsgleichungen können dann die Kräfte und Momente berechnet werden und so die Bewegung analysiert werden. Im Gegensatz dazu sind bei der Vorwärtssimulation Messungen nur zur Validierung des Modells nötig. Es werden denn auch nicht speziell für jede Simulation Meßdaten benötigt, aufgrund dessen sind auch Simulationen unter verschiedenen Anfangsbedingungen möglich.

II.3 Das dreigelenkige Modell

Das hier verwendete Modell zur Simulation von mehrgelenkigen Bewegungen des Menschen in der Sagittalebene ist zweidimensional und besteht aus vier Segmenten (Fuß, Unterschenkel, Oberschenkel und Rumpf). Es hat drei Gelenke (Hüft-, Knie- und Sprunggelenk) und besitzt $f = 6$ Freiheitsgrade in der Ebene.

Abbildung II.1 zeigt das Modell, den Aufpunkt und die Segmentwinkel, welche als unabhängige Koordinaten des Vektors $\mathbf{y} = (x, z, \varphi_F, \varphi_U, \varphi_O, \varphi_R)$ gewählt

wurden. Die Gelenkwinkel ($\phi^{\text{Hüft}}$, ϕ^{Knie} , ϕ^{Sprung}) lassen sich durch die vier Segmentwinkel, die von der rechten Horizontalen aus gemessen werden, ausdrücken:

$$\phi^{\text{Hüft}} = \pi + \varphi_R - \varphi_O \quad (\text{II.2})$$

$$\phi^{\text{Knie}} = \pi + \varphi_U - \varphi_O \quad (\text{II.3})$$

$$\phi^{\text{Sprung}} = \pi + \varphi_U - \varphi_F \quad (\text{II.4})$$

Drehmomente an den Gelenken erzeugen Bewegungen des Modells. Die Modellsegmente wurden als starre Körper modelliert. Die verwendeten anthropometrischen Daten beruhen auf biomechanischen Daten von durchschnittlichen Menschen aus dem *Anthropometric Source Book* der NASA [NASA, 1978]. Das Programm *calcman* [Hahn, 1993] liefert nach Eingabe von Geschlecht, Körpergröße und Gesamtkörpermasse aus diesem Datenmaterial die Segmentmassen, Segmentlängen, Teilkörperschwerpunktskoordinaten und Trägheitsmomente für verschiedene Zerlegungen in Teilkörper (siehe Anhang A.1). Mit diesen Angaben kann dann, entsprechend der zu simulierenden Bewegung, ein geeignetes Computermodell erstellt werden. Die Massen und Trägheitsmomente stehen dann in der Massenmatrix des Bewegungsgleichungssystems.

Zur Berücksichtigung der am Fuß angreifenden Reaktionskräfte besitzt der Fuß zwei Kraftangriffspunkte, die Ferse und Ballen repräsentieren. Die Fußgeometrie wird durch drei Größen festgelegt: Der Abstand Sprunggelenk–Ballen, der Abstand Ballen–Ferse und der Winkel zwischen diesen beiden Verbindungen.

Der Fuß ist nicht nur gelenkig mit der Unterlage verbunden, wie bei anderen, vergleichbaren Modellen [Soest, 1993 a], [Pandy, 1990], sondern hat in der Ebene Translations- und Rotationsfreiheit; außerdem treten bei Bodenkontakt Reaktionskräfte auf. Eine möglichst realitätsnahe Modellierung des Fersen- bzw. Ballenpolsters ist wichtig, damit bei der Simulation realistische Kräfte auftreten und ein natürliches Bewegungsverhalten erzielt wird.

Die Bodenreaktionskraft entsteht durch Deformation des Fersenpolsters und/oder Ballenpolsters sowie durch die Deformation der Kontaktfläche. In Abbildung II.1 wird dies durch Feder–Dämpfer–Elemente symbolisiert.

Die vertikale Komponente F_z wird durch ein visco-elastisches Element beschrieben und ist abhängig von Deformation Δz und Deformationsgeschwindigkeit \dot{z} an der Kontaktstelle [Gruber, 1987]:

$$F_z = a_z \Delta z^{3.5} - b_z \dot{z}. \quad (\text{II.5})$$

Mit den konstanten Parametern a_z , b_z für Ballen, respektive Ferse. Vor allem durch die Verbreiterung der Ferse und des Ballens beim Kontakt mit der Umgebung ist die Kraft nichtlinear abhängig von der Deformation.

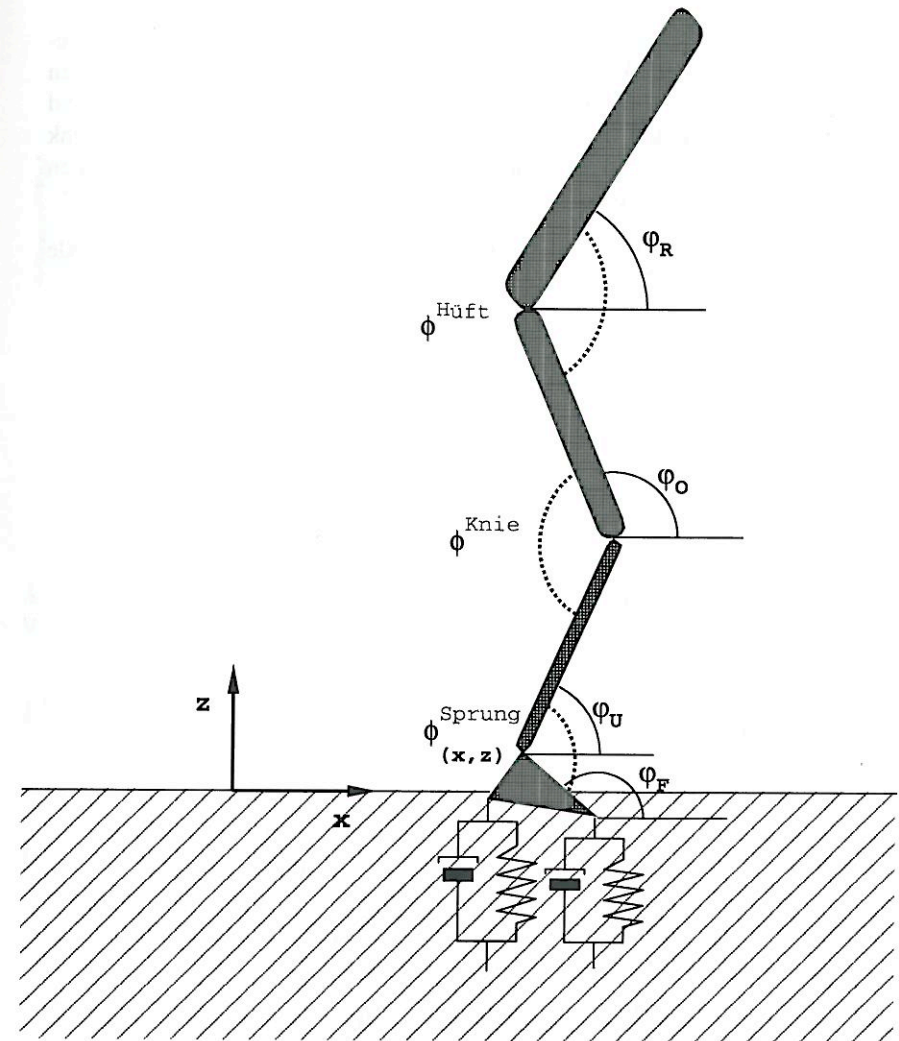


Abb. II.1: Das ebene viergliedrige Modell mit zwei Kraftangriffspunkten.

II.3.1 Gelenkwinkelbegrenzungen

Die Gelenke des zweidimensionalen Modells sind zunächst ideale Scharniergelenke. In der Realität wird die Bewegungsfreiheit von menschlichen Gelenken durch Zwangsbedingungen eingeschränkt. Während am Kniegelenk die Seiten- und Kreuzbänder die Bewegungsfreiheit begrenzen, geschieht dies im Sprunggelenk beim Hochziehen der Fußspitze durch knöchernerne Hemmung. In manchen Gelenken hemmen auch Weichteile wie das Muskelvolumen die Bewegung [Lippert, 1993].

In der Simulation sollen diese Gelenkwinkelbegrenzungen durch folgende Funktionen modelliert werden:

$$M_{\text{pas}}^j(\phi^j) = \begin{cases} \text{für } \phi_{\min}^j \leq \phi^j < (\phi_{\min}^j + \Delta\phi_{\min}^j) \\ \quad \mp M_{\min}^j \left[\frac{\Delta\phi_{\min}^j - (\phi^j - \phi_{\min}^j)}{\Delta\phi_{\min}^j} \right]^{k_{\min}^j} \\ \text{für } (\phi_{\min}^j + \Delta\phi_{\min}^j) \leq \phi^j \leq (\phi_{\max}^j - \Delta\phi_{\max}^j) \\ \quad 0 \\ \text{für } (\phi_{\max}^j - \Delta\phi_{\max}^j) < \phi^j \leq \phi_{\max}^j \\ \quad \pm M_{\max}^j \left[\frac{\Delta\phi_{\max}^j - (\phi_{\max}^j - \phi^j)}{\Delta\phi_{\max}^j} \right]^{k_{\max}^j} \end{cases} \quad (\text{II.6})$$

Der Index j spezifiziert das Gelenk. Jedes Gelenk hat zwei Anschläge, die bei den Gelenkwinkeln ϕ_{\min}^j und ϕ_{\max}^j liegen. Wenn während einer Bewegung ein Gelenkwinkel ϕ^j auftritt, der näher als $\Delta\phi_{\min}^j$ bzw. $\Delta\phi_{\max}^j$ an den Anschlag kommt, wird ein rücktreibendes Moment M_{pas}^j erzeugt. Durch M_{\min}^j bzw. M_{\max}^j wird die maximale Stärke des Drehmoments festgelegt. Die konstanten Parameter und das Vorzeichen müssen an die verschiedenen Gelenke angepaßt werden.

In Abbildung II.2 ist Gleichung (II.6) durch Kurven für die drei Gelenke des Modells aus Abbildung II.1 dargestellt. Man sieht, daß der maximale Anschlag bei allen Gelenken ungefähr bei 180 Grad eingestellt ist und der minimale bei mindestens 10 Grad. Über weite Winkelbereiche tritt kein Moment auf, erst wenige Grad vor dem Anschlag steigt es an. Das Vorzeichen ist für das Kniegelenk genau umgekehrt wie für Hüft- und Sprunggelenk.

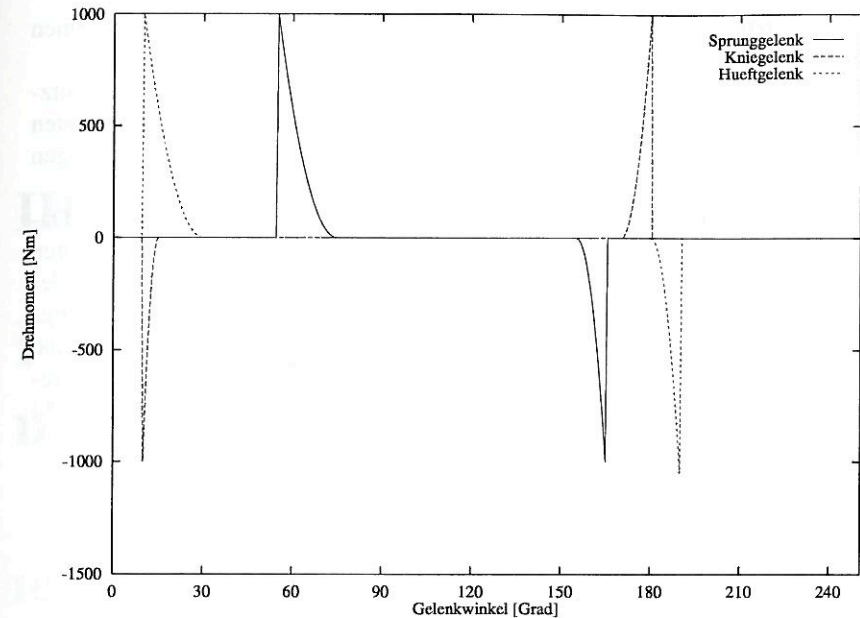


Abb. II.2: Anschlagmomente modellieren aus physikalischer Sicht die passive Wirkung der Bänder und Knochen im oberen und unteren Anschlag eines Gelenks.

II.4 Numerische Integration der Bewegungsgleichungen

Da nur für ganz wenige Spezialfälle analytische Lösungen gefunden werden können, werden die Bewegungsgleichungen numerisch gelöst. Es handelt sich dabei um ein Anfangswertproblem von gewöhnlichen Differentialgleichungen.

Die f Differentialgleichungen 2. Ordnung aus Gleichung (II.1) können durch

$$\dot{y} = v \quad (\text{II.7})$$

$$\dot{v} = M^{-1}(y, t) [QE(y, v, t) - K(y, v, t)] \quad (\text{II.8})$$

in ein System aus $2f$ Differentialgleichungen 1. Ordnung überführt werden.

Zu Beginn einer Integration wird ein kompletter Satz von Anfangskoordinaten und Anfangsgeschwindigkeiten benötigt.

Da die Matrix M von den Orten abhängt, muß sie nach jedem Integrations-

schritt invertiert werden. Ich habe zum Lösen von linearen Gleichungssystemen Routinen aus der Funktionenbibliothek LINPACK benutzt.

Die gängigen Integrationsverfahren unterscheiden sich in der Wahl der Stützstellen (feste oder variable Schrittweite) und in der Berechnung der Koordinaten und Geschwindigkeiten aus den an den Stützstellen erhaltenen Beschleunigungen (linear oder höhere Ordnung, Ordnung fehlerabhängig variabel).

Ich verwende den Code *de* von Shampine und Gordon [Shampine, 1975]. Dahinter verbirgt sich eine Prädiktor-Korrektor-Methode, ein Mehrschrittverfahren mit automatischer Ordnungs- und Schrittweitensteuerung, je nach geforderter Genauigkeit. Diese Integrationsroutine ist sehr komfortabel und gut geeignet für Bewegungen ohne harte Stöße. Durch die automatische Schrittweitensteuerung ist die Zahl der Aufrufe der Integrationsroutine *de* zwischen zwei Zeitschritten vom Benutzer nicht vorhersehbar. Die zu integrierenden Gleichungen dürfen deshalb nicht explizit von dieser Schrittweite oder von vorherigen Zeitschritten abhängen.

III

Fuzzy Logik, Neuronale Netze und Genetische Algorithmen

III.1 Fuzzy-Systeme und Neuronale Netze als Funktionenapproximierer

Seit einigen Jahren stehen mit Fuzzy-Systemen und künstlichen Neuronalen Netzen zwei sogenannte modell-freie Verfahren zur Verfügung. Bart Kosko beschreibt, wie beide Methoden als Funktionenapproximierer wirken [Kosko, 1992]:

Neural networks and fuzzy systems estimate input-output functions. Both are trainable dynamical systems. Sample data shapes and „programs“ their time evolution. Unlike statistical estimators, they estimate a function without a mathematical model of how outputs depend on inputs. They are model-free estimators. They „learn from experience“ with numerical and, sometimes, linguistic sample data. Neural and fuzzy systems encode sampled information in a parallel-distributed numerical framework.

Neuronale Netze lernen im allgemeinen aus Trainingsdaten, dagegen sind Fuzzy-Systeme regelbasiert. Obwohl funktionell äquivalent in der Verarbeitung von Information, sind die Voraussetzungen und Wirkungsweisen der zwei Methoden unterschiedlich.

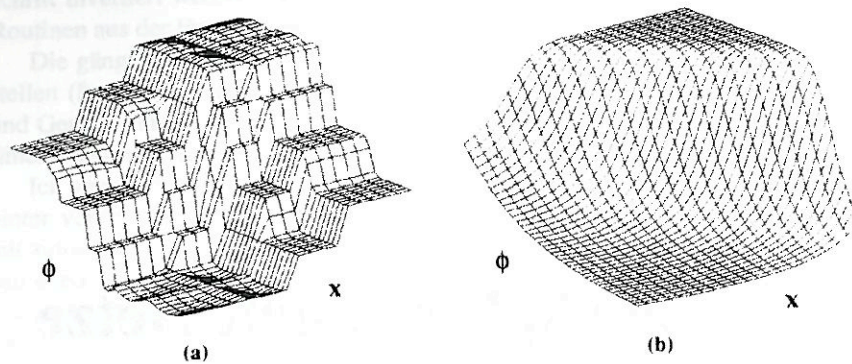


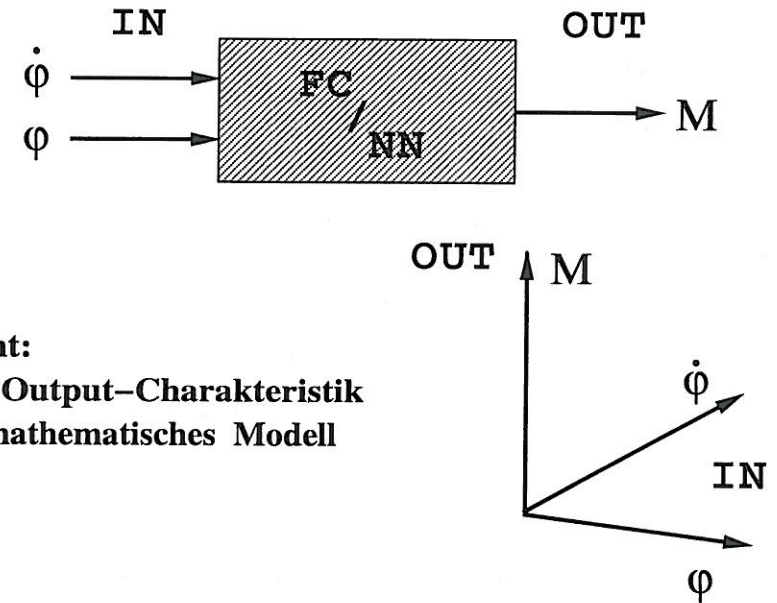
Abb. III.1: Beispiel für eine nichtlineare Input-Output-Charakteristik für dasselbe Problem, links als Ergebnis eines Fuzzy Controllers und rechts als Ergebnis eines Neuronales Netzes [Kosko, 1992].

Neural networks and fuzzy systems process inexact information and process it inexactly. Neural networks recognize ill-defined patterns without an explicit set of rules. Fuzzy systems estimate functions and control systems with partial descriptions of system behavior. Experts may provide this heuristic knowledge, or neural networks may adaptively infer it from sample data.

Für die Bearbeitung einer physikalischen Problemstellung sind solche Aspekte entscheidend. Abbildung III.1 soll ohne nähere Erläuterung einen optischen Eindruck davon geben, daß dasselbe Problem prinzipiell mit beiden Verfahren gelöst werden kann. Andererseits zeigt die Input-Output-Charakteristik auch deutliche Unterschiede.

Aus physikalischer Sicht sind Drehmomente in den Gelenken verantwortlich für eine aktive Bewegung. Für eine willentliche, zielgerichtete Bewegung sind die Drehmomente das Ergebnis eines komplexen neuro-muskulären Vorgangs. Obwohl heute umfangreiches mikroskopisches Detailwissen zur Verfügung steht und einzelne elementare Reflexe gut verstanden sind, gibt es kein allgemeines mathematisches Modell für die neuro-muskuläre Aktivierung und die resultierenden Drehmomente, die eine koordinierte Bewegung bewirken. Deshalb soll bei der Simulation von Bewegungen des Modells die Steuerung des mechanischen Systems von informationsverarbeitenden Systemen übernommen werden, die die Kontrollalgorithmen, ohne

mathematisches Modell (d.h. Differentialgleichungen), auf anderem Wege abschätzen.



Gesucht:
Input-Output-Charakteristik
ohne mathematisches Modell

Abb. III.2: Fuzzy-System und künstliches Neuronales Netz als Funktionenapproximierer. Als Beispiel sind zwei Eingangsgrößen und eine Ausgangsgröße eingetragen.

Abbildung III.2 soll diese Fragestellung nochmals schematisch verdeutlichen. Ein Fuzzy Controller oder ein Neuronales Netz sollen die Charakteristik zwischen Eingangs- und Ausgangsgrößen zur Verfügung stellen. Als Beispiel sind ein Winkel φ und eine Winkelgeschwindigkeit $\dot{\varphi}$ auf der Eingangsseite sowie ein Moment M am Ausgang eingetragen. Gesucht ist also analog zu Abbildung III.1 eine Input-Output-Charakteristik.

Zunächst sei noch ein technischer Aspekt erwähnt. Künstliche Neuronale Netze und Fuzzy-Regler können prinzipiell realisiert werden durch:

- Programmierung in einer konventionellen Programmiersprache,
- Verwendung spezieller Software-Werkzeuge,
- Verwendung einer speziellen Hardware (Fuzzy-Prozessor, Neuro-Chip).

Diese Arbeit bezieht sich ausschließlich auf eigene Programme, die in den Programmiersprachen FORTRAN und C erstellt wurden.

III.2 Fuzzy-Regelung

III.2.1 Fuzzy-Mengen

Der Begriff *Fuzzy* geht auf die 1965 von Lotfi Zadeh entwickelte *Fuzzy Set Theory* zurück, die im deutschen Sprachraum unter der *Theorie der unscharfen Mengen* bekannt geworden ist. Die Ausgangsfrage ist, wie man unsichere oder unscharfe Aspekte eines Problems in formalen Sprachen formulieren kann, damit sie maschinenverständlich werden.

Mit Unsicherheit oder Unschärfe kann beispielsweise folgendes gemeint sein:

- Fehlerbehaftete Daten aufgrund endlicher Meßgenauigkeit;
- Zufällige Ereignisse;
- Vage Beschreibungen, wie zum Beispiel Aussagen, die in natürlicher Sprache vorliegen und mit Attributen wie ziemlich klein, lauwarm, etwa zwei, teuer, unwahrscheinlich etc. ausgedrückt werden.

Zadeh führt die unscharfe Menge oder Fuzzy-Menge als axiomatische Struktur ein, um Information und Unschärfe mathematisch zu fassen [Kruse, 1993].

Definition 1 (Fuzzy-Menge) Eine Fuzzy-Menge μ von X ist eine Funktion von der Grundmenge bzw. dem Universum X in das Einheitsintervall, d. h.

$$\mu : X \rightarrow [0, 1].$$

$F(X)$ bezeichnet die Menge aller Fuzzy-Mengen von X .

Das Konzept unscharfer Mengen setzt voraus, daß die Zugehörigkeit von Objekten zu unscharfen Mengen quantifiziert werden kann. Ist die Grundmenge eine physikalische Größe, ist die Elementanzahl meist groß oder für X wird ein Kontinuum verwendet. Fuzzy-Mengen werden unter Verwendung der sie charakterisierenden Zugehörigkeitsfunktionen (membership functions) $\mu : X \rightarrow [0, 1]$ dargestellt.

Gebräuchliche Zugehörigkeitsfunktionen $\mu(x)$ für Anwendungen sind Trapezfunktionen, Dreiecksfunktionen oder Gaussfunktionen, da diese gut parametrisierbar sind. So kann jede trapezförmige unscharfe Menge mit vier Parameter

$a, b, c, d \in \mathbb{R}$ charakterisiert werden: ($x \in X$)

$$\mu_A(x) := \begin{cases} \frac{x-a}{b-a} & : a \leq x < b \\ 1 & : b \leq x \leq c \\ \frac{x-d}{c-d} & : c < x \leq d \\ 0 & : x < a \vee x > d \end{cases} \quad (\text{III.1})$$

Die Parameter $a \leq b \leq c \leq d$ sind dann dem konkret vorliegenden Modellierungsproblem anzupassen.

Es sei an dieser Stelle explizit darauf hingewiesen, daß eine Zugehörigkeitsfunktion nicht als Dichtefunktion im Sinne der Wahrscheinlichkeitstheorie interpretiert werden darf. Rein formal betrachtet muß der Flächeninhalt unterhalb des Graphen von μ nicht gleich 1 sein, und auch vom semantischen Standpunkt besteht ein wesentlicher Unterschied. Der Zugehörigkeitswert $\mu_A(x)$ mißt den Grad, mit welchem ein Element x zur Menge A gehört. Der Zugehörigkeitsgrad ist nicht die Wahrscheinlichkeit, mit der x zur Menge A gehört. (Ein angebissener Apfel gehört beispielsweise mit einem Grad von 0,9 zur Menge Äpfel, aber was ich esse, ist mit 100%-er Wahrscheinlichkeit ein Apfel.)

III.2.2 Basisverknüpfungen von Fuzzy-Mengen und Fuzzy-Logik

Die logischen Operatoren, die die unscharfen Mengen verknüpfen, sind nicht durch duale Wahrheitstabellen darstellbar, da $\mu(x)$ im normierten Fall jeden Wert zwischen 0 und 1 annehmen kann. Als mengentheoretische Basisoperatoren für Fuzzy-Mengen verwendet Zadeh folgende Operatoren:

$$\begin{array}{ll} \text{Durchschnitt:} & \mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \} \quad \text{für alle } x \in X \\ \text{Vereinigung:} & \mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \} \quad \text{für alle } x \in X \\ \text{Komplement:} & \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \text{für alle } x \in X \end{array}$$

Diese Operatoren zum Rechnen mit unscharfen Mengen können als Generalisierung der klassischen, zweiwertigen Operatoren angesehen werden. Zu betonen ist, daß diese Operatoren nicht die einzig möglichen sind, um unscharfe Mengen zu verknüpfen. Fuzzy Logik enthält also die klassische Logik als Grenzfall und erhebt somit mengentheoretisch einen allgemeineren Anspruch.

Was Zadeh aufgrund seiner Überlegungen über Unschärfe und Unsicherheit entwickelt hat, ist eine mehrwertige, kontinuierliche Logik. Solche Logiken sind schon seit längerer Zeit bekannt und gebräuchlich. Bereits 1922 diskutierte J. Lukasiewicz eine Logik mit Wahrheitswerten aus dem Einheitsintervall als Verallgemeinerung seiner dreiwertigen Logik, in der er neben den Werten 0 für falsch und 1 für

wahr auch den Wert 1/2 für unbestimmt betrachtete. Trotzdem hat Zadeh für sein mehrwertiges Mengenkalcul den neuen Begriff „fuzzy“ eingeführt, denn er hat vorausgesehen, daß seine *fuzzy set theory* der Grundstein für Konzepte und Techniken sein wird, deren Anwendungspotential weit über die reine Mathematik hinausgeht.

III.2.3 Fuzzy Control

Als eine der wichtigsten Anwendungen dieser unscharfen Logik ist Fuzzy Control oder Fuzzy-Regelung seit Mitte der siebziger Jahre zu Bedeutung gelangt. Fuzzy Control stellt eine neue Vorgehensweise zur Steuerung von komplexen, nichtlinearen Systemen dar. Durch die gezielte Verwendung imperfekter Informationen wird versucht, den Vorteil einer damit verbundenen Komplexitätsreduktion auszunutzen. Zadeh macht das an einem Beispiel aus dem Alltag deutlich: Das Einparken mit dem Auto ist ein Vorgang, der exakt vom eingeschlagenem Lenkrad, Geschwindigkeit usw. abhängt. Aber das Ziel, nämlich in vernünftiger Zeit in eine Parklücke zu fahren, kann nur erreicht werden, wenn bei der Ortsangabe auf Präzision verzichtet wird, denn wenn man mit dem Auto auf den Millimeter genau einparken müßte, würde man beliebig oft probieren müssen. Zadeh in [Reusch, 1993]:

The operative principle of fuzzy logic is: precision is costly. Minimize the precision needed to perform a task.

Eine zentrale Idee für die meisten Anwendungen von Fuzzy Logik ist das Konzept des unscharfen Schließens. Dazu wird das Wissen über das Verhalten eines Systems mittels linguistischer Ausdrücke in Regeln gefaßt, die der Fuzzy-Control-Formalismus auswertet. So versucht man das Input-Output-Verhalten eines Problems zu ermitteln, weil man zum Beispiel die dazugehörige Differentialgleichung nicht kennt. Genau so eine Aufgabe stellt die Bewegungssteuerung dar. Denn Stehen, Hüpfen, Laufen, Radfahren können die meisten Menschen, vielleicht auch noch die im Körper bewußt ablaufenden Vorgänge beschreiben, aber niemand kann die Steuerungsvorgänge in mathematische Funktionen fassen.

Im nächsten Abschnitt werden kurz die Grundzüge von Fuzzy-Regelung in der Form erklärt, wie ich sie zur Steuerung von biomechanischen Mehrkörpermodellen verwendet habe.

III.2.4 Wie funktioniert Fuzzy Control?

Von zentraler Bedeutung für die Kontrolle eines physikalischen Systems durch Fuzzy Control ist die Darstellung regelhafter Zusammenhänge in der Form:

IF <condition> THEN <action>

oder anders ausgedrückt für biologische Systeme

IF <stimulus> THEN <response>.

Die Ein- und Ausgangsgrößen eines Fuzzy Controllers nennt man linguistische Variablen. Entwirft man einen Regler, muß der Wertebereich und die Partitionierung der n linguistischen Variablen am Eingang festgelegt werden, die dann als Prämissen in den Regeln auftauchen. Genau das gleiche gilt für die l Konsequenzen B . Für jede linguistische Variable x_j werden auf der Basismenge X_j die p_j verschiedenen unscharfen Mengen $\mu_1 \dots \mu_{p_j}$ definiert und jede dieser Mengen mit einem linguistischen Term $A_1 \dots A_{p_j}$ assoziiert. Unter linguistischen Termen versteht man umgangssprachliche Ausdrücke wie *mittelgroß*, *etwa Null*, *ziemlich schnell*, *fast senkrecht* ..., mit denen die Werte der linguistischen Variablen qualitativ beschrieben werden können.

Die m Regeln verknüpfen die n Prämissen mit den l Konsequenzen in folgender Form:

if x_1 is A_1^1 and x_2 is A_2^1 ... and x_n is A_n^1 then y_1 is B_1^1 and ... y_l is B_l^1
: : : : :
if x_1 is A_1^m and x_2 is A_2^m ... and x_n is A_n^m then y_1 is B_1^m and ... y_l is B_l^m (III.2)

Die Auswertung der Regeln erfolgt durch eine Entscheidungslogik. Dieser Vorgang wird auch unscharfes Schließen oder Inferenz (decision making, inference) genannt.

Für ein System mit nur einer Konsequenz y wird dazu folgendermaßen vorgegangen. Der Wert α^i gibt den Erfüllungsgrad der i -ten Regel an, deren Prämissen konjunktiv verknüpft wurden und bestimmt somit die Gewichtung bzw. Aktivierung der Konsequenz dieser Regel:

$$\alpha^i = \min \{ \mu_1^i(x_1), \dots, \mu_n^i(x_n) \} \quad (III.3)$$

$$\mu_{out}^i(y) = \alpha^i \mu^i(y) \quad (III.4)$$

Nachdem die Entscheidungslogik jede einzelne Regel ($1 \leq i \leq m$) ausgewertet hat, muß sie die, bei der Auswertung der einzelnen Regeln erhaltenen Mengen, zu einer unscharfen Menge μ_{out} vereinigen:

$$\mu_{out}(y) = \max \{ \mu_{out}^1(y), \dots, \mu_{out}^m(y) \} \quad (III.5)$$

Aus dieser unscharfen Menge muß nun ein scharfer Wert y_{out} gewonnen werden, der als Stellwert verwendet werden kann. Eine gebräuchliche Methode für dieses sogenannte Defuzzifizieren ist die Schwerpunktmethode, d. h. der Flächenschwerpunkt der Vereinigungsmenge μ_{out} ist der gesuchte Endwert.

$$y_{out} = \frac{\int_{y \in Y} y \mu_{out}(y) dy}{\int_{y \in Y} \mu_{out}(y) dy} \quad (III.6)$$

Abbildung III.3 zeigt an einem willkürlichen Beispiel die Vorgehensweise. Jede der zwei Prämissen x_1, x_2 ist durch drei unscharfe Mengen kodiert und die Konsequenz y durch fünf unscharfe Mengen. Zwei Regeln der Regelbasis werden exemplarisch ausgewertet. Die dicken Striche in Abb. III.3 (a) und (b) zeigen an, welche x_1, x_2 Werte (z. B. Sensorsignale) im Moment als Regelgrößen anliegen. Sie werden fuzzifiziert, d. h. in Zugehörigkeitsgrade der unscharfen Mengen ausgedrückt. Mit diesen Werten werden jetzt alle Regeln ausgewertet. Man sagt die Regeln „feuern“. Sie feuern parallel, aber alle nur zu einem gewissen Grad. Die drei Diagramme einer Zeile zeigen die Auswertung einer Regel. Nach Gleichung (III.3) wird das Minimum der Zugehörigkeitswerte der Prämissen bestimmt und nach Gleichung (III.4) mit der unscharfen Menge multipliziert, die die Regel als Konsequenz vorsieht. Das letzte Diagramm zeigt die unscharfe Ausgangsmenge, die die Entscheidungslogik aus allen Regeln nach Gleichung (III.5) erzeugt. Der Pfeil zeigt an, wo der Flächenschwerpunkt liegt, d. h. welcher y -Wert nach der Defuzzifizierung als Stellgröße des Reglers ausgegeben wird.

