



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Technische Universität Darmstadt
FB20 Informatik
Fachgebiet Simulation und Systemoptimierung

Bachelor of Science Thesis - Bachelor Arbeit

Simulationsbasierte Optimierung einer bionisch inspirierten, unteraktuierten, flexiblen Laufmaschine

Simulation-based optimization of a bionic-inspired,
underactuated, flexible walking machine

von



Reimar Tausch

Betreuer: Dipl.-Tech. Math. Maximilian Stelzer

Darmstadt, 2006

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe.

Darmstadt, den 20. September 2006

Reimar Tausch

Kurzzusammenfassung — Elastizität ist in konventionell gebauten Robotern ein meist unerwünschter Nebeneffekt, der die Systemkontrolle erheblich erschweren kann. Deshalb wird diese Eigenschaft mittels starren kinematischen Ketten und komplexen Kontrollalgorithmen so stark wie möglich unterdrückt. Im Gegensatz dazu weist der menschliche Bewegungsapparat einen hohen Grad an Flexibilität und ausreichender Stabilität auf. In dieser Arbeit werden die Bewegungsmuster einer bionisch inspirierten Laufmaschine mit bewusst nachgiebig elastischen Beinen anhand eines Software-Modells optimiert. Dabei kommen im Zuge verschiedener Gütekriterien robuste numerische Optimierungsverfahren in teils sehr diffusen Suchräumen mit bis zu 12 Dimensionen zum Einsatz. Neben stabilen und langsamen Gehmustern werden auch extrem schnelle und agile Laufmuster gefunden, die sich dennoch als sehr effizient in Bezug auf die verbrauchte Energie herausstellen. Die Kontrollräume/Stabilitätsbereiche der optimierten Konfigurationen der elastischen Beineigenschaften werden untersucht und mittels Störungsszenarien getestet. Hierbei kann demonstriert werden, wie die Flexibilität des Designs optimal genutzt wird und der Roboter selbständig immer wieder zu angepasst stabilen Laufmustern zurückfindet.

Eingeplante Elastizität steigert die Agilität und Robustheit des Systems.

Inhaltsverzeichnis

Inhaltsverzeichnis	i
1 „Jena Walker 2“ Roboter	1
1.1 Einführung	1
1.2 Den Menschen als Vorbild	4
1.3 Mechanische Struktur und Eigenschaften	5
1.3.1 Bionisch inspiriert	6
1.3.2 Flexibel	6
1.3.3 Unteraktuiert	6
2 Simulink SimMechanics Modell	7
2.1 Einführung	7
2.2 Modellaufbau	8
2.2.1 Parameterblock	8
2.2.2 Hüftoszillator	9
2.2.3 Mechanik	10
2.2.4 Optimierungsschnittstelle	11
2.2.5 Visualisierungsschnittstelle	11
3 Optimierung	13
3.1 Einführung	13
3.1.1 Unbeschränktes, implizites Filtern	14
3.1.2 Nomad	14
3.2 Erstes Herantasten	15
3.2.1 Diskrete Optimierung mit bloßem Auge	17
3.2.2 Startpunkte finden	21
3.3 Optimierung über 9 Parameter	25
3.3.1 Formulierung von Optimierungszielen	26
3.3.2 Unbeschränkte Optimierung mit Imfil	28
3.3.3 Beschränkte Optimierung mit Nomad	32

3.4	Optimierung über 12 Parameter	39
3.4.1	Beschränkte Optimierung mit Nomad	39
3.5	Fazit	44
4	Analyse des Laufverhaltens	49
4.1	Einführung	49
4.2	Das Laufverhalten des JW2 Roboters	49
4.3	Das Laufverhalten des JW2 Simulink-Modells	50
4.4	Vergleichbare Laufmuster	51
4.4.1	Langsames Gehen	52
4.4.2	Schnelles Gehen	53
4.4.3	Elastisches, schnelles Laufen	53
4.5	Tendenzanalyse	54
5	Visualisierung des Kontrollraums	59
5.1	Einführung	59
5.2	Kontrollraum der Konfiguration Startpunkt1	59
5.3	Kontrollraum der Konfiguration Optipunkt9	61
5.4	Ausblick	63
6	Analyse zur Störungsanfälligkeit	65
6.1	Einführung	65
6.2	Szenarien	66
6.2.1	Treppensteigen	67
6.2.2	Hindernisblock	67
6.2.3	Sinuslaufen	68
6.3	Fazit der Arbeit und Ausblick	70
	Literaturverzeichnis	73
	Abbildungsverzeichnis	75

Kapitel 1

„Jena Walker 2“ Roboter

1.1 Einführung

Die Komplexität des menschlichen Laufapparats macht dessen Ergründung zu einer wahren Herausforderung. Muskulatur und Skelett bestehen aus mehreren Segmenten, die über Gelenke mit vielen verschiedenen Freiheitsgraden durch zahlreiche elastische Sehnen- und Muskelgruppen verbunden sind. Weiter gibt es sog. „Schwabbelmassen“, gedämpft-elastisch an das Skelett angekoppelte Gewebeteile, die besonders bei schnellen Bewegungen einen starken Einfluss auf die Reaktionskräfte und somit auf die Dynamik haben. Selbst die Knochen bringen ihre gewisse Elastizität mit sich.

Diese vielen Faktoren, die sich untereinander auf komplexe Weise beeinflussen, machen realitätsnahe-Modellierung, Kontrolle und Simulation eines solchen Systems zu einer fast unlösbaren Aufgabe. Bei Sensor-Feedback Methoden verschlechtert sich die Situation bei hohen Geschwindigkeiten, wie beim Rennen oder Sprinten, noch weiter. Die Störanfälligkeit der Sensoren / Sensorrauschen beeinträchtigt hier stärker die Feedback-Mechanismen und somit die Kontrolle des Systems.

Im letzten Jahrzehnt wurden einige Ansätze verwirklicht, die ganz ohne Sensor Feedback (Abb. 1.1)([12]) oder mit wenig auskommen ([6]) und so dieses Problem umgehen. Ein gemeinsames Manko dieser passiv dynamischen Laufmaschinen ist aber, dass während Stand- oder Belastungsphasen eines Beines das Knie komplett durchgestreckt sein muss, um das Gewicht tragen zu können. So werden mögliche Gangarten auf eine bestimmte Geschwindigkeit und Schrittfrequenz beschränkt. In der Realität können wir Menschen aber eine Vielzahl an Gehgeschwindigkeiten ($0 \frac{m}{s} - 5 \frac{m}{s}$) mit unterschiedlichen Schrittfrequenzen laufen. Diese Anpassungsfähigkeit kommt noch mehr zur Geltung bei den Übergängen vom Gehen zum Rennen. Hier wird schnell klar, dass man sich nicht auf steife



Abbildung 1.1: McGeers passiv dynamische Laufmaschine bewegt sich schwingend wie ein Pendel ohne Motorkraft und zusätzliche Kontrollen eine leicht schiefe Ebene herab ([12]).

Beine stützen sollte, sondern das Laufverhalten dynamisch nachgiebig gestalten muss ([4],[5]). Diese Idee wurde erfolgreich mit den ersten Hüpfrobotern (Abb. 1.2)([13]) umgesetzt.

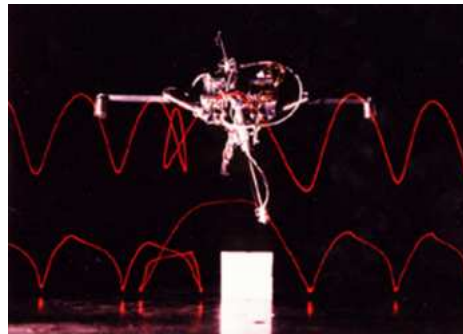


Abbildung 1.2: Raiberts selbst-balancierender einbeiniger Hüpfroboter beim Überwinden von Hindernissen ([13]).

Diese elastischen Roboter haben im Vergleich zu ihren starren Kollegen zwar wieder umgekehrt das Problem, einfach nur ruhig stehen zu bleiben oder langsam zu gehen, können sich aber stabil auf mehrere schnelle Bewegungsmuster einstellen, z.B. bei Vorgabe einer bestimmten Anfangsgeschwindigkeit oder Körperhaltung.

Die Entwicklung von Laufmaschinen und rennenden Robotern war so, in Bezug auf diese beiden unterschiedlichen Bein-Designs, bislang meist zweigeteilt: steife Beine zum Stehen und Gehen, aber nachgiebig elastische Beine zum Hüpfen und Rennen. Zunehmend zeigen Simulationsstudien, wie auch die des Laflabors in Jena ([9]), dass Gehen und Rennen sehr wohl auf den selben Beineigenschaften realisiert werden kann. Die natürliche, menschliche Fortbewegungsart entspricht grob gesagt dem Balancieren einer Masse auf zwei Trägern. Wenn man im Zuge von elastischem Gehen ein Bein als Feder abstrahiert, deren Ausdehnung und Stauchung einer Funktion genügt, die der Kniebeugung beim Menschen ent-

spricht, so können stabile Geh- und Rennmuster (nur durch Vorgabe bestimmter Aufttrittswinkel) erreicht werden (Abb. 1.3).

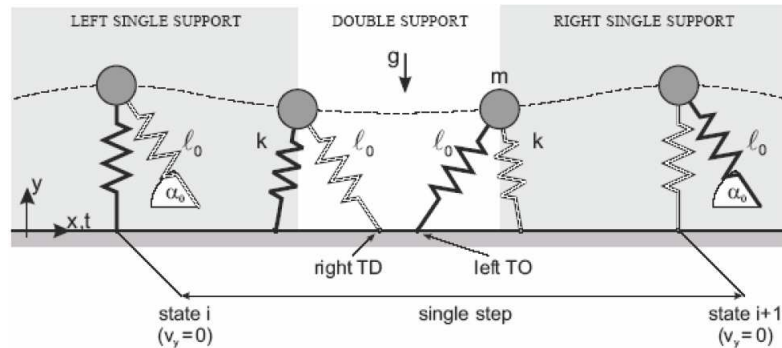


Abbildung 1.3: Das laufende Feder-Masse-Modell (FSU Jena, Lauflabor, A. Seyfarth).

Bei geringer Geschwindigkeit können Gehmuster mit Phasen, bei denen beide Beine Bodenkontakt haben, entdeckt werden, aber sobald eine gewisse Geschwindigkeit überschritten wird, stellen sich ohne Umschalten stabile Rennmuster ein mit Abstoßphasen, so dass höchstens nur ein Fuß Bodenkontakt hat.

Die Stabilität dieser Laufmuster kann mit der eines Fahrrads bei hoher Geschwindigkeit verglichen werden. Auch ohne Fahrer rollt das Rad weiter und kann sogar kleine Hindernisse überwinden. Die Stabilität beim Rennen ist ähnlich. Je schneller wir rennen, desto weniger entscheidend werden präzise Eigenschaften, wie die Beinsteifigkeit oder der Aufttrittswinkel/punkt. Kleinere Toleranzen sind im abgespielten Rennmuster nicht zu unterdrücken, sondern sollten von vorne herein in einem flexiblen, passiven Roboter-Design eingeplant sein.

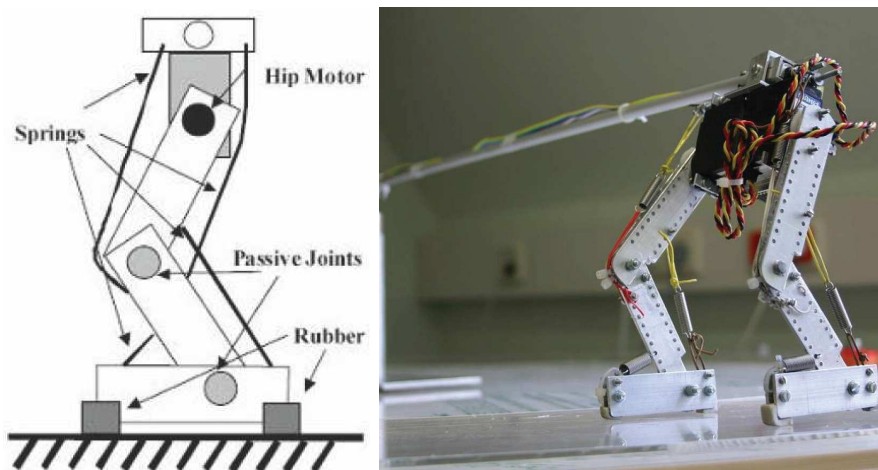


Abbildung 1.4: Der erste Jena Walker rennt mit Federn und Gummis passiv elastisch schnell an der Stange (FSU Jena, Lauflabor, A. Seyfarth).

Deshalb könnte es möglich sein, einen weitgehend passiv rennenden Roboter ohne oder mit wenig Sensor-Feedback zu bauen, der auch bei geringen Geschwindigkeiten stabil gehen kann. Beim ersten Jena Walker (Abb. 1.4)([11]) war es gelungen, auf einem unterteilten Beinaufbau mit elastisch bespannten Hüft-, Knie- und Fußgelenken passive Laufmuster zu demonstrieren. Es lohnt sich hier, das Potential elastischer Mechanismen des menschlichen Gehens und Rennens weiter zu untersuchen.

So wurden zwei Ansätze verfolgt ([14]): Erstens wurde eine komplexere zweibeinige Laufapparatur mit passiv elastischen, dreiteiligen Beinen und zwei DC Motoren an Hüftgelenken gebaut. Zweitens wurde dazu ein Modell in Matlab Simulink/SimMechanics erstellt, mit dem man geeignete Bein-Designs und Motoreinstellungen findet, um eine stabile Laufbewegung zu gewährleisten. Das Ziel dieser Arbeit ist hierbei in erster Linie die Optimierung von Laufmustern mittels numerischer Verfahren an diesem Software-Modell.



Abbildung 1.5: Einige der derzeit fortgeschrittensten humanoiden Roboter der Welt; von links nach rechts: Asimo (Honda), Johnnie (TUM), Qrio (Sony), HRP-2 (Kawada Industries). ([8])

1.2 Den Menschen als Vorbild

Die Beine fast aller heutigen humanoiden Robotern, die in der Lage sind, eine Reihe von unterschiedlichen Laufbewegungen durchzuführen (wie z.B. Asimo, Johnnie, Qrio, HRP-2 (Abb. 1.5)) bestehen aus starren kinematischen Ketten mit Drehgelenken und bedienen sich leistungsstarker, elektrischer Motoren mit festen Rotationsantriebsgängen. Obwohl bereits kurze Flugphasen erreicht wurden, ist man vom richtigen Joggen oder Rennen noch weit entfernt.

Elastizität ist in konventionell gebauten Robotern ein ungewollter Nebeneffekt, den es gilt, möglichst stark zu unterdrücken, da er große Schwierigkeiten

bei Berechnung von Position und Trajektorien verursacht. Der menschliche Bewegungsapparat ist im Gegensatz dazu aber nicht mit starren Rotationsgelenken und einfachen Stellmotoren ausgestattet, sondern besteht aus hochredundanten und nachgiebigen Aktuatoren und besitzt einen hohen Grad an Flexibilität und Stabilität, die hauptsächlich durch die lokalen Eigenschaften des Muskel-Skelett-Systems und den Reflexen garantiert wird.

Neben der Optimierung der Laufmuster ist das höhere Ziel dieser Arbeit herauszufinden, wie die Verwendung von Elastizität und Dämpfung im Design dazu beitragen kann, die Bewegungsfähigkeit humanoider Roboter zu verbessern. Das Hauptproblem hierbei ist, das, was für den Menschen die Reflexe und Eigenschaften der Muskulatur erledigen, so in mechanische Strukturen eines Roboters abzubilden, dass komplexe Voll-Feedback-Kontrollalgorithmen überflüssig werden. So könnte man Freiheitsgrade, die sonst von Kontrollalgorithmen zur Stabilisierung verwendet werden, redundant dazu benutzen, das Laufmuster leicht zu modifizieren (indem man z.B. Einstellungen an der Kniebeugung vornimmt), ohne dabei an Stabilität zu verlieren oder das Muster generell zu wechseln.

1.3 Mechanische Struktur und Eigenschaften

Das neu entwickelte, modulare Robotersystem BioLEG-2 (TETRA, Ilmenau) besteht aus starren Segmenten, Gelenkmodulen und einstellbaren, elastischen Federn, die Hüfte, Knie und Fußgelenk bespannen.

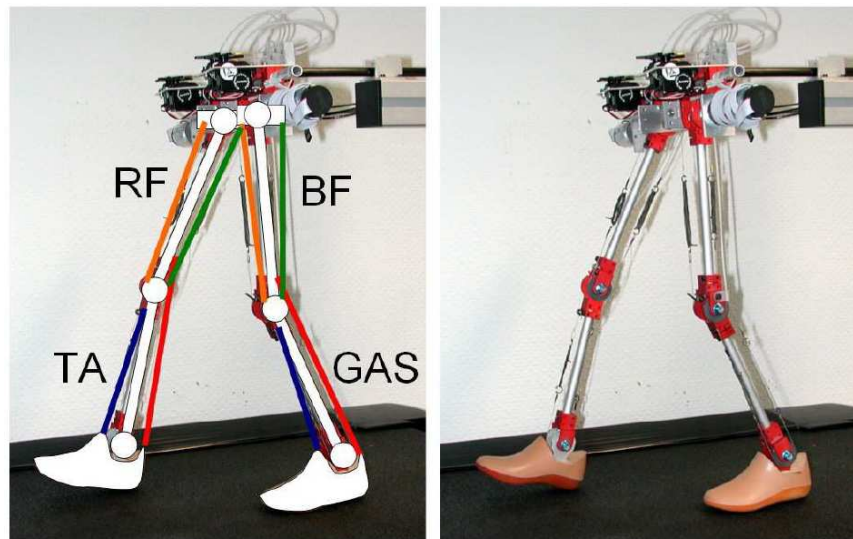


Abbildung 1.6: Jena Walker 2 Aufbau ([14])

1.3.1 Bionisch inspiriert

Der Roboter ist eine herunterskalierte Nachbildung (Roboter-Gesamtgewicht $2kg$) des menschlichen Laufapparats. Jedes Bein (Hüfthöhe $45cm$) enthält mit Oberschenkel, Schienbein und Fuß drei Teile.

1.3.2 Flexibel

Ähnlich wie beim Vorgänger (Abb. 1.4) repräsentieren vier Beinhauptmuskelgruppen die elastischen Strukturen des Jena Walker 2 Roboters (Abb. 1.6): Tibialis Anterior (TA), Gastrocnemius (GAS), Rectus Femoris (RF) und Biceps Femoris (BF).