Dieser intuitiv motivierte Ansatz geht auf Mamdani zurück, er garantiert glattes Reglerverhalten und kann auch auf einer formalen Basis hergeleitet werden [Kruse, 1993].

Im nächsten Abschnitt werden am Beispiel Balancieren mit drei Gelenken Fuzzy-Regelung und ein Kohonen-Netz als Vertreter für künstliche Neuronale Netze erklärt (ohne Anspruch auf Vollständigkeit oder Allgemeingültigkeit) und zur Steuerung des mechanischen Computermodells angewendet.

III.2.5 Beispiel: Balancieren auf drei Gelenken

Beim ersten Beispiel für den Einsatz von Fuzzy-Logik in Verbindung mit biomechanischen Mehrkörpersystemen geht es um die Berechnung der inneren Momente, die das Stehen des dreigelenkigen Modells in einer vorgegebenen Haltung ermöglichen. Dieses einfache Beispiel soll auch dazu dienen, die Unterschiede zu anderen Verfahren wie PID-Reglern und Neuronalen Netzen sichtbar zu machen und insbesondere Vor- und Nachteile aufzuzeigen. Die biologische Bedeutung der Problemlösung sei hier zurückgestellt.

Eingang:

Ausgang:

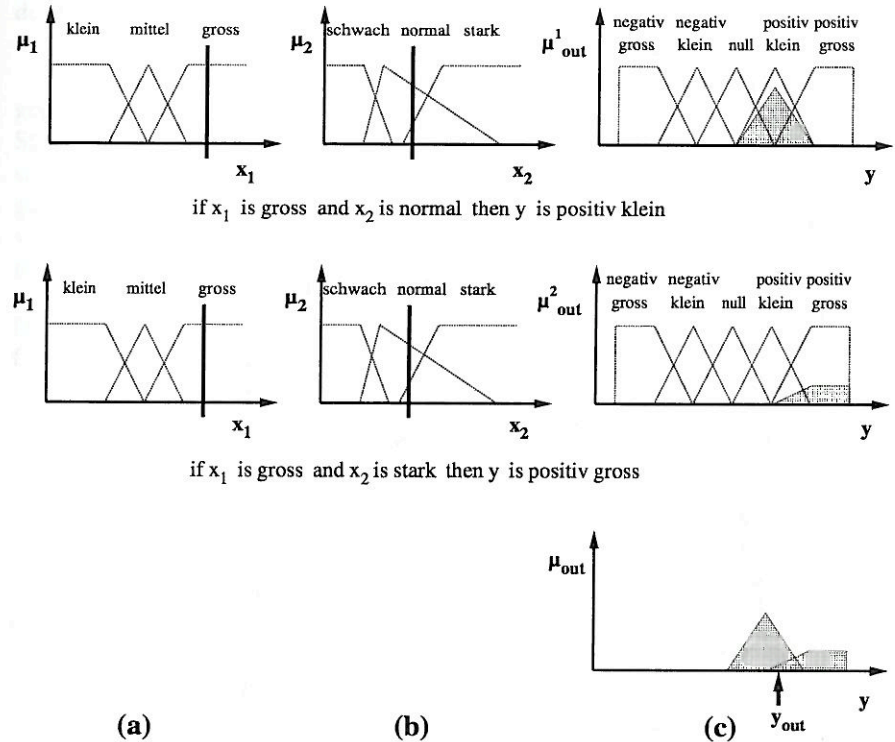


Abb. III.3: Beispiel für die Entscheidungsfindung (Inferenz) eines Fuzzy Controllers: Zwei Eingangsgrößen sind durch jeweils 3 Zugehörigkeitsfunktionen kodiert. Die dicken Striche sind die scharfen Regelgrößen. Zwei Regeln werden ausgewertet und die unterlegten Mengen stellen das Ergebnis der Minimum-Operationen dar. Das letzte Bild stellt das Endergebnis nach der Vereinigung als unscharfe Menge dar. Der scharfe Stellwert (Pfeil) wird durch Flächenschwerpunktbestimmung ermittelt.

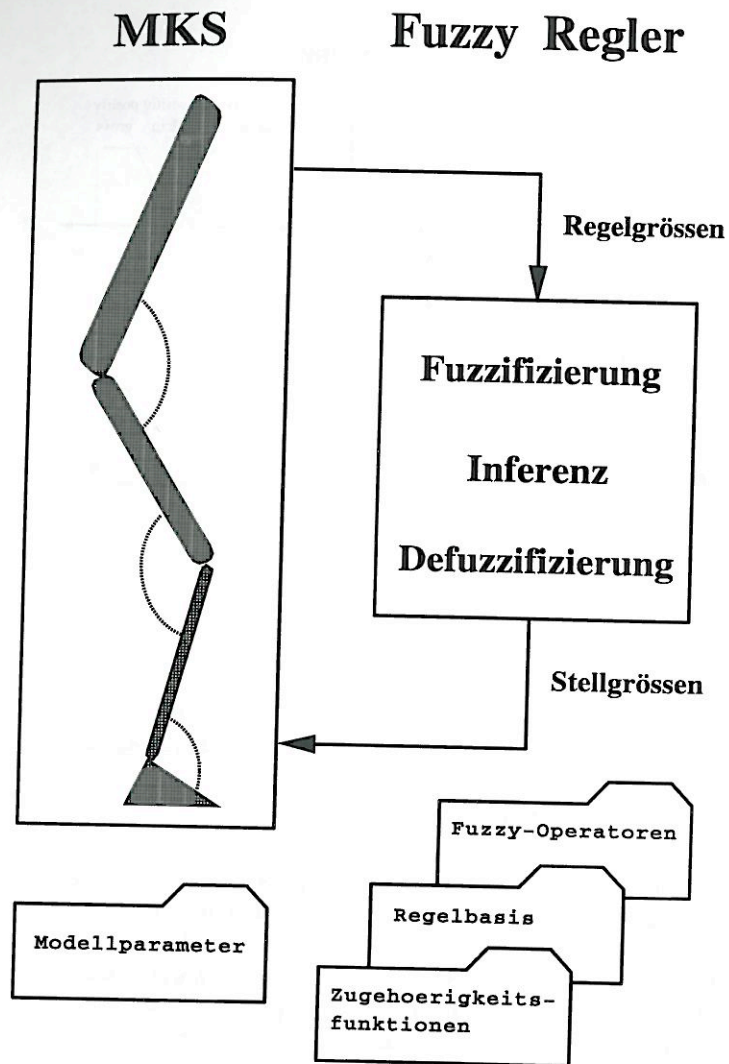


Abb. III.4: Mehrkörpersystem mit Prinzip des Fuzzy-Reglers.

Entwurf des Reglers

Das Regelungsproblem ist so gestellt, daß es sich für jedes der drei Gelenke um eine Balancieraufgabe nach dem Prinzip des invertierten Pendels handelt. Als Eingangsgrößen dienen jeweils Winkel und Winkelgeschwindigkeit. Die drei Momente in den Gelenken, die zum Stabilisieren in einer bestimmten, vorgegebenen Körperhaltung benötigt werden, werden dann unabhängig voneinander berechnet.

Der Entwurf des Reglers, d. h. die Festlegung der unscharfen Mengen und Regeln basiert auf Vorwissen (z. B. Meßwerte) und auf intuitivem Verständnis des Steuerungsproblems. Der menschliche Experte verfügt über heuristische Kenntnisse der zum Balancieren nötigen Steueraktionen und die Regeln der Balancieraufgabe sind gegeben durch Überlegungen der Art: „Wenn der Oberkörper sehr weit vorhängt und die Geschwindigkeit nach vorne zeigt, dann wirke ein stark aufrichtendes Moment in der Hüfte.“

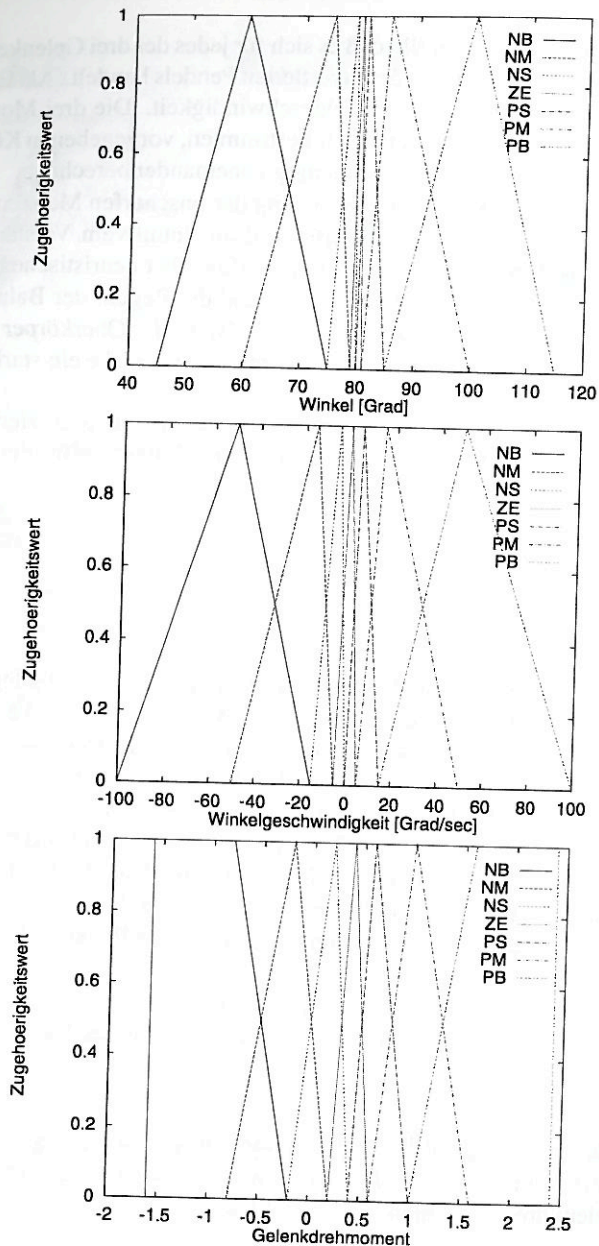
Als erstes müssen die umgangssprachlichen Ausdrücke quantifiziert werden. Dazu wähle ich für Ein- und Ausgangsgrößen jeweils 7 dreiecksförmige unscharfe Mengen, die ich mit folgenden Termen bezeichne:

NB (negative big)	PB (positive big)
NM (negative medium)	PM (positive medium)
NS (negative small)	PS (positive small)
ZE (zero)	

Dann muß die Lage und Form der Zugehörigkeitsfunktionen festgelegt werden. Abb. III.5 zeigt die Kodierung für die Prämissen Winkel φ und Winkelgeschwindigkeit $\dot{\varphi}$ sowie die Konsequenz Gesamtmoment M für das Hüftgelenk. Die Zahl 7 für die unscharfen Mengen war zwar willkürlich gewählt, aber so, daß die Zahl der Regeln überschaubar bleibt. Die Lage der unscharfen Mengen ist um den Bereich des Sollwinkels (liegt bei ZE) verteilt. Die unscharfen Mengen für die Winkelgeschwindigkeit sind um die Geschwindigkeit Null verteilt, denn das Modell soll zum Stillstand kommen. Die Mengen liegen im Bereich der 0 dichter, denn der Wechsel der Bewegungsrichtung ist beim Balancieren ein kritischer Punkt und soll deshalb mit mehr Regeln abgedeckt werden.

Die drei Gelenke haben etwas unterschiedliche Wertebereiche für die Zugehörigkeitsfunktionen, da die auftretenden Winkel und Winkelgeschwindigkeiten et-

Abb. III.5: (Seite 32) Zugehörigkeitsfunktionen für die Steuerung in der Hüfte: **Oben:** Rumpfwinkel in Grad, **Mitte:** Rumpfwinkelgeschwindigkeiten in Grad/s, **Unten:** Hüftgelenkdrehmoment in Nm/Modellmasse.



was unterschiedlich sind, aber die Regeln lauten für das Stehen für jedes Gelenk gleich, denn die Idee des Balancierens gilt gleichermaßen.

Systematisch kann eine Regelbasis für alle möglichen Kombinationen am Eingang aufgestellt werden. Diese 7×7 Regeln pro Gelenk kann man in einer sogenannten linguistischen Matrix darstellen:

Winkel vs. Winkelgeschwindigkeit

	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	ZE
NM	PB	PM	PS	PS	PS	ZE	NS
NS	PM	PS	PS	PS	ZE	NS	NS
ZE	PM	PS	PS	ZE	NS	NS	NM
PS	PS	PS	ZE	NS	NS	NS	NM
PM	PS	ZE	NS	NS	NS	NM	NB
PB	ZE	NS	NS	NM	NM	NB	NB

Der erste Matrixeintrag oben links entspricht also der oben genannten Regel. Die mittlere Regel entspricht dem Gleichgewicht in der Sollwinkelstellung, die angestrebt wird.

Die Anzahl der unscharfen Mengen, ihre Lage und ihre Form wurden zunächst durch Erfahrung festgelegt. Weitere Abschnitte dieser Arbeit werden Methoden erklären, die solche Parameter systematisch bestimmen und optimal einstellen können.

Da eine Fuzzy-Regelung vollkommen deterministisch ist, kann man ausrechnen und aufzeichnen (Abb. III.6), wie die Input-Output-Charakteristik für ein Gelenk aussieht. Deutlich erkennt man, wie die Fläche stückweise durch die Regeln aufgebaut wird. Die Kanten verraten auf welchen Achsenabschnitten die unscharfen Mengen liegen. Die Sollstellung liegt in der Mitte der Kontrollfläche. Bei Abweichungen von Sollwinkel und/oder Sollwinkelgeschwindigkeit innerhalb des durch die unscharfen Mengen abgedeckten Wertebereichs, werden durch die Regeln Momente berechnet, die das Modell in die gewünschte Stellung bewegen.

Visualisierung

Der Regler funktioniert nun wie in Abbildung III.4 schematisch angedeutet. Zu jedem Zeitschritt einer Simulation werden für jedes Gelenk die Regelgrößen fuzzifiziert, d. h. Winkel und Winkelgeschwindigkeit werden als Zugehörigkeitsgrade ausgedrückt. Dann wird die Inferenz ausgeführt und zum Schluß werden durch Defuzzifizieren die Werte für die inneren Momente ermittelt (Stellgrößen), mit denen ein Simulationsschritt ausgeführt wird.

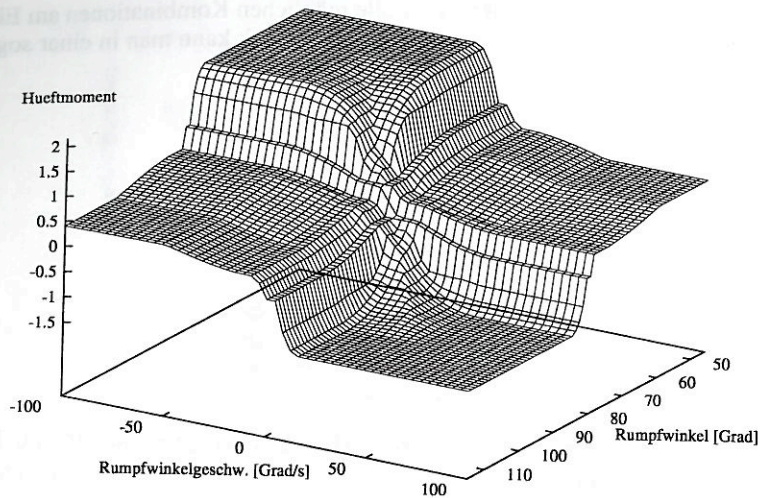


Abb. III.6: Input-Output-Charakteristik für das Hüftgelenk (Gelenkoment in Nm/Modellmasse)

Abb. III.7 zeigt eine Momentaufnahme des Modells und der Regeln während einer Simulation. Das System hat insgesamt $3 \times 7 \times 7$ Regeln für die drei Gelenke. Ein Eingangswert wird in einem logischen Schritt durch alle Regeln geschickt, die alle parallel, aber nur zu einem gewissen Grad aktiv sind. Durch die Visualisierung mittels Schieberegler, die wie obige Regelmatrix angeordnet und beschriftet sind, kann man auf einen Blick zu jedem Zeitpunkt sehen, welche Regeln zu welchem Grad aktiv sind. In diesem Beispiel mit 2 Eingangsvariablen und der speziellen Wahl von unscharfen Mengen, von denen sich immer zwei überlappen, können maximal vier Regeln gleichzeitig aktiviert sein, das heißt zu einem Grad α^i (Gleichung III.3) ungleich Null sein.

Diese übersichtliche Matrixdarstellung ist allerdings auf Systeme mit nur 2 Eingangs- und 1 Ausgangsvariablen beschränkt. Es ist aber durchaus wünschenswert mehr Prämissen zu haben. Beispielsweise um zusätzlich Informationen eines benachbarten Gelenks einzuberechnen.

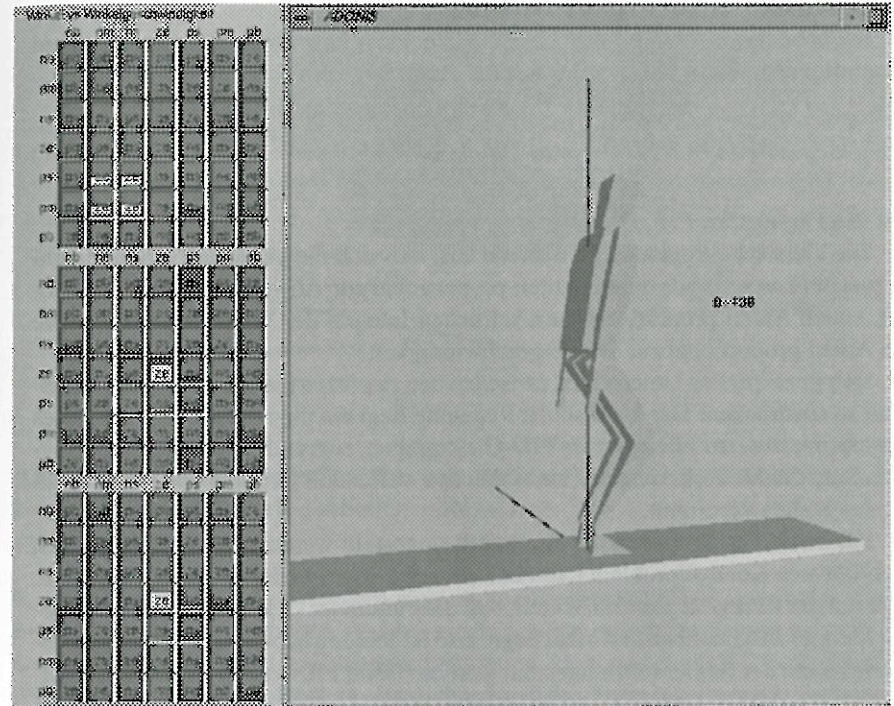


Abb. III.7: Visualisierung mit GL und FORMS auf einer SGI-Workstation. Für jedes der 3 Gelenke gibt es ein Tableau mit 7×7 Regeln. Zu jeder Regel gehört eine Schieberegler, die anzeigt, zu welchem Grad die Regel aktiviert ist (hell=aktiviert, dunkel=nicht aktiviert).

III.2.6 Vergleich mit PID-Regler

Das vorgestellte Beispiel eignet sich gut dazu, den konzeptionellen Unterschied verschiedener Methoden zu erfassen. Im folgenden werden die wichtigsten Unterschiede zwischen klassischer Regelung (PID), regelbasiertem Fuzzy-System und im nächsten Abschnitt neuronalem Netz mit unüberwachtem Lernen (Kohonen-Netz) angesprochen.

Ich habe das Stehen mit drei Gelenken auch auf konventionelle Art mit einem Proportional-Integral-Differential-Regler, durch folgende Gleichung bearbeitet:

$$M(t) = P(\varphi(t) - \varphi_0) + I \int_{t_0}^t (\varphi(\tau) - \varphi_0) d\tau + D\dot{\varphi}(t), \quad (\text{III.7})$$

mit den Parametern P , I , D und dem Sollwinkel φ_0 .

Die klassische Regelung geht davon aus, daß sich die gesuchten Drehmomente zusammensetzen aus einem Anteil proportional zur Abweichung vom Sollwinkel, einem Anteil proportional zum zeitlichen Integral der Winkeldifferenz sowie ein Anteil proportional zur Winkelgeschwindigkeit.

Den prinzipiellen Unterschied zwischen den Ansätzen kann man zusammenfassend so ausdrücken: Der klassischen Regelung liegt ein physikalisches Modell des Regelprozesses, zum Beispiel als PID-Gleichungen, zugrunde, während die Fuzzy-Regelung die Modellierung des menschlichen Verhaltens in Form von Faustregeln für das Stehen vornimmt.

Für beide Verfahren müssen Parameter eingestellt werden. Für ein komplexes, nichtlineares Kontrollproblem kann das ein schwerwiegendes Problem darstellen. Hier hat der Fuzzy-Regler den Vorteil, daß die Parameter (z. B. Lage einer unscharfen Menge) anschauliche und lokal begrenzte Bedeutung haben. Durch lokales Eingreifen kann der Regler fein eingestellt werden. Beim PID-Regler kommt es im allgemeinen bei fehlerhaft eingestellten Parametern P , I oder D zu sich aufschaukelnden Fehlern und damit zum Totalversagen der Regelung.

Diese Eigenschaft des Fuzzy-Reglers kann durch folgendes Vorgehen veranschaulicht werden. Streut man wissentlich falsche Informationen in das Fuzzy-System indem man die Regeln sabotiert, enthält ein Teil des Kontrollraums fehlerhafte Werte. In Abb. III.8 wurden im Vergleich zu Abb. III.6 zwei zentrale Regeln mit falschen Konsequenzen versehen. In der Simulation macht sich diese Manipulation so bemerkbar, daß das System keine stabile Endlage findet, d. h. das Modell bewegt sich wie „betrunken“.

Die wichtigsten Ziele des Fuzzy-Kalküls sind Fehlertoleranz und Robustheit. Stimmt die Annahme nicht ganz, so folgt nicht, daß die Ergebnisse der auf ihr beruhenden Schlußfolgerungen unbrauchbar werden.

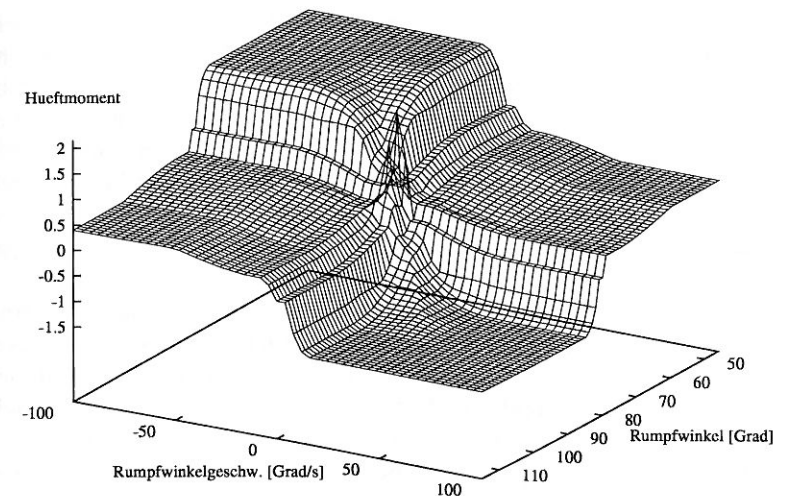


Abb. III.8: Input-Output-Charakteristik für das Hüftgelenk (Gelenkmoment in Nm/Modellmasse) mit zwei sabotierten Regeln.

Viele Regelprobleme erfordern eine on-line Steuerung. Der Regler sollte also möglichst in Echtzeit arbeiten, was bei einem System mit vielen Regeln normalerweise nicht der Fall ist. Um Rechenzeit zu sparen, kann man die I/O-Charakteristik (Kennfeld) eines Fuzzy-Reglers abspeichern und das Ergebnis nur noch auslesen, anstatt zu jedem Zeitschritt viele Regeln auszuwerten. Außerdem kann man das Kennfeld mit Splines glätten, wenn die Kanten unerwünscht sind, die durch die endliche Anzahl Regeln bedingt sind, und reduziert darüber hinaus die Datenmenge.

III.2.7 Vergleich mit Kohonen-Netz

Netzwerktypen und Trainingsmethoden

Künstliche Neuronale Netze bestehen aus diskreten Einzelementen, den Neuronen, die in einer Struktur angeordnet sind und sich gegenseitig beeinflussen. Heute gibt es circa 25 gebräuchliche Netzwerktypen und Trainingsmethoden. Eine

gute Übersicht der Komponenten und Architekturen neuronaler Modelle gibt Zell (1994).

Training heißt, daß die Gewichtungen der Wechselwirkungen zwischen Neuronen nach einer Lernregel trainiert werden. Ein trainiertes Netz kann als statisches System für die Lösung der vorgegebenen Aufgabe eingesetzt werden. Keine der heute bekannten Trainingsmethoden entspricht biologischem Lernen. Von einem System mit kognitiven Fähigkeiten sind künstliche Neuronale Netze um Größenordnungen entfernt, was die Zahl der Synapsen und die Geschwindigkeit angeht. Wenn ich im folgenden die Begriffe „Lernen“ oder „Training“ benutze, so ist damit nicht menschliches Bewegungslernen gemeint, sondern ausschließlich der Anpassungsvorgang in einem künstlichen Neuronalen Netz.

Wichtig ist die Unterscheidung zwischen Verfahren mit überwachtem Lernen (*supervised learning*) und unüberwachtem Lernen (*unsupervised learning*). Das erste basiert auf Trainingsdaten, die gelernt werden. Das trainierte Netz soll diese Daten reproduzieren können und neue Daten richtig interpretieren können. Beim unüberwachtem Lernen werden meist zufällig Trainingssituationen erzeugt und ein aufgabenabhängiges Bewertungskriterium beurteilt die Netzaktionen und steuert so den Lernvorgang.

Eine der wenigen Anwendungen von Neuronalen Netzen in der Biomechanik basiert auf überwachtem Lernen. Sepulveda (1993) benutzt ein dreilagiges Neuronales Netz mit Backpropagation um die Korrelation zwischen Elektromyogrammen und Gelenkwinkel bzw. -momente von Ganganalysedaten zu repräsentieren. Dazu werden EMG-Daten von 16 Muskeln der unteren Extremitäten auf drei Gelenkwinkel oder -momente abgebildet. Nach 54 000 Iterationen reproduziert das Netz die Trainingsdaten exakt. Für leicht veränderte Werte im Eingangsvektor (z.B. andere Muskelstärken) berechnet das trainierte Netz einen Output-Vektor (Winkel oder Momente), der plausibel erscheint. Weiterreichende Erkenntnisse können aus dem trainierten Netz nicht gewonnen werden.

Kohonen-Netz

Da ich das Steuerungsproblem beim Balancieren mit drei Gelenken ohne Vorgabe von Trainingsdaten lösen will, kommt nur ein Netz mit unüberwachtem Lernen in Frage. *Kohonen feature maps* oder *selbstorganisierende Karten* oder einfach kurz Kohonen-Netze arbeiten mit unüberwachtem Lernen. Es handelt sich dabei um Netze mit nur einer Schicht Neuronen, die allerdings in einem Gitter angeordnet sind. Das besondere an Kohonen-Netzen ist, daß eine Nachbarschaftsbeziehung zwischen den Neuronen definiert wird. Die Neuronen stehen also nicht in beliebiger gegenseitiger Verbindung, sondern die lateralen Wechselwirkungen sind

räumlich begrenzt und die Neuronenschicht wirkt als topographische Merkmalskarte. Die Erweiterung von Kohonens ursprünglichen Lernalgorithmus für Klassifizierungsprobleme durch Ritter und Schulden [Ritter, 1991] macht diesen Netztyp für Regelungsaufgaben brauchbar. Erst durch die Speicherung eines Ausgangswertes durch jedes Neuron entspricht die Bildung einer Kohonen-Karte dem Erlernen einer Steuerungsaufgabe.

Konkret habe ich ein zweidimensionales Kohonen-Netz mit einer Schicht aus 25×25 Neuronen programmiert. Der Systemzustand des Modells ist zu jedem Zeitpunkt durch die Winkel und Winkelgeschwindigkeiten der drei Gelenke gegeben. Entsprechend gehören zu jedem Neuron sechs verschiedene Synapsenstärken. Ein Eingangssignal für das Neuronengitter ist deshalb ein Vektor mit sechs Komponenten. Das Ausgangssignal sind die in den drei Gelenken benötigten Drehmomente, d. h. zu jedem Gitterpunkt im Kohonen-Netz gehört ein Ausgangsvektor mit drei Komponenten.

Die Algorithmen, nach denen sich die Merkmalskarte ausbildet und die Lernzyklen ablaufen, wurden in Anlehnung an Ritter und Schulden [Ritter, 1991] entwickelt. Zu jedem Zeitschritt einer Simulation wird das Neuron bestimmt, dessen Synapsenstärken dem Eingangssignal am ähnlichsten sind. Dazu wird bezüglich der euklidischen Vektornorm das Neuron mit der minimalen Differenz berechnet. Der Ausgangsvektor an diesem Neuron (Erregungszentrum) wird dann aus schon gelerntem Ausgangswert und einer zufälligen Komponente zusammengesetzt. Die zufällige Komponente ist um so kleiner, je länger schon gelernt wurde und außerdem auch, je besser das Eingangssignal auf ein Neuron paßt.

Durch eine gaußförmige räumliche Nachbarschaftsbeziehung werden bei Änderungen des Gewichtsvektors am Erregungszentrum auch die umliegenden Neuronen im Gitter beeinflusst. Die Definition von lokaler Nachbarschaft muß im Laufe des Lernens angepaßt werden, das bedeutet der Nachbarschaftsradius wird kleiner, je mehr schon gelernt wurde, damit das Gelernte nicht überschrieben wird.

Neurophysiologischer Hintergrund für die Anwendung eines zweidimensionalen Gitters auf das Problem der Bewegungssteuerung ist, daß auch im Motorkortex Bewegungskommandos auf zweidimensionale Erregungsorte abgebildet sind. Die Nachbarschaftsrelation bewirkt einen Zusammenhang zwischen Erregungsort und Bewegung oder anders ausgedrückt benachbarte Muster werden auf benachbarte Neuronen abgebildet (topologieerhaltende Abbildung).