1.3.3 Unteraktuiert

Bis auf die TA-Gruppe laufen alle Muskeln über mehrere Gelenke. Zusätzlich werden die Sprunggelenke von Schnüren umlaufen, die Reibung zwischen den sich bewegendenden Teilen und somit Gelenkdämpfung erzeugen. Das ist nötig, um Schwingungen des Fußes während der Schwungphasen des gesamten Beins zu vermeiden.

Servo-Motoren oberhalb der Hüfte werden benutzt, um die Ruhelänge der Federn via Bowdenzüge einzustellen. Das verhindert Verkürzung und Ausdehnen von GAS, RF und BF, die zu Haltungsproblemen in Knie- und Fußgelenk führen können. In der Hüfte sitzt an den DC-Motoren ein Sinusgenerator (Frequenz f , Amplitude A , Offset-Winkel O), der die alternierende Muskelaktivität an den Hüftgelenken imitiert. Die Flexibilität der Kopplung zwischen DC-Motor und Hüftgelenk ist konfigurierbar und erlaubt einen Drehgelenkspielraum von $10^\circ - 15^\circ$. Das reduziert die Stöße auf den Oberkörper und zeigt sich in zum Menschen vergleichbaren Hüftwinkelverläufen beim Gehen und Rennen. Dabei ist es wichtig zu erkennen, dass Hüftgelenkflexibilität nur dann Sinn macht, wenn das Bein selbst fähig ist, ein Laufmuster zu stabilisieren.

Die Trajektorie des Oberschenkels in Bezug zum Oberkörper kann ohne Probleme aus dem Sinusmuster der DC-Motoren abgeleitet werden. So legt der Hüftmotor nur die Frequenz f und die ungefähre Magnitude A der Hüftoszillation fest. Das Produkt der beiden Parameter $A \cdot f$ beschreibt ungefähr die erwünschte Vorwärtsgeschwindigkeit. Zur Vereinfachung ist der Bewegungsspielraum des Oberkörpers auf die Sagittalebene beschränkt. Torsodrehung und -neigung ist im momentanen Stadium noch nicht erlaubt. Der Roboter ist derzeit auf einer motorisierten Tretmühle bzw. Laufband installiert, um die Analyse der steady-state-Bewegungen zu erleichtern.

Kapitel 2

Simulink SimMechanics Modell

2.1 Einführung

In diesem Kapitel wird das auf Simulink basierende Software-Modell des Jena Walker 2 und dessen Umgebung beschrieben.

Das umfangreiche Softwarepaket Matlab/Simulink (The MathWorks, Inc.) bietet eine schnelle und leistungsstarke Umgebung zur numerischen Berechnung, Modellierung und Visualisierung von Daten aus dem Ingenieurbereich.

Matlab ist hauptsächlich für die effiziente Berechnungen mit Matrizen ausgelegt, woher sich auch der Name ableitet: MATrix LABoratory. Programmiert wird mit einer plattformunabhängigen Skriptsprache, die auf dem jeweiligen Computer interpretiert wird. Programme können als sogenannte Scripts oder Funktionen zu atomaren Einheiten verpackt werden, was das Erstellen von anwendungsorientierten Werkzeugkisten (Toolboxes) erlaubt. Viele solcher Pakete wie z.B. die Optimization Toolbox sind auch kommerziell erhältlich. Die meisten kostenlosen Optimierungsverfahren, wie auch das hier verwendete Implizite Filtern und das Nomad-Verfahren, stehen als Matlabcode im Internet zum Download zur Verfügung. Durch die vereinfachte, mathematisch orientierte Syntax der Matlab-Skriptsprache und die umfangreichen Funktionsbibliotheken für Statistik, Signalverarbeitung usw. ist die Erstellung entsprechender Programme wesentlich einfacher möglich als z.B. unter der effizienten Programmiersprache C. (Ferner gibt es sogar Schnittstellen, um C-Code einzubinden, sowie einen Übersetzer, mit dem aus einem Matlab-Script unabhängig lauffähigen C-Code erstellt werden kann. So wird in der Matlab-Umgebung auch das Entwickeln und Testen von mathematischen Modulen aufwendiger C-Projekte ermöglicht.) ([15])

Das auf Matlab basierende Simulink, eine zeitgesteuerte, visuelle Drag n' Drop Modellierungsumgebung mit vorgefertigten Modulblöcken, arbeitet mit seinen Integratoren nicht direkt auf Matlab, sondern ist in seinen Simulationsbereichen

sogar noch etwas schneller. Weiter bietet Simulink die Möglichkeit, Pakete, wie das hier verwendete SimMechanics, einzubinden um die Bausteinbibliothek zu erweitern. SimMechanics enthält, wie der Name schon sagt, vorgefertigte, mechanische Module, wie z.B. Festkörper, Gelenke, Aktuatoren und Sensoren bis hin zu einfachen Reibungsmodellen.

Für den Jena Walker 2 ([14]) wurde ein komplexes Simulink-Modell (A. Seyfarth) mit einem 2D Bodenkontaktmodell (H. Geyer) angefertigt, das sich auch Modulen aus SimMechanics bedient. Dieses wurde im Verlauf von Simulations- und Optimierungsstudien modifiziert und erweitert.

2.2 Modellaufbau

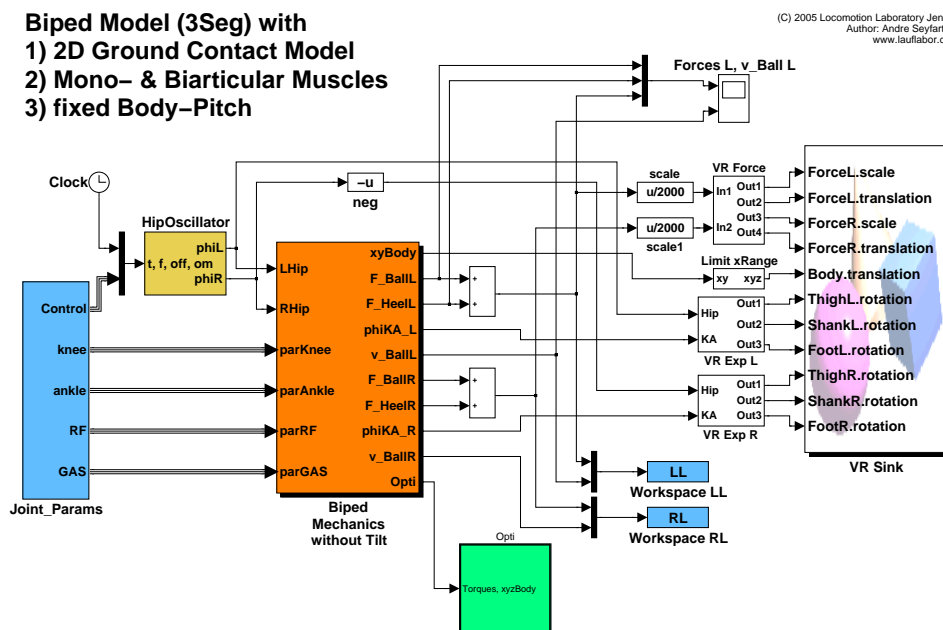


Abbildung 2.1: Simulink-Simmechanics-Modell des Jena Walker 2

Das Modell (Abb. 2.1) besteht aus vier Hauptkomponenten: dem Eingabeparameterblock, dem Hüftoszillator, der eigentlichen Mechanik und einer Optimierungsschnittstelle, sowie einer Visualisierungsschnittstelle. Im Folgenden werden die Aufgaben dieser in Subsysteme verpackten Komponenten kurz erklärt.

2.2.1 Parameterblock

Dieser Block (Abb. 2.1) bildet die Eingabemaske des Modells und besteht aus insgesamt 13 Parametern, die man in zwei Sparten unterteilen kann. Die ersten 3 Parameter (Frequenz f , Offset-Winkel α und Drehzahl ω) steuern den Hüft-

szillator, d.h. den eigentlichen Antrieb, und sind somit die ausschlaggebendsten Parameter, die das Basislaufmuster des Roboters festlegen. Die nächsten 10 Parameter charakterisieren die Mechanik des Roboters. Jeweils 3 Gelenkparameter (Drehsteifigkeit c_1 , c_2 , Ruhewinkel p_1 , p_2 und Drehdämpfung d_1 , d_2) für Knie- und Fußgelenk, sowie jeweils 2 Muskelparameter (Drehsteifigkeit c_{RF} , c_{GAS} , Ruhewinkel p_{RF} , p_{GAS}) für die beiden Hauptmuskeln (Federn im Modell). Diese Eingaben bestimmen also maßgeblich die Haltung und Elastizität der Beine.

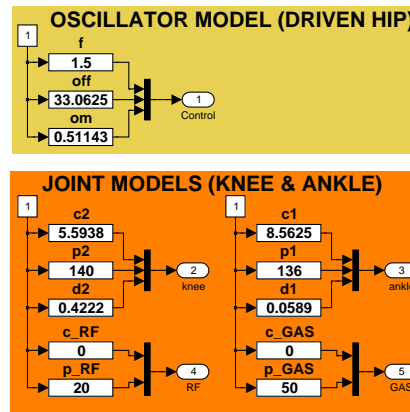


Abbildung 2.2: Der Parameterblock des Simulink-Modells

Parameter-Konfiguration

f	1	[Hz]	Frequenz (Hüfte)
α	2	[°]	Offset-Winkel (Hüfte)
ω	3	[rps]	Drehzahl (Hüfte)
c_1	4	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	5	[°]	Ruhewinkel (Fuß)
d_1	6	[Nms/°]	Drehdämpfung (Fuß)
c_2	7	[Nm/°]	Drehsteifigkeit (Knie)
p_2	8	[°]	Ruhewinkel (Knie)
d_2	9	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	10	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	11	[°]	Ruhewinkel (RF)
c_{GAS}	12	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	13	[°]	Ruhewinkel (GAS)

2.2.2 Hüftoszillator

Dieser Block ist ein typischer CPG (Central Pattern Generator), der aus den Eingabeparametern Frequenz f , Offset-Winkel α , Drehzahl ω , sowie der Simulationszeit t eine sinusförmige Ansteuerung für die Hüftgelenke erzeugt. Die Aus-

gabe des Oszillators ist hier im wesentlichen keine direkt erzeugte Kraft, sondern ein Vorgabewinkel an die Drehgelenke. Dieser Winkel wird von den Aktuatoren versucht zu realisieren, koste es was es wolle. Das kann zu unrealistisch hohen Drehmomenten führen, die man bei der Optimierung nicht außer acht lassen, sondern mit einer harten Schranke oder einem geeigneten Strafterm eindämmen sollte. Die durch den Eingabeparameter f vorgegebene Frequenz ist hier also genau auch die Frequenz der resultierenden Laufbewegung. Den Offset-Winkel kann man sich als die Neigung des Roboters nach vorne oder hinten vorstellen, während die Drehzahl die Leistungsfähigkeit des Motors (bzgl. der Rotationsgeschwindigkeit) beschreibt.

2.2.3 Mechanik

Das ist der eigentliche Hauptteil des Modells. Hier befindet sich die gesamte Mechanik, d.h. von der Hüfte über die kinematischen Ketten der Beine bis zum Bodenkontaktmodell. Der Jena Walker 2 befindet sich noch im Entwicklungsstadium und wird, damit er sich nicht dreht oder seitlich umkippt, an der Hüfte fixiert. Auch im dreidimensionalen XYZ-Koordinatensystem des Simulink-Modells ist es dem Körper der Hüfte nur gestattet, sich in XY-Richtung (Sagittalebene) zu bewegen, d.h. nur vor-zurück und auf-ab. Die Position des Modell-Roboters in Richtung der Z-Achse ist fest. Die Beine (Abb. 2.3) sind über Drehgelenke um die Z-Achse mit der Hüfte (50kg) verbunden. Wie auch beim Menschen bestehen sie aus Oberschenkel (je 8kg), Unterschenkel (je 4kg) und dem Fuß (je 1kg), ebenfalls jeweils durch Drehgelenke um die Z-Achse verbunden. Der Gesamtschwerpunkt des Systems liegt somit stets unmittelbar unter der Hüfte. Jeder Fuß besitzt 2 Kontaktpunkte, die Ferse und den Ballen. Aus Effizienzgründen sollte man die Gesamtanzahl der Kontaktpunkte auf ein wesentlich benötigtes Minimum reduzieren, deshalb wird hier auf zusätzliche Kontaktpunkte oder gar Flächen, und Kollisionsabfragen an anderen Körpern verzichtet. Die Berechnung der Haft- und Gleitreibung und deren Auswirkung bildet den Hauptrechenaufwand eines solchen Modells.

Die vier Hauptmuskelgruppen (Abb. 1.6) (RF, BF, GAS und TA) sind wie folgt modelliert: RF und BF sind nicht zuletzt wegen ihrer gegenspielerischen Wirkung zunächst zu einer effizienteren, zweigelenkigen Feder mit entsprechender Gleichung zusammengefasst. GAS wird ebenfalls durch eine Federgleichung im grünen Block (Abb. 2.3), der sich in der Feedback-Schleife zwischen Sensoren und Aktuatoren befindet, verkörpert. In diesem Block wird die Flexibilität der Gelenke sowie die Elastizität der virtuellen Muskeln berechnet bzw. passiv simuliert. Die separat explizite Wirkung von TA ist hierbei noch nicht modelliert (Abb. 2.4).

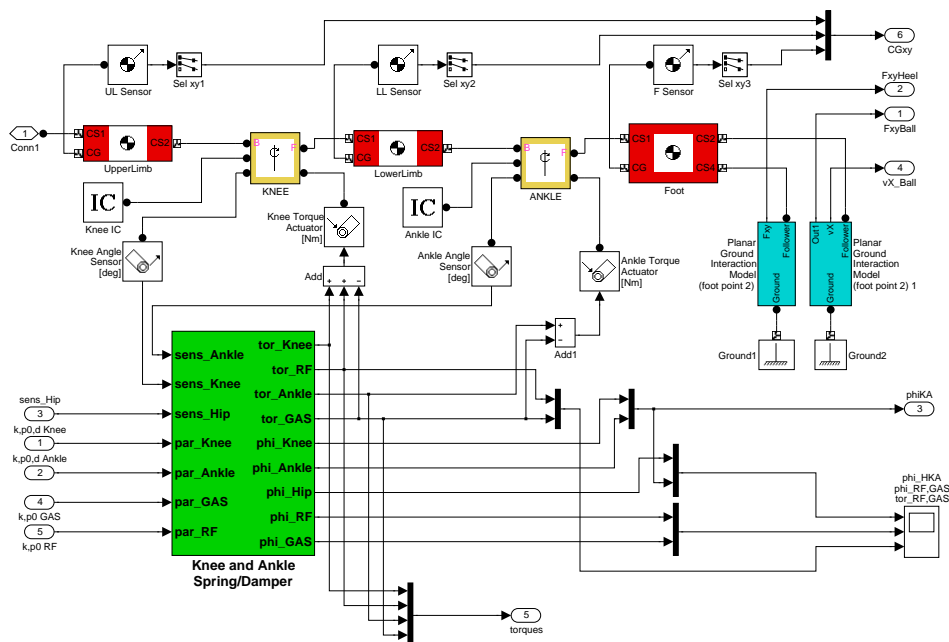


Abbildung 2.3: Die kinematische Kette des linken Beins.

2.2.4 Optimierungsschnittstelle

Hier werden für das Laufmuster charakteristische Werte, wie z.B. Positionen der Festkörper, Bodenkontakte, Geschwindigkeit, Drehmomente und andere Kriterien gemessen und für das jeweilige Optimierungsproblem aufbereitet. Von der Matlab-Konsole aus gesehen, bietet diese Read-Only-Schnittstelle zusammen mit der Write-Only-Schnittstelle des Parameterblocks den Ein- und Ausgang einer BlackBox und ermöglicht somit das einfache Ankoppeln eines robusten, auf Matlab basierenden Optimierers. Abhängig von der Gütefunktion φ mit ihren möglichen Straftermen und Nebenbedingung werden unterschiedliche Signale effizient verarbeitet und weitergeleitet.

2.2.5 Visualisierungsschnittstelle

Simulink SimMechanics bietet eine standardisiert verfügbare Visualisierung an. Diese ist sehr präzise, da sie zu jedem Zeitschritt des Simulationsintegrators, der je nach Einstellungen variabel in den Millisekunden-Bereich hineingehen kann, aktualisiert wird. Neben dem Manko an Konfigurierbarkeit, ist diese Alternative für komplexere Modelle und längere Simulationszeitspannen zu rechenintensiv.

Eine andere, schnellere Visualisierungsalternative bietet die hier verwendete Virtual-Reality-Toolbox (Abb. 2.4), mit der es möglich ist, die numerischen Ausgaben des Modells zu visualisieren und somit resultierende Laufmuster objektiv zu beurteilen. Es können Videos (.avi-Dateien), sowie 3D-Vektorgrafikverläufe

(.wrl-Dateien) verschiedener Szenarien aufgenommen und gespeichert werden. Natürlich sind Visualisierungen parallel zur Simulation immer rechenaufwändig und sind letztendlich nur für das menschliche Auge bestimmt. Deshalb sollten sie bei Optimierungsläufen deaktiviert und aus Effizienzgründen generell auf ein Minimum reduziert werden.

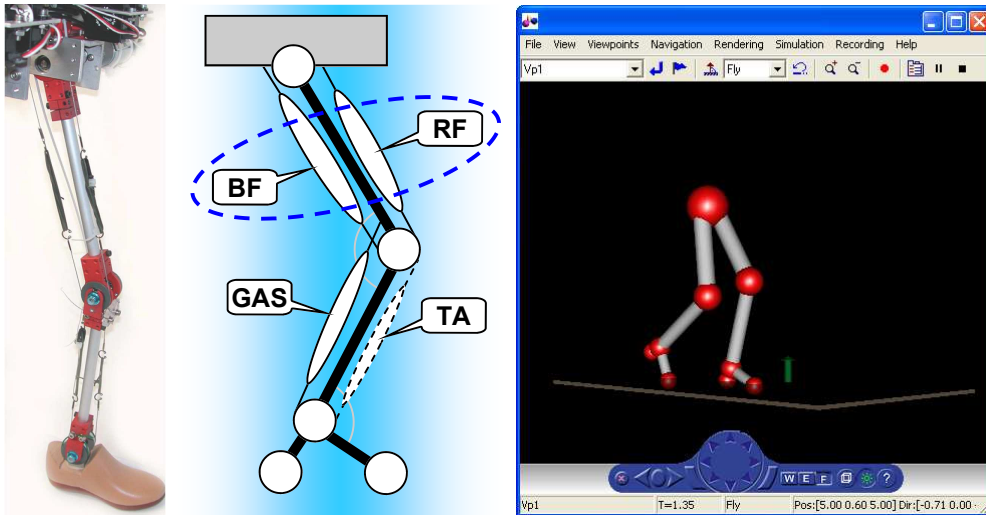


Abbildung 2.4: Aufbau des Modell-Beins und Visualisierung von Laufbewegungen

Mit der Einarbeitung unterschiedlicher Bodenstrukturen, um die Störungsanfälligkeit von Laufmustern zu testen, wurde die Visualisierung des Roboters und des Bodens wieder auf die wesentlich XY-Ebene beschränkt und über effizientere, grob getaktete Simulink S-Functions und Plots geregelt.