Wie sieht nun in der Simulation ein unüberwachter Lernprozeß aus? Der Lernvorgang beginnt mit zufällig vorgegeben Anfangssynapsenstärken und geht in Zyklen vor sich. Es gibt keine Trainingsdaten, sondern eine Bewertungsfunktion beurteilt den Lernerfolg jedes Simulationsschritts. In diesem Beispiel sind die Bewertungskriterien die Lage des Gesamtkörperschwerpunkts und die Gelenkwinkelstel-

Hueftmoment

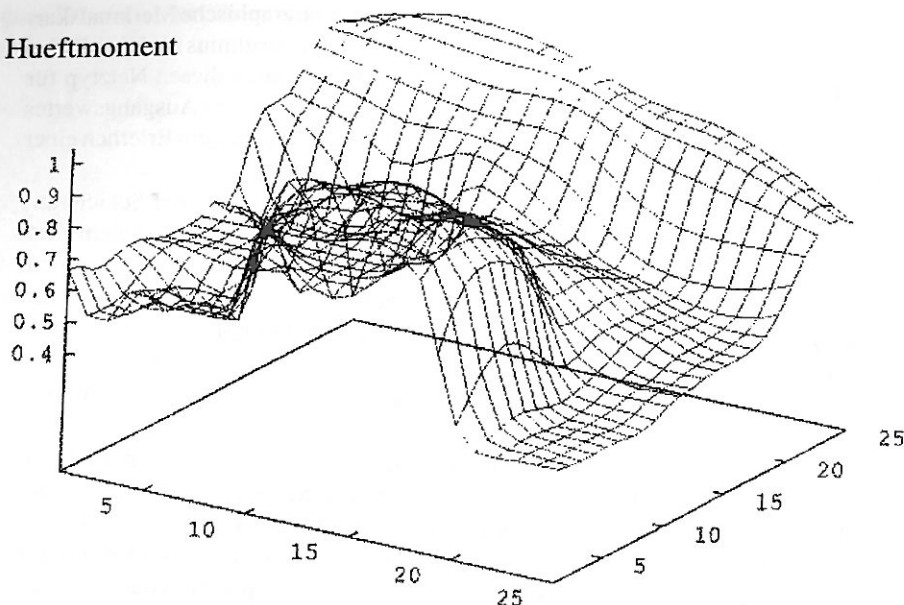


Abb. III.9: Hüftgelenkmoment (in Nm/Modellmasse) als Komponente des Ausgabevektors über dem 25×25 Gitter aufgetragen. Das Kohonen-Netz ist offensichtlich nur unzureichend trainiert.

lungen. Gelernt wird nur, wenn eine Bewegung besser bewertet wird als in den vorgegangenen Simulationen. Ein Abbruchkriterium bewirkt, daß Simulationen, die zu einer aussichtslosen Bewegung führen, sofort abgebrochen werden. Es werden solange neue Simulationen gestartet, bis eine bestimmte Anzahl Lernerfolge erzielt wurde. Die ersten Simulationen brechen meist schon nach wenigen Zeitschritten ab, denn die zufälligen Anfangsmomente bringen kaum sinnvolle Steueraktionen hervor. Werden aber einige Lernschritte erzielt, bildet sich durch die Nachbarschaftsrelation schnell eine Grobstruktur im Netz aus, welche sich dann laufend verbessert. Der Lernprozeß verlangsamt sich, wenn sich die Zahl der Lernschritte der vorgegebenen Maximalzahl nähert.

Eine ausführliche Darstellung der Arbeitsweise eines Kohonen-Netzes findet sich in Erhardt (1992). Mit einem dreigliedrigen Modell mit festgehaltenem Fußpunkt, also ohne Fersenpolster, hat M. Erhardt das Aufstehen aus der Hocke mit einem Kohonen-Netz simuliert. Bei dieser Bewegung überschneiden sich die Trajek-

torien im Winkel-Winkelgeschwindigkeitsraum nicht.

Ich konnte das Kohonen-Netz trainieren, so daß die gestellte Aufgabe, nämlich das Balancieren des dreigliedrigen Modells mit verschiedenen Störungen, für manche Anfangsbedingungen (aber nicht alle) richtig gelöst wurde. Im Balancierbeispiel zeigte sich schon nach 1000 Lernschritten ein Ergebnis. Trotzdem war das beste trainierte Netz, das ich erhielt, schlecht im Vergleich zum Ergebnis mit Fuzzy-Regelung. Ein wesentlicher Nachteil von Neuronalen Netzen ist, daß die Vorgänge im Netz während des Lernens nicht einsehbar und nicht beeinflussbar sind. Neuronale Netze bleiben also „black boxes“ und der Output kann nicht wie beim Fuzzy-Regler zu den Bestandteilen wie Fuzzy-Mengen und Regeln, die wiederum einen physikalischen Gehalt haben, zugeordnet werden. Häufig sah das Ergebnis eines Lernvorgangs wie in Abbildung III.9 aus. Ein gut trainiertes Netz sollte analog zur Abb. III.1 dem Kennfeld aus Abb. III.6 ähnlich sehen. Das Netz hat zwar offensichtlich im Vergleich zum rein zufälligen Zustand, mit dem es initialisiert war, etwas dazugelernt, ist in dieser Form aber unbrauchbar, weil man es lokal nicht weiter verbessern kann. Außerdem ist nicht klar, ob es sich nur um falsch eingestellte Netz- und Lernparameter handelt oder um ein prinzipielles Problem.

Kleine Bewegungen aufgrund des deformierbaren Fersenpolsters haben beim Lernprozess des Kohonen-Netzes erhebliche Schwierigkeiten bereitet. Im Gegensatz dazu ist schon im Entwurf des Fuzzy-Reglers berücksichtigt, daß kleine Änderungen am Eingang kleine Änderungen am Ausgang bewirken. Bei Änderung von äußeren Bedingungen, wie z. B. Stehen in einer anderen Position, muß ein Netz komplett neu gelernt werden. Beim Fuzzy-Regler muß man nur den Wertebereich der unscharfen Mengen verschieben und skalieren.

Da das mechanische Computermodell den Menschen repräsentieren soll und keinen Roboter, der Motoren in Gelenken hat, die stark genug sind jede Winkelstellung zu realisieren, habe ich versucht, beschränkte Drehmomente in das Bewertungskriterium einzuführen. Diese Nebenbedingung hat den Lernvorgang negativ beeinflusst, so daß sich kein stabiler Netzzustand mehr ausgebildet hat.

Viele dieser Schwierigkeiten hätten sich vielleicht mit etwas Aufwand und fortgeschrittenen Netzwerk-Methoden überwinden lassen. Da ich mit Fuzzy-Reglung auf einfache und direkte Weise bessere Ergebnisse erhalten habe, habe ich diesen Weg weiterverfolgt. Außerdem ist bei der Anwendung auf Bewegungssteuerungsprobleme ohne Datenvorgabe mehr Erkenntnis durch ein Fuzzy-System zu erwarten. Warum Neuronale Netze dann wieder interessant sind, wenn sowohl eine funktionierende Fuzzy-Regelung als auch repräsentative Trainingsdaten vorliegen, erläutert der nächste Abschnitt.

III.2.8 Adaptive Fuzzy-Systeme

Ein Fuzzy-Regler ist ein System, dessen Parameter problemangepaßt einzustellen sind. Für ein physikalisches Steuerungsproblem ist es oft wichtig, daß der Regler leicht anpaßungsfähig ist. Deshalb werden im folgenden die zwei wichtigsten Möglichkeiten, ein sogenanntes adaptives Fuzzy-System zu entwerfen, kurz erklärt.

Fuzzy Control + Neuronale Netze

Für den Anwendungsbereich der Methoden ist von Bedeutung, in wie weit Fuzzy Control und künstliche Neuronale Netze funktionell äquivalent sind, unter welchen Aspekten sie grundsätzlich verschiedene Ansätze darstellen, wo die Vor- und Nachteile liegen und schließlich wie die Kombination der beiden Techniken aussehen kann.

Ein Fuzzy-Regler legt die I/O-Charakteristik eines Systems durch die Wahl der unscharfen Mengen, die Regelbasis und der Operatoren für die logischen Verknüpfungen fest. Ein Neuronales Netz, das aus formalen Neuronen in einer Netzarchitektur aufgebaut ist, lernt durch einen Lernalgorithmus die I/O-Charakteristik. Es bestehen also Analogien in den Verfahren in der Hinsicht, daß die Input-Output-Funktionen aufgrund numerischer oder linguistischer Daten abgeschätzt werden.

Ist die Struktur eines Problems bekannt, ist es natürlich von Vorteil, wenn dieses Wissen direkt in einem Fuzzy-System umgesetzt werden kann, wobei bei größeren Systemen die Zahl der Parameter und Regeln stark ansteigen kann. Deshalb werden in der Praxis hauptsächlich 2 Prämissen-1 Konsequenz-Systeme eingesetzt.

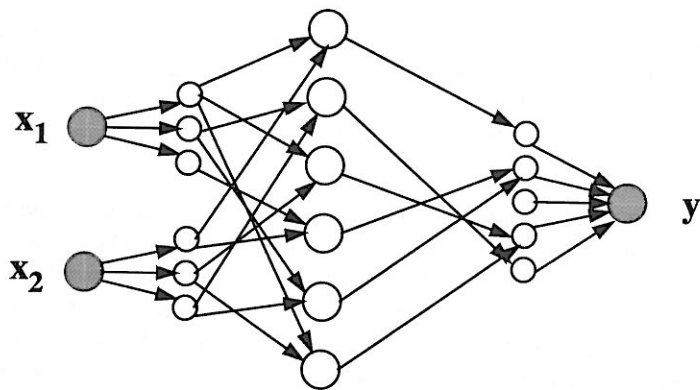


Abb. III.10: Beispiel für das Prinzip der Fuzzy-Neuro-Übersetzung.

Ist die Struktur nicht bekannt, liegen aber zum Training geeignete Daten vor, ist ein lernfähiges System wie ein Neuronales Netz von Vorteil. Dazu muß ein geeigneter Lernalgorithmus und ein passendes Netz gefunden werden. Allerdings ist das gelernte Wissen in der Regel nicht interpretierbar, so daß zum einen keine tiefere Einsicht in das gelernte Wissen resultiert. Zum anderen ist wegen dieser fehlenden Einsicht ein Nachoptimieren unmöglich.

Kombinationen von Fuzzy-Systemen und Neuronalen Netzen sollen die Vorteile beider Methoden vereinen und ihre Nachteile kompensieren. Solche *Fuzzy-Neuro*- oder *Neuro-Fuzzy*-Systeme sind Gegenstand aktueller Forschung.

Die funktionelle Äquivalenz läßt sich bei der Kombination ausnutzen, wenn man spezielle Netzwerk-Topologien benutzt, die es erlauben, das Netzverhalten durch Fuzzy-Regeln zu beschreiben und umgekehrt [Jang, 1993]. Anders ausgedrückt erfolgt die Adaption durch Konvertierung des Fuzzy-Systems in ein entsprechend strukturiertes Netz, das trainiert und dann wieder zurückkonvertiert wird. Eine schematische Darstellung der Fuzzy-Neuro-Umsetzung ist in Abbildung III.10 zu sehen. Kein Kohonen-Netz, sondern ein mehrschichtiges Netz repräsentiert durch die Gewichtung seiner Verbindungen das Fuzzy-System. Die erste Neuronenschicht entspricht den unscharfen Mengen der Prämissen, d. h. die Pfeile repräsentieren die Fuzzifizierung. Die mittlere Schicht entspricht der Regelbasis und die dritte Neuronenschicht repräsentiert die unscharfen Mengen der Konsequenz. Die Pfeile bedeuten Inferenz und Defuzzifizierung.

Ausgehend von einer existierenden Fuzzy-Regelung arbeitet ein solches Netz folgendermaßen: Anstatt mit zufälligen Anfangswerten wird das Netz mit Parametern initialisiert, die aus einer Fuzzy-Regelung stammen. Im nächsten Schritt wird das Netz anhand von Daten nachtrainiert. Während im Bereich von Trainingsdaten eine Anpassung stattfindet, bleibt in Bereichen ohne Daten der Initialzustand erhalten, was natürlich wesentlich besser ist als keine oder nur zufällige Informationen. Als letztes können aus den Netzparametern wieder Regeln extrahiert werden, die analysiert werden können und somit kann insbesondere das gelernte Wissen interpretiert werden.

Obwohl Neuronale Netze aufgrund der ursprünglich zugrunde liegenden Idee, das Gehirn in gewisser Weise nachzubilden, für die Steuerung in biomechanischen Systemen ausgezeichnet scheinen, gibt es doch gute Gründe auch Fuzzy-Regelung einzusetzen. Durch die Möglichkeit der Kombination sind Neuronale Netze und Fuzzy-Systeme aber nicht als konkurrierende, sondern als ergänzende Methoden zu sehen.

Allerdings sind für diese Neuro-Fuzzy-Systeme gute Trainingsdaten eine Hauptvoraussetzung, ich möchte aber Vorwärtssimulationen ohne Dateninput ausführen. Deshalb werden im folgenden Genetische Algorithmen vorgestellt, die un-

überwachtes Lernen darstellen, in dem Sinne, daß keine Trainingsdaten benötigt werden, sondern das Endergebnis einer Simulation bewertet wird.

Fuzzy Control + Genetische Algorithmen

Für das in Abschnitt III.2.5 behandelte Balance-Problem waren sowohl die Regeln bekannt als auch hinreichend gut die Wertebereiche der linguistischen Variablen. Diese Voraussetzungen sind aber bei den meisten realen Problemen nicht gegeben.

Ein anpassungsfähiges Fuzzy-System entsteht, wenn Parameter eines Fuzzy-Systems, oder gar eines Neuro-Fuzzy-Systems, mittels Genetischer Algorithmen optimiert werden. Unter Parameter verstehe ich Anzahl, Lage oder Breite von unscharfen Mengen oder auch die Konsequenz einer Regel.

Im nächsten Kapitel wird ausführlich erläutert, was man unter Genetischen Algorithmen versteht und wie sie funktionieren. Da es sich um eine Optimierungsmethode handelt, ist die Anwendung des adaptiven Systems aus Fuzzy Control und Genetische Algorithmen auf Probleme beschränkt, die sich durch ein oder mehrere Optimierungskriterien darstellen lassen.

Am Beispiel Standhochsprung wird in Abschnitt IV.1.2 der Einsatz von Genetischen Algorithmen zum Optimieren von Parametern eines Fuzzy-Systems gezeigt.

III.3 Optimieren mit Genetischen Algorithmen

Sowohl im Neuronalen Netz als auch im Fuzzy-System sind problemangepaßt Parameter einzustellen. Zum Beispiel um ein Netz zu konfigurieren, den Lernvorgang zu lenken, beziehungsweise unscharfe Mengen zu definieren, Fuzzy Operatoren festzulegen oder Regeln zu finden. In diesem Kapitel wird deshalb die Methode der Genetischen Algorithmen vorgestellt, die eine Möglichkeit darstellt, mit der solche Parameter durch Optimierung bestimmt werden können.

John Holland hat 1975 gezeigt, wie man evolutionäre Prozesse auf künstliche Systeme anwenden kann und hat Genetische Algorithmen als eine robuste, intelligente Suchtechnik in großen Parameterräumen eingeführt [Holland, 1975]. Was für lebende Organismen eine erfolgreiche Strategie ist, kann auch für das Optimieren mit einem Computer nützlich sein. Die Idee ist also, daß sich optimale Lösungen im Laufe der Evolution herausbilden, indem durch Kreuzung und Mutation neue Kombinationen von Erbanlagen entstehen, die einem Selektionsdruck ausgesetzt werden.

Optimieren heißt, das Minimum oder Maximum einer Funktion f zu suchen. Im folgenden werde ich einige Begriffe und die Funktionsweise von Genetischen Algorithmen erklären. Danach gehe ich auf die Leistungsfähigkeit und auf wesentliche Vorteile dieser Optimierungsmethode ein. Das ganze geschieht unter dem Gesichtspunkt, daß mein Ziel ist, durch eine Fuzzy Regelung eine bestimmte Bewegung mit dem mechanischen Mehrkörpermodell zu realisieren. Das Fuzzy-System wird für eine bestimmte Bewegung angepaßt, indem ein oder mehrere, für die Bewegung spezifische Kriterien, mit Genetischen Algorithmen optimiert werden.

III.3.1 Funktionsweise Genetischer Algorithmen

Wie die biologische Evolution benutzt auch die simulierte, künstliche Evolution das Konzept von Phänotyp und Genotyp. Der Genotyp ist die genetische Information eines Individuums, die in der Natur normalerweise in kodierter Form als DNS, d. h. als Chromosom vorliegt. Es gibt viele verschiedene Möglichkeiten der Repräsentation von Genotypen. Gebräuchlich sind Strings von binären Ziffern, von beliebigen Parametern oder von symbolischen Ausdrücken. Jeder String bzw. jedes Gen steht für eine potentielle Lösung des gestellten Problems.

Der Phänotyp entspricht beim Genetischen Algorithmus dem Wert der Funktion f , berechnet mit dem Gen des Individuums. Die Fitness ist ein Maß für die Fähigkeiten eines Individuums und wird normalerweise durch eine Bewertungsfunktion (objective function) als Ergebnis einer Simulation bestimmt. Die Fitness spielt bei der Selektion in einer Population eine Rolle. Selektion ist ein Prozeß, bei dem über

das Überleben und die Reproduktion der Individuen einer Generation entschieden wird. Reproduktion bedeutet, daß aus vorhandenen Genotypen neue Individuen entstehen. Da im Gegensatz zur Natur nur ein Chromosom pro Individuum vorhanden ist, werden Kinder durch *cross-over* erzeugt. Selektionsdruck heißt, daß sich die fähigsten, d. h. im Sinne der Problemstellung optimalen, Individuen bei der Fortpflanzung durchsetzen. Dazu kommt noch die Mutation, die mit einer gewissen Wahrscheinlichkeit zufällig Veränderungen in den Genanlagen einer Population bewirkt. Der wiederholte Kreislauf von Ausscheiden und Vermehrung von Individuen durch Selektion sowie Mutation über mehrere Generationen führt zu einer dramatischen Verbesserung der Fitness einer Population.

Unterschiedliche Probleme bedürfen unterschiedlicher Repräsentation. Die genetischen Operatoren wiederum müssen auf die jeweils gewählte Repräsentation angepaßt werden. Deshalb unterscheiden sich verschiedene Implementationen von Genetischen Algorithmen darin, mit welchem Algorithmus und mit welcher Wahrscheinlichkeit Paarung und Mutation auftreten, ob die Populationsgröße konstant bleibt und ob Duplikate möglich sind oder nicht usw.

Pseudocode

```

create initial random population;
evaluate fitness of each individual in population;
do
    select parents based on fitness;
    create children from parents by applying cross-over;
    perform mutation;
    evaluate fitness of the children;
    replace members of the population by children
        considering their fitness;
until some criteria met;

```

Aus dem Pseudocode ist zu ersehen, daß als erstes die Initialpopulation erzeugt werden muß. Dazu wird für alle n Individuen einer Population zufällig der Genotyp erzeugt. Abschnitt IV.1.2 zeigt im konkreten Beispiel, wie die Strings, die die Gene darstellen aussehen können.

Dann wird mit jedem Individuum eine Simulation ausgeführt und der Wert der Bewertungsfunktion für diese Simulation berechnet. In der nächsten Generation wird eine bestimmte Anzahl der schlechtesten Individuen durch Kinder ersetzt. Die Anzahl der Individuen der Population bleibt konstant. Dazu werden aus einem Elternpaar durch *cross-over* Kinder erzeugt. Die Fitness eines Individuums spielt dabei eine Rolle, mit welcher Wahrscheinlichkeit es als Elternteil fungiert. Wichtig ist,

daß sich auch schlechte Individuen mit einer nicht verschwindenden Wahrscheinlichkeit fortpflanzen. Außerdem findet mit einer gewissen Wahrscheinlichkeit p_m Mutation im Genmaterial der Kinder statt. Dabei muß ausgeschlossen werden, daß identische Gene in einer Population entstehen. Als nächstes werden die Werte der Bewertungsfunktion der Kinder ermittelt und die neue Population mit Kinder wieder nach der Fitness geordnet. Dann werden wieder die schlechten Individuen aussortiert und durch Kinder ersetzt usw. Ein Abbruchkriterium beendet diesen Prozeß. Das ist normalerweise der Fall, wenn entweder eine bestimmte Anzahl Generationen erreicht ist, oder das Verfahren konvergiert.

Das Populationskonzept bewirkt also, daß gleichzeitig viele Wege zum Optimum ausprobiert werden und dabei durch Vererbung Informationsaustausch stattfindet. Das im Laufe des simulierten Evolutionsprozesses erlangte Wissen über das Problem steckt verteilt in der ganzen Population und verhindert eine frühzeitige Stagnation während der Optimierung.

Genetische Operatoren

Genetische Algorithmen suchen nicht zufällig im Parameterraum, aber sie verwenden für Paarung und Mutation stochastische Elemente, um zusammen mit schon gewonnenen Informationen bessere Ergebnisse zu finden.

Zur Paarung werden die Eltern nach der sogenannten Roulette-Methode ausgewählt. Bei einer Population von n Individuen stelle man sich ein Roulette-Rad mit $\sum_{i=0}^n i$ Fächern vor. Die Verteilung ist so, daß dem besten Individuum n Fächer gehören, dem schlechtesten ein Fach gehört usw. Dann wird zufällig ein Fach ausgelost und das zugehörige Individuum nimmt als Elternteil am *Cross-over* teil. Sind zwei Elternteile gefunden, wird zufällig eine Bruchstelle im String der Eltern festgelegt (siehe Pfeil in Abb. III.11) und durch Neukombination zwei Kinder erzeugt. Entstehen dabei schon bestehende Genotypen, also Duplikate, wird das Verfahren wiederholt.

Es gibt auch noch andere Verfahren, um aus Eltern neue Gene zusammenzustellen. Beispielsweise kann für jede einzelne Position im String eines Kindes zufällig bestimmt werden, von welchem Elternteil die Erbinformation kommt.

Für meine Optimierungsprobleme hat sich die Vermehrung durch Kreuzung aber als überlegen erwiesen. Das ist wahrscheinlich darauf zurückzuführen, daß es beim *Cross-over* zu Wechselwirkung der Parameter aufgrund ihrer Position im String kommt. Denn die Wahrscheinlichkeit, daß auf dem String weit entfernte Positionen getrennt werden, ist entsprechend höher, als bei benachbarten.

Mutation bedeutet, das eine beliebige Position im String, zufällig, mit einer gewissen Wahrscheinlichkeit p_m auf einen anderen, beliebigen, aber zulässigen Wert

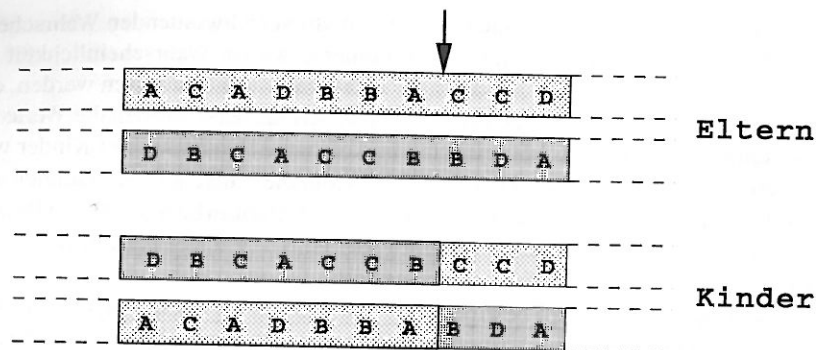


Abb. III.11: Cross-over

gesetzt wird (z. B. $p_m = 0,01$ bei einer Population von $n = 100$ Individuen). Mutation darf nicht zu häufig vorkommen, um nicht schon gebildete, optimale Stringteile wieder zu zerschlagen, aber auch nicht zu selten, damit das Genmaterial einer Population nicht verarmt und nur noch einen Teil vom Suchraum abdeckt.

Die Werte für Populationsgröße, Kinderzahl, Mutationswahrscheinlichkeit usw. müssen problemangepaßt eingestellt werden. Um die Wirkung der Operatoren zu verstehen und eventuell zu verbessern, erstellt man einen Fossilien-Report, d. h. man führt Buch über das Entstehen und Leben der Individuen und ihrer Fitness.

III.3.2 Empirische Analyse

Die Bewertungsfunktion

Die Bewertungsfunktion, die ein bestimmtes Systemverhalten charakterisieren soll, muß eindeutig sein. Genetische Algorithmen sind sehr erfindungsreich und belohnen bei einem schlecht formulierten Bewertungskriterium auch ein Verhalten, das eigentlich nicht erwünscht ist. (So war ich über die Vielfalt der möglichen Bewegungen meines dreigelenkigen Modells erstaunt, die durch minimale Schwerpunktsbewegung gekennzeichnet sind.)

Eine beispielhafte Entwicklung von Maximum (Optimum) und Mittelwert der Bewertung aller Individuen einer Population im Lauf der Generationen ist in Abb. III.12 zu sehen. Typisch ist, daß das Optimum (gestrichelte Kurve) oft über mehrere Generationen unverändert bleibt, bis ein Gen entsteht, welches ein besseres Ergebnis erzielt wird. Aber die über alle Individuen einer Population gemittelte Fitness nimmt tatsächlich im Laufe weniger Generationen dramatisch zu. Der

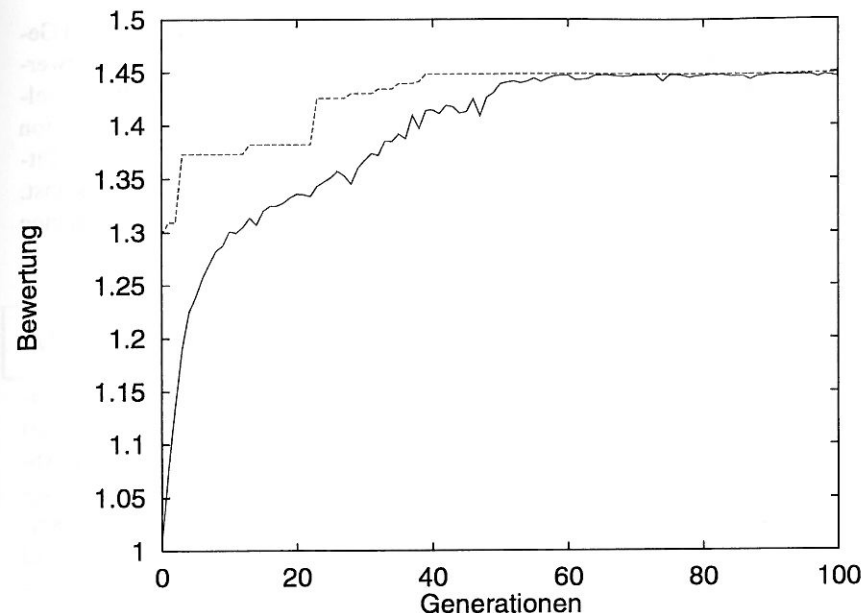


Abb. III.12: Maximum (gestrichelt) und Mittelwert (durchgezogen) einer Bewertung gegen Generationenzahl.

ganze Prozeß scheint zu konvergieren. Bei dieser Problemstellung mit einer Population aus 100 Individuen und 20 Kinder pro Generation, wird schon nach 40 Generationen, d. h. 900 Simulationen das beste Ergebnis gefunden. Ob es sich tatsächlich um das globale Optimum handelt, ist genausowenig wie bei anderen Optimierungsverfahren garantiert. Es empfiehlt sich deshalb, die gesamte Optimierung mehrmals mit verschiedenen, zufälligen Anfangspopulationen zu wiederholen. Die Wahrscheinlichkeit in einem lokalen Optimum steckenzubleiben ist geringer, wenn man mehrmals z. B. 40 Generationen startet, als mehrere hundert Generationen am Stück zu rechnen. Zum Schluß kann man noch eine neue Anfangspopulation mit den schon gefundenen, optimalen Genen mischen und schauen, ob sich noch weitere, bessere Lösungen durchsetzen.

Eine weitere, typische Eigenschaft der Entwicklung einer Population ist, daß sich die Differenz zwischen dem schlechtesten und dem besten Einzelergebnis innerhalb einer Population im Laufe der Generationen stark verringert. Würde man jetzt den oben erwähnten Reproduktionsvorgang mit dem absoluten Bewertungs-

wert steuern, hätten zum Beispiel beim Roulette-Verfahren schon nach wenigen Generationen fast alle Individuen die gleiche Chance Eltern zu werden, da ihre Bewertungswerte sehr nah beieinander liegen. Ein einfaches, aber probates Mittel, welches dieses Problem umgeht und über viele Generationen effektive Reproduktion garantiert, ist die lineare Skalierung. Das bedeutet, daß nur ein relative Maß als Fitness für den Selektionsdruck verwendet wird, aber nicht der Bewertungswert selbst. Dieses relative Maß ist der Rang, auf dem, in einer nach der Bewertung geordneten Reihenfolge, ein Individuum innerhalb einer Population steht.

Mehrzielige Optimierung

Viele physikalische Aufgabenstellungen zeichnen sich oft nicht nur durch ein Kriterium, sondern durch mehrere konkurrierende Kriterien aus, nach denen optimiert werden muß. Ein Beispiel für solch eine mehrzielige Optimierung findet sich in Abschnitt IV.1.4. In diesem Fall habe ich für die Individuen einer Population 2 Rangfolgen erstellt; einmal nach dem 1. Kriterium sortiert und einmal nach dem 2. Kriterium. Wie oben erwähnt, dient der Rang in der sortierten Folge als relatives Maß für die Bewertung. Es gibt nun beliebig viele Möglichkeiten aus mehreren Bewertungswerten eine Fitness zu berechnen. Sinnvoll und einfach sind zum Beispiel das Produkt, die Summe oder das Minimum (wenn 100 der Rang des besten Ergebnisses ist) der Ränge als Fitness zu nehmen. Alle folgenden Operationen wie Cross-over und Mutation funktionieren weiterhin, denn für sie spielt es keine Rolle, daß die Fitness aus zwei Bewertungskriterien gewonnen wurde.

Ein anderes häufig auftretendes Problem sind Nebenbedingungen, die eingehalten werden sollen. Eine Möglichkeit ist, Lösungen zu bestrafen (penalty functions), die Nebenbedingungen verletzen, z. B. Bewegungen, die bestimmte Gelenkwinkel überschreiten.

Grundsätzlich stellt sich die Frage nach einer Gewichtungsfunktion, wenn eine oder mehrere Optimierungs- oder Bestrafungskriterien eingesetzt werden.

III.3.3 Parallelisieren auf einem Workstation-Cluster

Eine Optimierung kann, je nach Problem, sehr viel Rechenzeit verbrauchen. Es kommt darauf an, wie lange die einzelne Simulation dauert und wie groß der Raum der möglichen Lösungen ist, in dem man suchen muß. Entscheidend ist, wieviele Simulationen ausgeführt werden müssen, um ein befriedigendes Ergebnis zu erhalten. Ein Vorteil von diesem Optimierungsverfahren ist, daß keine Gradienten berechnet werden müssen.

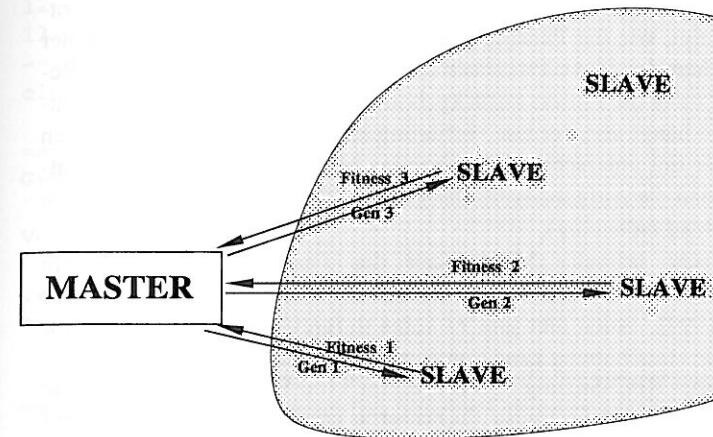


Abb. III.13: Das Master-Slave Prinzip

Bei dem vorgestellten biomechanischen Modell liegt die Rechenzeit pro Simulation zur Bestimmung der Bewertung für die Optimierung einer Bewegung typischerweise bei einigen Sekunden bis einer Minute, je nach Problemstellung. In der Regel wird es sich um mehrere hundert bis einige tausend Simulationen handeln und insgesamt kommt man also auf einige Stunden Rechenzeit für einen Optimierungsprozeß.

Der größte Vorteil gegenüber anderen Optimierungsverfahren liegt darin, daß sich Genetische Algorithmen aufgrund des Populationskonzepts sehr gut zum Parallelisieren eignen, da die Bewertungsfunktion der einzelnen Individuen unabhängig voneinander bestimmt werden kann. Bei biomechanischen Optimierungsaufgaben ist die Berechnung dieser Funktion in der Regel so aufwendig, daß der Rechenaufwand zur Verwaltung der Parallelisierung dagegen vernachlässigbar ist.