Kapitel 3

Optimierung

3.1 Einführung

Aufgrund des momentanen Entwicklungsstandes der Architektur des Modells mit seinen 13 Eingabeparametern und der Schwierigkeit, die Verhaltensweise des Modells als höhere Ableitungen (Änderungsprognosen) einer Gütefunktion zu formulieren, ist die BlackBox-Optimierung hier der erste Ansatz. Für die WhiteBox-Optimierung müsste man präzise hinter die Kulissen des Modells schauen und zu möglichen Parameterkonfigurationen vorhersagen können, wie sich das Ändern der Konfiguration, bzgl. jeder Dimension auf die Ausgabe des Modells auswirkt. So könnte man zu dieser Ausgabe, die den Wert der Zielfunktion bildet bzgl. der man optimiert, Gradienten oder sogar Hesse-Matrizen finden oder approximieren. Diese vorhandene Information würde dann den Einsatz effizienter, gradientenbasierter Optimierungsverfahren (Newton, SQP) ermöglichen. Liegen aber diese Änderungsinformationen, wie beim diesem Modell, nicht explizit vor bzw. ist die Zielfunktion nicht differenzierbar, so muss sich das geeignete Optimierungsverfahren selbst, z.B. mit Hilfe eines Simplex (Punktmuster) und der somit verbundenen höheren Anzahl an Zielfunktionsauswertungen, ein 'Bild' über den Verlauf des Gütekriteriums machen. Verfahren diesen Typs (Nelder-Mead, Nomad) eignen sich für die BlackBox-Optimierung und sind neben ihrem Nachteil bzgl. der Rechenintensivität meist robuster (weniger anfällig auf Störungen des Gütekriteriums) als die gradientenbasierten, effizienteren Verfahren und finden sich so auch in diffusen Suchräumen zurecht.

Folgende Verfahren wurden zur Optimierung der Laufmuster des Jena Walker 2 im Simulink SimMechanics Modell ausgewählt:

3.1.1 Unbeschränktes, implizites Filtern

Das implizite Filtern ([10]) basiert auf der Methode des steilsten Abstiegs. Hierzu wird ein Gradient $\nabla\varphi(p^{(k)})$, der senkrecht zur Höhenlinie steht, im Iterationspunkt $p^{(k)}$ approximiert. Aus der notwendigen Abstiegsbedingung $(d^{(k)}) \cdot \nabla\varphi(p^{(k)}) < 0$ kann dann die Suchrichtung $d^{(k)} = -\nabla\varphi(p^{(k)})$ hergeleitet werden. Die Schrittweite kann z.B. durch eine ungenaue Liniensuche bestimmt werden. So kommt man relativ schnell in die Nähe eines lokalen Minimums, läuft aber Gefahr, sich dort schon fest zu verfangen und ein mögliches, besseres Optimum zu verpassen (Abb. 3.1). Weiter konvergiert ein solches Verfahren in der Nähe des lokalen Minimum auf flachem Terrain meist recht langsam. Man erreicht also ziemlich schnell eine Verbesserung, sollte sich mit dieser aber nicht gleich zufrieden geben. Möglichkeiten wären z.B.: mehrere Läufe von unterschiedlichen Startlösungen aus, sowie ein Umschalten/Koppeln zwischen unterschiedlichen Optimierungsverfahren, um insgesamt eine bessere Konvergenz zu erreichen.

Das implizite Filtern ist allerdings schon auf zerklüftete Landschaften bzw. gestörte Gütekriterien ausgelegt. Seit Neuem besitzt es auch die Fähigkeit zur Parallelisierbarkeit und zum Umgang mit Nebenbedingungen. Bei dieser Arbeit kommt aber nur die unbeschränkte Variante des Verfahrens zum Einsatz.

3.1.2 Nomad

NOMAD (Nonlinear Optimization for Mixed vAriables and Derivatives) ([1]) ist ein robustes, gradientenfreies Simplex-Verfahren, das sich einer Reihe von verschiedenen Mustersuchverfahren bedient, um Optimierungsprobleme mit nichtlinearen Nebenbedingungen lösen zu können. Wie andere Mustersuchverfahren (Nelder-Mead, MDS) besitzt auch dieses einen 'globaleren Konvergenzcharakter'. Ein Punktmuster (Simplex) bewegt sich meist durch eine Reihe geometrischer Operationen, wie z.B. Falten und Strecken, ins Tal hinab, überwindet auf seinem Weg kleinere, schlechte, lokale Minima und verdichtet sich schließlich bei einem guten Optimum. Solche Verfahren sind folglich robust gegenüber Unstetigkeiten im Gütekriterium und ihre Anwendung kann leicht, ohne genaue Kenntnis des Modells, erfolgen, da kein Gradient verlangt wird. Dennoch ist die hohe Anzahl an Zielfunktionsauswertungen, um den Simplex zu bilden und zu transformieren, ein Nachteil, der besonders bei hoher Auswertungsdauer der Zielfunktion sehr schwerwiegend wird. Deshalb sind solche Verfahren bei direkter Optimierung an Hardware mit aufwendiger Simulation oft nicht die erste Wahl. Aufgrund eines meist geometrisch polylinearem Simplex kann es zu Problemen bei der Verarbeitung von nichtlinearen, aktiven Nebenbedingungen kommen, wenn der Simplex in eine ungünstige Form gezwungen wird. Nomad ist jedoch mit den in dieser Arbeit gestellten Nebenbedingungen relativ gut zurechtgekommen.

Als Terminierungskriterien für die verwendeten Verfahren wird hier Mindest-

fortschritt bzw. Konvergenztoleranz und eine maximale Anzahl an Funktionsauswertungen verwendet.

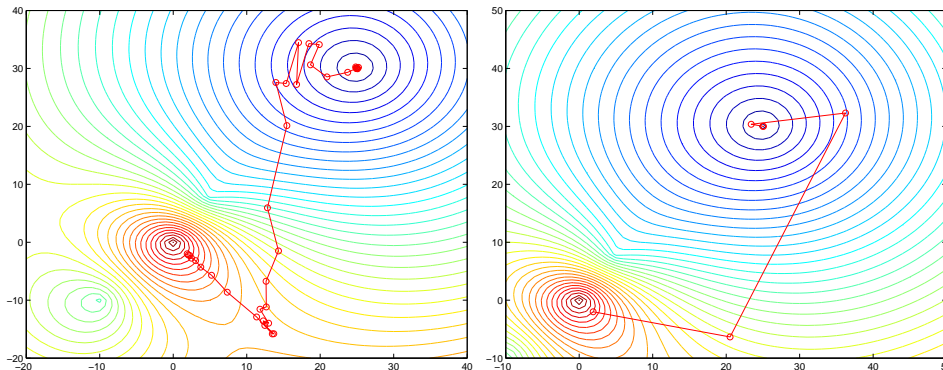


Abbildung 3.1: Hier wird eine Variante des Weber-Problems einmal vom Nelder-Mead-Simplex-Verfahren (erstes Bild) und einmal vom Newton-Verfahren mit BFGS Update (zweites Bild) gelöst. Der erste, etwas exotisch anmutende Pfad wird durch die geometrischen Faltungsoperationen des Simplex beeinflusst. Liegt, wie in diesem Falle, Gradienteninformation vor, so ist das Newton-Verfahren, nach der hier gut erkennbaren Methode des steilsten Abstiegs, wesentlich schneller in der Nähe des nächsten lokalen Minimums.

3.2 Erstes Herantasten

Hat man es mit einem Software-Modell zu tun, so ist es leicht und unbedingt nötig, den Optimierungskreislauf (Abb. 3.2) voll zu automatisieren. Hierbei müssen die Schnittstellen des Modells zu denen des Optimierers passen. Der Optimierer iteriert dann gemäß seines Verfahrens über die Parameterkonfigurationen des Modells und sollte nur dann abbrechen, wenn eines der gesetzten Terminierungskriterien erfüllt ist. Modelle sollten also insbesondere in der Lage sein, eine große Anzahl an sequentiellen, unabhängigen Simulationsläufen für zulässige Eingabeparameter fehlerfrei und ohne Ändern ihres Verhaltens auszuführen.

Der Optimierer prüft, bevor er die eigentliche Zielfunktion auswertet, erst die Nebenbedingungen. Sind diese verletzt, so macht es keinen Sinn, die Zielfunktion auszuwerten, denn man könnte einen Fehler (z.B. eine jetzt zugelassene Division durch Null) während der Simulation riskieren und so den Optimierungskreislauf zum Absturz bringen. Sind die Nebenbedingungen erfüllt, so wird die Zielfunktion mit der zugelassenen Konfiguration ausgewertet und das Ergebnis dem Wissensraum des Optimierers hinzugefügt.

Zu Beachten ist hierbei, dass in beiden Phasen, Überprüfung der Nebenbedingungen und Auswertung der Zielfunktion, es zu einer Modell-Simulation kommen kann. Man betrachte folgendes Optimierungsproblem:

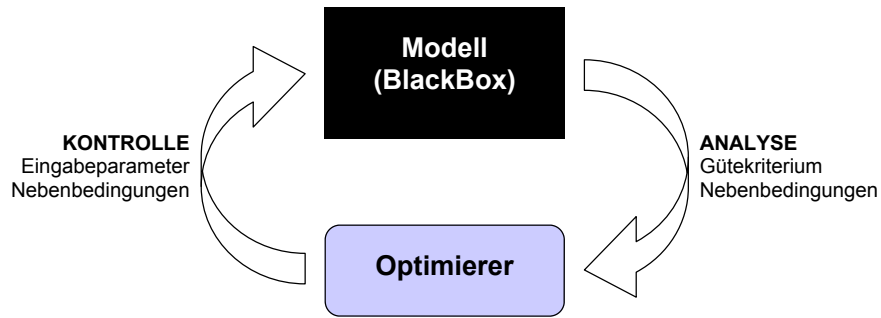


Abbildung 3.2: Der Optimierungskreislauf

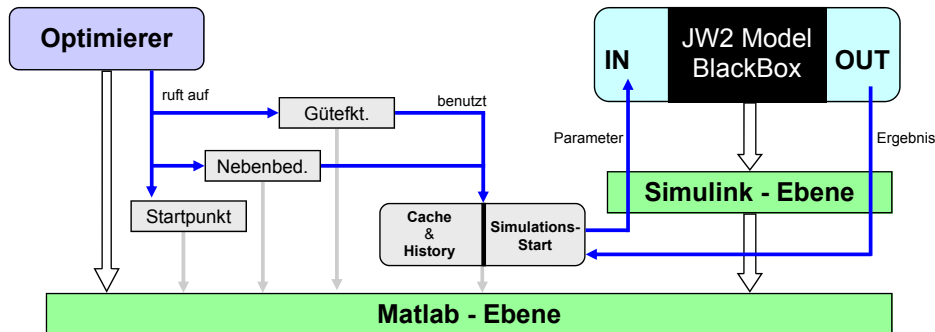


Abbildung 3.3: Architektur der Optimierung

Beispiel:

$$\begin{aligned} & \min_p -v(p) \\ \text{s.t. } & p > 0 \\ & |m(p)| < 500 \end{aligned}$$

Hier soll die resultierende Geschwindigkeit $v(p)$ unter Bedingungen an die Parameter p und die auftretenden Momente $m(p)$ maximiert werden. Für $v(p)$ und $m(p)$ ist eine Simulation nötig. Es ist daher naheliegend, dem Optimierer eine Steuerung mit Cache der folgenden Art vorzusetzen (Abb. 3.3)+(Pseudocode):

```
public boolean checkConstraints(Parameter p)
{
    if (p > 0)
    {
        prepare(p);
    }
}
```

```

        return (Cache.getTorques(p)<500);
    }
    else return false;
}

public double objectiveFunction(Parameter p)
{
    prepare(p);
    return -Cache.getVelocity(p);
}

private void prepare(Parameter p)
{
    if (!Cache.exists(p))
    {
        all_results = simulate(model,p);
        Cache.add(all_results,p);
    }
}

```

So wird die zeitaufwändige Simulation nur dann ausgeführt, wenn sie auch unbedingt nötig ist. Man betrachte hierzu auch das Sequenzdiagramm (Abb. 3.4), das die drei folgenden Szenarien darstellt.

1. Iteration: Das Optimierungsverfahren holt sich seinen Startpunkt (dieser sollte natürlich im zulässigen Bereich liegen). Überprüft die erfüllten Nebenbedingung (hier z.B. $(p_0 > 0) = 1$ und $(|m(p_0)| < 500) = 1$) und wertet dann die eigentliche Gütefunktion $\varphi(p_0)$ aus. Da für die letztere Nebenbedingung bereits eine Simulation gestartet wurde, können die Daten zu $\varphi(p_0)$ jetzt aus dem Cache genommen werden.

2. Iteration: In dieser Iteration sind die simulationsunabhängigen Nebenbedingungen bereits verletzt. (z.B. $(p_1 > 0) = 0$). Es ist folglich keine Simulation nötig.

3. Iteration: Der schlechteste Fall; während die ersten Schranken hier erfüllt sind (z.B. $(p_2 > 0) = 1$), geben die simulationsabhängigen Nebenbedingungen 0 zurück $(|m(p_2)| < 500) = 0$. D.h. es wurde eine Simulation unter Zeitaufwand durchgeführt, aber das Wissen des Optimierers nicht wesentlich erweitert.

Um die Anzahl der Fälle, die der 3. Iteration entsprechen, so klein wie möglich zu halten, sollte man simulationsabhängige Schranken nicht zu streng setzen.

3.2.1 Diskrete Optimierung mit bloßem Auge

Um erfolgreich zu optimieren, sollte man die Beschaffenheit des Bildbereichs der Gütefunktion zumindest grob kennen. Das Modellverhalten sowie das ausgewähl-

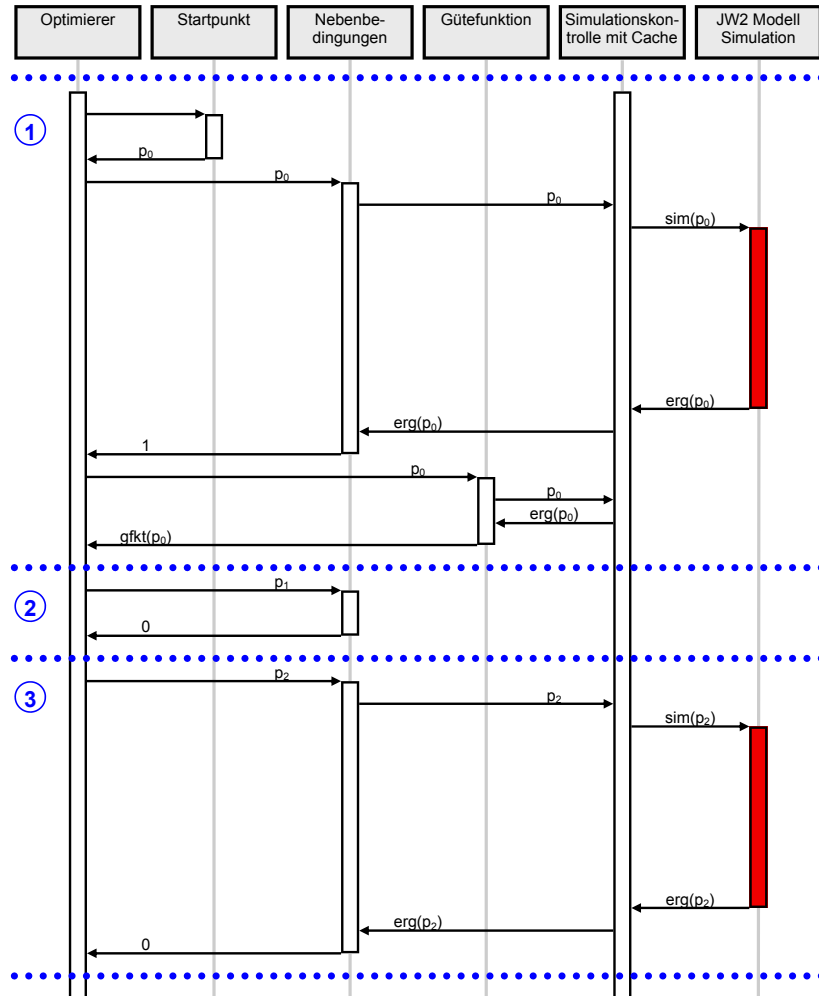


Abbildung 3.4: Sequenzdiagramm für drei mögliche Iterationen des Optimierers

te Gütekriterium selbst sind hierbei wichtig und beeinflussen maßgeblich die Wahl des richtigen Optimierungsverfahrens. Mit steigender Anzahl von Eingabeparametern erhöht sich die Dimension des Wertebereichs der Zielfunktion und somit auch die Schwierigkeit, den Bildraum grob zu ergründen oder gar zu visualisieren.

Startpunkt1			
f	1	[Hz]	Frequenz (Hüfte)
α	20	[°]	Offset-Winkel (Hüfte)
ω	0.3	[rps]	Drehzahl (Hüfte)
c_1	4	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	120	[°]	Ruhewinkel (Fuß)
d_1	0.1	[Nms/°]	Drehdämpfung (Fuß)
c_2	8	[Nm/°]	Drehsteifigkeit (Knie)
p_2	140	[°]	Ruhewinkel (Knie)
d_2	0.1	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	20	[°]	Ruhewinkel (RF)
c_{GAS}	0	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	50	[°]	Ruhewinkel (GAS)

Es bietet sich also an, zunächst klein zu beginnen und alle Eingaben gemäß einer Startkonfiguration (Startpunkt1) zu fixieren, sich z.B. nur die dominantesten Parameter herauszupicken, hier die Motorparameter Frequenz f und Drehzahl ω , und unabhängig davon ein simples Gütekriterium, wie die Geschwindigkeit, darzustellen (Abb. 3.5).

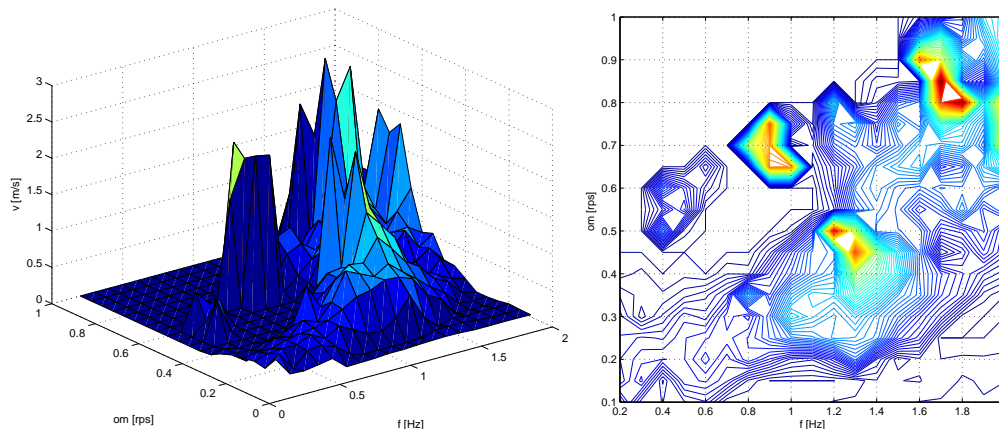


Abbildung 3.5: Geschwindigkeit der Hüfte v [m/s] in Abhängigkeit von Frequenz f [Hz] und Drehzahl ω [rps] des Hüftoszillators; Auf dem Höhenlinienbild sind drei steile, rote Maxima erkennbar. (insg. 19x19 Messpunkte; Simulationsdauer 6h auf einem Intel Pentium M 1.73 GHz, 1GB RAM)

Neben der Veranschaulichung der Landschaft, auf der das Optimierungsverfahren arbeitet, dient eine solche Auswertungsreihe dazu, den Simulations- bzw. Optimierungsaufwand abzuschätzen und um zu testen, ob ein automatisierter Optimierungskreislauf in der Lage ist, fehlerfrei zu laufen.

Die Simulationszeitspanne beträgt hier, wie auch in den folgenden Versuchen, jeweils immer 10 Sekunden [0s,10s] bei einer stark schwankenden, tatsächlich

gebrauchten CPU-Zeit von im Schnitt 30 Sekunden. Messungen für den Optimierer, die das Laufmuster charakterisieren sollen, wie z.B. Drehmomente und verbrauchte Energie pro Meter, stammen meist aus dem Intervall [8s,10s], in dem der Roboter seine Laufbewegungen/Gangart bereits gefunden und hinreichend stabilisiert hat.

Zwei der ersten Probleme, die bei der Auswertung des Gitters (Abb. 3.5) aufgetaucht sind und behoben wurden, seien hier kurz aufgeführt:

Hohe CPU-Rechenzeiten durch Solver-Schwierigkeiten: Die variable Schrittweite der Integrierten während der Simulation in Simulink ist in komplexeren Modellen empfehlenswert, da der Solver selber entscheiden kann, bei kritischen Situationen während der Simulation präziser zu werden oder bei "Langeweile" eine größere Schrittweite vorzunehmen, um so CPU-Zeit zu sparen. Das bringt aber auch Risiken mit sich. Angenommen, es herrscht während einer Simulationszeitspanne über 10 Sekunden durchgehend ein kritischer Zustand, z.B. wenn der Roboter zusammengebrochen ist, die Motoren weiterlaufen und ständig alle Arten rechenintensiver Reibungskräfte erzeugt werden, dann werden bei einer aktivierten Mindestschrittweite von 10^{-4} Sekunden 10^5 unnötig aufwändige Schritte ausgeführt. So haben einzelne Simulationen über eine Zeitspanne von 10 Sekunden zu bestimmten Eingabekonfigurationen mehr als 10 Minuten CPU-Rechenzeit in Anspruch genommen. Am Ende kamen meist zwar realistische, aber keine brauchbar guten Werte heraus, die eine so unverhältnismäßig hohe CPU-Zeit rechtfertigen. Eine mögliche Lösung ist das frühzeitige Erkennen einer ungewollten Situation durch zusätzliche Nebenbedingungen (an z.B. Roboterhaltung oder CPU-Zeit) und deren Überprüfung dynamisch zur Simulationslaufzeit. Bei Verletzung erfolgt dann ein Herunterfahren der Motoren bzw. ein Abbrechen der Simulation durch einen kontrolliert erzeugten Fehler, der an richtiger Stelle abgefangen und behandelt (z.B. mit hohem Zielfunktionswert bestraft) wird.

Unrealistische Werte durch unerkant ungeeignete Eingaben: Ein Modell muss normalerweise nicht dem Anspruch genügen, in allen Lagen perfekt zu reagieren wie die Realität, die es abzubilden gilt. Meistens sind Modelle nur für bestimmte Szenarien und Parameterbereiche hinreichend annähernd realistisch gebaut. Das macht sie weniger komplex und effizienter. Werden solche vorgegebenen Eingabebereiche verletzt oder sind nicht eindeutig bekannt, so kann das erhebliche Auswirkungen auf die Ausgabe bzw. auf den Wert eines Gütekriteriums haben, z.B. $v_p(t_f) = \pm 10000 \frac{m}{s}$ als resultierende Endgeschwindigkeit. Das Optimierungsverfahren stürzt sich natürlich gleich auf so ein extremes, aber falsches Ergebnis. Eine unmittelbare Lösung dieses Problem wäre natürlich ein vorgeschalteter Filter, der unrealistisch gute Zielfunktionswerte in sehr schlechte umwandelt und diese an den Optimierer weitergibt. Der Optimierer wiederum wird gemäß seines

Verfahrens darauf reagieren und diesen Parameterbereich in Zukunft meiden. Dieser Lösungsansatz bekämpft aber offensichtlich nicht die Ursache und verhindert somit nicht, dass das Modell mit einer unerkannt ungeeigneten Konfiguration ausgewertet wird. Bei Software-Modellen machen Konstrukte, wie das `try/catch`-Statement einem das Leben leichter, aber bei Optimierung an Hardware, können unerkannt ungeeignete Eingaben zu einer Beschädigung des Roboters führen. Die bessere Alternative ist offenbar, den Fehler frühzeitig zu erkennen und zu eliminieren. Das erfordert meistens aber eine präzise Modellkenntnis und eine entsprechend starke Verschärfung der Parameterschranken/Nebenbedingungen. Aber wenn man den zulässigen Bereich zu eng wählt, verpasst das Optimierungsverfahren wiederum vielleicht ein gutes, realistisches Optimum.

3.2.2 Startpunkte finden

Im Höhenlinienbild (Abb. 3.5) sind drei rote, ziemlich steile, charakteristische Maxima zu erkennen, die potentielle Startlösungen für Optimierungsläufe sein könnten. Diese gilt es im Folgenden weiter zu untersuchen.

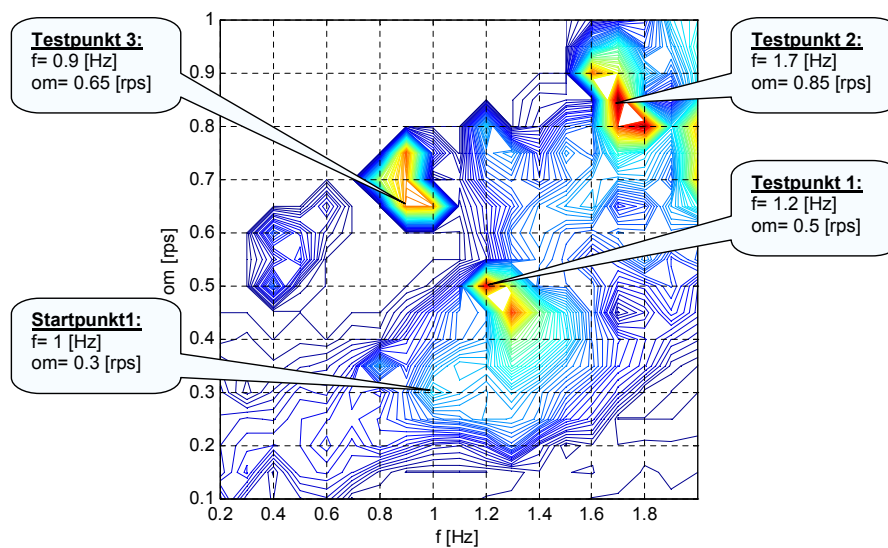


Abbildung 3.6: Koordinaten möglicher Startlösungen - Testpunkte

Diese drei Konfigurationen unterscheiden sich nur in den Hüftoszillatorparametern f und ω . Parameter wie α , p , d und c , die Haltung des Roboters und elastischen Eigenschaften der Beine beeinflussen, waren fixiert und sind folglich die selben. Stellt man jeweils eine Konfiguration ein und betrachtet den Verlauf des Gütekriteriums (Abb. 3.7), hier die Geschwindigkeit v , über 100 Sekunden, so fällt auf, dass in allen drei Testpunkten bereits eine erstaunliche Verbesserung um mehr als +300% erreicht wurde.

Testpunkt1			Testpunkt2			Testpunkt3		
f	1.2	[Hz]	f	1.7	[Hz]	f	0.9	[Hz]
α	20	[°]	α	20	[°]	α	20	[°]
ω	0.5	[rps]	ω	0.85	[rps]	ω	0.65	[rps]
c_1	4	[Nm/°]	c_1	4	[Nm/°]	c_1	4	[Nm/°]
p_1	120	[°]	p_1	120	[°]	p_1	120	[°]
d_1	0.1	[Nms/°]	d_1	0.1	[Nms/°]	d_1	0.1	[Nms/°]
c_2	8	[Nm/°]	c_2	8	[Nm/°]	c_2	8	[Nm/°]
p_2	140	[°]	p_2	140	[°]	p_2	140	[°]
d_2	0.1	[Nms/°]	d_2	0.1	[Nms/°]	d_2	0.1	[Nms/°]
c_{RF}	0	[Nm/°]	c_{RF}	0	[Nm/°]	c_{RF}	0	[Nm/°]
p_{RF}	20	[°]	p_{RF}	20	[°]	p_{RF}	20	[°]
c_{GAS}	0	[Nm/°]	c_{GAS}	0	[Nm/°]	c_{GAS}	0	[Nm/°]
p_{GAS}	50	[°]	p_{GAS}	50	[°]	p_{GAS}	50	[°]

Tabelle 3.1: Konfigurationen der Testpunkte

Betrachtet man die Momente in Knie- und Fußgelenk (Abb. 3.8) so fällt die aufgrund der gleichen Beinlastizität vielleicht erwartete Ähnlichkeit auf. Die Drehmomente der vier Konfigurationen liegen trotz unterschiedlicher Geschwindigkeiten alle ungefähr im selben Bereich knapp unter 200Nm. Die Ausschläge der Kurven zu den drei Testpunkte mit ihren vergleichbar hohen Geschwindigkeiten sehen fast gleich aus, bis auf die unterschiedliche, direkt aus der Eingabefrequenz ableitbare Periodenlänge der Schrittbewegung (z.B. bei Testpunkt3 $P = \frac{1}{f} = \frac{1}{0.9}\text{s} \approx 1.11\text{s}$).

Wie Visualisierungen der Laufbewegungen gezeigt haben, erzeugen alle drei Testkonfigurationen eine Art Hopslerlauf, ausgezeichnet durch ein energisches Abstoßen und anschließender vom Schwungbein unterstützter langer Flugphase mit schließlich fast beidbeinigem Landen.

Testpunkt2 ist besonders signifikant. Bei etwas genauerem Hinsehen entdeckt man, dass bei dieser Konfiguration, anders als bei Testpunkt1 oder Testpunkt3, nur die blaue Kurve $\text{tor}_{\text{Knee}}\mathbf{R}$ periodisch stark ausschlägt. Hier findet der Roboter selbstständig ein anderes, unsymmetrisches aber stabiles Laufmuster, das zu der periodischen Ansteuerung vom Hüftoszillator passt. Dieser einseitige Hopslerlauf (Abb. 3.9) zeichnet sich dadurch aus, dass der Roboter nach seiner Absprunghase einen kurzen Schritte in der Luft macht, um dann nach einer fast beidbeinigen Landung wieder mit dem gleichen bei Bein abzuspringen.

Auf den ersten Blick machen die 3 Testpunkte einen sehr guten Eindruck. Aber eine diskrete Optimierung mit bloßem Auge an einem Schaubild ersetzt keines Falls ein richtiges Optimierungsverfahren und birgt einige Tücken.

Zu beachten ist natürlich, dass die 3 Maxima im visualisierten f - ω -Raum

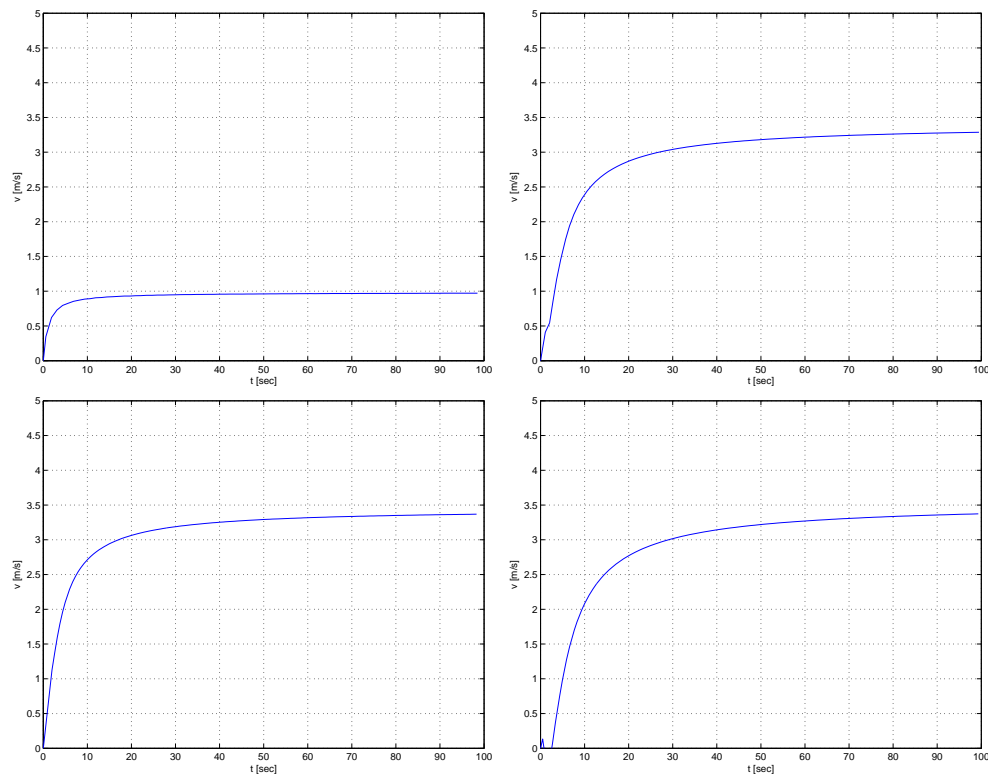


Abbildung 3.7: Hier über 100 Sekunden simuliert, der Verlauf der Geschwindigkeit v zu Startpunkt1 (links-oben), Testpunkt1 (rechts-oben), Testpunkt2 (links-unten) und Testpunkt3 (rechts-unten).

nicht notwendiger Weise auch im gesamten 13-Parameter-Raum Extrema sind. Auch eine unangebrachte Visualisierung aller paarweisen Parameterkombinationen stellt nicht zwingend die Charakteristik des Zusammenspiels im gesamten Raum dar.

Weiter besteht z.B. immer die Chance, dass zwischen zwei diskret aufeinander folgenden Gitterpunkten im Schaubild ein noch viel besseres Optimum liegt und übersehen wird. Ein Verfahren mit automatischer Schrittweitensteuerung findet in der Regel auch solche Optima problemlos.

Der eigentliche Hauptgrund, warum man sich mit diesen Testpunkten nicht zufrieden geben sollte, ist jedoch ein anderer: Drehmomentverläufe in den Knie- und Fußgelenken helfen zwar bei der Beurteilung der Ergonomie des Laufmusters, (Sind die Beine des Roboters geeignet stabil genug gebaut?), aber spiegeln im Fall des Jena Walkers 2, der seinen Antrieb in der Hüfte hat, nicht die tatsächlich gebrauchte Motorleistung wieder.

Schaut man sich nämlich die Momente an den beiden Hüftgelenken (Abb. 3.10) an, so entdeckt man schnell, wo der Haken an der Sache ist. Wie in der Einleitung zum Simulink-Modell erwähnt, versuchen die Aktuatoren an den Hüft-

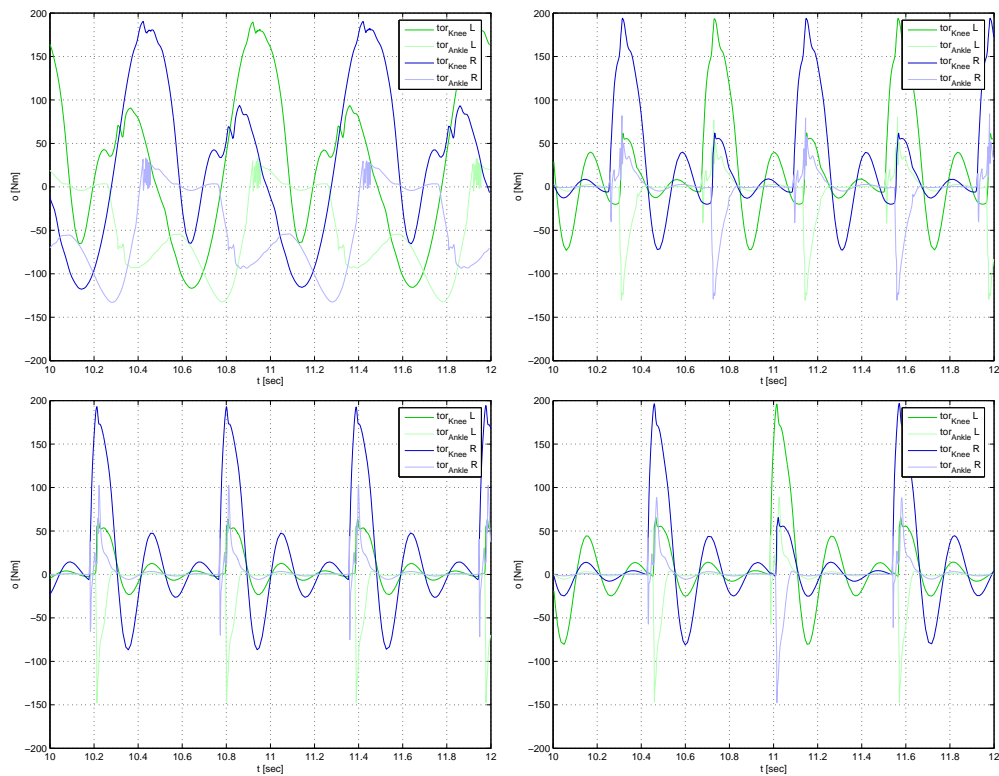


Abbildung 3.8: Hier im Intervall [10sec,12sec] simuliert, der Verlauf der Drehmomente in Knie- und Fußgelenk zu Startpunkt1 (links-oben), Testpunkt1 (rechts-oben), Testpunkt2 (links-unten) und Testpunkt3 (rechts-unten).