Paralleles Arbeiten ist auf vernetzten Workstations möglich, die zu einem virtuellen Multiprozessorsystem zusammengefaßt werden können. Als Toolkit zur Parallelisierung in solch einem Workstation-Cluster wurde PVM 3.2 (Parallel Virtual Machine) verwendet, da es einfach zu installieren und zu handhaben ist. Das System wurde am Oak Ridge National Laboratory entwickelt und ist Public Domain Software. PVM ist aus Sicht des Benutzers eine Bibliothek zur Kommunikation in einem parallelen Rechner mittels Message-Passing. Mit dem Master-Slave-Prinzip

wird erreicht, daß die Berechnungen mit steigender Prozessorzahl schneller beendet sind, ohne daß mehr Programmieraufwand für den Genetischen Algorithmus entsteht. Abb. III.13 zeigt, wie die Parallelisierung erreicht wird. Konkret stellt der Master-Prozeß die Gene den Slave-Prozessen zur Verfügung, die ihrerseits nach Beendigung ihrer Rechnung die Fitness zurückgeben. Die Anzahl der parallel laufenden Slave-Prozessen hängt von der zur Verfügung stehenden Anzahl Prozessoren ab. In einem Cluster mit unterschiedlich schnellen Maschinen und vielen Benutzern ist es wichtig, daß die Aufgabenverteilung durch PVM nicht statisch, sondern dynamisch erfolgt (load balancing).

Eine andere, sehr schnelle Möglichkeit des Parallelisierens in einem Workstation-Cluster stellt das Verteilte Rechnen mit DFN-RPC (Remote Procedure Call) dar [Herold, 1994]. Auch hier gibt es keine tiefgreifenden Änderungen für den Benutzer und die Handhabung kann komfortabel gestaltet werden.

Das Protokoll einer RPC-Sitzung von insgesamt 1100 Simulationen (Calls) in einem Cluster von 13 Silicon Graphics Prozessoren unter IRIX 4 zeigt, wie auf die einzelnen Server verteilt wurde. Aufgrund der nicht baugleichen Prozessoren und der unterschiedlichen Belastung durch andere Prozesse ergibt sich cirka der Faktor 3 in der Rechenzeit für eine Simulation auf den unterschiedlichen Servern (CPU/Call/sec). Die dynamische Verteilung bewirkt, daß schnelle Server tatsächlich erheblich mehr Simulationen (Calls) bearbeiten haben als langsame. Auf das Programm zur Verwaltung der Parallelisierung entfallen nur 0.6 % der verbrauchten CPU-Zeit. Die gesamte Rechnung hatte 6 mal mehr CPU-Zeit zur Verfügung, als tatsächlich Zeit für den Benutzer vergangen ist. Das scheint auf den ersten Blick bei 13 Servern kein guter Schnitt zu sein, aber dabei darf man nicht vergessen, daß auf einer einzelnen Workstation niemals 100 % der CPU-Zeit für die Rechnung zur Verfügung stehen.

```
*****
DFN-RPC Protocol of clnt_gen_al:      50 generations
```

Server	Calls	CPU-time/sec	CPU/Call/sec
1 nova	93	925.10	9.95
2 nova	85	877.43	10.32
3 nova	95	918.59	9.67
4 nova	84	874.73	10.41
5 rigel	124	561.28	4.53
6 ceres	113	545.58	4.83
7 cygnus	111	541.92	4.88
8 koma	101	541.57	5.36
9 virgo	60	766.31	12.77
10 pluto	57	706.13	12.39

11 adonis	58	737.62	12.72
12 einstein	60	710.61	11.84
13 aorta	59	739.73	12.54

clnt_gen_al			58.54
=====			
Gesamt:	1100		9505.14
Verstrichene Weltzeit:			1580.00
Verhaeltnis CPU/Weltzeit:			6.02

Die Möglichkeit zu Parallelisieren bringt nicht nur den Vorteil der Schnelligkeit, sondern kann auch genutzt werden, um den Optimierungsalgorithmus selbst effizienter zu machen. Denn es ist für den Austausch von Information durch Vererbung vorteilhafter, ganze Subpopulationen parallel zu rechnen, die immer wieder Individuen austauschen, als nur Individuen parallel zu rechnen.

IV

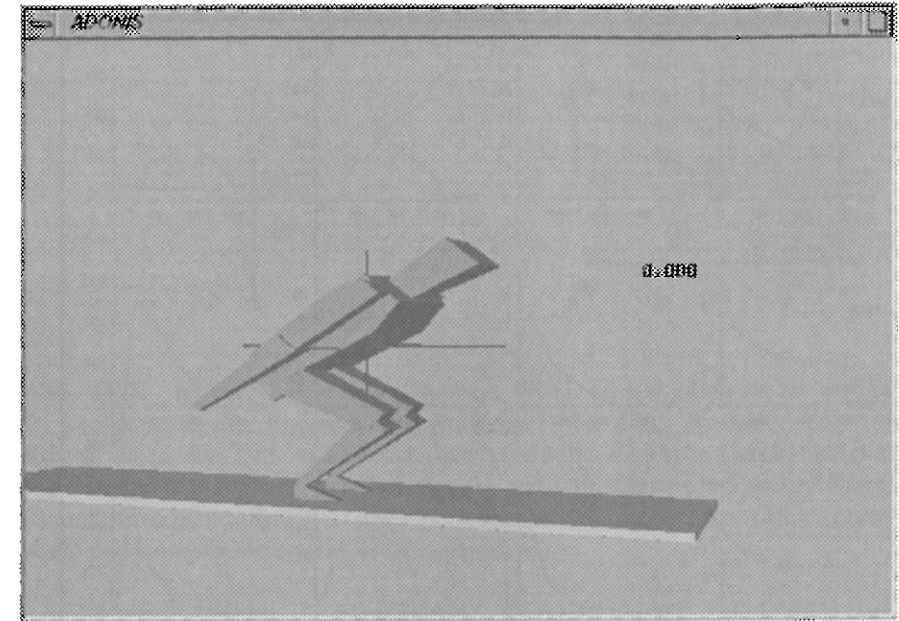
Simulation von Standhochsprüngen

IV.1 Simulation des maximalen Standhochsprungs

In der biomechanischen Forschung ist der maximale Standhochsprung sowohl für experimentelle Untersuchungen, etwa mit optischen Methoden und Kraftmeßplatten, als auch für Simulationen ein beliebtes Thema. Ein Grund dafür ist, daß diese Bewegung durch ein eindeutiges, gut meßbares Kriterium, die Sprunghöhe, charakterisiert ist. Durch Gleichung (IV.1) ist die Höhe h_{CM} , die der Gesamtkörperschwerpunkt nach dem Absprung im Aphel der Flugphase erreichen wird, festgelegt. In diese Gleichung gehen die vertikale Komponente des Schwerpunkts z_{CM} und seine vertikale Geschwindigkeit \dot{z}_{CM} zum Absprungszeitpunkt t_{off} ein.

$$h_{CM} = z_{CM}(t_{off}) + \frac{1}{2g} \dot{z}_{CM}^2(t_{off}) \quad (IV.1)$$

Für eine große Sprunghöhe sollte sowohl $z_{CM}(t_{off})$ als auch $\dot{z}_{CM}(t_{off})$ maximal sein. Das Ziel beim maximalen Standhochsprung ist deshalb, daß der Körperschwerpunkt beim Abheben möglichst hoch liegen sollte, d. h. die Haltung beim Absprung möglichst gestreckt sein sollte. Während dies unmittelbar einleuchtet, ist nicht so klar, wie gleichzeitig die in der Beschleunigungsphase gewonnene Rotationsgeschwindigkeit optimal in vertikale Geschwindigkeit umgesetzt wird und wie die Gelenke gleichzeitig vor Überdehnung geschützt werden.



IV.1.1 Experimentelle Ergebnisse

Bobbert et al. haben Messungen an Sportlern zum maximalen Standhochsprung an der Freien Universität Amsterdam vorgenommen. Die Kurven in Abbildung IV.1 und IV.2 wurden aus dem Artikel *Coordination in vertical jumping* entnommen [Bobbert, 1988].

Die Zeitverläufe verschiedener Größen beruhen auf Auswertungen von mehreren Sprüngen mit angelegten Armen von 10 männlichen Probanden, die im Mittel 1,93 m groß und 84 kg schwer waren. Die Kurven zeigen also die Mittelwerte aus mehreren Versuchen von mehreren Probanden. Dabei wurden Bodenreaktionskraftmessungen und Aufnahmen mit einer 16-mm Kamera mit 100 Hz Aufnahme Frequenz gemacht. Daraus wurden Segmentwinkel extrahiert und durch Differenzieren Segment- und Gelenkwinkelgeschwindigkeiten und -beschleunigungen berechnet. Weiter wurden Teil- und Gesamtkörperschwerpunktskoordinaten berechnet sowie Moment und Leistung von Hüft-, Knie- und Sprunggelenk berechnet.

Wenn man aus dem Stand maximal hoch springen will, geht man zuerst in die Hocke und holt Schwung (countermovement jump). Sowohl Beschleunigungsweg als auch Beschleunigungsdauer werden verlängert und dadurch wird auch eine hö-

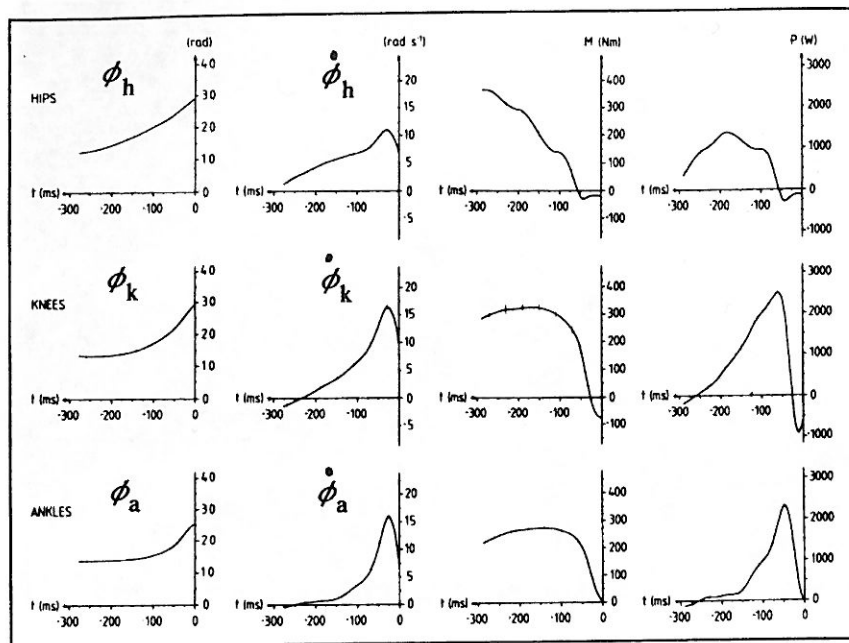


Abb. IV.1: Gelenkwinkel, Gelenkwinkelgeschwindigkeit, Gesamtmoment und Leistung (von links nach rechts) über die Zeit an Hüft-, Knie- und Sprunggelenk (von oben nach unten) nach gemittelten Messungen von [Bobbert, 1988].

here vertikale Geschwindigkeit beim Absprung erreicht. Die Abbildungen IV.1 und IV.2 zeigen Kurven aus der Beschleunigungsphase, nach dem Schwungholen. Das ist die Zeit zwischen dem Beginn der Aufwärtsbewegung des Rumpfes t_{on} , nach dem Schwungholen aus der Hocke, und dem Absprungszeitpunkt t_{off} , wenn sowohl die Fersen als auch die Zehen den Bodenkontakt verloren haben. Diese Beschleunigungsphase dauerte bei allen Sprüngen zwischen 300 und 350 Millisekunden. Es handelt sich somit um eine kurze, explosive Bewegung.

Um sich die Sprunghöhe klarzumachen, die man beim Hochspringen ohne Anlauf erzielen kann, kann man selbst die Übung *jump and reach* ausführen. Dazu stellt man an einer Wand beim aufrechten Stand die maximale Höhe der Fingerspitzen der ausgestreckten Hand fest. Dann erfolgt der Sprung und die Finger zeigen an der Wand die erzielte Höhendifferenz an. Ein großgewachsener Sportler überwindet bei dieser Übung ca. 60–70 cm. Die von Bobbert et al. gemessenen Sprünge

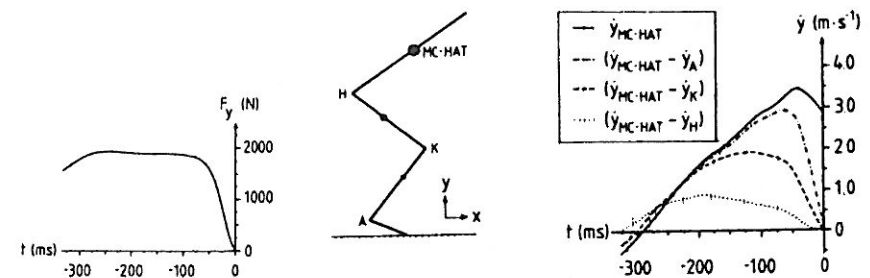


Abb. IV.2: Meßergebnisse aus [Bobbert, 1988] Links: Gemittelte vertikale Komponente der Bodenreaktionskraft über die Zeit; Rechts: Die vier Kurven über die Zeit entsprechen der gemittelte vertikale Geschwindigkeit vom Rumpfschwerpunkt (durchgezogen), vertikale Geschwindigkeitsdifferenz Rumpfschwerpunkt–Knöchel (strich-punktiert), Rumpfschwerpunkt–Knie (gestrichelt) und Rumpfschwerpunkt–Hüfte (gepunktet).

wurden allerdings mit angelegten Armen durchgeführt und die durchschnittlich gewonnene Höhe betrug deshalb im Mittel nur 39 cm, was bei den Probanden einer durchschnittlichen, absoluten Schwerpunkthöhe von 1,48 m entspricht.

IV.1.2 Simulation

Gesucht ist eine Steuerung, die die Gesamtdrehmomente im Hüft-, Knie- und Sprunggelenk berechnet, die auf der rechten Seite der Bewegungsgleichungen (Gl. II.1) eingehen.

In der Realität erzeugt der menschliche Bewegungsapparat diese Drehmomente durch neuro-muskuläre Komponenten. Die Drehmomente sind in komplizierter Weise abhängig vom Zustand des Bewegungsapparats. Selbst wenn man von mikroskopischen Größen absieht und nur wie in meiner Modellierung auf der Ebene von ebenen Gelenken arbeitet, bleiben viele Fragen offen. Das Moment im Gelenk j ist sicher nichtlinear abhängig von den Gelenkwinkeln und Gelenkwinkelgeschwindigkeiten $M^j = M^j(\varphi^j, \dot{\varphi}^j, \varphi^{j-1}, \dot{\varphi}^{j-1}, \varphi^{j+1}, \dot{\varphi}^{j+1}, \dots)$. Insbesondere sind die maximal möglichen Gelenkmomente und die maximal mögliche zeitliche Änderung der Momente von Gelenkwinkelstellung und -geschwindigkeit abhängig. Aus Experimenten sind einzelne Werte von maximalen isometrischen Momenten bei bestimmten Winkelstellungen bekannt, aber es gibt keine konsistenten Datensätze, aus

denen man allgemein die Abhängigkeiten und damit Funktionen für die M^j rekonstruieren könnte [Delp, 1990].

Für die Berechnung der Drehmomente muß ich deshalb im folgenden stark vereinfachende Annahmen machen, die zum Teil nur für die spezielle Bewegungsaufgabe Standhochsprung sinnvoll sind.

Den Ablauf eines Standhochsprungs behandle ich in zwei Phasen unterteilt, die Beschleunigungsphase und die Flugphase. Der entscheidende Bewegungsabschnitt, der auch mit den Messungen verglichen werden soll, ist die Beschleunigung aus der Hocke bis zum Abheben des Modells. Die nachfolgende Flugphase dient nur zur Beurteilung des Absprungs.

Beschleunigungsphase

Während der Beschleunigungsphase wird das innere Gesamtmoment M_{ges}^j in einem Gelenk j nach folgenden Überlegungen berechnet.

Zu jedem Zeitpunkt können die statischen Gelenkmomente M_{stat}^j berechnet werden, die zum Halten in der gerade eingenommenen Stellung gebraucht werden. Diese statischen Momente sind abhängig von den Winkelstellungen der über dem Gelenk stehenden Segmente. Die Segmentwinkel werden gegen die Horizontale gemessen (siehe Abb. II.1) und mit den Indizes R für Rumpf, O für Oberschenkel und U für Unterschenkel bezeichnet. Die Haltemomente können folglich für jeden Simulationsschritt aus folgenden Gleichungen berechnet werden:

$$M_{\text{stat}}^{\text{Hüft}}(\varphi_R) = \frac{1}{2} l_R \cos \varphi_R m_R g \quad (\text{IV.2})$$

$$M_{\text{stat}}^{\text{Knie}}(\varphi_R, \varphi_O) = \left(\frac{1}{2} l_O \cos \varphi_O m_O + \left(l_O \cos \varphi_O + \frac{1}{2} l_R \cos \varphi_R \right) m_R \right) g \quad (\text{IV.3})$$

$$M_{\text{stat}}^{\text{Sprung}}(\varphi_R, \varphi_O, \varphi_U) = \left(\frac{1}{2} l_U m_U \cos \varphi_U + \left(l_U \cos \varphi_U + \frac{1}{2} l_O \cos \varphi_O \right) m_O + \left(l_U \cos \varphi_U + l_O \cos \varphi_O + \frac{1}{2} l_R \cos \varphi_R \right) m_R \right) g \quad (\text{IV.4})$$

Im folgenden bezeichne ich die Momente, die die dynamische, zweckgebundene

Bewegung bewirken, als aktive Momente. Für die Beschleunigungsphase mache ich den speziellen Ansatz, daß diese Momente für jedes Gelenk das Produkt aus Haltemoment und einer Größe $f^j(\dot{\varphi}^j)$ sind.

$$M_{\text{akt}}^j(\varphi^j, \dot{\varphi}^j) = f^j(\dot{\varphi}^j) M_{\text{stat}}^j \quad (\text{IV.5})$$

Der von der Gelenkwinkelgeschwindigkeit abhängige dimensionslose Multiplikator $f^j(\dot{\varphi}^j)$ wird für jedes Gelenk j durch Fuzzy Control berechnet.

Um das Gesamtmoment im Gelenk j zu erhalten, muß zu diesem sogenannten aktiven Drehmoment noch das passive Moment $M_{\text{pas}}^j(\varphi^j)$ (siehe Gleichung (II.6)) hinzugerechnet werden, das allerdings nur in extremen Gelenkwinkelbereichen von Null verschieden ist.

$$M_{\text{ges}}^j = M_{\text{akt}}^j(\varphi^j, \dot{\varphi}^j) + M_{\text{pas}}^j(\varphi^j) \quad (\text{IV.6})$$

Fuzzy-Regelung

Der Multiplikator $f^j(\dot{\varphi}^j)$ in Gleichung (IV.5) wird aus einem Fuzzy-System mit einer Prämisse und einer Konsequenz berechnet:

IF <Gelenkwinkelgeschwindigkeit> THEN <Multiplikator>.

Da das statische Moment stark winkelabhängig ist, soll der Multiplikator nur die Winkelgeschwindigkeitsabhängigkeit berücksichtigen. Für jedes Gelenk wird die Winkelgeschwindigkeit durch n unscharfe Mengen repräsentiert, der Multiplikator durch l unscharfe Mengen. Es gibt also $3n$ Regeln für die drei Gelenke, die analog zu dem in Abschnitt III.2.4 erklärten Verfahren ausgewertet werden.

Beim Standhochsprung bringt der Bewegungsapparat Kräfte und Momente entgegen der Schwerkraft auf, was in der Simulation durch Drehmomente in den Gelenken nach Gleichung (IV.6) realisiert wird. Die aktiven Momente (Gl. IV.5) sind sicher zu jedem Zeitpunkt und in jeder Zwischenstellung größer, als die entsprechenden statischen Momente (Gl. IV.2), damit es zu der explosiven Bewegung kommt. Andererseits können die Momente nicht beliebig groß werden, denn sie sind durch die Anatomie und Physiologie des Bewegungsapparats beschränkt.

In den Entwurf des Fuzzy-Reglers gehen deshalb folgende a-priori-Annahmen ein:

1. EXPLOSIVE AUFWÄRTSBEWEGUNG:

Die Bewegung soll im wesentlichen nach oben gerichtet sein, d. h. die interessanten Gelenkwinkelgeschwindigkeiten sind positiv (Extension). Außerdem ist der Standhochsprung eine beschleunigte Bewegung, d. h. die aktiven Momente werden größer oder gleich den Haltemomenten sein. Daraus folgt, daß

$$M_{\text{ges}}^j = M_{\text{flug}}^j(\varphi^j, \dot{\varphi}^j) + M_{\text{pas}}^j(\varphi^j) \quad (\text{IV.8})$$

$$M_{\text{flug}}^j(\varphi^j, \dot{\varphi}^j) = \mp(c^j h^j k^j \dot{\varphi}^j - d^j h^j(\pi - \varphi^j)) \quad (\text{IV.9})$$

$$h^j = \begin{cases} 0 & : \varphi^j \geq \pi \\ 1 & : \text{sonst} \end{cases} \quad (\text{IV.10})$$

$$k^j = \begin{cases} 0 & : \dot{\varphi}^j \geq 0 \\ 1 & : \text{sonst} \end{cases} \quad (\text{IV.11})$$

mit gelenkspezifischen Vorzeichen sowie Konstanten c^j und d^j .

Aus der Definition von h^j ersieht man, daß für Gelenkwinkel größer 180° nur die Anschlagsmomente wirken. Durch deren Einfluß wird sich der Winkel wieder verkleinern und $M_{\text{flug}}^j(\varphi^j, \dot{\varphi}^j)$ wirkt proportional zum Differenzwinkel dagegen. Ist außerdem die Winkelgeschwindigkeit negativ, wirkt aufgrund der Definition von k^j ein Term proportional zur Winkelgeschwindigkeit der Flexion entgegen.

Landung

Die Landung nach dem Sprung wird an dieser Stelle nicht simuliert. Durch die kurzzeitig auftretenden starken Verzögerungen beim Aufkommen auf den Boden bei der Landung werden an das Modell andere Anforderungen gestellt, als beim Aufwärtsbeschleunigen und Abheben zum Sprung. Mit dem Schwabbelmassen-Modell [Gruber, 1987] und einer analytischen Funktion, die den Momentenaufbau beim Abbremsen steuert [Ruder, 1994], wurden Landungen nach Niedersprüngen erfolgreich simuliert.

IV.1.3 Diskussion der Simulationsergebnisse

Um meine Simulationsergebnisse mit den Kurven von Bobbert (1988) vergleichen zu können, hat das Modell die anthropometrischen Daten einer männlichen, 84 kg schweren und 1,93 m großen Person und es entspricht damit dem Mittel der Probanden, mit denen Bobbert seine Messungen durchgeführt hat.

Die Simulation beginnt mit Anfangswerten für die Segmentwinkel und -winkelgeschwindigkeiten, die aus den Meßkurven abgeschätzt wurden.

Empirisch wurde die Anzahl der Zugehörigkeitsfunktionen für die Multiplikatoren auf 5 festgelegt und für die Gelenkwinkelgeschwindigkeiten auf 6 Zugehörigkeitsfunktionen. Aufgrund der a-priori-Annahmen wurden ihre Lagen definiert.

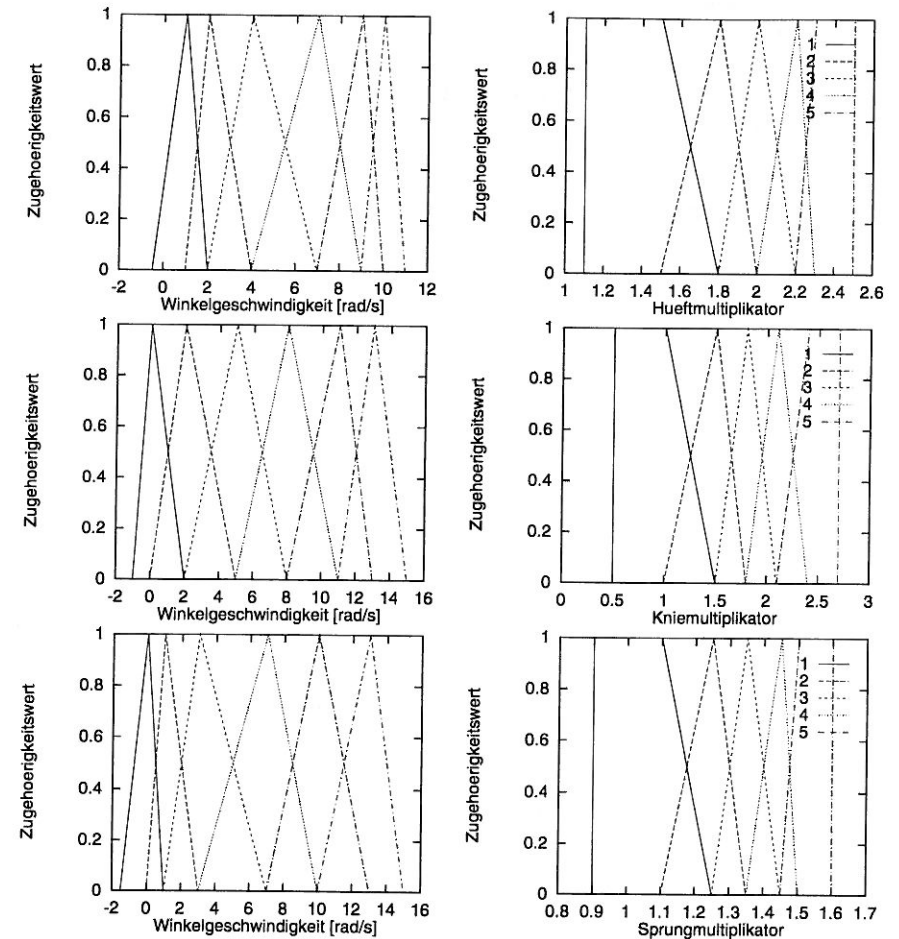


Abb. IV.3: Links: Je 6 Fuzzy sets für die Prämissen φ^j und Rechts: je 5 Fuzzy sets für die Konsequenz f^j . Oben: Hüftgelenk, Mitte: Kniegelenk, Unten: Sprunggelenk.

Abbildung IV.3 zeigt konkret die Zugehörigkeitsfunktionen, die zu der Simulation gehören, mit der die im folgenden Abschnitt behandelten Ergebnisse gewonnen wurden. Die unscharfen Mengen für die Multiplikatoren werden in Abbildung IV.3 einfach von links nach rechts durchnummeriert. Die Winkelgeschwindigkeiten werden mit Z (zero), T (tiny), S (small), N (normal), L (large) und H (huge) bezeichnet.

Das Optimierungsproblem besteht somit aus einem String der Länge 18. Durch Optimierung auf maximale Höhe wurde ein String gefunden, der der besten (unüberwacht gelernten) Regelbasis entspricht:

	Z	T	S	N	L	H
Hueftgelenk:	2	4	5	5	1	5
Kniegelenk :	5	5	5	2	4	5
Sprunggelenk:	4	5	5	5	5	3

Oder als Regeln ausgedrückt:

if $\dot{\varphi}^{hueft}$ is Z then f^{hueft} is 2
if $\dot{\varphi}^{hueft}$ is T then f^{hueft} is 4
if $\dot{\varphi}^{hueft}$ is S then f^{hueft} is 5
if $\dot{\varphi}^{hueft}$ is N then f^{hueft} is 5
if $\dot{\varphi}^{hueft}$ is L then f^{hueft} is 1
if $\dot{\varphi}^{hueft}$ is H then f^{hueft} is 5
if $\dot{\varphi}^{knie}$ is Z then f^{knie} is 5
if $\dot{\varphi}^{knie}$ is T then f^{knie} is 5
if $\dot{\varphi}^{knie}$ is S then f^{knie} is 5
if $\dot{\varphi}^{knie}$ is N then f^{knie} is 2
if $\dot{\varphi}^{knie}$ is L then f^{knie} is 4
if $\dot{\varphi}^{knie}$ is H then f^{knie} is 5
if $\dot{\varphi}^{sprung}$ is Z then f^{sprung} is 4
if $\dot{\varphi}^{sprung}$ is T then f^{sprung} is 5
if $\dot{\varphi}^{sprung}$ is S then f^{sprung} is 5
if $\dot{\varphi}^{sprung}$ is N then f^{sprung} is 5
if $\dot{\varphi}^{sprung}$ is L then f^{sprung} is 5
if $\dot{\varphi}^{sprung}$ is H then f^{sprung} is 3

Mit dieser Regelbasis wurde folgendes Ergebnis gewonnen: Die simulierte Bewegung hat eine Beschleunigungsphase von 300 ms Dauer, die Höhe vom Gesamtkörperschwerpunkt erreicht 1,51 m, was einer Sprunghöhe von 40 cm entspricht. Die Segmente bewegen sich koordiniert und die Bewegungssequenz in Abb. IV.4 und IV.5 zeigt, daß zuerst der Oberkörper etwas aufgerichtet wird. Erst dann kann sich auch das Knie strecken und erst kurz vor dem Abheben entwickelt das Sprung-

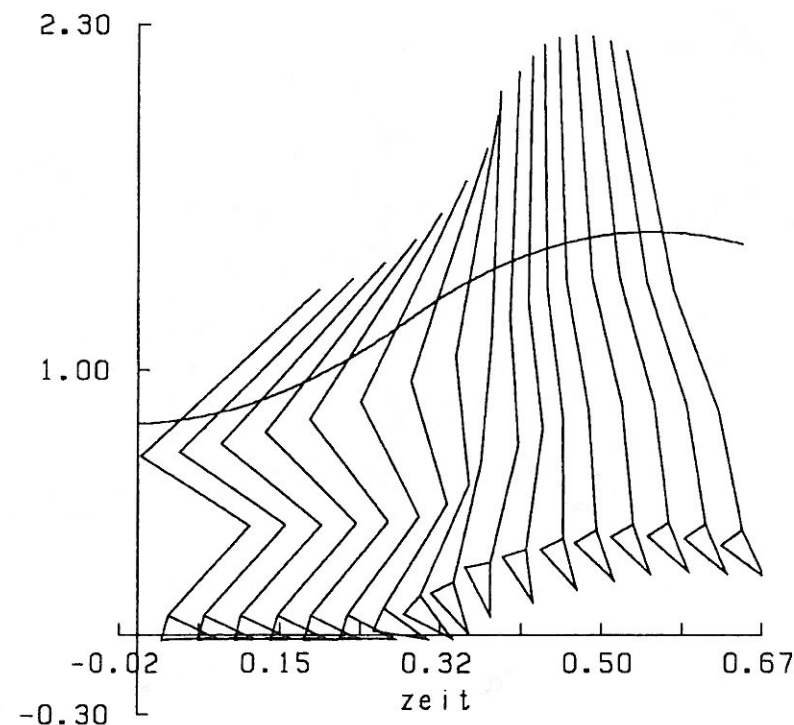


Abb. IV.4: Pseudo-Stroboskopische Darstellung eines simulierten Sprungs mit Schwerpunktskurve.

gelenk sein volles Drehmoment. Diese Eigenschaften der simulierten Bewegung sind in guter Übereinstimmung mit den von Bobbert gemessenen Sprüngen.

Die Zeitverläufe der Geschwindigkeiten, Gesamtmomente und der Bodenreaktionskraft sind nicht identisch mit den Meßkurven, aber die Werte dieser Größen sind alle in realistischen Bereichen, was ein Blick auf die Größen im Einzelnen zeigt.