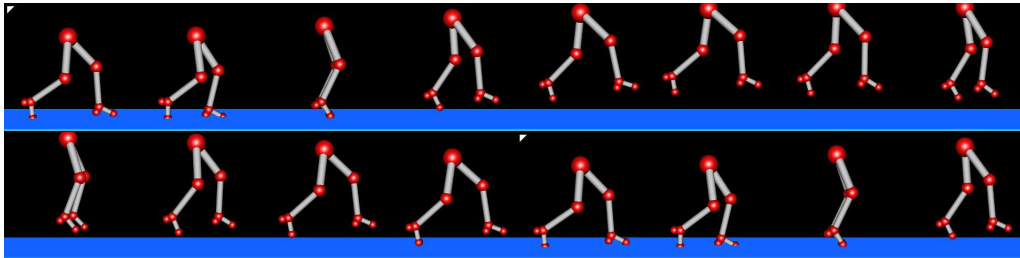


Abbildung 3.9: Hopslerlauf der Konfigurationen Testpunkt2

gelenken das vom Oszillator vorgegebene Muster um jeden Preis zu realisieren. Somit wird die extreme, anfangs so positiv wirkende Geschwindigkeitssteigerung von ca. 300% mit einer Steigerung von ca. 800% der Drehmomente in der Hüfte erkauft.

Das inspiriert dazu, die elastischen Fähigkeiten besser anzupassen um so eventuell die Stöße auf die Hüfte zu verringern.

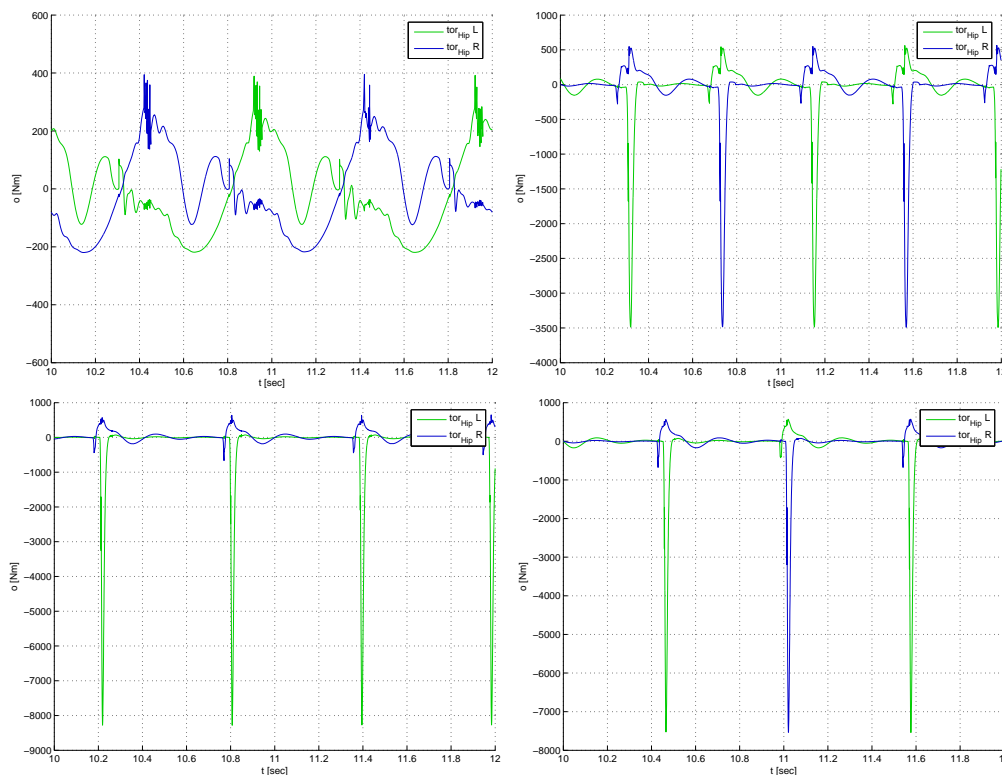


Abbildung 3.10: Hier im Intervall [10sec,12sec] simuliert, der Verlauf der Drehmomente in den beiden Hüftgelenken zu Startpunkt1 (links-oben), Testpunkt1 (rechts-oben), Testpunkt2 (links-unten) und Testpunkt3 (rechts-unten).

3.3 Optimierung über 9 Parameter

Es müssen mit den Hüftoszillatorparametern, die maßgeblich das Basismuster der Laufbewegung bestimmen, auch die elastischen Eigenschaften der Beine entsprechend angepasst werden, um hohe Drehmomente, besonders in den Hüftgelenken, zu vermeiden.

Experimente haben gezeigt, dass die ersten 9 Parameter, für Hüftoszillator, Kniegelenk und Fußgelenk, die vorerst verheißungsvollsten Eingaben bzgl. der Optimierung sind. Die Parameter der Muskeln scheinen eher feineren als starken Einfluss auf die Laufbewegung des Modells zu haben, da eben die Hauptelastizität schon direkt in den Gelenken selbst modelliert ist (c_1 , c_2).

Nach all diesen ersten Erkenntnissen ist die Bildung neuer Kriterien möglich. Einige Optimierungskriterien, mit denen experimentiert wurde, werden hier vorgestellt und zugehörige Vor- und Nachteile kurz erläutert. Die spätere Präsentation von numerischen Optimierungsergebnissen beschränkt sich dann auf die signifikantesten und erfolgreichsten Ansätze.

3.3.1 Formulierung von Optimierungszielen

In der Optimierungsschnittstelle des Simulink-Modells werden die anfallenden Simulationsdaten gemäß des aktuellen Gütekriteriums und dessen Nebenbedingungen für das Optimierungsverfahren aufbereitet und zusammengestellt. Simulationübergreifende Prozesse, wie z.B. das Schreiben einer History oder das dynamische Modifizieren eines Gütekriteriums (z.B. durch Abkühlungsfaktor beim Simulated Annealing), passieren entsprechend von der alles umfassenden Matlabebene aus (Abb. 3.3). Die Vielfalt der am Modell messbaren Daten ermöglicht das Bilden unzähliger Optimierungskriterien.

Legt man z.B. auf eine schnelles und symmetrisches Laufen (bei Beine gleichbelastet), so bietet sich das folgende Gütekriterium an.

$$\min_p \varphi(p) = -v_p(t_f) + \rho \cdot |M_{RLeg,p}(t) - M_{LLeg,p}(t + \frac{1}{2f})|$$

Hierbei wird auf das bisherige Kriterium an die Geschwindigkeit ein Strafterm addiert. Dieser Strafterm wächst, wenn die phasenverschobene Deckungsgleichheit der Drehmomentverläufe nachlässt. Bei diesen kombinierten Kriterien besteht jedoch generell das Problem der Wahl des richtigen Multiplikators ρ , oder gar einer Multiplikatorfunktion $\rho(p)$. Einfaches Skalieren auf den ungefähr gleichen Wertebereich genügt in der Regel nicht, um treffende Optimierungsergebnisse zu erzielen.

Statt solcher Straftermansätze bietet sich auch das Verwenden von festen Schranken an, das allerdings ein fähiges Optimierungsverfahren erfordert. Will man z.B. Flugphasen auf der Suche nach dem schnellst möglichen Gehen verhindern, bietet sich dieses Optimierungsproblem an:

$$\min_p \varphi(p) = -v(p) \quad \text{s.t.} \quad \sum_{i,t} \theta_{p,i}(t) \geq 1$$

Wobei $\theta_{p,i}(t)$ eine binäre Rechteckfunktion des i -ten Kontaktpunkts zur Konfiguration p ist, die beschreibt, ob zum Zeitpunkt t Bodenkontakt besteht.

Allerdings sind binäre Nebenbedingungen vom Optimierer generell schwer zu bewältigen, da er nur diskret 0 oder 1 vorge setzt bekommt und nicht weiß, wie nah er genau dem Übergang ist (Problem 3. Iteration (Abb. 3.4)). Abhilfe können hierbei Barrierefunktionen schaffen, die in der Nähe des Übergangs (außen oder innen) sehr steil werden und so die Optimierungspunkte in den zulässigen Bereich schieben. Hierbei wird die betroffene harte Schranke eliminiert und mit dem Gütekriterium entsprechend verrechnet. Z.B. könnte man so, anstatt die Flugphasen generell streng binär zu verbieten, im Sinne einer äußeren Straffunktion die Dauer der Flugphasen quadratisch oder gar exponentiell bestrafen. Zu beachten ist hierbei aber, dass die Lösung dieses Optimierungsproblems nicht zwingend auch zulässig gemäß der eliminierten, binären Nebenbedingung ist.

Das Ganze ist also ein geschicktes Abwägen zwischen der Komplexität der Nebenbedingungen und der des Gütekriteriums, in Bezug auf Optimierungsverfahren und Modellverhalten.

Die eben allgemein angesprochenen Probleme äußerten sich auch bei unseren Versuchen in hohen Funktionsauswertungsanzahlen und teils schlechten (hier nicht aufgeführten) Optimierungsergebnissen. Exotischere Ansätze wurden deshalb nicht weiter verfolgt. Effektiver erwiesen sich die vom Optimalsteuerungsproblem aus der dynamischen Optimierung motivierten Standardkriterien bzgl. der Zeit und Energie.

Kriterien eines Optimalsteuerungsproblem: Gesucht ist die optimale Steuerung u , die das Gütekriterium J minimiert. Weiter definiert x in der Regel einen Zustand bzw. zusätzliche Parameter. Hier wird die gängige Formulierung aus der dynamischen Optimierung verwendet mit dem Mayer-Term ϕ und dem Lagrange-Term L .

$$\min_u J[x, u] = \phi(x(t_f), t_f) + \int_0^{t_f} L(x(t), u(t), t) dt$$

Minimale Endzeit für eine bestimmte Bewegung (t_f frei)

$$(\phi = 0) \wedge (L = 1) \Rightarrow \min_u J_T[u] = t_f$$

Maximale Endgeschwindigkeit (t_f fest)

$$(\phi = -v(t_f)) \wedge (L = 0) \Rightarrow \min_u J_V[u] = -v(t_f)$$

Minimale Energie (t_f fest)

$$(\phi = 0) \wedge (L = u^T(t) \cdot u(t)) \Rightarrow \min_u J_E[u] = \int_0^{t_f} u^T(t) \cdot u(t) dt$$

Bei diesen Kriterien sind jedoch keine einzelnen Werte als Lösung gesucht, sondern komplette Funktionsverläufe bzw. Trajektorien (Abb. 3.11). Die Architektur des Simulink-Modells mit dem Parameterblock als Eingabeschnittstelle nimmt aber nicht direkt den optimalen Verlauf der Steuerung entgegen, sondern die optimale Parameterkonfiguration, die den Verlauf impliziert. Aufgrund des als CPG fungierenden Hüftoszillators (f, α, ω), der sich um die periodische Ansteuerung kümmert, kann man das Optimalsteuerungsproblem wieder auf die statische Optimierung zurückführen.

Rückführung auf die statische Optimierung: Gesucht ist die Parameterkonfiguration p , die die optimale Steuerung u_p (Drehmomentverlauf) impliziert.

Maximale Endgeschwindigkeit (t_f fest)

$$\min_p J_V[p] = -v_p(t_f)$$

Minimale Energie (t_f fest)

$$\min_u J_E[p] = \int_0^{t_f} u_p^T(t) \cdot u_p(t) dt$$

Somit kommt man auf die bereits vorweggenommene Idee, neben der Geschwindigkeit die Drehmomente in der Hüfte direkt mit einer Nebenbedingung zu beschränken oder gar deren Minimierung als Pseudoenergie zum Hauptkriterium zu machen.



Abbildung 3.11: Industrieroboter, wie dieser (KR 6 ARC, KUKA Roboter GmbH), haben oft vorbestimmte, einseitige Arbeiten zu verrichten. Statt Flexibilität steht hier meist die Präzision im Vordergrund. Es müssen vorbestimmte Koordinaten oft zeit- oder energieminimal erreicht werden. Gesucht ist hierbei der optimale Steuerungsverlauf.

3.3.2 Unbeschränkte Optimierung mit Imfil

Der erste simple erfolgreiche Ansatz war ein Optimierungslauf von den initialen Werten des Startpunkt1 aus mit einem Gütekriterium, dessen Lösung die Endgeschwindigkeit maximieren soll.

Imfil-Problem 1

$$\begin{aligned} p_0 &= \text{Startpunkt1} \\ \min_p \quad &\varphi(p) = -v_p(t_f) \\ \text{s.t.} \quad &- \end{aligned}$$

Das implizite Filtern scheint sich hierbei relativ schnell in einem lokalen Optimum verfangen zu haben, wie auch Vergleiche mit späteren Ergebnissen zeigen. Die Konfiguration des Ergebnisses, hier Optipunkt1 genannt, liegt nicht weit von der Startlösung entfernt und die Laufmuster sind sich folglich auch sehr ähnlich, wie im nächsten Kapitel weiter untersucht wird.

Optipunkt1			
f	1.1241	[Hz]	Frequenz (Hüfte)
α	19.7817	[°]	Offset-Winkel (Hüfte)
ω	0.3888	[rps]	Drehzahl (Hüfte)
c_1	5.3839	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	119.9802	[°]	Ruhewinkel (Fuß)
d_1	1.0514	[Nms/°]	Drehdämpfung (Fuß)
c_2	8.0190	[Nm/°]	Drehsteifigkeit (Knie)
p_2	140.0232	[°]	Ruhewinkel (Knie)
d_2	0.3853	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	20	[°]	Ruhewinkel (RF)
c_{GAS}	0	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	50	[°]	Ruhewinkel (GAS)

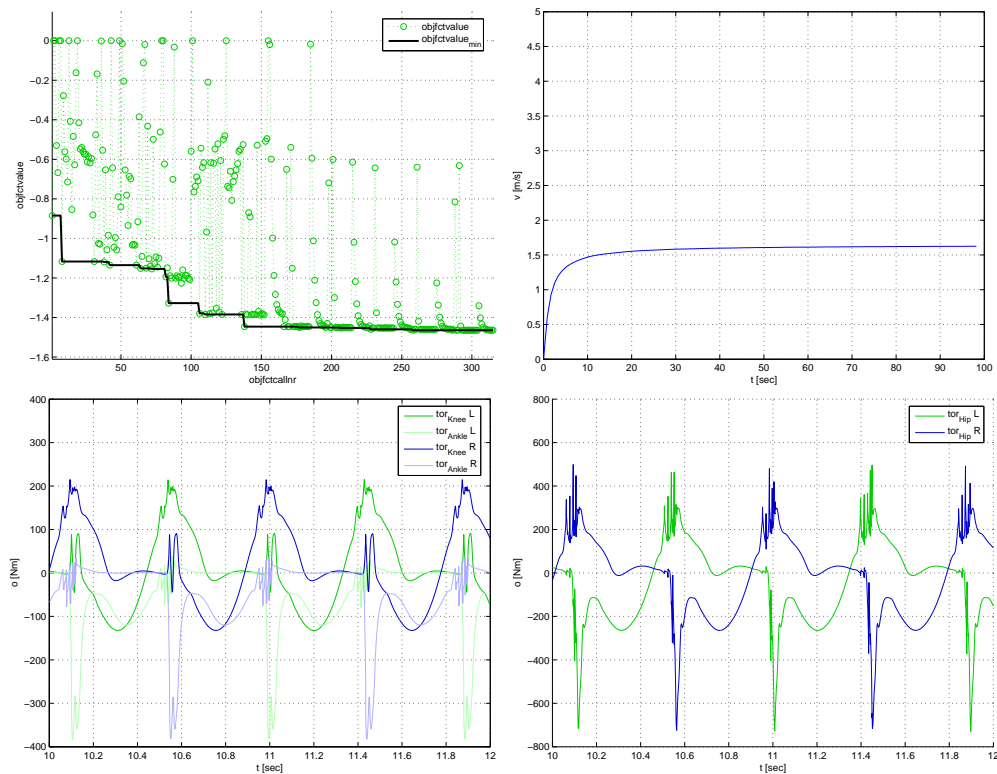


Abbildung 3.12: Optipunkt1: Optimierungskonvergenz (links oben), Geschwindigkeitsverlauf (rechts oben), Drehmomente in Knie- und Fußgelenk (links unten), Drehmomente in den Hüftgelenken (rechts unten).

Nach 150 Gütefunktionsauswertungen ist Optipunkt1 eigentlich schon erreicht (Abb. 3.12). Es folgen aber weitere 150 Aufrufe, in denen sich das Verfahren von unterschiedlichen Richtungen aus immer wieder im gleichen lokalen Minimum

fängt. (Ein Beispiel für die bereits angesprochene langsame Konvergenz des Impliziten Filterns in unmittelbarer Nähe des Optimums.)

Die Geschwindigkeit der Ergebniskonfiguration beträgt nach 10 Sekunden schon ca. $1.5 \frac{\text{m}}{\text{s}}$ und ist somit eine Verbesserung um mehr als 50%. Betrachtet man die Drehmomente im Knie- und Fußgelenk, so fällt auf, dass neben den nur unmerklich höheren Momenten im Knie, das Fußgelenk kurzzeitig mehr als doppelt so hohe Drehmomente aufweist. Die für den Antrieb entscheidenden Momente in den Hüftgelenken steigen um mehr als 200%. Während das Vorziehen des Schwungbeins kaum höhere Kraft erfordert (Ausschläge im positiven Bereich), entstehen beim Aufsetzen des Fußes vor dem Körper und anschließendem zurückziehen deutlich höhere Kräfte (Ausschlag nach unten). Es ergibt sich folgenden Bilanz:

Δ Bilanz:	
Startpunkt1 \implies Optipunkt1	
$v_p(t_f)$	+70%
$epm_p(t)$	-9%
$\ u_{hip,p}\ _\infty$	+80%
$\ u_{knee,p}(t)\ _\infty$	+11%
$\ u_{ankle,p}(t)\ _\infty$	+200%

$v_p(t_f)$	Geschwindigkeit nach 100 Sekunden
$epm_p(t)$	verbrauchte Energie pro Meter
$\ u_{hip,p}(t)\ _\infty$	max. Ausschlags des Betrags der Drehmomente im Hüftgelenk
$\ u_{knee,p}(t)\ _\infty$	max. Ausschlags des Betrags der Drehmomente im Kniegelenk
$\ u_{ankle,p}(t)\ _\infty$	max. Ausschlags des Betrags der Drehmomente im Fußgelenk

(Zu beachten ist hierbei, dass diese Bilanz nicht direkt den Optimierungfortschritt bewerten soll, sondern die Verbesserung der resultierenden Laufbewegung. So ist die verbrauchte Energie mit der zurückgelegten Strecke verrechnet und $t_f = 100s \neq 10s$ bei der Geschwindigkeitsmessung.)