Abbildung IV.6 zeigt die vertikale Bodenreaktionskraft eines simulierten Sprungs als Summe aus der Kraft auf Ferse und auf Ballen. Zu Beginn der Simulation sieht man eine Kraftspitze, die von dem abrupten Beginn der Bewegung her-

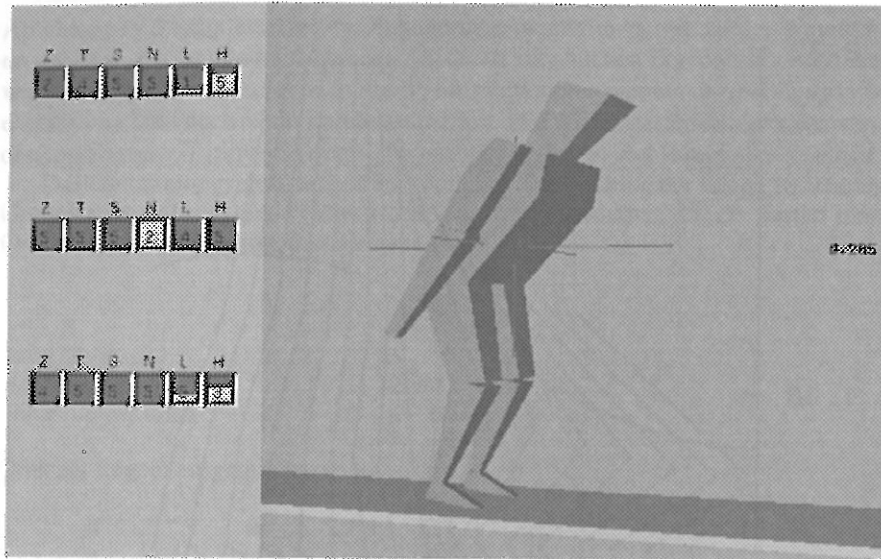


Abb. IV.5: Animation mit Darstellung der Regelaktivitäten.

rührt. Dann stellt sich für 200 ms eine annähernd konstante Kraft von circa 1500 N ein, was dem 1,8-fachen der Körpergewichtskraft entspricht. Da der Modellfuß ein starrer Körper ist und abgesehen von Fersen- und Ballenpolster keine elastischen Eigenschaften hat, sieht man nach 200 Millisekunden das Abheben der Ferse deutlich als Einbruch in der Bodenreaktionskurve. Im Vergleich zur gemessenen Kurve (Abb. IV.2) zeigt die Simulation eine um circa 300 N geringere Kraft.

Die vertikale Geschwindigkeit des Rumpfschwerpunkts und die Differenzen zu den Geschwindigkeiten der Gelenkpunkte in Abbildung IV.6 sind interessante Kurven, weil 70 % der Körpermasse im Rumpf vereinigt sind und die Umwandlung von Rotationsgeschwindigkeit in vertikale Geschwindigkeit entscheidend für die Sprunghöhe ist. Der maximale Wert der simulierten Rumpfschwerpunktsgeschwindigkeit ist kleiner als der im Mittel gemessene Wert. Außerdem erreichen die Differenzen ihr Maximum fast gleichzeitig und nicht nacheinander, wie in Abb. IV.2.

Die Kurve der Gelenkwinkel in Abbildung IV.7 links oben zeigt, daß alle Gelenkwinkel auf 180 Grad zustreben. Der zeitliche Verlauf zeigt auch, daß zuerst das Hüftgelenk, dann das Kniegelenk und erst 100 Millisekunden vor dem Abheben das Sprunggelenk in Streckung geht. Dies stimmt mit den Meßergebnissen überein.

Der zeitliche Verlauf der Gelenkwinkelgeschwindigkeiten (links unten) zeigt

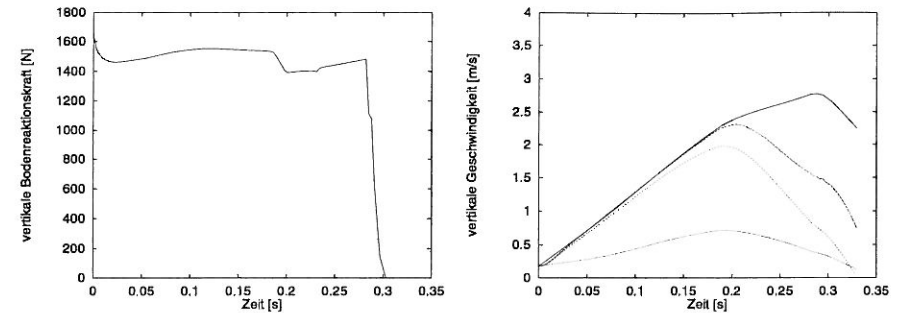


Abb. IV.6: Simulationsergebnisse: **Links:** Vertikale Komponente der Bodenreaktionskraft. **Rechts:** Die vier Kurven sind analog zu Abbildung IV.2 von oben nach unten: Vertikale Geschwindigkeit vom Rumpfschwerpunkt, vertikale Geschwindigkeitsdifferenz Rumpfschwerpunkt-Knöchel, Rumpfschwerpunkt-Knie und Rumpfschwerpunkt-Hüfte.

noch deutlicher die proximal zu distal Reihenfolge der Segmentbewegung. Alle drei Gelenke erreichen dann fast gleichzeitig ihre maximale Geschwindigkeit, die danach schnell abfällt. Der Vergleich mit den Meßwerten zeigt gute Übereinstimmung, nur im Kniegelenk sind die auftretenden Geschwindigkeiten um einige rad/s in der simulierten Bewegung geringer. Der Anstieg der Winkelgeschwindigkeiten ist allerdings nicht streng monoton.

Abbildung IV.7 rechts oben zeigt die Kurven der Gesamtdrehmomente der Gelenke. In der analog zu Bobbert gewählten Darstellung entsprechen positive Werte der Drehmomente Extension in der Hüfte, Extension im Knie sowie Plantarflexion am Fuß. (Tatsächlich hat das Moment im Knie ein negatives Vorzeichen. Die Segmentmassenschwerpunkte liegen auf der x -Achse hinter dem Kniegelenk, aber vor dem Hüft- und Sprunggelenk.) Während das Gesamtdrehmoment in der Hüfte zu Beginn der Beschleunigungsphase groß ist und dann abnimmt, steigt das Moment im Sprunggelenk noch leicht an und fällt dann steil ab. Die Werte des Knie Moments sind im Vergleich zur Messung zu klein, während die Kurvenform übereinstimmt. Ein Blick auf die Regelbasis (s.v.) zeigt, daß als Konsequenz häufig die 5 vorkommt, die den größten zugelassenen Multiplikatorwerten entspricht. (Da auf maximale Höhe optimiert wurde, war das auch zu erwarten.) Die zeitlichen Verläufe der Momente innerhalb physiologisch vernünftigen Wertebereichen zeigen, daß der Fuzzy-Ansatz für die Multiplikatoren funktioniert.

Abbildung IV.7 rechts unten zeigt die Leistung, die für jedes Gelenk aus dem

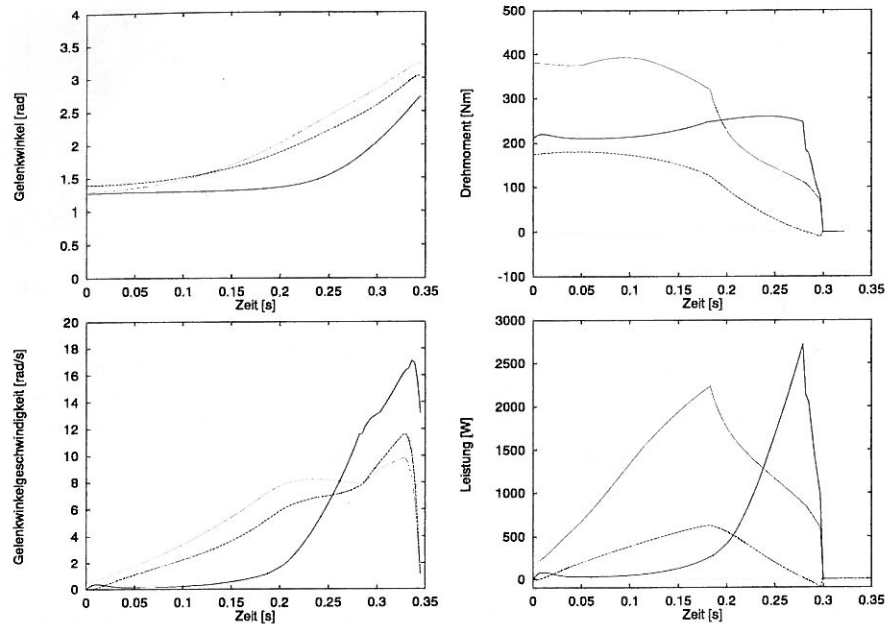


Abb. IV.7: Simulationsergebnisse: links oben: Gelenkwinkel, links unten: Gelenkwinkelgeschwindigkeiten, rechts oben: Gelenkmomente rechts unten: Leistung; Hüftgelenk gepunktet, Kniegelenk gestrichelt, Sprunggelenk durchgezogen.

Produkt von Gesamtmoment und Winkelgeschwindigkeit berechnet wurde.

$$P^j(t) = M_{\text{ges}}^j(t) \dot{\varphi}^j(t) \quad (\text{IV.12})$$

Da im Kniegelenk sowohl kleinere Geschwindigkeiten als auch ein geringeres Moment in der Simulation auftraten, ist die Leistung erheblich geringer als die gemessene. In Übereinstimmung mit der Messung liegt das Maximum in der Leistungskurve des Hüftgelenks bei der Halbzeit der Beschleunigung, während für das Sprunggelenk die Leistung erst 150 ms vor dem Abheben stark ansteigt und mit dem Abheben auf Null abfällt.

In Abb. IV.8 sind die zeitlichen Verläufe der Multiplikatoren f^j zu sehen, die durch die Fuzzy-Regelung berechnet wurden. Im Kniegelenk wird mit Beginn der Beschleunigung auch schon das maximale Moment erreicht. Es entspricht dem 2,4-fachen des statischen Haltemoments und dauert 200 ms an. Im Hüftgelenk wird erst nach 100 ms das Maximum erreicht. Der Multiplikator beträgt dann 2,3. Der Fak-

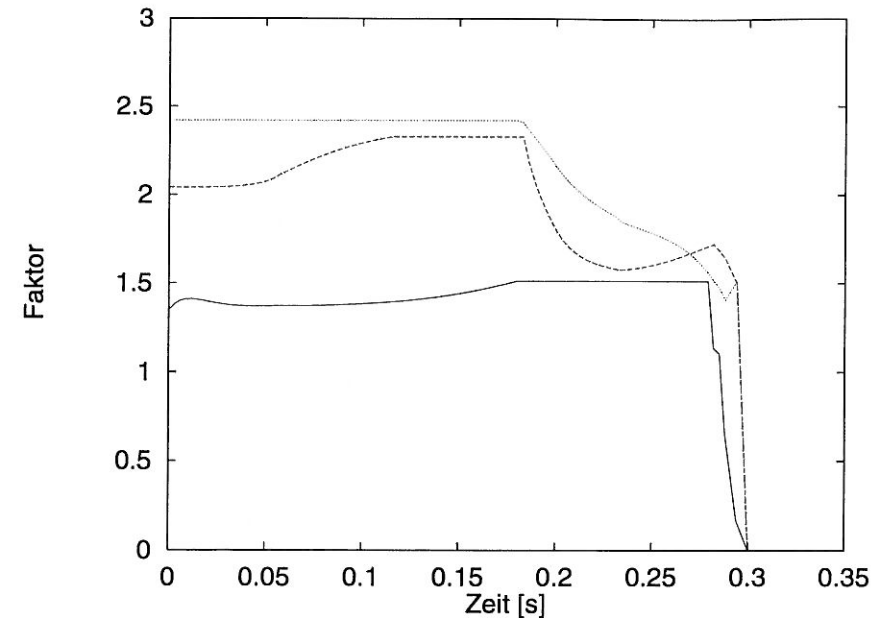


Abb. IV.8: Zeitverläufe der Faktoren $f^j(\dot{\varphi}^j)$ aus Gleichung IV.5 für das Hüftgelenk gestrichelt, Kniegelenk gepunktet und Sprunggelenk durchgezogen.

tor für das Sprunggelenk ist während der ganzen Beschleunigungsphase annähernd konstant, und erreicht erst in den letzten 100 Millisekunden vor den Abheben einen Maximalwert von 1,5.

Wichtig ist die Größenordnung der Faktoren. Interessant ist, daß einige Personen, die ich um ihre Meinung gebeten habe, die Werte der Faktoren auf 100 oder mehr geschätzt haben. Dabei liegen sie eben nur zwischen 1 und maximal 2,5. Absolut ist das Gesamtmoment für das Hüftgelenk am größten, der Fuzzy-Faktor ist aber für das Knie am höchsten. Die Haltemomente geben nur den Anteil wieder, der gegen die Schwerkraft aufgebracht werden muß.

In Abschnitt III.2.5 wurde erklärt, daß Fuzzy-Inferenz vollkommen deterministisch ist und man sich das Kennfeld der Regelung aufzeichnen kann. Genau das ist in Abbildung IV.9 für das Hüftgelenk getan. Die Eingangsgrößen sind Rumpfwinkel und Hüftwinkelgeschwindigkeit. Darüber ist das Hüftmoment als Produkt aus Fuzzy-Faktor und Haltemoment aufgetragen. Die Fläche läßt sich leicht interpretieren: Entlang der Winkelgeschwindigkeit Null kann man die sinusförmige Winkel-

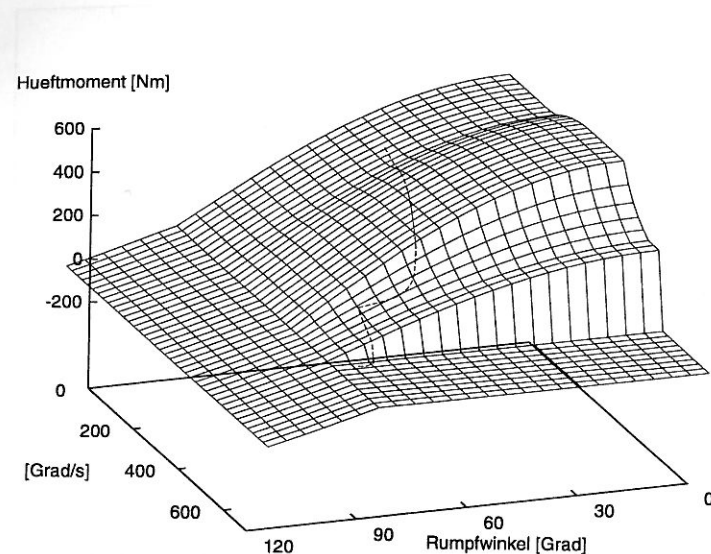


Abb. IV.9: I/O-Fläche der Hüfte berechnet mit dem optimierten Regelsatz sowie die Trajektorie einer Simulation mit diesem Regelsatz dargestellt im Raum von Rumpfwinkel φ_R – Hüftwinkelgeschwindigkeit $\dot{\varphi}^{\text{Hüft}}$ –Hüftmoment $M^{\text{Hüft}}$.

abhängigkeit aufgrund des Haltemoments aus Gl. IV.2 sehen. In der anderen Richtung wird die Fläche durch die 6 Regeln bestimmt, was aufgrund der wenigen unscharfen Mengen, die als Konsequenz zur Verfügung standen, Kanten in der Fläche zur Folge hat. Diese dreidimensionale Darstellung des Kennfeldes ist allerdings nur für das oberste Segment möglich, für alle anderen Segmente ist das Moment von mehr als einem Winkel abhängig.

Als gestrichelte Trajektorie ist der Verlauf des Hüftmoments während der Simulation eingetragen. Sie entspricht damit der Kurve für das Hüftmoment aus Abb. IV.7 rechts oben. Die Trajektorie beginnt bei 42 Grad Rumpfwinkel und 30 Grad/s Hüftwinkelgeschwindigkeit. Der Regelsatz wurde durch Optimieren mit diesen Anfangsbedingungen gewonnen. Trotzdem sollte die Steuerung auch bei veränderten Anfangsbedingungen so gut funktionieren, daß ein Sprung simuliert wird, wenn auch nicht mehr mit optimaler Höhe (siehe auch Abschnitt IV.3).

Auffällig ist der Knick im Kennfeld entlang der 90 Grad Linie. Für Hüftwinkel

größer 90 Grad muß sich das Vorzeichen des Moments umkehren, um den Rumpf wieder aufzurichten. In diesem Fall besteht die Steuerung nur aus einem einfachen Moment proportional zur Winkeldifferenz zur Vertikalen, ohne Fuzzy-Regelung. Dieser Winkelbereich sollte bei den optimierten Sprungsimulationen auch nicht vorkommen. Erstens verhindern die Anschlagmomente extreme Winkelbereiche und zweitens soll ein optimierter Sprung nicht in Rücklage geraten. Tatsächlich sieht man, daß die Trajektorie vor der 90 Grad Linie abbricht und das Modell abhebt. Allerdings kommt es nach dem Absprung, in der Flugphase, in Rücklage, wie Abbildung IV.4 zeigt.

Wenn man sich den optimierten String auf Seite 64 ansieht, erkennt man, daß das Fuzzy-System vereinfacht werden könnte. In einigen benachbarten Positionen steht derselbe Inhalt, d. h. Konsequenz, und es werden auch nicht alle der 5 möglichen Zugehörigkeitsfunktion pro Gelenkmoment genutzt. In der Tat konnte ich mit einem System von nur 5 Zugehörigkeitsfunktionen für jede Prämisse und 4 für jede Konsequenz ähnliche Ergebnisse erzielen. Für dieses kleinere System habe ich außerdem noch die Lage der unscharfen Mengen optimiert. Dieses relativ aufwendige Vorgehen bringt ein paar Millimeter Höhe und verändert vor allem die Zeitverläufe. Da ich keine Vergleichsdaten habe, um zu entscheiden, ob die Simulation dadurch wirklich besser wird, das heißt realen Sprüngen näher kommt, wurde dies nicht weiter verfolgt.

Der gewählte Ansatz, der die Gelenkwinkelgeschwindigkeit direkt zur Ausgabe abbildet, beruht auf der Annahme, daß während der Beschleunigungsphase die Geschwindigkeit monoton ansteigt. Versuche mit Ansätzen, in denen Größen wie Schwerpunktgeschwindigkeit, Winkel oder Winkelgeschwindigkeit benachbarter Gelenke zusätzlich als 2. Prämisse in eine Fuzzy-Regelung eingehen, brachten keine Verbesserung.

Vergleich mit Simulationsergebnissen anderer Arbeitsgruppen

In der Einleitung wurde erwähnt, daß es auch von anderen Arbeitsgruppen [Soest, 1993 a], [Soest, 1993 b], [Pandy, 1990] Standhochsprungsimulationen gibt. Sie basieren auf einem ähnlichen mechanischen Modell, werden aber nicht auf der Ebene der Gesamtmomente gesteuert, sondern durch das Stimulationsmuster von Muskelgruppen. Den Muskeln liegt im wesentlichen das Hill'sche Muskelmodell zugrunde. Für willentliche Bewegungen ist dieser Ansatz mit dynamischen Muskelmodellen und der Aktivierung als Input naheliegend. Dieses Vorgehen hat aber das Problem, daß der muskuläre Input nicht gemessen werden kann, sondern auch abgeschätzt werden muß. Dazu werden die Zeitpunkte der Aktivierung von Null auf Eins (*bang bang control*) numerisch optimiert. Die Kriterien dazu sind aus ma-

thematischen Gründen und nicht aufgrund biologischer Überlegung gewählt. Als Ergebnis können Muskelkräfte resultieren, die nicht realisierbar sind oder bei dieser speziellen Bewegung nicht vorkommen. Durch Vergleich mit EMG-Daten wird versucht, die Gültigkeit des durch Optimierung gefundenen Aktivierungsmusters zu legitimieren.

Die Ergebnisse dieser Arbeitsgruppen sind aufgrund des unterschiedlichen mechanischen Modells und Steuerungsansatzes nicht direkt mit meinen vergleichbar, aber es zeigen sich doch erstaunlich viele übereinstimmende Erfahrungen beim Simulieren.

Die simulierten Sprünge sind qualitativ vergleichbar. Ich kann aufgrund fehlender Muskelmodelle keine Aussagen über einzelne Muskeln machen. Die drei Gelenke werden von m Muskeln überspannt, wobei $m > 3$ gilt und somit im Nachhinein die Zusammensetzung der Gesamtgelenkmomente aus den Muskelmomenten ohne Zusatzannahmen nicht eindeutig zu ermitteln ist.

Die anderen Arbeitsgruppen können allerdings mit ihrem Ansatz nicht das Maximum und Abfallen in den Winkelgeschwindigkeiten kurz vor dem Absprung reproduzieren. Sie erklären dieses charakteristische Verhalten aber auch übereinstimmend als funktionelle Eigenschaft zum Schutz der Gelenke. Soest hat versucht, eine Bestrafung aufgrund der summierten Rotationsenergie der Segmente einzuführen. Er bemerkt [Soest, 1993 a], daß es ihm nicht gelungen ist, durch Berücksichtigung dieser Nebenbedingung im Optimierungsprozeß den gewünschten Kurvenverlauf zu erreichen. Diese Tatsache stimmt mit meiner Erfahrung überein. Beim Versuch, den Abfall in den Winkelgeschwindigkeiten durch Optimierung zu erzwingen, traten in meinen Simulationen immer insgesamt geringere Winkelgeschwindigkeiten auf, d. h. die Strategie der gesamten Beschleunigungsphase wurde geändert und es zeigt sich eben nicht der typische zeitliche Verlauf der Winkelgeschwindigkeiten mit einem deutlichen Maximum.

Eine Lösung ist, wie in meinem Ansatz verwirklicht, den Abbruch der Beschleunigungsphase und damit den Abfall der Geschwindigkeiten beim Erreichen bestimmter Winkelgeschwindigkeiten herbeizuführen (siehe IV.1.2). Dies läßt sich auch physiologisch plausibel durch einen Schutzreflex erklären. Tatsächlich erreichen die durch Optimierung - ohne weitere Vorgabe - gefundenen Sprünge die Winkelgeschwindigkeitsmaxima synchron in allen drei Gelenken. Diese Beobachtung deutet darauf hin, daß sich allgemein physiologische Randbedingungen nicht einfach durch Optimierungskriterien ersetzen lassen.

Verfrühter Absprung

Ein Verhalten, das sowohl meine simulierten Sprünge, als auch die Simulationen der anderen Arbeiten zeigten, ist, daß die volle Streckung des Körpers erst nach dem Absprung erreicht wird und nicht wie im Experiment zeitgleich mit dem Absprung. In Abb. IV.6 und IV.7 erkennt man den verfrühten Absprung: Laut Bodenreaktionskraftkurve erfolgt der Absprung nach 0,3 Sekunden. Die Gelenkwinkel erreichen aber erst circa 0,05 Sekunden später 180 Grad. Auch die Maxima in den Gelenkwinkelgeschwindigkeiten werden erst nach dem Absprung erreicht.

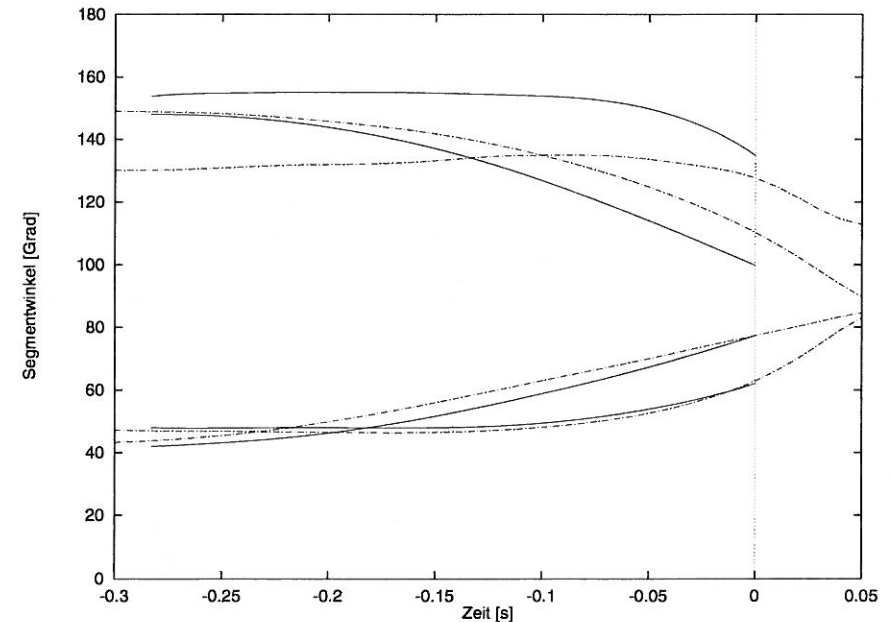


Abb. IV.10: Simulierte (durchgezogen) und gemessene (gestrichelt) Segmentwinkelverläufe. Entlang der 0-Sekunden-Achse sind von oben nach unten jeweils die zwei zusammengehörenden Kurven in der Reihenfolge Fuß, Oberschenkel, Rumpf und Unterschenkel abgebildet.

Eine weitere Abbildung soll darstellen, daß die Simulationen auch gegenüber realen Sprüngen ein um circa 0,05 Sekunden verfrühtes Absprungsverhalten zeigen. In Abbildung IV.10 sind sowohl gemessene als auch simulierte Zeitverläufe der 4 Segmentwinkel eingetragen. Die Meßdaten wurden mir freundlicherweise von

A. van Soest zur Verfügung gestellt. Sie stammen von Messungen von Standhochsprüngen aus der tiefen Hocke (ohne vorheriges Schwungholen) [Soest, 1992]. Die Zeitachse der Abbildung muß folgendermaßen interpretiert werden: Der Zeitpunkt Null markiert den Absprung des Modells in der simulierten Bewegung. Die gemessenen Kurven wurden verschoben eingezeichnet, der Absprung findet dadurch 0,05 Sekunden später statt. Tatsächlich stimmen die zeitlichen Verläufe von Rumpf- und Unterschenkelwinkel in dieser Darstellung sehr gut überein. Der gemessene Fußwinkel ist schon von Beginn an in der gemessenen Kurve kleiner, während das Modell platt auf dem Boden stehend beginnt. Alle anderen Anfangswerte stimmen bis auf wenige Grad überein.

Van Soest erklärt das verfrühte Absprungsverhalten durch die starren, nichtelastischen Segmente des Modells im Unterschied zu den realen, flexiblen Segmenten des menschlichen Körpers. Zajac (1993) weist darauf hin, daß zur Aufwärtsbeschleunigung nur die Energiespeicherung im Fußgewölbe eine Rolle spielt, aber nicht wesentlich die Energiespeicherungen in den anderen Segmenten. Aus dieser Einschränkung durch die Starre-Körper-Modellierung der Segmente folgt, daß unabhängig von der Wahl des Steuerungsansatzes für bessere Simulationsergebnisse vor allen Dingen ein verbessertes Fußmodell gefunden werden muß. Ein starrer Fuß mit Polster genügt nicht, sondern eine detailliertere Fußmodellierung muß die flexible Struktur des Fußes mit seiner Fähigkeit zur Energiespeicherung beim Abspringen berücksichtigen.

Spezielle Winkelabhängigkeit

Mein Ansatz für die Momentenberechnung während der Beschleunigungsphase beruht wesentlich auf einer ganz speziellen Winkelabhängigkeit, die durch die statischen Haltemomente vorgegeben ist. Dieses Vorgehen stellt eine grobe Vereinfachung dar und limitiert die Aussagekraft der Simulationsergebnisse. Deshalb müssen die Folgen dieser speziellen Annahme diskutiert werden.

Die Haltemomente sind nur von den Segmentwinkeln abhängig (Gl. IV.2). In Wirklichkeit sind aber für die Kraftentwicklung des Bewegungsapparats nicht die Winkel gegen die Horizontale, sondern die Winkel der Segmente zueinander (Gelenkwinkel) entscheidend. Während Segmentwinkel eindeutig sind, kann ein bestimmter Gelenkwinkelwert durch verschiedene Kombinationen von Segmentwinkeln realisiert werden. Abbildung IV.11 stellt das Haltemoment im Knie in Abhängigkeit von Hüftgelenks- und Kniegelenkwinkel dar. Die Höhenlinien für das Haltemoment im Knie wurden bei festgehaltenem Unterschenkelwinkel berechnet. Die Strichmännchen sollen eine Vorstellung von den zu den Eckpunkten der Abbildung

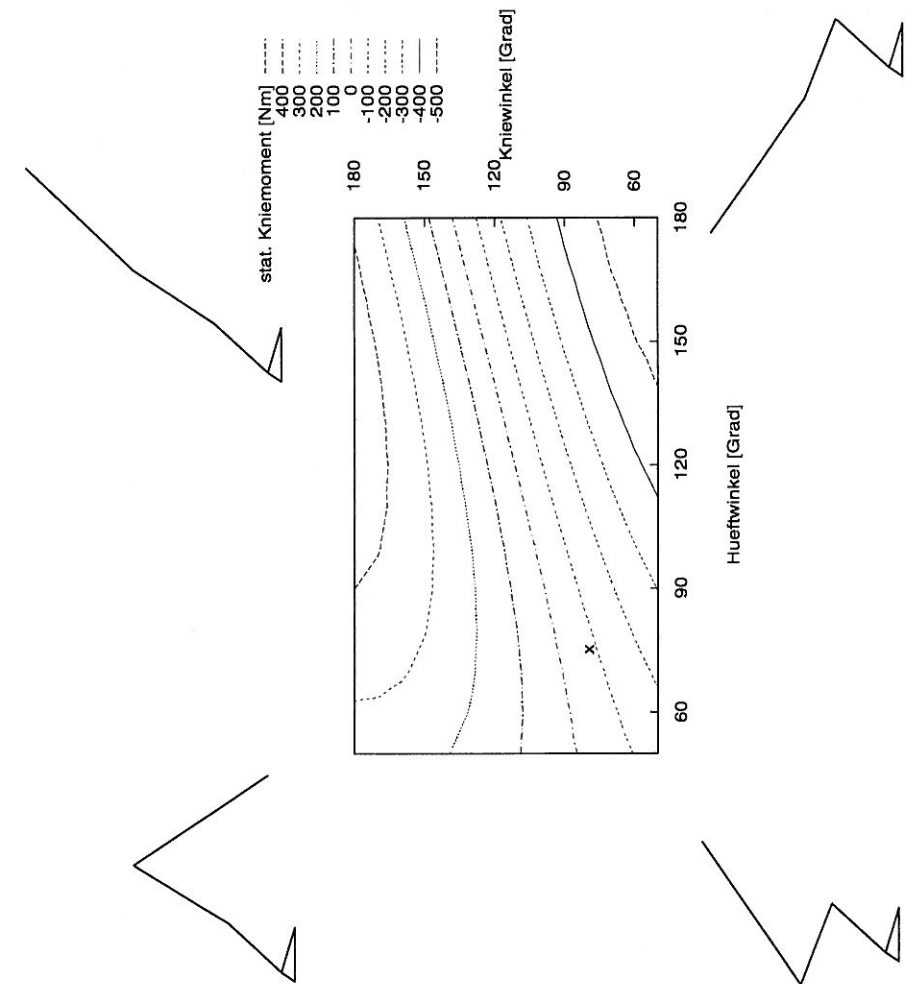


Abb. IV.11: Höhenlinien des statischen Kniemoments in Abhängigkeit von Kniegelenks- und Hüftgelenkwinkel. Der Unterschenkelwinkel ist dabei auf 48 Grad festgehalten. Das Kreuz bezeichnet den Punkt, der zum Simulationsbeginn gehört. Die Strichfiguren veranschaulichen die Körperstellungen, die zum dargestellten Winkelbereich gehören.

gehörenden Körperstellungen geben. Das Kreuz soll veranschaulichen, an welchem Punkt eine Sprungsimulation unter den bisher gewählten Anfangsbedingungen beginnt. Das Haltemoment im Knie ist unabhängig vom Unterschenkelwinkel. Da sich aber der Kniegelenkwinkel sehr wohl mit dem Unterschenkelwinkel ändert, verschieben sich die Höhenlinien nach oben oder unten, wenn man die gleiche Abbildung mit einem anderen Unterschenkelwinkel macht. Steht der Unterschenkel beispielsweise senkrecht, läuft die 0-Nm-Linie zur rechten oberen Ecke der Abbildung, weil dann das Modell genau aufrecht steht.