Da bzgl. des Gütekriterium die laut den Testpunkten möglichen, hohen Geschwindigkeiten nicht annähernd erreicht wurden und mit einem erneuten Starten des Verfahrens vom Optipunkt1 auch keine weitere Verbesserung erzielt wurde, fällt die Wahl auf Testpunkt3 als neue Startlösung. Bei dieser Konfiguration wird trotz anfänglicher Probleme, die Spielraum zur Optimierung lassen, mit die höchste Geschwindigkeit erzielt (Abb. 3.7).

Hier ist das Optimierungskriterium des 2. Laufs (gleiches Verfahren, gleiche Gütefunktion, aber unterschiedliche Startlösung).

Imfil-Problem 2

$$\begin{aligned}
 p_0 &= \text{Testpunkt3} \\
 \min_p \quad &\varphi(p) = -v_p(t_f) \\
 \text{s.t.} \quad &-
 \end{aligned}$$

Optipunkt2

f	1.2911	[Hz]	Frequenz (Hüfte)
α	20.9709	[°]	Offset-Winkel (Hüfte)
ω	0.4915	[rps]	Drehzahl (Hüfte)
c_1	4.8745	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	120.0118	[°]	Ruhewinkel (Fuß)
d_1	0.0892	[Nms/°]	Drehdämpfung (Fuß)
c_2	7.9909	[Nm/°]	Drehsteifigkeit (Knie)
p_2	139.9174	[°]	Ruhewinkel (Knie)
d_2	0.0895	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	20	[°]	Ruhewinkel (RF)
c_{GAS}	0	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	50	[°]	Ruhewinkel (GAS)

Im Konvergenzbild (Abb. 3.13) ist wieder zu erkennen, dass bereits nach der Hälfte der Auswertungen ein zum endgültigen Optimum vergleichbares Ergebnis vorliegt. Danach scheinen sich zahlreiche Iterationen in umliegenden, schlechteren, lokalen Minima zu verfangen. Dies lässt auf einen recht zerklüfteten Bildraum des Zielfunktional schließen. Gestartet wird bei einer Endgeschwindigkeit von ca. $v(t_f = 10\text{s}) = 2 \frac{\text{m}}{\text{s}}$ der Konfiguration Testpunkt3 (Abb. 3.7). Im Konvergenzbild und Geschwindigkeitsverlauf zu Optipunkt2 (Abb. 3.13) erkennt man: $v(t_f = 10\text{s}) = 2.7 \frac{\text{m}}{\text{s}}$. Die anfänglichen Probleme des Roboters, das richtige Laufmuster zu finden, wurden wegoptimiert und nach 100 Sekunden konvergiert die Geschwindigkeit zum bis jetzt höchsten Wert von $v(t_f = 100\text{s}) = 3.5 \frac{\text{m}}{\text{s}}$. Aus den Drehmomentschaubildern wird wieder die Ähnlichkeit zur Startlösung sichtbar. Betrachtet man aber die Motorparameter f und ω genauer, so stellt man fest, das sich Optipunkt2 weit von der Startlösung Testpunkt3 entfernt hat und in der Nähe von Testpunkt1 liegt. Die Frequenz hat sich etwas erhöht, während die erforderliche Motordrehzahl erfreulicher Weise gesunken ist. Die Ausschläge der Drehmomente an den Hüftgelenken in den negativen Bereich, die beim Auftreten nach Flugphasen entstehen, sind zwar immer noch recht hoch, aber haben sich ohne das Verwenden von Schranken oder Bestrafung deutlich verringert.

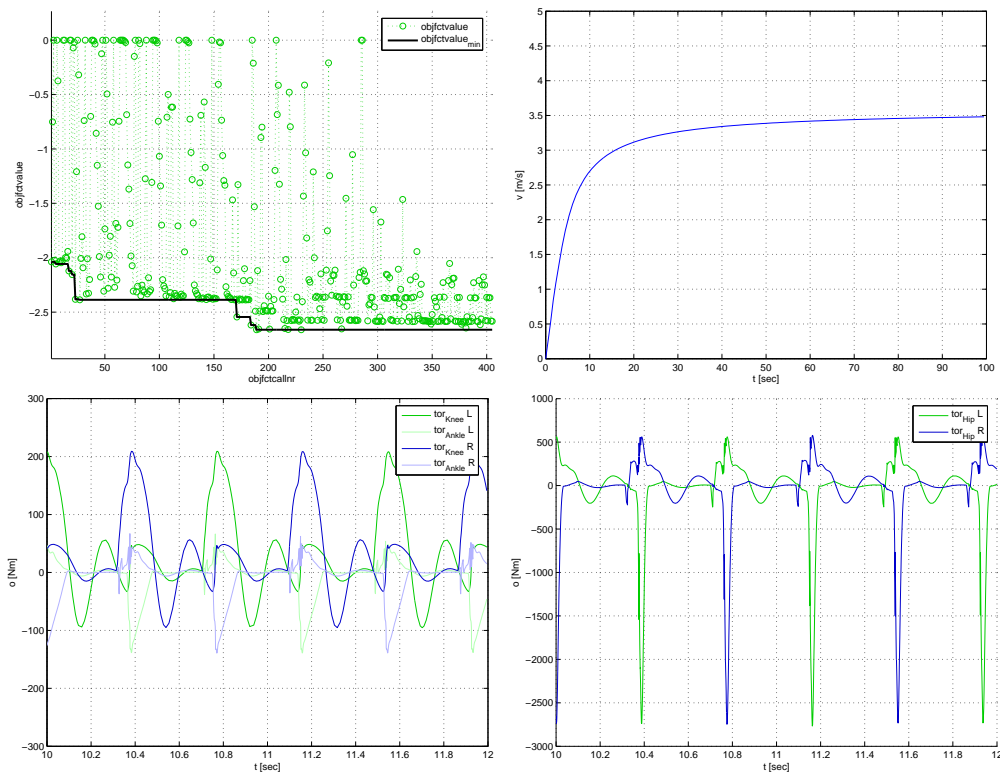


Abbildung 3.13: Optipunkt2: Optimierungskonvergenz (links oben), Geschwindigkeitsverlauf (rechts oben), Drehmomente in Knie- und Fußgelenk (links unten), Drehmomente in den Hüftgelenken (rechts unten).

Δ Bilanz:	
Testpunkt3 \implies Optipunkt2	
$v_p(t_f)$	+6%
$epm_p(t)$	-67%
$\ u_{hip,p}\ _\infty$	-63%
$\ u_{knee,p}(t)\ _\infty$	+11%
$\ u_{ankle,p}(t)\ _\infty$	-13%

Diese Ergebnisse und die Tatsache, dass die Elastischen Fähigkeiten der Beine im Anbetracht der Knie- und Fußgelenkparameter kaum modifiziert/ausgenützt wurden, motivieren zu weiteren Optimierungsstudien mit anderen Verfahren.

3.3.3 Beschränkte Optimierung mit Nomad

Es muss möglich sein, auch ohne extreme Drehmomente in der Hüfte eine akzeptable Geschwindigkeit zu erreichen, z.B. bei einem Laufmuster, dessen Flugphasen flacher und kürzer sind, sodass die Stöße auf die Hüfte verringert werden.

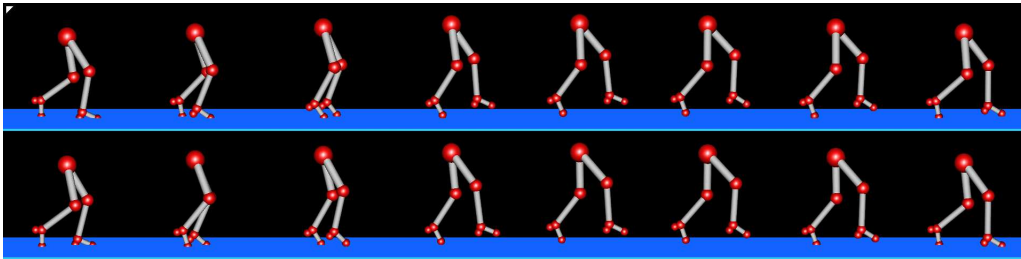


Abbildung 3.14: Hopslerlauf der Konfigurationen Optipunkt2

Die Fähigkeit des Nomad-Optimierungsverfahrens, mit nichtlinearen Nebenbedingungen umgehen zu können, erlaubt es, wie bereits angekündigt, Optimierungsprobleme der folgenden Art zu formulieren:

Nomad-Problem 1

$$\begin{aligned}
 p_0 &= \text{Startpunkt1} \\
 \min_p \quad & \varphi(p) = -v_p(t_f) \\
 \text{s.t.} \quad & |u_{hip,p}(t)| < 500[\text{Nm}] \quad t \in [t_0, t_f]
 \end{aligned}$$

Das Kriterium ist identisch zum ersten Ansatz des Impliziten Filterns, bis auf die Schranke, die den Raum des Optimierers eindämmt. Der Betrag der Hüftdrehmomente von Startpunkt1 ist stets kleiner als 400Nm und erfüllt somit die neu eingeführte, simulationsbasierte Nebenbedingung.

Nach den bisherigen Erkenntnissen wäre offenbar eine bzgl. des Gütekriteriums noch schlechtere Lösung als Optipunkt1 zu erwarten, denn schon die Hüftdrehmomente dieser Konfiguration verletzen die Nebenbedingung. Umso so überraschender war das Ergebnis des folgenden Optimierungslaufs.

(Es sei hier nur kurz angemerkt, dass auch die Testpunkte 1-3 sich nicht als Startpunkte für dieses Kriterium eignen, denn Startlösungen sollten vor allem eines sein: zulässig, wie jede andere Lösung auch.)

Optipunkt5			
f	1.5	[Hz]	Frequenz (Hüfte)
α	21	[°]	Offset-Winkel (Hüfte)
ω	0.51143	[rps]	Drehzahl (Hüfte)
c_1	9.5625	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	136	[°]	Ruhewinkel (Fuß)
d_1	0.0375	[Nms/°]	Drehdämpfung (Fuß)
c_2	4	[Nm/°]	Drehsteifigkeit (Knie)
p_2	140	[°]	Ruhewinkel (Knie)
d_2	0.26602	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	20	[°]	Ruhewinkel (RF)
c_{GAS}	0	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	50	[°]	Ruhewinkel (GAS)

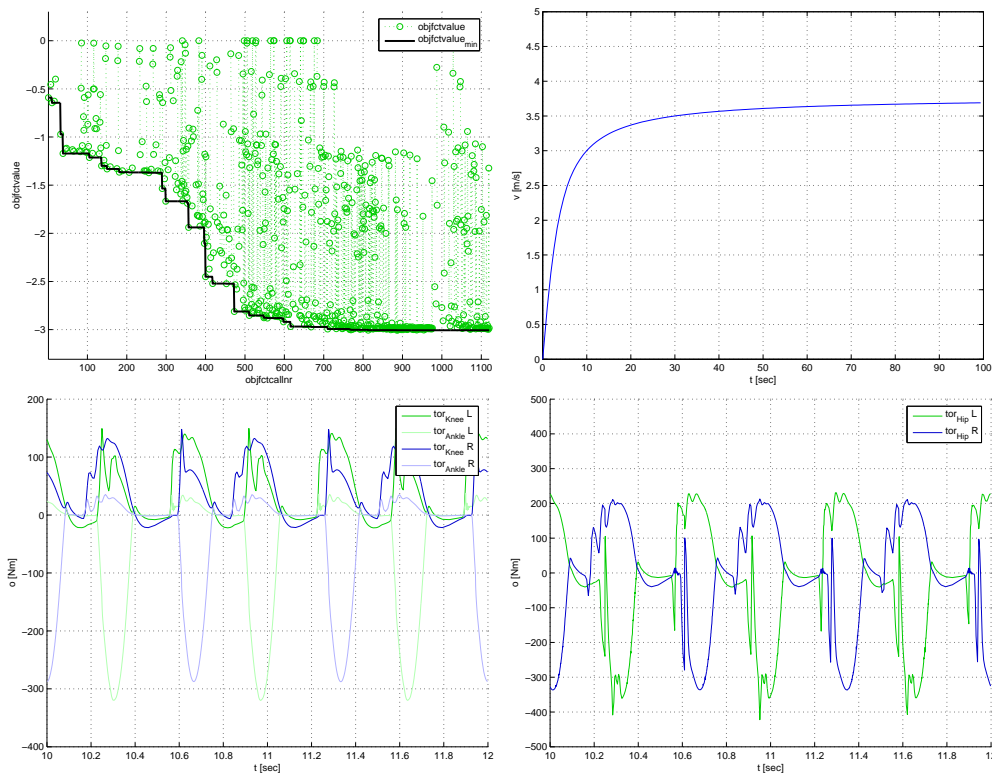


Abbildung 3.15: Optipunkt5: Optimierungskonvergenz (links oben), Geschwindigkeitsverlauf (rechts oben), Drehmomente in Knie- und Fußgelenk (links unten), Drehmomente in den Hüftgelenken (rechts unten).

Wie bereits angekündigt, brauchen Mustersuchverfahren, wie Nomad, generell mehr Zielfunktionsauswertungen, sind aber dadurch auch robuster. Die hohe Geschwindigkeit zu den kleinen Drehmomenten (Abb. 3.15) rechtfertigt jedoch

die mehr als 1000 Aufrufe der Gütefunktion. Hier liegt sogar die bisherige Rekordgeschwindigkeit von $v(t_f = 100\text{s}) = 3.7 \frac{\text{m}}{\text{s}}$ vor. Im Vergleich zur Startlösung hat sich die Geschwindigkeit ungefähr verdreifacht.

Betrachtet man die Parameter der betroffenen Konfigurationen, so fällt auf, dass, anders als beim Impliziten Filtern, die elastischen Eigenschaften der Knie- und Fußgelenke sich deutlich geändert haben. Neben der um 50% erhöhten Frequenz des Hüftoszillators hat sich die Drehsteifigkeit im Fußgelenk mehr als verdoppelt, aber die im Knie halbiert. Während die Haltung des Knies gleichgeblieben ist, hat sich der Ruhewinkel des Fußgelenks um 16° vergrößert. Das bewirkt einen durchgestreckteren, härter federnden Fuß und somit ein Aufrichten auf den Fußballen. Folglich erhöht sich die gesamte Elastizität des Beins und die Stöße auf die Hüfte werden effektiv absorbiert. Während sich so die Drehmomente an den Fußgelenken folglich verdoppelt haben, sind die an den Hüftgelenken ungefähr auf dem gleichen Level geblieben, wie bei der Startlösung. Insgesamt wurde so eine enorme Verbesserung im Vergleich zu den Testpunkten erzielt und die Laufbewegung ist in Betracht auf die Motorleistung wesentlich schonender und realisierbarer geworden.

Δ Bilanz:	
Startpunkt1 \implies Optipunkt5	
$v_p(t_f)$	+270%
$epm_p(t)$	-70%
$\ u_{hip,p}\ _\infty$	+0%
$\ u_{knee,p}(t)\ _\infty$	-21%
$\ u_{ankle,p}(t)\ _\infty$	138%

Die hohe Geschwindigkeit trotz der drosselnden Nebenbedingung (die nicht einmal deutlich aktiv wird) hat an dieser Stelle dazu motiviert, einen unbeschränkten Optimierungslauf mit dem Nomad-Verfahren ebenfalls von Startpunkt1 aus durchzuführen um zu erfahren, wie schnell der Roboter generell sein kann. Erstaunlicher Weise hat das Verfahren die identische Konfiguration Optipunkt5 als Lösung zurückgegeben. Hohe Geschwindigkeit muss also unter geschickter Verwendung von elastischem Potential nicht unbedingt hohe Drehmomente bedeuten, wie dieses Beispiel zeigt.

Da also vorerst die schnellste Konfiguration für 9 Parameter gefunden schien, war es jetzt interessant, die Momente unter eventuellen Geschwindigkeitseinbußen weiter zu reduzieren. Hierzu bietet sich folgendes Optimierungskriterium an:

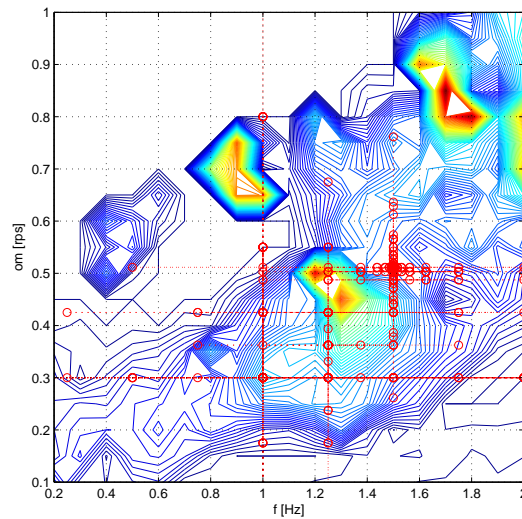


Abbildung 3.16: Auf diesem Bild sieht man, wie das Nomad-Verfahren die Parameter f und ω variiert und zum Optimum hin verdichtet. Zu beachten ist, das 7 weitere Parameter jetzt das Gütekriterium φ beeinflussen, und deshalb das Optimum sich nicht mehr mit dem hier nochmals unterlegten Auswertungsgitter der Testpunkte (bzgl. nur 2 Parameter) deckt.

Nomad-Problem 2

$$\begin{aligned}
 p_0 &= \text{Optipunkt5} \\
 \min_p \quad &\varphi(p) = \int_{t_0}^{t_f} u_{hip,p}^2(t) dt \\
 \text{s.t.} \quad &v_p(t_f) > 2 \left[\frac{\text{m}}{\text{s}} \right]
 \end{aligned}$$

In der Optimierungsschnittstelle werden die Quadrate der an den beiden Hüftgelenken auftretenden Drehmomente über die Simulationszeit integriert, und so eine Aussage gebildet über die verbrauchte Pseudoenergie des von der aktuellen Konfiguration implizierten Laufmusters. Die Startlösung Optipunkt5 ist bzgl. der Mindestendgeschwindigkeitsforderung sicherlich mehr als zulässig. Aufgrund des vorhandenen Optimierungsspielraums kann man davon ausgehen, dass die Drehmomente in der Gütefunktion weiter minimiert werden können.

Optipunkt9			
f	1.5	[Hz]	Frequenz (Hüfte)
α	33.0625	[°]	Offset-Winkel (Hüfte)
ω	0.51143	[rps]	Drehzahl (Hüfte)
c_1	8.5625	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	136	[°]	Ruhewinkel (Fuß)
d_1	0.058984	[Nms/°]	Drehdämpfung (Fuß)
c_2	5.5938	[Nm/°]	Drehsteifigkeit (Knie)
p_2	140	[°]	Ruhewinkel (Knie)
d_2	0.42227	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	20	[°]	Ruhewinkel (RF)
c_{GAS}	0	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	50	[°]	Ruhewinkel (GAS)

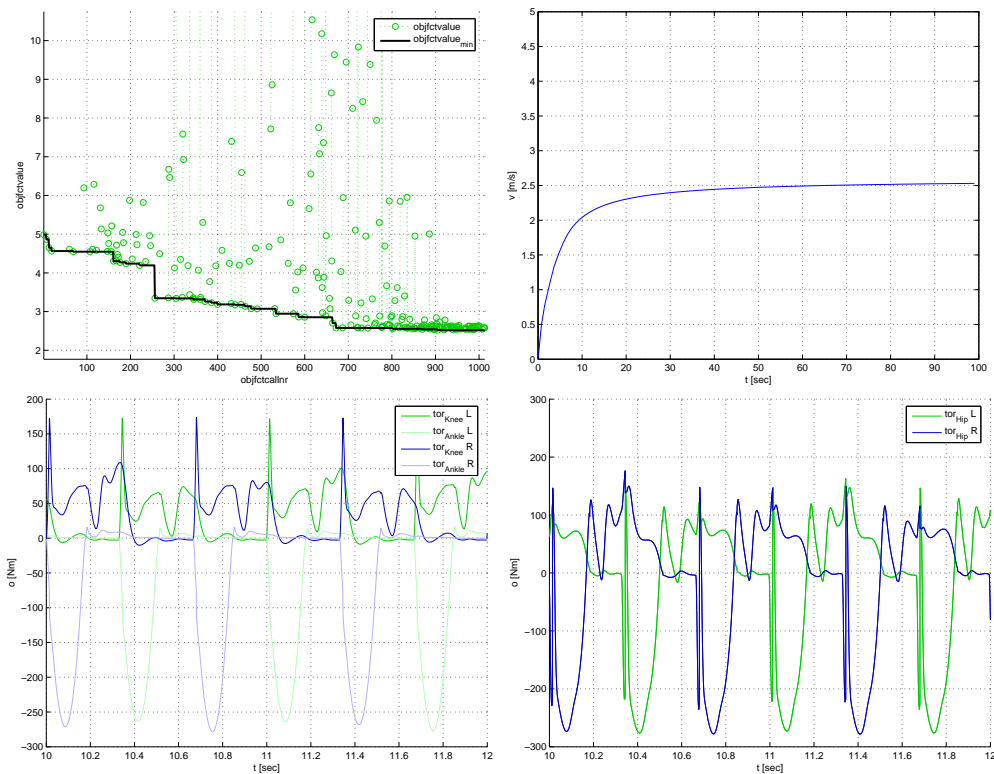


Abbildung 3.17: Optipunkt9: Optimierungskonvergenz (links oben), Geschwindigkeitsverlauf (rechts oben), Drehmomente in Knie- und Fußgelenk (links unten), Drehmomente in den Hüftgelenken (rechts unten).

Nach der gewohnt hohen Anzahl von ca. 1000 Zielfunktionsauswertungen konvergiert das Nomad-Verfahren nahe der Schranke zum unzulässigen Bereich. Während des Optimierungslaufs lieferten, wie aus dem Konvergenzbild (Abb.

3.17) ersichtlich, einige Aufrufe der Gütefunktion sehr hohe, hier aus Skalierungsgründen abgeschnittene Werte an Pseudoenergieeinheiten zurück. Das liegt hauptsächlich daran, dass der Bildraum der Drehmomente, und somit auch verstärkt der Energie, sich als wesentlich zerklüfteter herausgestellt hat als der Bildraum der Geschwindigkeit (siehe auch Kapitel Kontrollraum).

Betrachtet man wieder die Parameter der betroffenen Konfigurationen so ist besonders überraschend, dass sich die Motorparameter f und ω überhaupt nicht verändert haben, obwohl das Kriterium an die minimale verbrauchte Energie eigentlich hauptsächlich den Motor adressiert. Am signifikantesten ist die Änderung des Offset-Winkels α in der Hüfte um ganze $+12^\circ$, die eine verstärkte Rücklage des Roboters zur Folge hat, während die Haltung des Knie- und Fußgelenks gleichgeblieben ist. Weiter gibt es wieder eine Drehmomentverlagerung um ca. 10% vom Fußgelenk zurück auf das Knie, dessen Drehsteifigkeit und Dämpfung gestiegen ist. Unter dem Strich ist das ein Geschwindigkeitsverlust und eine Einsparung bzgl. der Antriebsmomente um jeweils 30%. Die über die Simulationszeit verbrauchte Energie wurde entsprechend um sogar knappe 50% verringert (Konvergenzbild). Die Beine bewegen sich anders als bei der Startlösung wieder nach einem symmetrischen Muster (Abb. 3.18). Trotz des nach wie vor sprunghaft federnden Verhaltens, sind die Flugphasen fast verschwunden.

An dem Geschwindigkeitsverlauf (Abb. 3.17) kann man erkennen, dass anders als bei dem ersten Nomad-Optimierungslauf der Spielraum bezüglich der Nebenbedingung komplett ausgeschöpft wurde. So liegt die Geschwindigkeit nach Simulationsende $t_f = 10\text{s}$ nur ganz knapp über $2\frac{m}{s}$ im zulässigen Bereich. D.h. weiteres Reduzieren der Momente geht auf jeden Fall einher mit Geschwindigkeitsverlust.

Δ Bilanz:	
Optipunkt5 \implies Optipunkt9	
$v_p(t_f)$	-30%
$epm_p(t)$	-8%
$\ u_{hip,p}\ _\infty$	-30%
$\ u_{knee,p}(t)\ _\infty$	+13%
$\ u_{ankle,p}(t)\ _\infty$	-13%

Das kann man natürlich auch noch weiter treiben und die Schranke zur Drosselung der Geschwindigkeit tiefer setzen, um zu schauen, in welcher Relation das Drehmoment minimiert werden. Es hat sich jedoch gezeigt, dass ab dem Unterschreiten eines bestimmten Grunddrehmoments die Geschwindigkeit überproportional stark sinkt und die Bewegung so wieder sehr unwirtschaftlich wird.

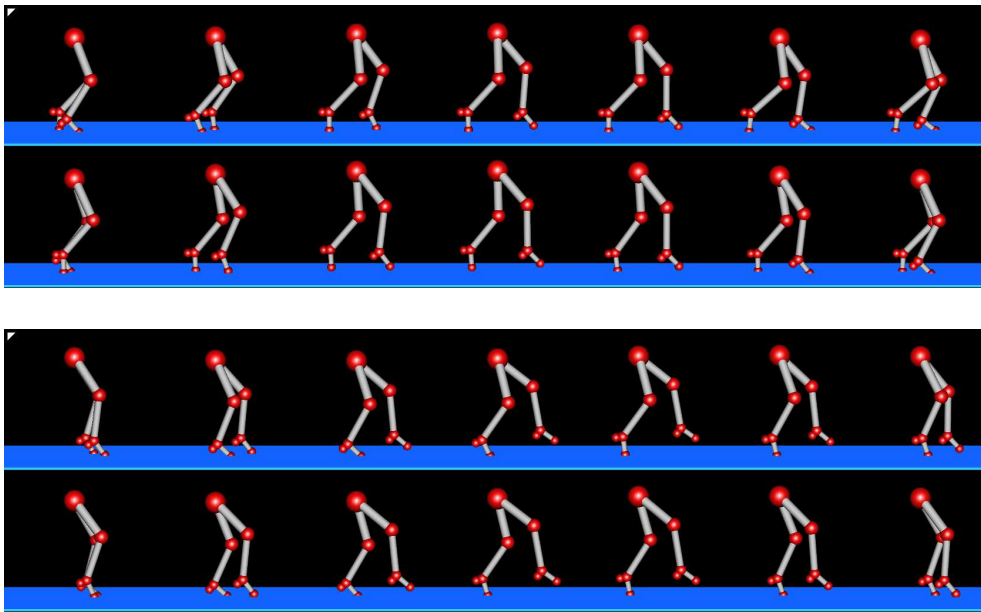


Abbildung 3.18: Laufmuster der Konfigurationen Optipunkt5 (obere Sequenz) und Optipunkt9 (untere Sequenz);

3.4 Optimierung über 12 Parameter

Nachdem man sehen konnte, wie durch passende Einstellungen der Haltung, Drehsteifigkeit und Dämpfung in Knie- und Fußgelenk der Motor in der Hüfte entscheidend entlastet wurde, so ist es jetzt natürlich interessant zu erfahren, in welchen Umfang die Muskeln GAS, RF weiteren Einfluss auf das Laufmuster ausüben.

Die zusätzlichen Eingabeparameter c_{RF} , p_{RF} , c_{GAS} und p_{GAS} erweitern den Optimierungsraum also um 4 Dimensionen. Motiviert von dem realen Roboter mit seinen eher langsamen, fest eingebauten Motoren, wird bei den folgenden zwei Studien auf die Optimierung der max. Drehzahl ω bewusst verzichtet. Jetzt interessieren neben der Frequenz f , welche die Umschaltzeiten der Motoren, und somit die Schrittweite festlegt, nur noch die Parameter zu den passiven Eigenschaften (Haltung α, p , Dämpfung d und Steifigkeit c). Der Optimierungsraum wächst somit auf insgesamt 12 Dimensionen an.

3.4.1 Beschränkte Optimierung mit Nomad

Im folgenden werden zwei weitere Ergebniskonfigurationen präsentiert, deren Gütekriterien vergleichbar sind zu denen von Optipunkt5 und Optipunkt9. Deshalb bietet sich zunächst dieses Kriterium an:

Nomad-Problem 3 $p_0 =$ Startpunkt1 $\min_p \quad \varphi(p) = -v_p(t_f)$ s.t. $|u_{hip,p}(t)| < 1000[\text{Nm}] \quad t \in [t_0, t_f]$

Die obere Schranke an die Drehmomente in der Hüfte ist hierbei bewusst etwas höher gesetzt. Denn beim Optimierungslauf zur Konfiguration Optipunkt5 war auf dem Weg zum Minimum, das am Ende zwar sehr klar im zulässigen Bereich lag, eine überdurchschnittlich hohe Anzahl an knappen Nebenbedingungsverletzungen (600Nm) zu verzeichnen. Um dem Optimierer auch diese Gütefunktionsauswertungen als Information zu gönnen und um eventuelle, neue Pfade zu besseren Optima zu finden, wurden die Schranken entspannt.

Schaut man sich die Parameter der Ergebniskonfiguration an, so fällt auf, dass das Verfahren wieder bei der gleichen Frequenz $f = 1.5[\text{Hz}]$ der vorherigen 2 Studien gelandet ist. Die Drehzahl ω beträgt natürlich den Wert der Startlösung, da dem Optimierer die Kontrolle über diesen Parameter entrissen wurde. Neben der hohen Drehsteifigkeit c_1 und dem extrem weiten Ruhewinkel p_1 des Sprunggelenks, der wieder auf ein ausgestreckte Fußhaltung deutet, wurde das Kniegelenk erneut etwas weicher nimmt die gewohnte Haltung von 140° ein. Muskeln beugen und strecken federnd Gliedmaßen über die Gelenke. Deshalb sind die Muskeln hier einfach als Federn abstrahiert mit einem Ruhewinkel und einer Drehsteifigkeit und werden gemäß ihres Wirkungsgrads auf die Gelenke, die sie bespannen, addiert. Bis jetzt waren die Muskeln aufgrund ihrer nicht vorhandenen Steifigkeit $c = 0$ völlig deaktiviert. In der Konfiguration zu Optipunkt14 ist am Wadenmuskel (GAS) eine deutliche maximale Drehsteifigkeitserhöhung erfolgt und auch der Ruhewinkel hat sich um ca. 30° geweitet. Der Gebrauch der Wadenmuskulatur deutet wieder auf eine verstärkte Aktivität des Sprunggelenks hin, während die Oberschenkelmuskulatur (RF), wie auch das Knie, eine eher passive Rolle einnimmt.

Optipunkt14			
f	1.5	[Hz]	Frequenz (Hüfte)
α	27	[°]	Offset-Winkel (Hüfte)
ω	0.3	[rps]	Drehzahl (Hüfte)
c_1	9.75	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	152.25	[°]	Ruhewinkel (Fuß)
d_1	0.00039063	[Nms/°]	Drehdämpfung (Fuß)
c_2	2	[Nm/°]	Drehsteifigkeit (Knie)
p_2	140	[°]	Ruhewinkel (Knie)
d_2	0.095605	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0.16406	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	42.7813	[°]	Ruhewinkel (RF)
c_{GAS}	9.75	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	82	[°]	Ruhewinkel (GAS)

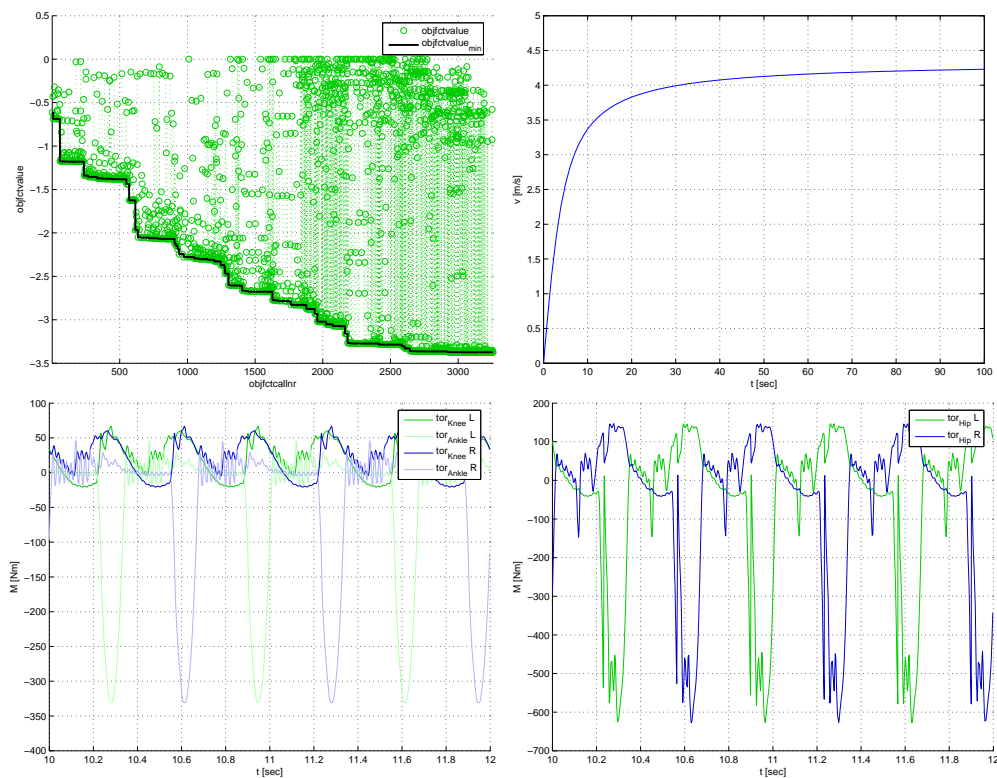


Abbildung 3.19: Optipunkt14: Optimierungskonvergenz (links oben), Geschwindigkeitsverlauf (rechts oben), Drehmomente in Knie- und Fußgelenk (links unten), Drehmomente in den Hüftgelenken (rechts unten).

Der Optimierungsraum wurde um 3 Dimensionen erweitert und im Konvergenzbild (Abb. 3.19) dieses Optimierungslaufs ist eine rund 3mal so hohe Funktionsauswertungsanzahl erkennbar. Diese Proportion ist eigentlich eine schlechte

Bilanz, und deutet auf einen zerklüfteten, schwierigen Bildraum hin. Die vielen schlechten Funktionsauswertungen gegen Ende des Optimierungslaufs, zeigen die Anfälligkeit der Laufbewegung mit der Rekordgeschwindigkeit von 4.2[Hz]. Die Optimierung wird so zu einer Gratwanderung zwischen hoher Geschwindigkeit und Zusammenbruch. In den Schaubildern der Drehmomente wird, neben der üblichen Tendenz zu starken Ausschlägen im Sprunggelenk, ein weiterer Effekt sichtbar. Nach dem Peak, der durch das Auftreten des Fußballens erzeugt wird, folgen deutliche Schwingungen der Drehmomente im Fußgelenk. Diese sind durch ein Hinterherschleifen des Vorderfußes über den Boden erklärbar. So muss das elastische Sprunggelenk immer wieder gegen die Haft- bzw. Gleitreibung des Bodens gemäß seiner Steifigkeit Drehmoment aufwenden, um in seinen Ruhewinkel zurück zu finden. Auch das ist ein weiteres Beispiel für die Störungstoleranz des flexiblen Designs. Bei einem starren Sprunggelenk wäre das Schleifen sicherlich nicht in dieser selbst-kontrollierenden Form möglich. Die vergleichbar hohen Hüftdrehmomente lassen aber darauf schließen, dass ein Lauf im Modell mit aktivierten Muskeln anstrengender ist als ohne. Die Momente liegen aber wiederum merklich unter der geforderten Schranke von 1000[Nm].

△ Bilanz:

Startpunkt1 \implies Optipunkt14

$v_p(t_f)$	+320%
$epm_p(t)$	-56%
$\ u_{hip,p}\ _\infty$	+53%
$\ u_{knee,p}(t)\ _\infty$	-63%
$\ u_{ankle,p}(t)\ _\infty$	+154%

Das Laufmuster der Konfiguration Optipunkt14 scheint das maximale an Geschwindigkeit zu sein, was der aktuelle Modell-Roboter liefern kann. Eben gerade deshalb ist diese Laufbewegung extrem anfällig. Das motiviert dazu, ausgehend von Optipunkt14 die Geschwindigkeit im Zuge der Energieminimierung zu drosseln. Dazu bietet sich das zu Optipunkt5 identische Kriterium an:

Nomad-Problem 4

$$\begin{aligned}
 p_0 &= \text{Optipunkt14} \\
 \min_p \quad & \varphi(p) = \int_{t_0}^{t_f} u_{hip,p}^2(t) dt \\
 \text{s.t.} \quad & v_p(t_f) > 2\left[\frac{\text{m}}{\text{s}}\right]
 \end{aligned}$$

Betrachtet man die Ergebniskonfiguration zu Optipunkt15, so fällt die bis jetzt höchste Frequenz auf. Interessant ist aber besonders wieder die Rückverlagerung der Steifigkeit vom Fußgelenk auf das Kniegelenk im Zuge der Energieminimierung. Und analog dazu wird auch die Oberschenkelmuskulatur etwas aktiver.