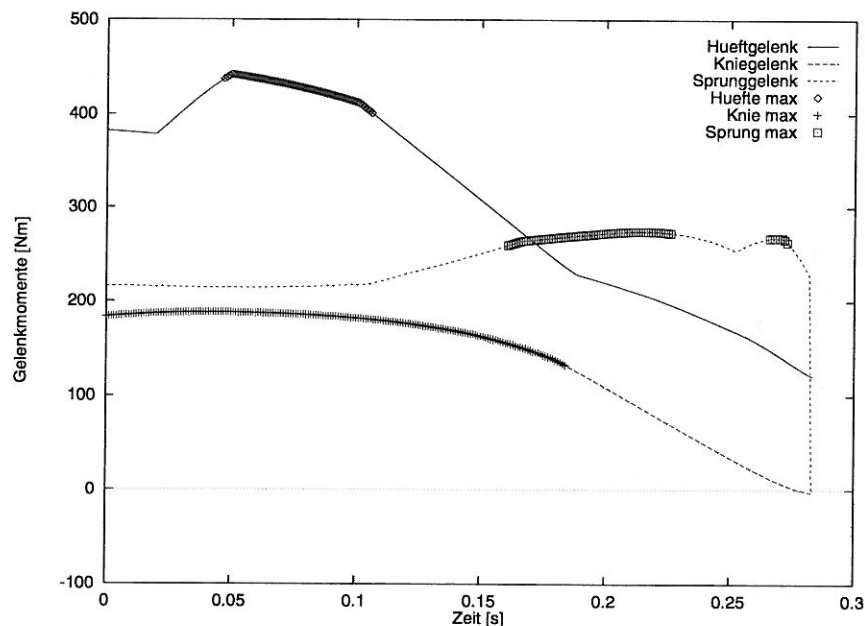


Abb. IV.12: Gelenkmomente: Die markierten Bereiche zeigen in welchen Kurvenabschnitten tatsächlich Maximalmomente wirksam sind.

Für die Suche nach der optimalen Bewegung ist es sehr wichtig, daß die möglichen Momentenwerte nach oben beschränkt sind und so das Modell die zur Verfügung stehenden Ressourcen optimal ausnutzen muß. In der Realität sind diese Maximalmomente wieder von Gelenkwinkel und -geschwindigkeit abhängig. Für meine Simulationen sind die maximal möglichen Momente durch den größten Wert des Faktors aus der Fuzzy-Regelung multipliziert mit dem Haltemoment gegeben. Zum Beispiel ergeben die Werte in Abb. IV.11 $\times 2,5$ die maximalen Kniemomen-

te. Allerdings sagt das nichts darüber aus, wann und wo welche Maximalmomente während der Simulation eines Sprungs tatsächlich angenommen werden. Deshalb sind in Abb. IV.12 an einem Simulationsbeispiel die Bereiche hervorgehoben, in denen Maximalmomente erreicht werden.

Vergleich mit Meßergebnissen

Bei der Bewertung der simulierten Ergebnisse im Vergleich mit den gemessenen Ergebnissen müssen einige Einschränkungen berücksichtigt werden. Ein grundsätzlicher Unterschied ist, daß Bobberts Kurven sehr glatt sind, weil sie über viele Versuche gemittelt sind und die Filmdaten gefiltert und geglättet wurden. Der Vergleich mit den Daten einer Simulation kann nur qualitativ erfolgen.

Trotzdem läßt sich ein Teil der Differenzen zwischen Messungen und Simulation erklären. Mein Modell stellt die Anthropometrie der Versuchspersonen nur angenähert dar, da die Segmentgrößen nur aus Körpermasse, -länge, Geschlecht und aus statistischen Daten berechnet werden (siehe S. 16). Besonders sensitiv sind die Ergebnisse gegenüber der Fußlänge, was durch die Hebelwirkung auf das gesamte darüberliegende System zu erklären ist. Das Sprunggelenk ist für die Beschleunigung in der Endphase, kurz vor dem Abheben, wenn bereits hohe Geschwindigkeiten erreicht sind, entscheidend.

Unklar ist, warum die Bodenreaktionskraft bei Bobbert um 300 N größer ist. Abzüglich der Gewichtskraft ist sie somit um ca. 30 % höher als die simulierte Bodenreaktionskraft. Aus vertikaler Kraft und Dauer des Bodenkontakts läßt sich die Absprunggeschwindigkeit berechnen, denn der Kraftstoß ist gleich der Impulsänderung:

$$\int_{t_{\text{on}}}^{t_{\text{off}}} (F_z - m_{\text{ges}}g) dt = p_z(t_{\text{off}}) - p_z(t_{\text{on}}) \\ = m_{\text{ges}} (\dot{z}_{\text{CM}}(t_{\text{off}}) - \dot{z}_{\text{CM}}(t_{\text{on}})) \quad (\text{IV.13})$$

Berechnet man aus dem Kraftstoß die Flughöhe, stimmt diese im simulierten Fall mit der tatsächlich erreichten überein und müßte bei Bobbert also grob abgeschätzt um 5 cm größer sein. Allerdings kann ich nicht den Anfangsimpuls der gemessenen Sprünge abschätzen. Die Probanden führen einen *countermovement jump* aus, das heißt, es ist gut möglich, daß sich der Schwerpunkt noch nach unten bewegt, wenn per Definition die Beschleunigungsphase beginnt. Weiter stellt sich die Frage, welcher Wert für die Schwerpunkthöhe beim letzten Bodenkontakt $z_{\text{CM}}(t_{\text{off}})$ von Bobbert gemessen wurde. Nigg (1977) berichtet von Differenzen von ca. 6 cm für diese Größe bei von verschiedenen Autoren gemessenen Standhochsprüngen.

Der simulierte Sprung hat eine Absprunggeschwindigkeit von $\dot{z}_{CM} = 2,53$ m/s. Soest (1992) gibt Werte von 2,59 m/s für seine Simulation und 2,67 m/s für seine Experimente an.

Es gibt mehrere (ähnliche) Regelbasen, die ungefähr gleich hohe Sprünge erzielen, was ja auch der Flexibilität des Menschen in der Bewegungsausführung entspricht. Die Rücklage in der Flugphase zeigen alle simulierten Sprünge, die große Sprunghöhen erreichen. Um einen Sprung mit Landung zu simulieren, reicht es offensichtlich nicht aus, die Beschleunigungsphase nur auf vertikale Sprunghöhe zu optimieren.

IV.1.4 Mehrzielige Optimierung

Beim Standhochsprung wird Rotationsenergie in Lageenergie umgewandelt. Mit der Sprunghöhe als Optimierungskriterium wird nur die vertikale Position des Körperschwerpunkts bewertet. Wie schon erwähnt, zeigten die nur auf Höhe optimierten Sprünge nach dem Abheben eine ungünstige Rücklage. Realistische Absprünge, nach denen auch eine Landung möglich ist, können durch zusätzliche Kriterien erreicht werden.

In Abschnitt III.3.2 wurde beschrieben, wie Genetische Algorithmen gleichzeitig mehrere Zielfunktionen optimieren können.

Drehimpuls

Der Drehimpuls, der zum Zeitpunkt des Abhebens relativ zum Schwerpunkt vorhanden ist, bestimmt die Rotation des Modells nach dem Abheben. Der Bahndrehimpuls des Gesamtkörperschwerpunkts bezüglich des Koordinatenursprungs ist:

$$L_{CM} = m_{ges}(\mathbf{r}_{CM} \times \dot{\mathbf{r}}_{CM}). \quad (IV.14)$$

Im zweidimensionalen Fall gibt es nur eine Drehimpulskomponente:

$$L_{CM} = m_{ges}(z_{CM}\dot{x}_{CM} - x_{CM}\dot{z}_{CM}). \quad (IV.15)$$

Der Gesamtdrehimpuls der 4 Segmente lautet:

$$L_{ges} = \sum_{i=1}^4 m_i(z_i\dot{x}_i - x_i\dot{z}_i) - \sum_{i=1}^4 \theta_i\dot{\varphi}_i \quad (IV.16)$$

mit den Segmentwinkeln φ_i , die von der Horizontalen aus gemessen werden.

Der Eigendrehimpuls oder Relativdrehimpuls des Modells bezüglich des Gesamtkörperschwerpunkts berechnet sich zu:

$$L_{Rel} = L_{ges} - L_{CM}. \quad (IV.17)$$

Tatsächlich verschwindet die Rücklage, wenn bei der Optimierung nicht nur maximale Höhe, sondern auch minimaler Drehimpulsbetrag relativ zum Schwerpunkt zum Zeitpunkt des Abhebens gefordert wird.

$$|L_{Rel}(t_{off})| = \min \quad (IV.18)$$

Obwohl in die Zielfunktion und damit in die Optimierung nur die Werte vom Zeitpunkt t_{off} eingehen, beeinflusst diese Zielfunktion den ganzen Bewegungsablauf.

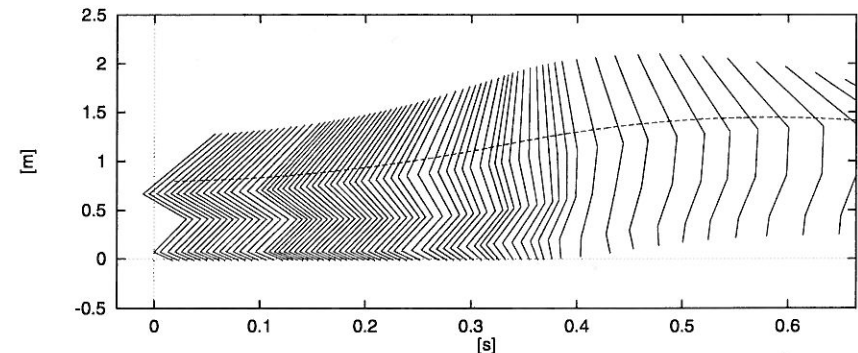


Abb. IV.13: Pseudo-Stroboskopische Darstellung mit Schwerpunktskurve für 1,44 m Sprunghöhe und maximalen negativen relativen Drehimpuls.

Versuchsweise habe ich auch genau das Gegenteil, nämlich maximalen negativen relativen Drehimpuls neben maximaler Höhe gefordert. Das Ergebnis der Optimierung ist ein Sprung, der mit 1,44 m relativ hoch ist und tatsächlich an einen Flop im Hochsprung erinnert (Abb. IV.13).

Statistik

Bei der Optimierung auf minimalen relativen Drehimpulsbetrag (Gl. IV.17) kommt es wesentlich auf die Gewichtung bzw. Verrechnung mit den anderen Kriterien an, wie klein der optimale Wert denn nun wirklich ist. Es ist deshalb interessant im

Zusammenhang mit mehreren Zielfunktionen den Optimierungsvorgang zu analysieren. Zum Beispiel wie sich die Zusammensetzung des Genmaterials einer Population im Laufe der Generationen ändert. Dazu muß man während der Optimierung über das Auftreten und die Zusammensetzung der Chromosomen Statistik führen.

Eine beispielhafte Entwicklung eines willkürlich ausgewählten Optimierungslaufs mit den Kriterien $h = \max$ und $|L_{\text{rel}}| = \min$ zeigt folgende typische Eigenschaften. Generation 0 ist die zufällig erzeugte Anfangspopulation. Eine Population hat 100 Individuen, deshalb ist die Summe in jeder Zeile 100. Die Tabelle zeigt, wie oft jede der fünf möglichen Konsequenzen bei Gelenklänge 18 an jeder der 6 Positionen pro Gelenk auftritt.

• Generation 0:

	1	2	3	4	5
Z	22	19	13	23	23
T	28	19	13	26	14
S	21	20	21	21	17
N	25	12	26	22	15
L	28	14	29	21	8
H	20	27	20	16	17
Z	26	12	23	20	19
T	18	16	19	24	23
S	28	22	13	21	16
N	21	17	23	20	19
L	27	16	16	19	22
H	19	26	16	19	20
Z	20	19	19	25	17
T	13	25	24	16	22
S	15	24	24	15	22
N	23	22	19	19	17
L	22	19	21	18	20
H	19	18	23	22	18

• Generation 50:

	1	2	3	4	5
Z	0	0	6	45	49
T	4	3	0	27	66
S	92	0	5	3	0
N	3	0	41	7	49
L	46	3	47	4	0
H	1	78	4	16	1
Z	0	0	40	54	6
T	0	0	8	84	8
S	27	0	3	7	63
N	0	7	60	21	12
L	5	4	5	0	86
H	0	5	10	85	0
Z	9	4	0	86	1
T	0	0	19	2	79
S	0	0	18	2	80
N	5	35	0	60	0
L	78	0	0	22	0
H	9	9	45	20	17

Man sieht, daß die Anfangspopulation der Generation 0 statistisch verteilt erzeugt worden ist, so daß jeder mögliche Wert an einer Position etwa gleich oft vorkommt. Dagegen sieht die Tabelle nach 50 Generationen ganz anders aus. In diesem Beispiel steht an insgesamt 26 Positionen die Null, was bedeutet, daß kein einziges Chromosom der Population diese unscharfe Menge als Konsequenz für diese Regel

mehr enthält. Das Genmaterial ist verarmt, nur durch Mutation könnte wieder die Konsequenz an dieser Stelle auftauchen. In manchen Spalten haben sehr viele Regeln dieselbe Konsequenz, es wird aber in der Reproduktionsroutine explizit ausgeschlossen, daß sich innerhalb einer Population zwei völlig identische Chromosome befinden.

Wie sehen jetzt die dazugehörigen Bewertungen aus? Aufgelistet sind Mittelwerte, Minimum und Maximum der zwei Bewertungsfunktionen innerhalb einer Population.

• Generation: 0

Höhe [m]: avg = 1.255, max = 1.458, min = 0.901

Rel. Drehimpuls [Nms]: avg = 26.337, max = 64.269, min = 1.992

• Generation: 50

Höhe [m]: avg = 1.433, max = 1.460, min = 1.362

Rel. Drehimpuls [Nms]: avg = 2.585, max = 29.888, min = 1.807E-02

Die besten Einzelergebnisse bei der Bewertung durch das Produkt $\text{Rang}(h = \max) \times \text{Rang}(|L_{\text{rel}}| = \min)$:

• Generation: 0

$h = 1.419 \text{ m}$ und $|L_{\text{rel}}| = 1.992 \text{ Nms}$

• Generation: 50

$h = 1.446 \text{ m}$ und $|L_{\text{rel}}| = 0.151 \text{ Nms}$

Während sich die Höhe des besten Einzelergebnisses nur um 3 cm verbessert hat, ist der relative Drehimpuls um den Faktor 10 kleiner geworden. Dieses Verhalten ist stark abhängig von der gewählten Verrechnungsart der zwei Bewertungsfunktionen.

Gelenkwinkelbeschleunigungen

Hohe Gelenkwinkelbeschleunigungen beim Abheben würden die Gelenke belasten. Deshalb fordere ich in einem Bewertungskriterium, daß die Summe der Gelenkwinkelbeschleunigungen zum Absprungszeitpunkt minimal sein soll.

$$\sum_{j=1}^3 |\ddot{\varphi}^j(t_{\text{off}})| = \min \quad (\text{IV.19})$$

Tatsächlich sehen die dadurch gewonnenen Sprünge in der Animation außerordentlich realistisch aus und die Rücklage verschwindet. Die zeitliche Entwicklung der Beschleunigungen macht allerdings einen schlechten Eindruck. In der menschlichen Bewegung kommen keine beliebig großen Gradienten in den Beschleunigungen vor und deshalb sollten sie auch in der Simulation begrenzt sein. Die Momente müssen folglich nicht nur auf einen Maximalwert begrenzt sein, sondern auch die zeitliche Änderung der Momente muß beschränkt sein.

IV.1.5 Maximalkraft und Schnellkraft

Die von einem Muskel maximal zu entwickelnde Kraft hängt von seinem physiologischen Querschnitt ab. Als Näherungswert gilt je Quadratzentimeter Muskelquerschnitt können 50–100 N Muskelkraft entwickelt werden. Ein Muskel von 100 cm² Querschnitt kann demnach eine Kraft von maximal 10 000 N aufbringen. Damit kann man grob die Momente abschätzen. Mit 5 cm Hebelarm und 10 000 N kommt man auf 500 Nm. Dieser Wert darf beim Hochsprung laut Messung maximal in der Hüfte auftreten. Im Knie bewegen sich die Hebelarme für die zum Springen wichtigsten Muskeln zwischen 8 mm in gebeugter Haltung und 40 mm bei gestrecktem Knie [van Leeuwen, 1992].

Für das Training eines Sportlers ist nicht nur die Maximalkraft wichtig, sondern auch die Schnellkraft. Die Schnellkraft ist keine Kraft, sondern die zeitliche Änderung der Kraft, das heißt Kraftanstieg bzw. -abfall. Diese Größe ist von der Vorgeschichte des Muskels und vielen anderen Faktoren abhängig.

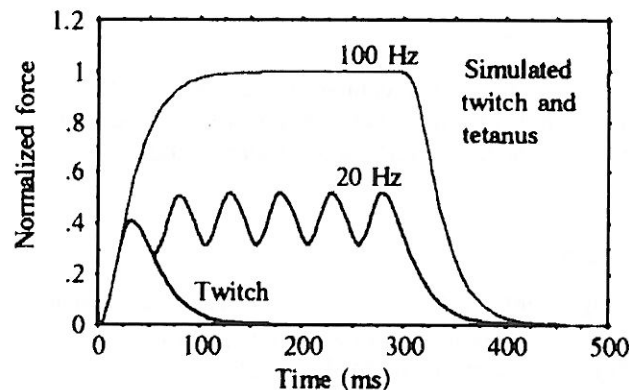


Abb. IV.14: Antwort einer schnellen Muskelfaser auf Stimulationen mit unterschiedlichen Frequenzen [van Leeuwen, 1992].

Für einzelne Muskelfasern gibt es Kurven, die den Zusammenhang zwischen Zeit nach tetanischer Anregung und Kraftanstieg zum Maximalwert erkennen lassen [van Leeuwen, 1992]. Nach Abbildung IV.14 kann eine Muskelfaser in circa 100 ms ihre volle Kraft erreichen. Dasselbe gilt nach Ende der Stimulation für den Rückgang der Kraft. Da aber während der Bewegung kurzzeitig nicht unbedingt von Null auf Maximalkraft beschleunigt wird, kann als Richtwert für einen ganzen Muskel ein Wert von 50 ms angenommen werden. Diese Überlegungen habe ich dazu verwendet, die maximal mögliche zeitliche Änderung des Moments in den Gelenken abzuschätzen. Damit ergibt sich ein Wert von 10 000 Nm/s für die kurzzeitige Änderung des obigen Moments von 500 Nm. Aus den gemessenen Kurven von Bobbert et al. kann man Werte von 7500 Nm/s abschätzen. Durch das Filtern und Glätten der Meßdaten sind aber wahrscheinlich größere Steigungen in den Momentenkurven herausgefallen.

Während der Simulation wird die Momentenänderung berechnet mit:

$$\left| \frac{dM_{\text{ges}}^j}{dt} \right| = \left| \frac{M_{\text{ges}}^j(t) - M_{\text{ges}}^j(t - \Delta t)}{\Delta t} \right| \quad (\text{IV.20})$$

mit einem typischen Wert von 1 ms für Δt . Die Simulation wird abgebrochen, wenn ein gelenkspezifischer Maximalwert überschritten wird.

$$\left| \frac{dM_{\text{ges}}^j}{dt} \right| < \Delta M_{\text{max}}^j \quad (\text{IV.21})$$

mit Hüftgelenk: $\Delta M_{\text{max}}^{\text{Hüft}} = 10\,000 \text{ Nm/s}$, Kniegelenk: $\Delta M_{\text{max}}^{\text{Knie}} = 7500 \text{ Nm/s}$ und Sprunggelenk: $\Delta M_{\text{max}}^{\text{Sprung}} = 15\,000 \text{ Nm/s}$.

Die Optimierung hat erst dann Regelbasen für Sprünge mit der gewünschten Winkelgeschwindigkeit und damit Sprunghöhe gefunden, nachdem der Maximalwert im Sprunggelenk auf 15 000 Nm/s erhöht wurde. Die Simulationsergebnisse des maximalen Standhochsprungs mit beschränkter Momentenänderung werden in Abschnitt IV.2.1 im Vergleich mit anderen Sprüngen mit geringerer Sprunghöhe diskutiert.

Die Schnellkraft hat einen entscheidenden Einfluß auf eine dynamische Bewegung und sollte eigentlich direkt in die Fuzzy-Regelung eingehen. Die Abhängigkeit von der Vorgeschichte und der Ermüdung beispielsweise wäre sehr gut geeignet, um durch ein Fuzzy-System modelliert zu werden, dazu fehlen aber die geeigneten Daten.

IV.2 Submaximale Standhochsprünge

Der maximale Standhochsprung hat als oberstes Ziel die Sprunghöhe zu optimieren.

Eine interessante Frage ist, was passiert, wenn nicht alle vorhandenen Ressourcen des Bewegungsapparats in einem Sprung zum Erreichen einer großen Sprunghöhe eingesetzt werden müssen, sondern mehr im Sinne einer Alltagsbewegung ein beliebiger Standhochsprung mit geringerer Höhe ausgeführt wird. Wodurch zeichnet sich ein Standhochsprung eigentlich aus?

Bei einem linearen Problem kann man mit der Simplexmethode durch Zielfunktion und Nebenbedingungen ein Gebiet definiert werden, in dem alle möglichen Lösungen liegen. Kuo (1993) nutzt diese Möglichkeit aus, indem er das Stehen als quasilineares Problem formuliert. Beim Springen handelt es sich um ein nichtlineares Problem und kann nicht durch lineares Programmieren gelöst werden. Deshalb kann ich nicht einfach den Bereich angeben, in dem sich alle möglichen Sprünge des Modells befinden.

Was ich tun kann, sind Beispiele von submaximalen Sprüngen zu rechnen. Im folgenden werden Sprünge mit einer gewünschten Sprunghöhe simuliert, indem als Optimierungskriterium die Sprunghöhe vorgegeben wird. Die Ergebnisse von verschiedenen Sprunghöhen werden mit den Simulationen von maximalen Standhochsprüngen verglichen.

IV.2.1 Vergleich von Sprunghöhe 151 cm, 140 cm und 133 cm

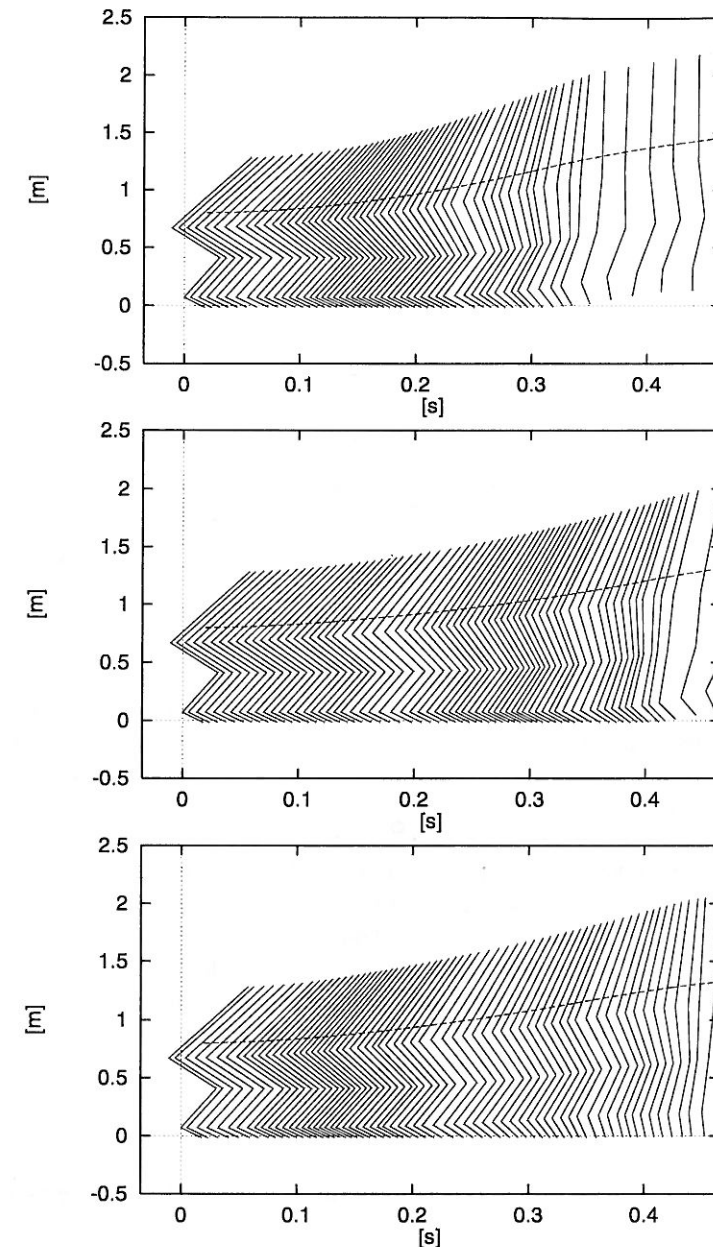
Einen Standhochsprung mit 1,51 m Sprunghöhe habe ich durch Optimieren auf maximale Höhe mit den in Abschnitt IV.1.5 angegebenen maximalen, gelenkspezifischen Momentenänderungen als Nebenbedingung erhalten. Wie sehen jetzt Beispiele von Sprüngen geringerer Höhe aus, die unter denselben Anfangs- und Nebenbedingungen gewonnen wurden?

Eine Regelbasis für einen Sprung mit der Sprunghöhe 1,4 m habe ich durch die Optimierung mit folgendem Kriterium erhalten:

$$|h - 1.4 \text{ m}| = \min \quad (\text{IV.22})$$

Das Ergebnis ist ein Sprung, der um 0,1 mm von der Sprunghöhe 140 cm abweicht. Wichtiger als das genaue Erreichen der vorgegebenen Sprunghöhe ist die

Abb. IV.15: (Seite 85) Pseudo-Stroboskopische Darstellung mit Schwerpunktskurve für die Sprunghöhen 1,51 m (**Oben**), 1,40 m (**Mitte**) und 1,33 m (**Unten**).



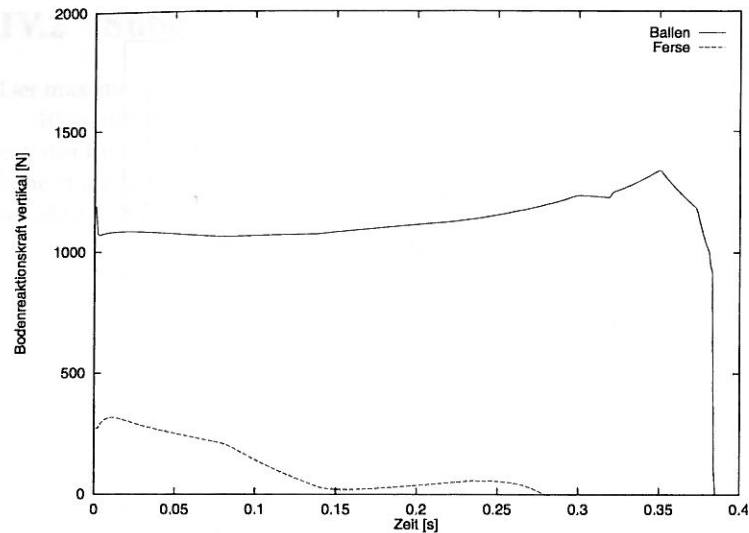


Abb. IV.16: Simulierte vertikale Bodenreaktionskraft an Ferse und Ballen für die Sprunghöhe 1,40 m.

Frage, ob die Koordination der Segmentbewegungen stimmt. Im Gegensatz zum maximalen Sprung ($h=1,51$ m) unter den gleichen Bedingungen, bei dem die Momente möglichst optimal eingesetzt werden, um eine große Höhe zu erreichen, hat das Modell bei einer 11 cm niedrigeren Sprunghöhe sozusagen Reserven übrig. Hier liegt der grundsätzliche Unterschied zum maximalen Sprung. Während beim maximalen Sprung die proximal zu distal Reihenfolge der Segmentbewegung durch die beschränkten Momente und die beschränkte Schnellkraft erzwungen wurde, hat das Modell beim submaximalen Sprung mehr Freiheiten.

Trotzdem ergab sich ohne weitere Vorgaben ein sehr guter Sprung. Die vertikale Bodenreaktionskräfte an Ferse und Ballen zeigen einen plausiblen Verlauf (siehe Abb. IV.16). Die horizontale Position des Schwerpunkts liegt zu Beginn hinter dem Ballen und schiebt sich während der Beschleunigungsphase langsam nach vorne. Diese Eigenschaft zeigten übrigens alle guten Sprünge, unabhängig von der Sprunghöhe.

Analog dazu wurde durch Optimierung eine Regelbasis für einen Sprung mit der Sprunghöhe 1,33 m gefunden.

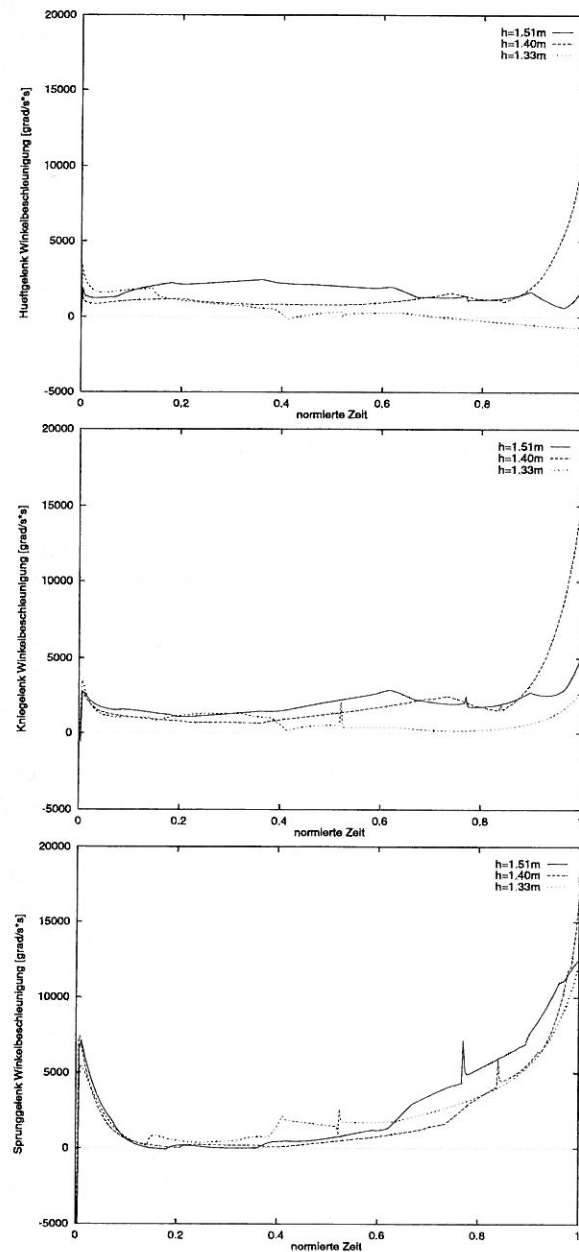
Abbildung IV.15 zeigt Strichdarstellungen für den maximalen Sprung und die zwei verschiedenen submaximalen Sprünge.

Abbildung IV.17 zeigt die dazugehörigen Gelenkwinkelbeschleunigungen. Für jedes Gelenk sind die Kurven der drei Sprunghöhen in ein Diagramm gezeichnet. Die Gelenkwinkelbeschleunigungen sind über eine, auf den Absprung normierte Zeitskala aufgetragen, da die Dauer der Beschleunigungsphase der drei Sprünge nicht identisch ist. Zum Teil liegen die Winkelbeschleunigungen der verschiedenen Sprunghöhen sehr dicht zusammen. (Dies konnte ich auch für weitere, hier nicht abgebildeten Sprünge, feststellen.) Für eine koordinierte Aufwärtsbewegung scheinen die möglichen Beschleunigungswerte vor allem am Anfang der Bewegung sehr eng begrenzt zu sein. Dies entspricht auch der Erfahrung, denn in der gehockten Stellung bewirkt zuviel Drehmoment in einem Gelenk sofort das Umkippen nach vorne, bzw. hinten, und da der Schwerpunkt so niedrig liegt, bleibt kaum Zeit zum Abfangen. Außerdem ist die Hocke eine ungünstige Haltung, für die die Kräfte auch im statischen Fall nicht beliebig lange aufgebracht werden können, und so nur begrenzt Momente zum Ausgleichen zur Verfügung stehen. Im Übrigen stimmen die Werte nicht überall mit der Erwartung überein, daß für den höheren Sprung tendenziell größere Beschleunigungen auftreten müssen.

Warum die Beschleunigungen kurz vor dem Abheben stark differieren, läßt sich besser verstehen, wenn man die Beschleunigungen gegen die vertikale Schwerpunktslage aufträgt. Die Abb. IV.18 zeigt für jedes Gelenk jeweils die Winkelbeschleunigungen der drei Sprünge gegen die vertikale Position des Gesamtkörperschwerpunkts aufgetragen. Die Kurven beginnen alle bei 80 cm, wo sich der Schwerpunkt in der Ausgangsstellung befindet. Bei 1,11 m ist die Standhöhe eingezeichnet, die zum Vergleich die Lage des Schwerpunkts des Modells beim aufrechten Stand angibt. Die Kurven brechen bei unterschiedlichen vertikalen Schwerpunktswerten ab, was unterschiedlichen Absprungshaltungen entspricht. Die Absprungsgeschwindigkeiten betragen 2,53 m/s (1,51 m), 1,94 m/s (1,40 m) und 1,82 m/s (1,33 m).

Wenn sich das Modell voll auf dem Fuß aufstellt, kann es fast 20 cm Höhe gewinnen. Bei einer maximal möglichen Sprunghöhe von 40 cm ist das die Hälfte. Die Sprunghöhe 1,33 m ist so gewählt, daß das Modell auf jeden Fall gerade noch abspringen muß. In die Sprunghöhe (Gl. IV.1) geht sowohl die Absprungshöhe als auch die Absprungsgeschwindigkeit ein. Es gibt für eine bestimmte Sprunghöhe also verschiedene Möglichkeiten der Kombination von Absprungshöhe und

Abb. IV.17: (Seite 88) Winkelbeschleunigungen gegen normierte Beschleunigungsphase für 1,51 m, 1,40 m und 1,33 m Sprunghöhe. **Oben:** Hüftgelenk, **Mitte:** Kniegelenk, **Unten:** Sprunggelenk.



-geschwindigkeit. Startet man mehrere Optimierungen erhält man tatsächlich unterschiedliche Absprungsbedingungen bei gleicher vorgegebener Sprunghöhe.

Auffallend in Abbildung IV.17 und IV.18 sind die Peaks, die mittendrin in fast allen Beschleunigungskurven erscheinen. Diese Spitzen sind in der Tat völlig unphysiologisch und unerwünscht. Sie rühren von der Fuzzy Regelung mit einer relativ kleinen Anzahl Regeln her und es gibt mehrere Möglichkeiten solche Artefakte zu vermeiden. (Nur der erste Ausschlag am Beginn der Bewegung kommt durch die deformierbaren Fußpolster und den abrupten Bewegungsbeginn).

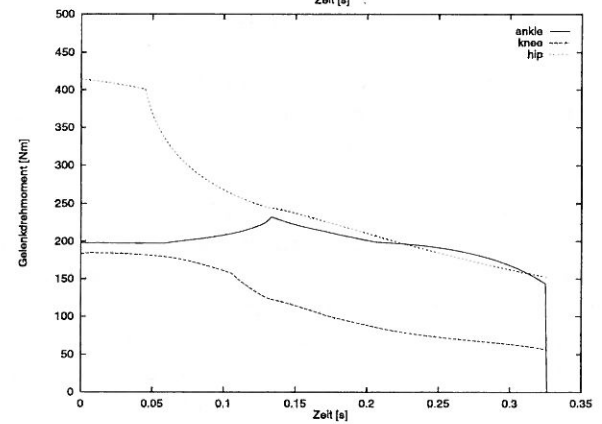
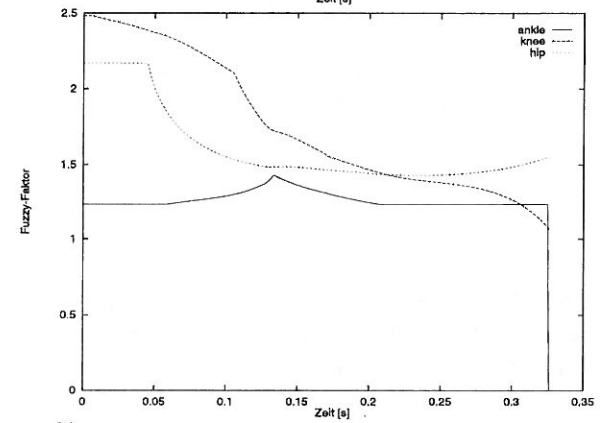
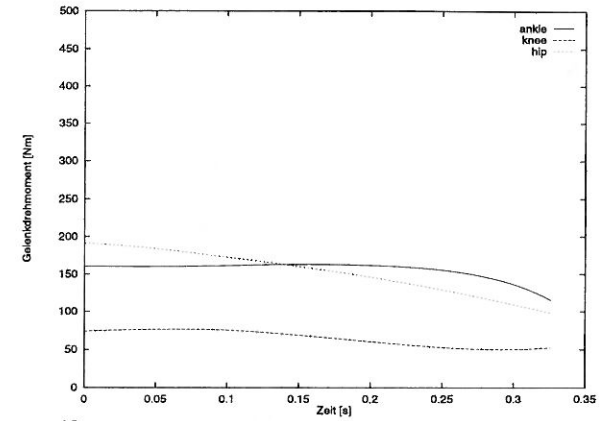
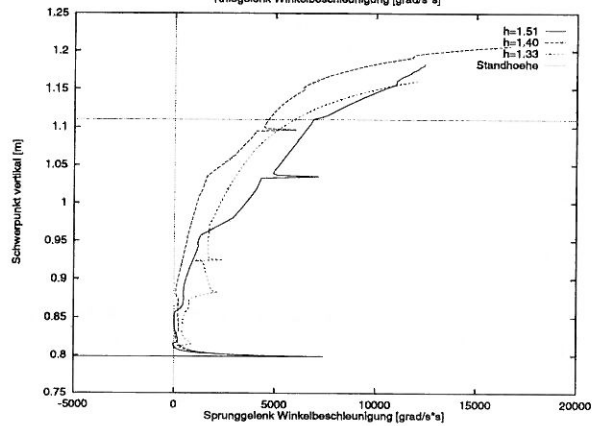
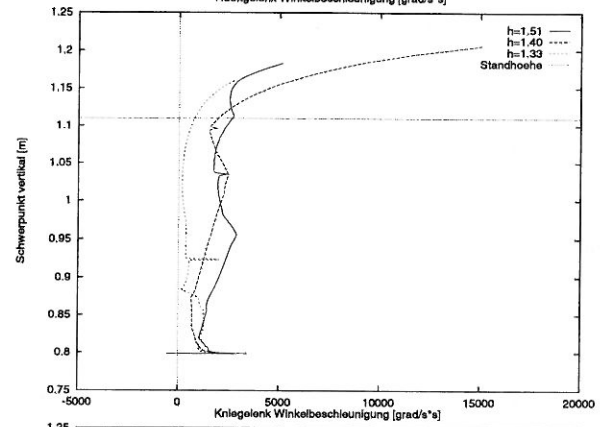
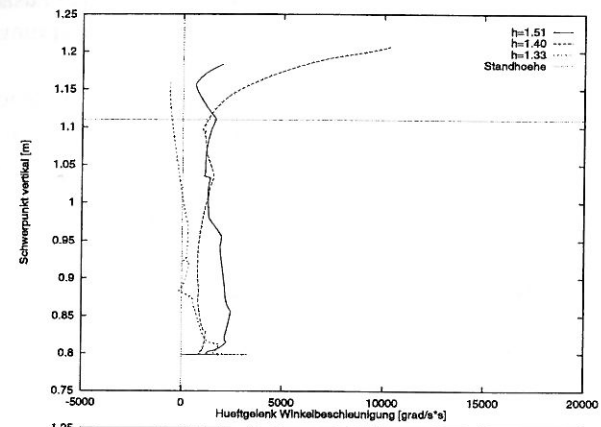
Ein Blick auf die Regelbasen zeigt die Ursache.

- maximaler Sprung: $h=1,51\text{ m}$, $t=0,284\text{ s}$
 Hüftgelenk: 3 5 5 2 4 5
 Kniegelenk: 5 5 5 3 1 2
 Sprunggelenk: 3 5 5 4 5 3
- submaximaler Sprung: $h=1,40\text{ m}$, $t=0,382\text{ s}$
 Hüftgelenk: 2 2 1 2 4 5
 Kniegelenk: 5 2 5 5 1 4
 Sprunggelenk: 2 2 3 4 2 1
- submaximaler Sprung: $h=1,33\text{ m}$, $t=0,327\text{ s}$
 Hüftgelenk: 4 4 1 4 2 1
 Kniegelenk: 5 4 1 3 2 3
 Sprunggelenk: 2 4 2 2 2 2

Die Zahlenkombinationen der Regelbasen sind nicht sehr anschaulich. Aber zwei Dinge kann man doch herauslesen. Zum ersten sind für den maximalen Sprung die Hälfte aller Positionen mit einer 5, der größt möglichen Zahl, besetzt. Für Sprunghöhe 1,33 m findet sich nur noch eine 5 und diese Entwicklung entspricht

Abb. IV.18: (Seite 90) Gelenkwinkelbeschleunigungen gegen vertikale Schwerpunktsposition für 1,51 m, 1,40 m und 1,33 m Sprunghöhe. **Oben:** Hüftgelenk, **Mitte:** Kniegelenk, **Unten:** Sprunggelenk.

Abb. IV.19: (Seite 91) Zeitverläufe für Sprunghöhe 1,33 m. **Oben:** Haltemomente M_{stat}^j , **Mitte:** dimensionslose Fuzzy-Faktoren $f^j(\dot{\varphi}^j)$, **Unten:** Gesamtmomente $M_{\text{ges}}^j(\varphi^j, \dot{\varphi}^j)$.



der Erwartung. Zweitens sieht man, daß oft an benachbarten Positionen, d. h. Winkelgeschwindigkeitswerte keine benachbarten Zahlen, d. h. Momentenwerte stehen. Das führt zu Kanten im Kennfeld. Am Beispiel der Simulation mit Sprunghöhe 1,33 m sind die Peaks gut zu erklären. Abbildung IV.19 zeigt im mittleren Diagramm die Kurven der Fuzzy-Faktoren. In den Regeln für das Sprunggelenk steht überall die Konsequenz 2 nur an der Stelle, die zur zweiten unscharfen Menge der Winkelgeschwindigkeit gehört, steht eine 4. Der Knick, der dadurch in der Kurve entsteht, führt auch zu einem Knick in der Momentenkurve (Abb. IV.19 unten), was sich wiederum in den Beschleunigungen deutlich als Peak bemerkbar macht.

In Abbildung IV.9 ist ein Beispiel für ein Kennfeld zu sehen, daß durch eine Regelbasis erzeugt wird, samt einer Trajektorie, die während einer Simulation mit bestimmten Anfangsbedingungen durchlaufen wird. Auch dort sind solche Kanten zu sehen, die zu den Knicken in den Kurven führen.

Eine Möglichkeit diese Kanten zu beseitigen, ist das Kennfeld einer Regelbasis durch Splines zu glätten und als *look up table* abzuspeichern. Dann wird während der Simulation nicht mehr zu jedem Zeitschritt die Fuzzy-Inferenz ausgeführt, sondern nur noch die Werte ausgelesen. Dies hat sogar den Vorteil, daß es schneller als die Berechnung ist. Für mehrere Eingangsvariablen sind die Splines aber nicht mehr trivial und außerdem braucht man viel Speicherplatz für das Kennfeld.

Eine weitere Möglichkeit wäre unmittelbar im Fuzzy-Ansatz solche Kanten auszuschließen. Dazu sind auf jeden Fall mehr unscharfe Mengen und mehr Regeln notwendig. Gaußfunktionen anstatt dreiecksförmige Zugehörigkeitsfunktionen geben glattere Kurven. Mit Genetischen Algorithmen kann dazu die optimale Anzahl, die Lage und die Form der unscharfen Mengen ermittelt werden.

Auf diese Maßnahmen habe ich hier verzichtet, denn ohne Anhaltspunkte aus Messungen von realen Sprüngen würde es sich nur um Schönheitskorrekturen handeln, aber keine weitere Erkenntnis bringen.

Abbildung IV.19 macht außerdem anschaulich, wie die Gelenkmomente als Produkt von Haltemoment und Fuzzy-Faktor zustandekommen. Das obere Diagramm zeigt, daß die Form der Haltemomentkurven durch die trigonometrischen Funktionen der Segmentwinkel bedingt ist. Die Anschlagmomente werden bei diesen Sprüngen während der Beschleunigungsphase nicht gebraucht. Dies ist eine weitere Eigenschaft aller optimierten Sprünge. Für die Simulationen während des Optimierungsprozesses spielen die Anschlagmomente aber sehr wohl eine Rolle.

IV.3 Standhochsprünge unter variierenden Startbedingungen

Der Mensch kann problemlos Standhochsprünge aus in gewissem Rahmen variierenden Startbedingungen ausführen. Eine der wichtigsten Gründe für die Anwendung von Fuzzy-Systemen zur Steuerung von Mehrkörpersystemen ist die Toleranz gegenüber Abweichungen in den Anfangsbedingungen oder kleinen Störungen.

Alle bisher gezeigten Standhochsprungsimulationen begannen in derselben Hockstellung, wie auf Seite 54 abgebildet. Oder anders ausgedrückt, die Vektoren $y(t_{on})$ und $\dot{y}(t_{on})$ wurden immer mit denselben Werten initialisiert. Ein Blick auf das Kennfeld in Abbildung IV.9 zeigt, daß die Trajektorie auch an einem anderen Punkt auf der Fläche beginnen könnte, d. h. eine Bewegung des Modells mit etwas anderen Anfangsbedingungen simuliert werden könnte. Allerdings wurde die Regelbasis, die diesem Kennfeld zugrunde liegt, aufgrund von Optimierung mit nur einer speziellen Anfangsbedingung gewonnen. Die Frage ist nun, ist der Steuerungsansatz so gut, daß auch unter etwas abweichenden Anfangsbedingungen, ohne neu optimieren zu müssen, eine vernünftige, wenn auch nicht optimale Bewegung resultiert.

Anhand eines Beispiels habe ich den Einfluß von verschiedenen Abweichungen untersucht und das Ergebnis sowohl als Strichfiguren (Absprungshaltung) in Abbildung IV.20, als auch als Tabelle (Sprunghöhe, Absprungszeit) dargestellt. Die Abbildung ist folgendermaßen zu verstehen. Für jeden Sprung ist nur ein Strichmännchen gezeichnet, und zwar in der Haltung, in der es zur Zeit t_{off} den Bodenkontakt verliert. Die Fußspitze zeigt auf der x -Achse die Zeit an, zu der das Modell abspringt. Der Referenzsprung ist ein submaximaler Sprung, der in der Animation sehr gut koordiniert aussieht. (Warum die volle Streckung erst nach dem Absprung erreicht wird, wurde beim maximalen Standhochsprung diskutiert.)

Die Art der Abweichungen sind zum einen Segmentwinkel- und Segmentwinkelgeschwindigkeitsdifferenzen. Zum anderen Unterschiede in Segmentmassen und Segmentträgheitsmomenten. Folgende Sprünge mit variierenden Startbedingungen wurden mit der Regelbasis des Referenzsprungs simuliert.

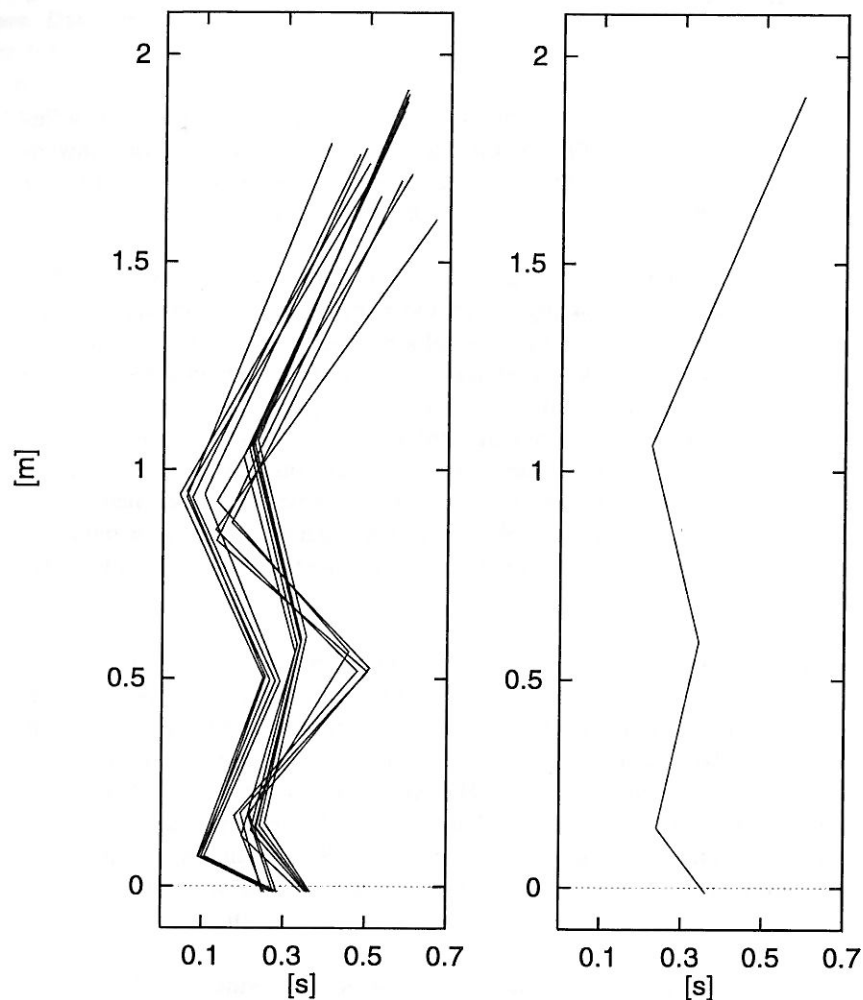


Abb. IV.20: Links: Absprunghaltungen bei verschiedene Anfangsbedingungen (siehe Tabelle auf Seite 93). Rechts: Referenzsprung. Die Fußspitze gibt auf der x -Achse den Absprungszeitpunkt an.

Nr.	Art der Abweichung bzgl. Referenzsprung	Sprunghöhe h	Zeit t_{off}
1.	Referenzsprung	1,384 m	0,362 s
2.	Rumpfwinkel $+5^\circ$	1,223 m	0,287 s
3.	Rumpfwinkel -5°	1,263 m	0,286 s
4.	Rumpfwinkelgeschw. $+30^\circ/s$	1,222 m	0,280 s
5.	Rumpfwinkelgeschw. $-60^\circ/s$	1,152 m	0,257 s
6.	Rumpf $+5^\circ$ und $+30^\circ/s$	1,222 m	0,275 s
7.	Unterschenkelwinkel $+5^\circ$	1,210 m	0,280 s
8.	Unterschenkelwinkel -5°	1,173 m	0,251 s
9.	Unterschenkelwinkelgeschw. $+30^\circ/s$	1,218 m	0,271 s
10.	Unterschenkelwinkelgeschw. $-30^\circ/s$	1,208 m	0,278 s
11.	Rumpfmasse $+5\%$	1,406 m	0,367 s
12.	Rumpfmasse -5%	1,358 m	0,355 s
13.	Rumpfrägheitsmoment $+5\%$	1,386 m	0,363 s
14.	Rumpfrägheitsmoment -5%	1,380 m	0,360 s
15.	doppelte Fußmasse	1,311 m	0,345 s
16.	doppeltes Fußträgheitsmoment	1,378 m	0,359 s
17.	Oberschenkel 10 cm länger	1,326 m	0,307 s

Bei fast allen Sprüngen ist die Beschleunigungsdauer kürzer als beim Referenzsprung. Der Absprung erfolgt offensichtlich bevor eine optimale Haltung eingenommen wurde und die Sprunghöhe ist geringer. Nur Sprung 11 und 13 haben eine längere Beschleunigungsdauer und erreichen auch eine größere Sprunghöhe als der Referenzsprung.

Neben der Flexibilität gegenüber abweichender Anfangsstellungen ist die Toleranz gegenüber Veränderungen in Masse und Trägheitsmomenten wichtig, denn ein Sportler z.B. verbessert durch Training nicht nur Kraft, Schnellkraft und Ausdauer sondern verändert eben auch seine Massenverteilung. Dagegen ist die Änderung von Segmentlängen mehr eine akademische Übung.

Die Absprungshaltungen in Abb. IV.20 kann man in drei Gruppen einteilen.

- Sprünge, die sich nur wenig vom Referenzsprung unterscheiden (Nr. 11-16).
- Sprünge, bei denen sich der Fuß kaum aufgerichtet hat und folglich das Sprunggelenk ungenügend beschleunigt hat (Nr. 2,4,6,7,9).
- Sprünge, bei denen der Fuß stark aufgerichtet ist und das Sprunggelenk zu stark beschleunigt hat, so daß der restliche Körper noch nicht ausreichend aufgerichtet ist (Nr. 3,5,8,10).

Die Art der Abweichung ist korreliert mit diesen drei Gruppen von Absprungshaltungen. Zur ersten Gruppe gehören alle Sprünge mit Abweichungen in Segment-

massen und -trägheitsmomenten (5 % Rumpfmasse entsprechen 3 kg). Sie zeigen geringe Unterschiede zum Referenzsprung und das ist auch nicht erstaunlich, denn diese Modellparameter gehen in die Gleichungen der Haltemomente und somit direkt in die Momentenberechnung ein. Nr. 11 und 13 erreichen sogar eine größere Sprunghöhe.

In der zweiten Gruppe finden sich die Sprünge, deren Abweichung in Winkel und Winkelgeschwindigkeit in die Richtung abweichen, in die die Bewegung gehen soll. Das scheint kein Vorteil zu sein.

Die dritte Gruppe beginnt mit Differenzen in den Startbedingungen, die der gewünschten Bewegung entgegengesetzt sind. Die zwei Sprünge mit der geringsten Höhe lassen sich aus ihren extrem ungünstigen Startbedingungen erklären. Nr. 5 hat am Simulationsbeginn eine Winkelgeschwindigkeit, die den Rumpf nach unten bewegt. Nr. 8 hat einen kleineren Unterschenkelwinkel, so daß das ganze Modell bei Bewegungsbeginn noch tiefer in der Hocke sitzt.

Alle Simulationen erreichen eine vertikale Schwerpunktsposition über der Standhöhe von 1,11 m. Außerdem werden während der Beschleunigungsphase keine extremen Winkelbereiche erreicht. Folglich bewirkt die für eine spezielle Modell- und Startbedingung optimierte Regelbasis auch für die Beispiele mit variierten Modell- und Startbedingungen eine Sprungbewegung.

Wenn es wirklich auf die Flexibilität der Steuerung ankommt, kann die optimale Lösung auch bezüglich einem Satz von Anfangsbedingungen gesucht werden. Noch besser ist es, die Optimierung mit Simulationen unter verrauschten Anfangsbedingungen durchzuführen.

IV.4 Hypothetische Prinzipien für Bewegungskoordination

Nur sehr wenige Bewegungen haben ein so eindeutiges Optimierungskriterium wie der maximale Standhochsprung. Für die meisten Bewegungsaufgaben ist das, oder sind die Zielkriterien, wenn es sie überhaupt gibt, weder eindeutig noch ersichtlich. Für die Simulation von Bewegung ist es aber oft unverzichtbar, den Zweck einer Bewegungsaufgabe in einer mathematischen Form zu formulieren.

Hier einige Beispiele für hypothetische Prinzipien, die in der Biomechanik für die Ausführung von willentlichen Bewegungen formuliert wurden:

- Minimaler Energieverbrauch
- Minimale Summe der Momente
- Minimaler Jerk
- Minimale Bewegungsdauer

Solche Hypothesen sind attraktiv, konnten aber experimentell bisher für keine Bewegung bestätigt werden [Johansson, 1991]. Ich nehme diese hypothetischen Prinzipien als Anregung, um Optimierungskriterien zu formulieren und an einem Beispiel den Einfluß auf meine Standhochsprungsimulationen zu untersuchen. Allerdings kann ich im Rahmen dieser Arbeit nicht klären, ob solche Kriterien tatsächlich einer Bewegung zu Grunde liegen und damit das Vorgehen überhaupt sinnvoll ist.

IV.4.1 Jerk

Unter *jerk* versteht man die Änderung der Beschleunigung einer Größe r , also die dritte Ableitung nach der Zeit [Schneider, 1989b].

Die biomechanische Hypothese *minimaler jerk* geht davon aus, daß der Mensch die Minimierung von Beschleunigungsänderungen anstrebt und somit ein Kriterium für eine gut gelernte, glatte Bewegung ist. Will man diese Hypothese für eine Simulation anwenden, stellt sich zunächst die Frage, welche Variable r in welchem Koordinatensystem dreimal abgeleitet werden soll. Weiter muß beantwortet werden, ob beispielsweise die Absolutwerte, die Betragsquadrate oder das Zeitintegral des *jerk* minimiert werden sollen.

Ich will feststellen, ob sich mit meinem Modell für Sprünge mit vorgegebener Sprunghöhe Unterschiede feststellen lassen, zwischen Optimierungen mit und ohne Jerk-Kriterium.

Für die Sprungsimulationen habe ich die dritte Ableitung des Gelenkwinkels vom Gelenk mit dem Index j gebildet:

$$J^j = \frac{d^3 \varphi^j}{dt^3} \quad (IV.23)$$

Unter *jerk cost* versteht man das Zeitintegral über das Betragsquadrat des jerks:

$$J_c^j = \int_{t_{on}}^{t_{off}} \left| \frac{d^3 \varphi^j}{dt^3} \right|^2 dt \quad (IV.24)$$

Es soll ein Sprung mit dem schon in Abschnitt IV.2.1 vorgestellten Sprung von 1,4 m Sprunghöhe verglichen werden. Dazu wird durch Optimieren mit dem Kriterium IV.22 für die Sprunghöhe und dem Kriterium, daß die Summe der *jerk costs* der drei Gelenke minimal sein soll, eine neue Regelbasis gewonnen.

$$\sum_{j=1}^3 J_c^j = \min \quad (IV.25)$$

In der Animation ist kein signifikanter Unterschied zwischen den beiden Sprüngen zu sehen, obwohl die Regelbasen fast nichts gemeinsam haben. Außerdem ist der Wert von $\sum J_c^j$ im optimierten Fall tatsächlich nur halb so groß. Die auftretenden maximalen Absolutwerte für J^j sind allerdings vergleichbar. Erstaunlicherweise ist die Dauer der Beschleunigungsphase nicht verlängert. Wo der signifikante Unterschied liegt, der für die unterschiedlichen *jerk costs* verantwortlich ist, wird aus den Kurvendarstellungen der Jerks und Momente nicht klar. Abb. IV.21 zeigt die Gelenkmomente der zwei Sprünge zum Vergleich. Man sieht, daß gleich zu Beginn unterschiedlich große Momente aufgrund unterschiedlicher Regeln wirken.

Lasse ich die Nebenbedingung beschränkte Momentenänderung weg, treten - wie zu erwarten ist - durch die Forderung nach Minimierung der *jerk costs* unrealistisch hohe, schmale *jerk*-Peaks auf. Im Übrigen ist der Steuerungsansatz zu grob, um eingehendere Untersuchungen zu diesem Thema zu erlauben.

IV.5 Animation und Performance

Der optische Eindruck durch die Animation ist ein sehr wichtiges Mittel zur Beurteilung einer simulierten Bewegung, die die Koordination mehrerer Gelenke voraussetzt. Der menschliche Beobachter erkennt unnatürliche Bewegungen in einer Animation sofort. Kleine Unstimmigkeiten, die aus den Zeitverläufen der Koordinaten nicht zu ersehen sind, springen in der Animation sofort ins Auge.

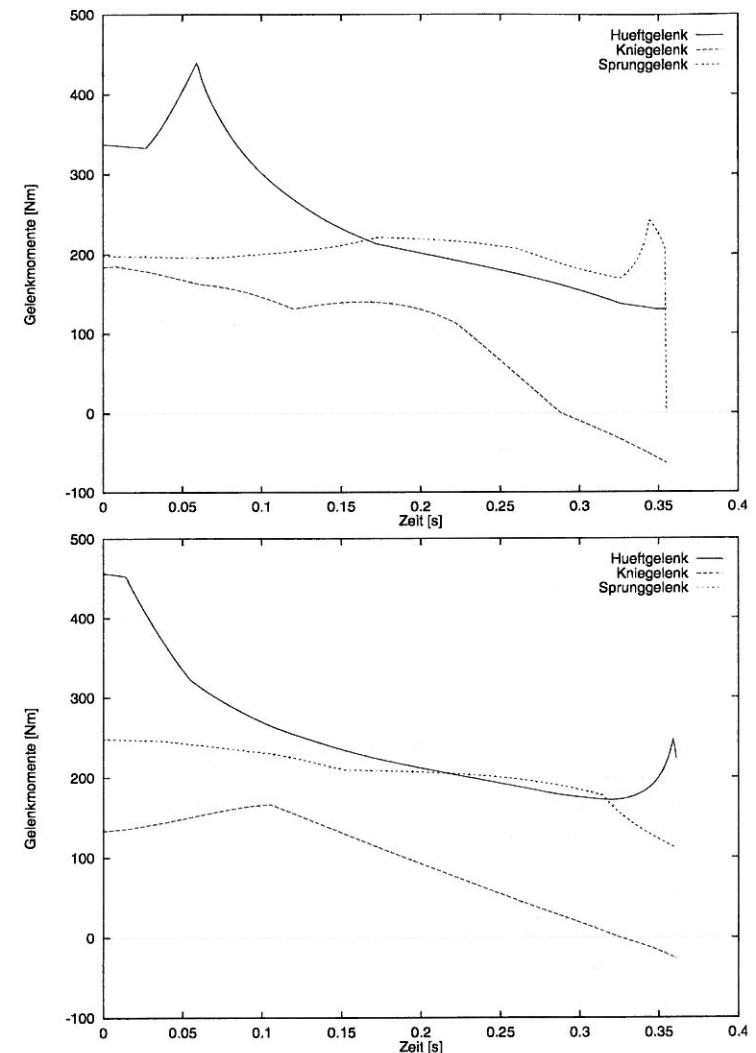


Abb. IV.21: Gelenkmomente (1,40 m): Vergleich zwischen einem nur auf Höhe optimiertem Sprung (unten) und einem auf Höhe und minimale Summe der *jerk costs* optimiertem Sprung (oben).

Die Abbildungen auf Seite 101 zeigen eine Animation in Form von einer Sequenz von Bildern, wie sie von einer Silicon Graphics Workstation dargestellt werden. Die Visualisierung erfolgte mit dem 3D-Grafik-Paket GL, der Graphics Library von Silicon Graphics. Rechts neben der Figur ist die Zeit in Sekunden eingeblendet und das Kreuz bezeichnet die Lage des Gesamtkörperschwerpunkts. Bei der on-line Animation kann durch interaktiven Eingriff über die Tastatur die Simulation angehalten werden und dann das Modell auf dem Bildschirm verschoben, gezoomt etc. werden.

Ich habe einige Dutzend Optimierungen mit verschiedenen Kriterien und Nebenbedingungen, mit Modifikationen in den Anfangsbedingungen und in der Fuzzy Regelung berechnet. Als Animationen habe ich hunderte verschiedener Sprünge beobachtet. Die Frage was ein guter Sprung ist, läßt sich abgesehen von den Fällen, in denen unphysiologische Werte auftreten, anhand der Daten nicht einfach entscheiden. Betrachtet man die Animationen, kann man „Haltungsnoten“ verteilen, aber quantifizieren läßt sich nur schwer, was einen gut koordinierten Sprung ausmacht.

IV.5.1 Programmstruktur

Das Struktogramm soll einen Überblick geben, wie das Programm für eine animierte on-line Simulation des Mehrkörpermodells aussieht und den Zusammenhang zu den erwähnten Methoden und Gleichungen herstellen.

Einlesen von Parameter für das Modell

Parameter für die Fuzzy Regelung

Anfangswerte (y, \dot{y})

Integrationsparameter (Dauer, Schrittweite, Fehler)

Regeln (Chromosome vom Genetischen Algorithmus)

Grafik initialisieren

Integrationsschritt $i=0$

do

Kontaktprüfung und Berechnung der Kräfte

Gelenkwinkel und -geschwindigkeiten berechnen

Prüfen welche Bewegungsphase vorliegt

Haltemomente und Anschlagmomente berechnen

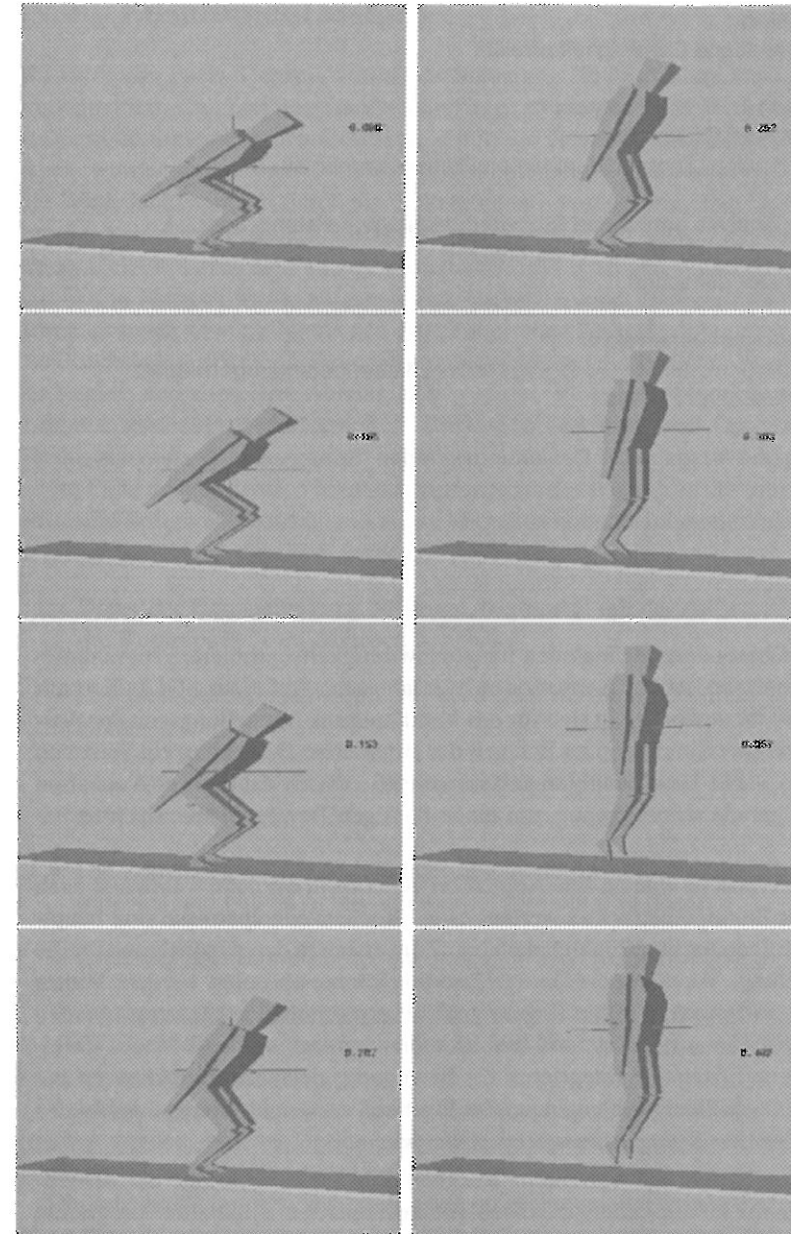
Faktoren mit Fuzzy Control berechnen

Gesamtmomente berechnen

do

Bewegungsgleichungen zur Verfügung stellen

Beschleunigungen und Geschwindigkeiten berechnen



Integrieren

until vorgegebene Genauigkeit erreicht

Koordinaten an Grafik übergeben

Darstellung auf Bildschirm

Dialog: Anhalten, Zoomen durch interaktives Kommando

Daten zur Analyse berechnen (Schwerpunktlage, Leistung, Jerk ...)

Auslesen der Daten für Kurvendarstellung

$i=i+1$ nächster Zeitschritt

until Integrationsdauer erreicht

oder Abbruchkriterium (Fuss abgehoben, Nebenbedingungen verletzt)

oder interaktiver Abbruch

Berechnung und Ausgabe der Bewertungskriterien (Sprunghöhe, Drehimpuls ...)

Performance

Da ich keine besondere Maßnahmen für geschwindigkeitsoptimierte Programmierung ergriffen habe, ist die Animation relativ langsam. Auf einer SGI-Indigo mit einem R4000 Prozessor ergibt sich für das Verhältnis aus Darstellungszeit der Animation bei on-line Simulation zu Echtzeit der simulierten Bewegung ein Wert von 30/1. Eine typische Beschleunigungsdauer von 0,3 s dauert dann in der Animation 9 s. Das ist gerade schnell genug, um einen flüssigen Bewegungseindruck zu bekommen.

Erstrebenswert ist eine on-line Animation in Echtzeit, die dann wahlweise auch in beliebiger Zeitlupe betrachtet werden kann. Rechenzeiterparnisse sind hauptsächlich an folgenden drei Punkten denkbar. Zum ersten in den Algorithmen für die Fuzzy-Regelung. Vor allem die Defuzzifizierung könnte optimiert werden. Weiter könnte man ausnutzen, daß die Regeln unabhängig simultan ausgewertet werden können bzw schon ausgewertet als *look up table* abgelegt werden können. Zweitens kann beim Lösen und Integrieren der Bewegungsgleichungen Rechenzeit gespart werden, wenn hier problemangepaßte Routinen verwendet werden. Als letztes kann noch das Grafikprogramm optimiert werden.

Erste Ergebnisse mit Verteiltem Rechnen mit DFN-RPC und der shared-memory Technik in Verbindung mit einem biomechanischen Simulationsprogramm sind in [Allrutz, 1994] nachzulesen.

IV.6 Zusammenfassung

Mit Hilfe von Fuzzy Control wurde eine Steuerung für das mehrgelenkige Modell implementiert. Das Ziel war Standhochsprünge zu simulieren. Für die Berechnung der Gesamtmomente in den Gelenken, die in die Newton-Euler-Gleichungen eingehen, wurden einige stark vereinfachende Annahmen gemacht. So ist eine spezielle Winkelabhängigkeit durch die Haltemomente vorgegeben. Die Dynamik des Sprungs wird durch Faktoren, die in Abhängigkeit von den Winkelgeschwindigkeiten der Gelenke durch das Fuzzy-Kalkül berechnet werden, bestimmt. Die Lage der unscharfen Mengen wurde aufgrund von Auswertung der Meßergebnisse anderer Arbeitsgruppen sowie allgemeinen physiologischen Nebenbedingungen definiert. Die Beschränkung der Gelenkmomente auf gelenkspezifische Werte ist im Entwurf des Reglers enthalten. Die Abhängigkeit von der Winkelgeschwindigkeit wird als kombinatorisches Problem formuliert. Dieses Problem wird durch Optimierung mit Genetischen Algorithmen gelöst.

Im Falle des maximalen Standhochsprungs ist das wichtigste Optimierungskriterium die Sprunghöhe und ohne weitere Vorgaben konnten folgende Charakteristika dieser Bewegung simuliert werden:

- Dauer der Beschleunigung 300 msec, beginnend mit der Aufwärtsbewegung des Rumpfes bis zum Abheben der Füße.
- Der Gesamtkörperschwerpunkt überwindet bezüglich der Standhöhe 0,4 m Höhe.
- Eine annähernd konstante Bodenreaktionskraft während der Beschleunigungsphase.
- Die Zeitabfolge der Bewegung der Segmente Rumpf, Oberschenkel, Unterschenkel und Fuß ist richtig koordiniert (proximal zu distal).
- Gleichzeitiges Erreichen der maximalen Winkelgeschwindigkeiten.

Die Animation der Bewegung und die Visualisierung der Aktivität der Fuzzy-Regeln sind wichtige Analyseinstrumente, die zur Beurteilung einer simulierten Bewegung eingesetzt wurden.

Auch Sprünge mit geringerer Sprunghöhe konnten mit diesem Steuerungsansatz simuliert werden. Nur das Optimierungskriterium muß angepaßt werden. Es lagen allerdings keine Meßdaten zum Vergleich vor.

Obwohl der Fuzzy-Regler nur mit wenigen, diskreten unscharfen Mengen arbeitet, resultieren daraus differenzierte Zeitverläufe einer Steuergröße für die Gelenk-

momente. Die Flexibilität der Steuerung wurde unter verschiedenen Startbedingungen ausgetestet. Die winkelgeschwindigkeitsabhängigen Multiplikatoren werden durch den Genetischen Algorithmus unüberwacht gelernt. Wenn man sich die Zwischenergebnisse der Optimierung ansieht, hat man in der Tat den Eindruck das Lernen der Bewegung in Bezug auf das verwendete Kriterium (maximale Höhe, minimaler Drehimpuls etc) zu beobachten. Genetische Algorithmen sind als Optimierungsmethode für diese Problemstellung ausgezeichnet geeignet. So gut, daß das Ergebnis oft die Unzulänglichkeiten des Modells aufdeckt und „pathologische“ Situationen herausfindet. Deshalb können Fragestellungen, die das Lernen betreffen, nur mit einem sehr guten und getesteten Modell angegangen werden.

Eine wesentliche Einschränkung für die weitere Anwendung der Steuerung auf andere Bewegungen stellt die spezielle Winkelabhängigkeit der Gelenkmomente durch die Berechnung auf der Basis der Haltemomente dar. Dieses Vorgehen stellt nur eine unzureichende Modellierung der tatsächlichen komplizierten Abhängigkeit der Gelenkmomente von Gelenkwinkeln und Gelenkwinkelgeschwindigkeiten dar. Vor allem Maximalmomente und maximale Gradienten der Momente müssen sich besser am biologischen Vorbild orientieren.

Prinzipiell ist es nicht möglich, aus dem vorgestellten Ansatz und den damit gewonnenen Simulationsergebnissen Rückschlüsse auf die krafterzeugenden oder energietransportierenden, physiologischen Strukturen des Körpers zu ziehen. Die Gesamtgelenkmomente lassen sich nicht einfach ohne Zusatzannahmen in Muskelkräfte zerlegen. Dieses Problem ist hochgradig redundant, denn jedes Gelenk wird von mehreren Muskeln bewegt, manche Muskeln ziehen über mehrere Gelenke, Agonisten und Antagonisten sind gleichzeitig aktiv, Energie wird elastisch in Sehnen gespeichert, Hebelarme sind winkelabhängig etc.

Die Steuerung in der hier verwendeten Form auf der Ebene von Gesamtgelenkmomenten ist insbesondere geeignet, die Eigenschaften des mechanischen Modells bei einer speziellen dynamischen Bewegung zu erforschen und zu verbessern. Der vorzeitige Absprung des Modells zeigt die Limitation der Simulationen durch die Modellierung der Segmente als starre Körper. Die Ergebnisse meiner Simulationen zeigen deutlich, daß für Sprungsimulationen ein besseres Fußmodell, nicht nur mit Polster, sondern mit einer elastischen Struktur entwickelt werden muß. Diese Beobachtung steht auch im Einklang mit Simulationsergebnissen anderer Arbeitsgruppen. Erst wenn diese Fehlerquelle behoben ist, ist es sinnvoll, eine detailliertere Steuerung für Sprungsimulationen einzusetzen.

IV.7 Ausblick

Der Ausblick auf Fragestellungen und Möglichkeiten der Vorwärtssimulation von biomechanischen Mehrkörpermodellen mit Fuzzy-Regelung hat viele Aspekte. Zwei Punkte will ich konkretisieren:

Der erste Punkt geht von der Voraussetzung aus, daß geeignete Vergleichsdaten von realen Bewegungen zur Verfügung stehen. In diesem Fall kann eine bestehende Fuzzy-Regelung mit einem Fuzzy-Neuro-Verfahren verbessert werden (siehe Abschnitt III.2.8). Da sich die, aufgrund von Meßdaten gelernte Veränderung in den Regeln manifestiert, kann sie physikalisch interpretiert werden. Auf diese Weise können Abhängigkeiten und Korrelationen aufgedeckt werden.

Eine Alternative baut darauf auf, daß es wünschenswert ist, für ein Problem das Fuzzy-System und seine Parameter von Anfang an **systematisch** zu gewinnen. Tatsächlich zeigt M. Sugeno [Sugeno, 1993], wie eine systematische Analyse von Daten mittels unscharfer Datenanalyse vorgenommen werden kann. Aus den numerischen Input-Output Daten wird ein Fuzzy Modell gewonnen, ohne a-priori-Kenntnisse über das System zu benötigen. Mit diesem sogenannten Fuzzy-Clustering kann man aus Meßdaten ein Fuzzy-System gewinnen, das zur Vorwärtssimulation geeignet ist und außerdem Rückschlüsse auf die Struktur des Problems zuläßt. Dieses Verfahren ist also in gewisser Weise analog zur Inversen Dynamik. Der interessanteste Punkt an diesem Ansatz ist die Strukturidentifikation. Aus möglichen Kandidaten von Eingabevariablen werden die Variablen bzw. die Kombination von Variablen systematisch ermittelt, die die Daten am besten reproduzieren. Zuerst wird die Partition vom Ausgaberaum durch Fuzzy Clustering ermittelt. Daraus folgen die Regeln und die Eingaberaumpartition. Erst ganz zum Schluß wird Parameteridentifikation durch Optimierung vorgenommen. Dieses Verfahren steht für zeitabhängige Probleme erst in den Anfängen.

Der zweite Punkt bezieht sich auf den Gesichtspunkt, mit Fuzzy-Regelung einen wichtigen, nichtlinearen Parameter, über den ein gewisses qualitatives und intuitives Wissen vorliegt, in einen größeren Modellzusammenhang einzubauen. Während die eben erwähnten Verfahren davon ausgehen, möglichst die ganze Steuerung auf Fuzzy-Basis zu stellen, soll jetzt nur gezielt ein Parameter einer Steuerung fuzzifiziert werden. (So wie in den Standhochsprungsimulationen nur die f^j durch Fuzzy Control berechnet wurden.)

Interessant ist in diesem Zusammenhang an das Modell von [Taga, 1991] anzuknüpfen. Taga benutzt zur Vorwärtssimulation ein dem hier vorgestellten Modell vergleichbares, allerdings zweibeiniges Mehrkörpersystem. Der Bewegungsapparat wird auch durch Newton-Euler-Gleichungen beschrieben und die Bodenreaktionskraft wird durch ein Feder-Dämpfer-System berechnet.

Angeregt durch die Erkenntnisse aus Versuchen mit dezerebrierten Katzen hat Taga eine Strategie zur Bewegungskontrolle beim Gehen und Laufen entwickelt. Zentrales Element ist ein spinaler Lokomotionsgenerator (Neural Rhythm Generator), der aus Differentialgleichungen für gekoppelte neurale Oszillatoren besteht. Zu jedem Gelenk gehört ein Oszillator, der aus einem Flexor-Neuron und einem Extensor-Neuron besteht, die sich gegenseitig hemmen. Das Flexor-Neuron eines Gelenks ist inhibitorisch mit dem Flexor-Neuron des anderen Beines verbunden usw. Auf diese Weise generiert der spinale Lokomotionsgenerator die alternierende und synchronisierte Bewegung der Gelenke beim Gehen.

In die Gleichungen des spinalen Generators gehen außerdem noch folgende Informationen ein:

- 1) Der Einfluß von supraspinalen lokomotorischen Regionen geht in Form einer additiven Konstanten in die Differentialgleichungen ein. Dadurch ändert sich die Amplitude, nicht aber die Frequenz der Generatoraktivität.
- 2) Feedback durch die Sensoren der Propriozeption (Haut, Muskeln, Gelenke), die die Stellung und Bewegung der Segmente und Gelenke erfassen, sowie Feedback über externe Ereignisse wie beispielsweise Bodenkontakt.

Der spinale Lokomotionsgenerator (Oszillatorgleichungen) wechselwirkt mit dem Bewegungsapparat (Newton-Euler-Gleichungen) in einer Art und Weise, daß Bewegung sensorische Konsequenzen hat und umgekehrt. Denn einerseits werden die wirkenden Drehmomente durch den Zustand der Neuronen bestimmt und andererseits ist der Zustand der Neuronen durch den sensorischen Feedback, also dem Zustand des Bewegungssystems, abhängig.

Wie hängt jetzt die Bewegung von den drei Hierarchiestufen supraspinaler Faktor, spinaler Generator und sensorischer Feedback ab?

- Der supraspinale Faktor bestimmt die Geschwindigkeit der rhythmischen Bewegung und ist somit auch für den Phasenübergang von Gehen zu Laufen verantwortlich.
- Der spinale Generator verursacht die rhythmische Bewegung durch Drehmomente in den Gelenken.
- Der sensorische Feedback ermöglicht stabile Bewegungszyklen und in gewissem Maße auch eine Adaption an veränderte externe Bedingungen wie unebenes Gelände.

Dieser Ansatz von Taga ist in mehrfacher Hinsicht neu und bemerkenswert für die funktionelle Biomechanik. Der Ansatz des Neural Rhythm Generators als Modell der spinalen Kontrolle und eines einfachen supraspinalen Einflusses ist geeignet, viele bekannte neurophysiologische Eigenheiten von periodischen Bewegungen zu reproduzieren. Die Verschaltung der Oszillator-Neurone und ihre Zeitkonstanten sind vom Modell und der Bewegungsaufgabe abhängig. Es gehen keine Parameter

ein, die stark interindividuell variieren.

Den wichtigste Anknüpfungspunkt für Modifikationen sehe ich im Feedback. Denn es ist nicht geklärt, in welchem funktionellen Zusammenhang der sensorische Feedback mit dem Bewegungsgenerator steht. Taga hat für das Gehen durch *trial and error* eine einfache Möglichkeit gefunden, wobei zu beachten ist, daß das Gehen auch ganz ohne diesen Feedback möglich, wenn auch nicht in jedem Fall stabil, ist.

Es stellt sich die Frage, wie der sensorische Feedback für eine andere Bewegung, z. B. dem Auf- und Ab-Hüpfen, lauten muß. Hier genügt die spinale Kontrolle vielleicht nicht mehr, denn es erfordert eine wiederholte Gleichgewichtskorrektur. Trotzdem wäre es interessant zu untersuchen, ob propriozeptive (Position, Bewegung) und exterozeptive (visuelle, somatische, vestibulare) Informationen auf die Bewegung Einfluß nehmen können. Fuzzy Control stellt eine einfache und direkte Möglichkeit dar, diese sensorischen Informationen mit der Bewegung zu koppeln.

A

Anhang

A.1 Anthropometrische Daten

Ausgabe des Programms *calcman* [Hahn, 1993]:

Anthropometrische Daten des menschlichen Koerpers
=====

Alle Angaben in MKS Groessen!

Laengen in m, Massen in kg, Traegheitsmomente in kgm²

ACHTUNG: Segmente von Arm+Bein sind mit doppelter Masse und
und doppeltem Traegheitsmoment behaftet!

Referenzpunkte fuer die Schwerpunkte wie im Sourcecode beschrieben

Eingabedaten:

Geschlecht: m Koerpergroesse: 1.93 m Koerpermasse: 84.00 kg

Oberschenkel:

Masse: 17.717 kg
Laenge: 0.4728 m
Schwerpunkt: x: 0.0000 y: 0.1938 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:0.316969

Unterschenkel:

Masse: 7.152 kg
Laenge: 0.4672 m
Schwerpunkt: x: 0.0000 y: 0.2056 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:0.095458

Fuss:

Masse: 2.116 kg
Laenge: 0.2994 m
Schwerpunkt: x: 0.0690 y: 0.0693 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:0.009189
Fusshoehe: 0.1080 m

Ferse:

Masse: 0.466 kg
Laenge: 0.1134 m
Schwerpunkt: x: -0.0230 y: 0.0720 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:0.000246
Winkel: 0.309665 rad

Ballen:

Masse: 1.651 kg
Laenge: 0.1846 m
Schwerpunkt: x: 0.0950 y: 0.0685 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:0.002894
Winkel: 0.945952 rad

Unterschenkel & Fuss:

Masse: 9.269 kg
Laenge: 0.5752 m
Schwerpunkt: x: 0.0000 y: 0.2811 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:0.283459

Kopf & Hals & Rumpf & Arme:

Masse: 57.015 kg
Laenge: 0.8821 m
Schwerpunkt: x: 0.0000 y: 0.2853 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:3.237704

gesamter Koerper:

Masse: 84.000 kg
Laenge: 1.9300 m
Schwerpunkt: x: 0.0000 y: 1.1175 z: 0.0000
Traegheitsmomente: xx: 0.000000 yy:0.000000 zz:14.004621

sonstige Daten:

Federlaenge im Fussgewoelbe: 0.118073 m
Laenge der Handflaeche: 0.124 m
Schulterhoehe: 1.598 m
Schultergelenkhoehe: 1.573 m
Kopfumfang: 0.610 m
Halsumfang: 0.431 m

A.2 Eingabedatei

```

Kommentarzeile
ja                               Ausgabe ja/nein
jumperg.dat                     Ausgabedateiname
Modellparameter (beidbeinig) aus calcman m.1.93.84.0
0.458d0 0.485d0 0.92d0 11, 12, 13 [m]
Fuss:
0.20d0 0.205d0 24.0d0 2.5d0 0.02d0 L0, LF, beta, M0, T0
Massen und Traegheitsmomente: [kg] [kgm^2]
MIK  MIW  THETAIK  THETAIW
7.1d0 0.0  0.14d0  0.0    UNTERSCHENKEL
17.4d0 0.0  0.42d0  0.0    OBERSCHENKEL
57.0d0 0.0  3.9d0  0.0    RUMPF
0.0  0.0  0.00  0.0    OBERARM
0.0  0.0  0.00  0.0    UNTERARM
ja                               Anschlaege ja/nein
Anfangskoordinaten und Anfangsgeschwindigkeiten
0.0  0.0          XF, XFP [m] [m/s]
0.07d0 0.0       ZF, ZFP Knoechelhoeh+Fersenpolster
48.d0  0.0       PHK1, PHK1P [grad] [grad/s]
148.d0  0.0       PHK2, PHK2P
42.d0  30.0      PHK3, PHK3P
153.80d0 0.0     PH0, PH0P
Integrationsdaten [s]
0.6d0 0.001d0 1.d-6 1.d-6 dau, hde, relf, absf

```

Literaturverzeichnis

- [Allrutz, 1994] R. Allrutz: Anwendungen des DFN-RPC in der Physik, *Dissertation* Tübingen, 1994.
- [Bandemer, 1993] H. Bandemer and S. Gottwald: Einführung in Fuzzy-Methoden, Akademie Verlag, 1993.
- [Beek, 1989] J.P. Beek: Juggling Dynamics, *Dissertation*, Free University Press, Amsterdam, 1989.
- [Bobbert, 1988] F. Bobbert, G. van Ingen Schenau: Coordination in vertical jumping, *Journal of Biomechanics*, 21,3,249–262, 1988.
- [Delp, 1990] S.L. Delp: Surgery Simulation: A Computer Graphics System to Analyze and Design Musculoskeletal Reconstructions of the Lower Limb, *Dissertation*, Stanford University, 1990.
- [Erhardt, 1992] M. Erhardt: Computersimulation für lernfähige biomechanische Modelle, *Diplomarbeit* Tübingen, 1992.
- [Gottwald, 1993] S. Gottwald: Fuzzy Sets and Fuzzy Logic, Vieweg, 1993.
- [Gruber, 1987] K. Gruber: Entwicklung eines Modells zur Berechnung der Kräfte im Knie- und Hüftgelenk bei sportlichen Bewegungsabläufen mit hohen Beschleunigungen, *Dissertation* Tübingen, 1987.
- [Hahn, 1993] U. Hahn: Entwicklung mehrgliedriger Modelle zur realistischen Simulation dynamischer Prozesse in biologischen Systemen, *Diplomarbeit* Tübingen, 1993.
- [Herold, 1994] H. Herold, A. Geyer: Verteilte Anwendungen über Hochgeschwindigkeitsdatenkommunikation, DFN-Vorhaben TK 558-VA 005.3, 3. Meilensteinbericht, März 1994.

- [Holland, 1975] J. H. Holland: *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Mich. 1975.
- [Hospach, 1993] F. Hospach: *Computersimulationen von Abstürzen mittels biomechanischer Ganzkörpermodelle*, *Dissertation* Tübingen, 1993.
- [Jang, 1993] J.-S. Roger Jang: *Functional Equivalence between Radial Basis Function Networks and Fuzzy Inference Systems*, *IEEE Transactions on Neural Networks*, Jan. 1993.
- [Johansson, 1991] R. Johansson, M. Magnusson: *Optimal Coordination and Control of Posture and Locomotion*, *Mathematical Bioscience* 103, 203-244, 1991.
- [Kandel, 1991] E. Kandel, J. Schwartz, T. Jessell, *Principles of Neural Science*, Elsevier, 1991.
- [Kosko, 1992] B. Kosko: *Neural Networks and Fuzzy Systems, a Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, 1992.
- [Krebs, 1991] T. Krebs: *Biomechanische Simulationen an mehrgliedrigen Modellen zur Untersuchung von Bewegungsabläufen der Maus*, *Diplomarbeit* Tübingen, 1991.
- [Krieg, 1992] M. Krieg: *Simulation und Steuerung biomechanischer Mehrkörpersysteme*, *Diplomarbeit* Tübingen, 1992.
- [Kruse, 1993] R. Kruse: *Fuzzy-Systeme*. B.G. Teubner Stuttgart, 1993
- [Kuo, 1993] A. Kuo, F. Zajac: *A Biomechanical Analysis of Muscle Strength as a Limiting Factor in Standing Posture*, *Journal of Biomechanics*, Vol. 26, Supplement 1, 137-150, 1993.
- [Lippert, 1993] H. Lippert: *Lehrbuch Anatomie*, Urban & Schwarzenberg, 1993.
- [McGeer, 1990] T. McGeer: *Passive Dynamic Walking*, *The International Journal of Robotics Research*, Vol. 9, No. 2, April 1990.
- [NASA, 1978] *Anthropometric Source Book*, NASA Reference Publication 1024, National Technical Information Service, Springfield, 1978.
- [NEWEUL, 1988] *Programmsystem NEWEUL'88*, Institut B für Mechanik, Universität Stuttgart, 1988.

- [Nigg, 1977] B. Nigg: *Biomechanik*, Juris Druck und Verlag Zürich, 1977.
- [Pandy, 1990] M. Pandy et al.: *An Optimal Control Model for Maximum-Height Human Jumping*, *Journal of Biomechanics*, Vol. 23, No. 12, 1185-1198, 1990
- [Pandy, 1991] M. Pandy, F. Zajac: *Optimal Muscular Coordination Strategie for Jumping*, *Journal of Biomechanics*, Vol. 24, No. 1, 1-10, 1991.
- [Reusch, 1993] B. Reusch (Hrsg.): *Fuzzy Logic, 2. Dortmunder Fuzzy Tage 1992*, Springer, 1993.
- [Ritter, 1991] H. Ritter, T. Martinetz, K. Schulten: *Neuronale Netze: eine Einführung in die Neuroinformatik selbstorganisierter Netzwerke*, Addison-Wesley, 1991.
- [Ruder, 1994] H. Ruder et al.: *Die Dynamik der Körpermassen: Einfache Modelle zur Simulation von Körperbewegungen und Aufprallvorgängen*. In *Biomechanik-Rekonstruktion* (Oehmichen und König, Hsg) Schmidt-Römhild, Lübeck, 1994.
- [Schneider, 1989b] K. Schneider, R. F. Zernicke: *Jerk-Cost Modulations During the Practice of Rapid Arm Movements*, *Biological Cybernetics*, 60, 221-230, 1989.
- [Sepulveda, 1993] F. Sepulveda, D. Wells, C. Vaughan: *A Neural Network Representation of Electromyography and Joint Dynamics in Human Gait*, *Journal of Biomechanics*, Vol 26, Nr 2, 101-109, 1993.
- [Shampine, 1975] L. Shampine, M. Gordon: *Computer Solution of Ordinary Differential Equation*, Freeman and Co, San Francisco, 1975.
- [Soest, 1992] A. van Soest: *Jumping from structure to control*, *Dissertation*, Freie Universität Amsterdam, 1992.
- [Soest, 1993 a] A. van Soest et. al: *The influence of the biarticularity of the gastrocnemius muscle on vertical-jumping achievement*, *Journal of Biomechanics* 26, 1, 1-8, 1993
- [Soest, 1993 b] A. van Soest, M. Bobbert: *The contribution of muscle properties in the control of explosive movements*, *Biological Cybernetics* 69, 195-204, 1993.

- [Sugeno, 1993] M. Sugeno, T. Yasukawa: A Fuzzy-Logic-based Approach to Qualitative Modeling. IEEE Transactions on Fuzzy Systems, Vol.1, No.1, Feb. 1993.
- [Taga, 1991] G. Taga and Y. Yamaguchi and H. Shimizu, Self-organized control of bipedal locomotion by neuronal oscillators in unpredictable environment, Biological Cybernetics 65, 147-159, 1991.
- [van Leeuwen, 1992] J.L. van Leeuwen: Muscle Function in Locomotion, in McN. Alexander (Ed.): Advances in Comparative and Environmental Physiology, Vol. 11, Springer, 1992.
- [Widmayer, 1990] K. Widmayer: Simulation von Bewegungsabläufen beim Menschen mit Hilfe von Mehrkörpersystemen, *Diplomarbeit* Tübingen, 1990.
- [Zadeh, 1973] L. Zadeh: Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man, and Cybernetics, SMC-1, 1973.
- [Zajac, 1993] F. Zajac: Muscle Coordination of Movement: A Perspective, Journal of Biomechanics, Vol. 26, Supplement 1, 102-124, 1993.
- [Zell, 1994] A. Zell: Simulation Neuronaler Netze, Addison-Wesley, 1994.

Danksagung

Ich bedanke mich bei Prof. Ruder für die Möglichkeit an seinem Lehrstuhl auf dem Gebiet der Biomechanik zu promovieren. Insbesondere bedanke ich mich für das Vertrauen, das mir zuteil wurde und für die Freiheit beim Forschen, die ich während der Entstehungszeit dieser Dissertation genossen habe.

Ich bedanke mich bei den „Altmeistern“ der Biomechanik Karin Gruber und Frank Hospach sowie bei Jörg Subke, Thomas Krebs, Uli Hahn, Michael Günther, Andreas Boose, Oana Schützler und allen anderen Biomechanikern für gute Zusammenarbeit und Freundschaft.

Ich bedanke mich bei allen anderen KollegInnen am Lehrstuhl, die neben ihrer eigenen Forschung auch Zeit für mein *ill-posed problem* „Warum tun die Rechner nicht was ich will?!“ gefunden haben.

Ich bedanke mich bei Prof. Klaus Schneider für die Diskussion über die Anwendung von Softcomputing-Methoden in der Biomechanik.

Ich bedanke mich bei Susanne Huber und Gaby Zinßmeister, die meine Arbeit sowohl inhaltlich kritisch betrachteten als auch Lust und Frust beim Promovieren mit mir teilten.

Ich bedanke mich beim Stiftungsverband Regenbogen e. V. für die zweieinhalbjährige finanzielle Unterstützung mit einem Promotionsstipendium aus Mitteln der Begabtenförderung des Bundesministeriums für Bildung und Wissenschaft. Insbesondere bedanke ich mich bei Jutta Helm und Beate Kuboth-Asbach vom Studienwerk des Buntstift e. V. für die persönliche Betreuung.

Lebenslauf

Karin Ulrike Widmayer, geboren am 12. März 1964 in Sindelfingen

Eltern:

Theo Widmayer

Lieselotte Widmayer, geb. Kienle

Schulausbildung:

1970-1974

Grundschule Renningen

1974-1983

Johannes-Kepler-Gymnasium Weil der Stadt

Universitätsausbildung:

ab WS 1983

Physikstudium an der

Eberhard-Karls-Universität Tübingen

7.3.1986

Vordiplom

10.1.1991

Diplom

Januar 1991–

Promotion am Lehr- und Forschungsbereich

November 1994

Theoretische Astrophysik und Computational Physics

unter Anleitung von Prof. Dr. H. Ruder

Computersimulationen mit biomechanischen

Mehrkörpermodellen und Fuzzy-Regelung zur

Untersuchung der Bewegungskoordination

am Beispiel Standhochsprung

Stipendium:

1.4.1992–

Promotionsstipendium vom

31.9.1994

Stiftungsverband Regenbogen e. V.

Akademische Lehrer an der Universität Tübingen:

H. Kaul, M. Wolff, M. Baumann, H. Lichte, W.-D. Hartmann,

G. Staudt, H. Ruder, A. Schwenk, F. Lenz, H. Müther,

A. Fäßler, O. Lutz, G. Wagner, J. Strähle, H. Herold,

K.-H. Herrmann, F. Gönnerwein, K.-H. Gaukler, R. Schief.