Optipunkt15			
f	1.8906	[Hz]	Frequenz (Hüfte)
α	23	[°]	Offset-Winkel (Hüfte)
ω	0.3	[rps]	Drehzahl (Hüfte)
c_1	8.75	[Nm/°]	Drehsteifigkeit (Fuß)
p_1	148.5	[°]	Ruhewinkel (Fuß)
d_1	0.056909	[Nms/°]	Drehdämpfung (Fuß)
c_2	5.6719	[Nm/°]	Drehsteifigkeit (Knie)
p_2	140	[°]	Ruhewinkel (Knie)
d_2	0.2626	[Nms/°]	Drehdämpfung (Knie)
c_{RF}	0.25757	[Nm/°]	Drehsteifigkeit (RF)
p_{RF}	36.3438	[°]	Ruhewinkel (RF)
c_{GAS}	9.75	[Nm/°]	Drehsteifigkeit (GAS)
p_{GAS}	82	[°]	Ruhewinkel (GAS)

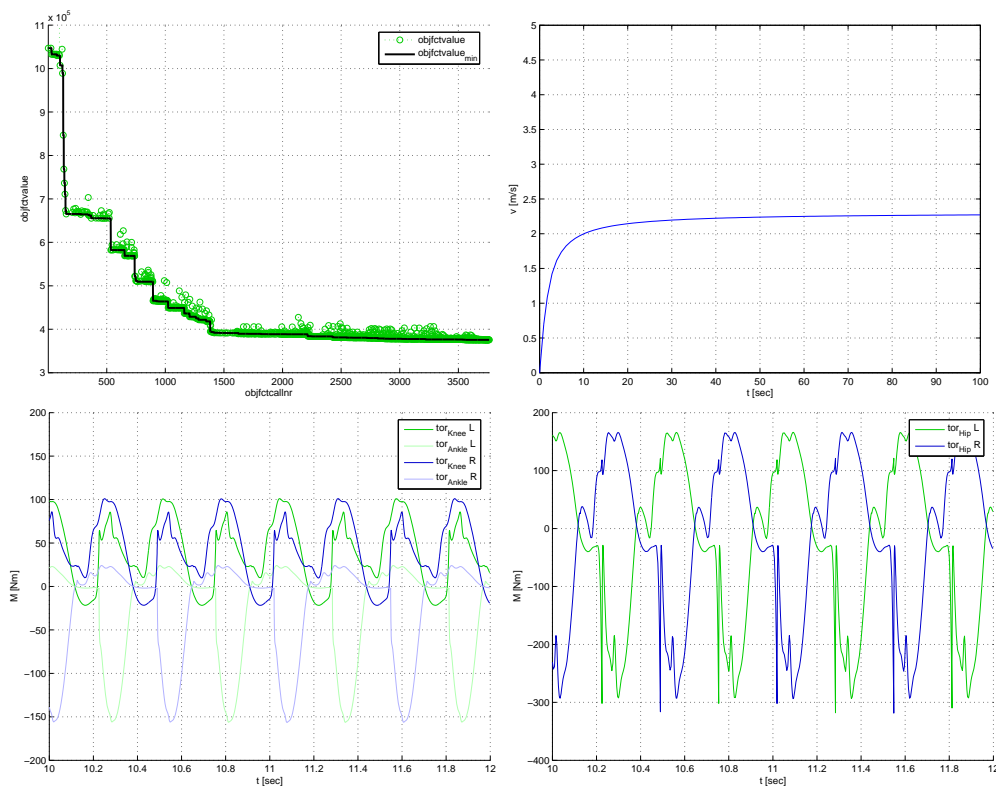


Abbildung 3.20: Optipunkt15: Optimierungskonvergenz (links oben), Geschwindigkeitsverlauf (rechts oben), Drehmomente in Knie- und Fußgelenk (links unten), Drehmomente in den Hüftgelenken (rechts unten).

Im Konvergenzschaubild (Abb. 3.20) ist ein krasser Unterschied zum letzten Optimierungslauf erkennbar. Neben der vergleichbar hoch gebliebenen Anzahl an

Funktionsauswertungen, scheint der Bildraum der verbrauchten Energie bei diesen Laufmustern wesentlich flacher beschaffen zu sein. Die Werte liegen sehr nahe bei einander und von Gratwanderung kann in diesem Sinne keine Rede mehr sein. Offensichtlich drosselt das Optimierungsverfahren die Geschwindigkeit soweit, bis die Nebenbedingung aktiv wird (siehe Geschwindigkeitsverlauf: nach 10s Simulationszeit genau $2\frac{\text{m}}{\text{s}}$). So wird jetzt im Konvergenzbild dieses Optimierungslaufs, die Schwierigkeit des Nomad-Verfahrens an nichtlinearen, aktiven Nebenbedingungen erkennbar. Die Schranke ist bereits nach 1500 Funktionsauswertungen erreicht. Der Simplex faltet sich aber weitere 2000 Auswertungen lang an dieser Schranke hin und her, bei denen keine merkliche Verbesserung zu verzeichnen ist. (Das unterstreicht allgemein nochmals die langsame Konvergenz von Simplexverfahren entlang nichtlinearer, aktiver Nebenbedingungen.) In den Drehmomentvisualisierungen ist beobachtbar, dass die Schleifschwingungen weitgehend verschwunden sind. Die Bewegung wirkt runder und somit stabiler. Die Drehmomente in Hüft- und Sprunggelenk sind jeweils um die Hälfte gesunken, während die im Knie leicht gestiegen sind.

Δ Bilanz:	
Optipunkt14 \implies Optipunkt15	
$v_p(t_f)$	-45%
$epm_p(t)$	-16%
$\ u_{hip,p}\ _\infty$	-51%
$\ u_{knee,p}(t)\ _\infty$	+43%
$\ u_{ankle,p}(t)\ _\infty$	-55%

Betrachtet man die Visualisierungen der Laufmuster zu Optipunkt14-15 (Abb. 3.21), so wird nochmals klar, warum diese Bewegungen, die sich durch eine extrem gespannte Wadenmuskulatur auszeichnen, so schnell, aber auch so anfällig sind. Der Schwerpunkt bei diesen Bewegungen liegt extrem weit oben. Mit kleinen Impulsen stößt der Roboter sich fast nur durch die hohe Drehsteifigkeit im Fußgelenk ab, und zieht den anderen Fuß schleifend nach vorne. Im Bodenkontaktschaubild (Abb. 3.22) ist überraschend erkennbar, dass es bei der Konfiguration zu Optipunkt14 alle drei Sorten von Laufphasen gibt: Einbeinkontakt-, Zweibeinkontakt- und Flugphasen. Bei dem hochfrequenten Trippeln des Optipunkt15 entstehen keine Flugphasen mehr, aber sehr wohl kleinere Rutschphasen.

3.5 Fazit

In diesem Kapitel konnte man feststellen, wie wichtig für den Modell-Roboter eine geeignete Justierung der passiv elastischen Parameter des Beins ist. So konnten die Drehmomente an den Hüftgelenken hinab auf die Elastizität der Beine ausgelagert werden.

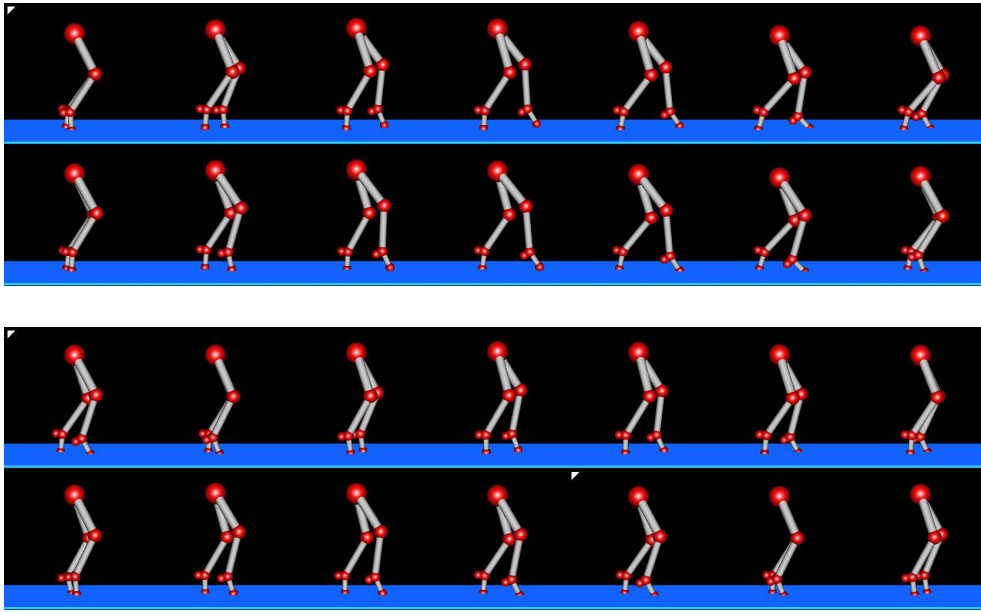


Abbildung 3.21: Laufmuster der Konfigurationen Optipunkt14 (obere Sequenz) und Optipunkt15 (untere Sequenz); Wie eine balancierte Masse auf zwei Federn.

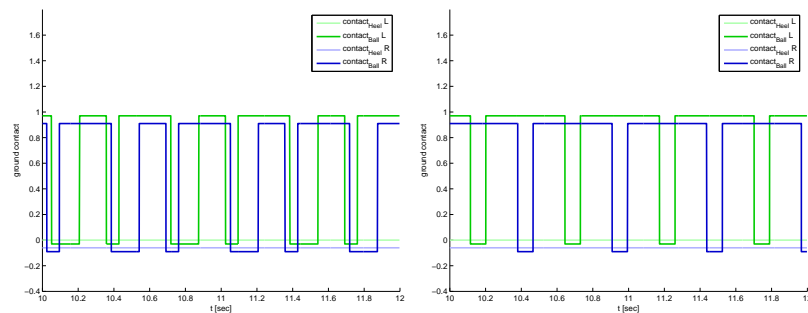


Abbildung 3.22: Bodenkontakt der Konfigurationen Optipunkt14 (links) und Optipunkt15 (rechts);

Das Sprunggelenk kam verstärkt zum Einsatz, während das Knie eher passiv blieb. Der Modell-Roboter benutzte weitgehend nur einen Kontaktpunkt pro federndem Bein, was an die Feder-Masse-Abstraktion erinnert (Abb. 1.3). Die Frequenz wurde tendenziell erhöht und die Schrittweite somit verkürzt. Erfreulich war auch zu sehen, dass trotz vergleichbar niedriger Motordrehzahl sehr hohe Geschwindigkeiten realisierbar sind.

Das Nomad-Verfahren hat sich hierbei als ein robustes Werkzeug erwiesen, das langsam, aber sicher verschiedene Optimierungskriterien auf sehr zerklüfteten Räumen bewältigte. Diese Robustheit wurde allerdings durch eine hohe Anzahl an Zielfunktionsauswertungen erkauft. Somit eignet sich das Verfahren leider nicht für die Optimierung direkt am realen Roboter, denn dort möchte man z.B. schon

nach 50 Auswertungen ein entsprechend gutes Ergebnis haben.

Es bietet sich also an, den realen Roboter zunächst gemäß einer im Modell ermittelten Konfiguration grob einzustellen, dann die Anzahl der Eingabeparameter zu reduzieren (Kontrollraum), und ausgehend von dieser Startlösung eine niedrigdimensionale Feinoptimierung direkt an der Hardware vorzunehmen. Ist die Beschaffenheit des Kontrollraums hinreichend glatt, so können effiziente, gradientenbasierte Optimierungsverfahren angewendet werden und führen schnell zu guten Lösungen. Bei der Untersuchung des Kontrollraums wird dieser Aspekt nochmals behandelt.

Konfiguration p	$v_p(t_f)$	$epm_p(t)$	$\ u_{hip,p}(t)\ _\infty$	$\ u_{knee,p}(t)\ _\infty$	$\ u_{ankle,p}(t)\ _\infty$
Startpunkt1	1.0 $\left[\frac{\text{m}}{\text{s}}\right]$	$4.3 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	400 [Nm]	190 [Nm]	130 [Nm]
Testpunkt1	3.3 $\left[\frac{\text{m}}{\text{s}}\right]$	$1.3 \cdot 10^5 \left[\frac{\text{e}}{\text{m}}\right]$	3500 [Nm]	190 [Nm]	130 [Nm]
Testpunkt2	3.3 $\left[\frac{\text{m}}{\text{s}}\right]$	$3.1 \cdot 10^5 \left[\frac{\text{e}}{\text{m}}\right]$	8200 [Nm]	190 [Nm]	150 [Nm]
Testpunkt3	3.3 $\left[\frac{\text{m}}{\text{s}}\right]$	$3.0 \cdot 10^5 \left[\frac{\text{e}}{\text{m}}\right]$	7500 [Nm]	190 [Nm]	150 [Nm]
Optipunkt1	1.7 $\left[\frac{\text{m}}{\text{s}}\right]$	$3.9 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	720 [Nm]	210 [Nm]	390 [Nm]
Optipunkt2	3.5 $\left[\frac{\text{m}}{\text{s}}\right]$	$9.9 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	2750 [Nm]	210 [Nm]	130 [Nm]
Optipunkt5	3.7 $\left[\frac{\text{m}}{\text{s}}\right]$	$1.3 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	400 [Nm]	150 [Nm]	310 [Nm]
Optipunkt9	2.6 $\left[\frac{\text{m}}{\text{s}}\right]$	$1.2 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	280 [Nm]	170 [Nm]	270 [Nm]
Optipunkt14	4.2 $\left[\frac{\text{m}}{\text{s}}\right]$	$1.9 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	610 [Nm]	70 [Nm]	330 [Nm]
Optipunkt15	2.3 $\left[\frac{\text{m}}{\text{s}}\right]$	$1.6 \cdot 10^4 \left[\frac{\text{e}}{\text{m}}\right]$	300 [Nm]	100 [Nm]	150 [Nm]

 $v_p(t_f)$

Geschwindigkeit nach 100 Sekunden

 $epm_p(t)$

verbrauchte Energie pro Meter

 $\|u_{hip,p}(t)\|_\infty$

max. Ausschlags des Betrags der Drehmomente im Hüftgelenk

 $\|u_{knee,p}(t)\|_\infty$

max. Ausschlags des Betrags der Drehmomente im Kniegelenk

 $\|u_{ankle,p}(t)\|_\infty$

max. Ausschlags des Betrags der Drehmomente im Fußgelenk

Tabelle 3.2: Messdaten zu den Konfigurationen

Kapitel 4

Analyse des Laufverhaltens

4.1 Einführung

Die Laufmuster der im letzten Kapitel präsentierten numerischen Ergebnisse der Optimierung werden hier untersucht und mit den Experimenten am richtigen Roboter verglichen. Eine solche Analyse gibt unter anderem darüber Auskunft, wie gut das Modell die Realität abbildet und wo die Schranken liegen. Erfreulicherweise passen die gewonnenen Erkenntnisse beider Seiten sehr gut zueinander.

4.2 Das Laufverhalten des JW2 Roboters

Nach sorgfältiger Justierung der elastischen Stränge (die GAS, TA, BF und RF simulieren) war der Roboter schon in der Lage, vom Hüftmotor animierte Schrittbewegungen durchzuführen. Interessanter Weise konnten sogar bei Null-Geschwindigkeit Laufmuster erzielt werden, die dem auf der Stelle Gehen eines Menschen sehr ähneln. Die Servomotoren sind auch für das Verändern der Haltung der Beine verantwortlich, z.B. für das Einstellen der Kniebeugung oder der Ausstreckung des Fußgelenks (Sohlenkrümmung).

Mit zunehmender Geschwindigkeit findet der Roboter durch Regelung der Schrittfrequenz und der Amplitude des Hüftoszillators eine entsprechend angepasste Bewegung, ganz ohne die Justierungen der elastischen Fähigkeiten des Beins verändern zu müssen. Bei ca. $1 \frac{m}{s}$ wird die maximale Gehgeschwindigkeit erreicht. Hier kann durch weiteres Erhöhen der Schrittfrequenz f auf Kosten der Amplitude der Übergang zum Rennen erzwungen werden. Zugegeben ist aufgrund der Drehmomentbeschränkung der Servomotoren nur Joggen mit nahezu durchgestreckten Kniegelenken möglich. Um diesen Nachteil zu kompensieren, wird durch Einstellung der GAS-Steifigkeit eine ausgestrecktere Fußhaltung genutzt. So können kurze Flugphasen erzielt werden.

Indem man die Phasenrelation entsprechend verändert, können beidbeinige Hüpfbewegungen realisiert werden. Hierbei agieren beide Kniegelenke gleichzeitig zusammen und generieren genug Kraft, um mit dem unterstützten Körper, sogar bei gebeugten Knien, wiederholt abzuspringen. Das demonstriert das elastische Beinverhalten, welches ebenso Gehen, Joggen, wie auch Hüpfmuster generieren kann.

4.3 Das Laufverhalten des JW2 Simulink-Modells

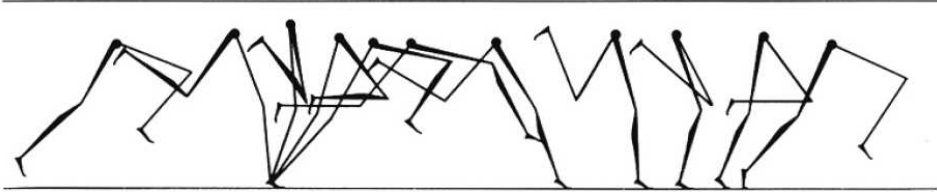
Auch am Modell war ein breites Spektrum an Laufmustern realisierbar, die der Roboter in Bezug auf die Hüftoszillation selbstständig gefunden hat und ziemlich sicher zu konstant bleibender Endgeschwindigkeit und konstantem Energieverbrauch pro Meter konvergiert ist. Maßgeblich für die Basismuster ist hierbei in erster Linie die Frequenz f , der Offset-Winkel α und die Drehzahl des Motors ω , aus denen eine sinusförmige Ansteuerung der Hüftdrehgelenke berechnet wird und somit auch die Amplitude, die im Gegensatz zum realen Roboter nicht direkt als Eingabeparameter zur Verfügung steht. Die übrigen Parameter der elastischen Eigenschaften der Beine kamen vor allem, wie auch beim realen Roboter, im Zuge von Beschränkungen an die Drehmomente in der Hüfte zum Einsatz und wurden von Optimierungsverfahren geschickt ausgenützt. So war auch im Verlaufe der Optimierungsstudien mit zunehmender Geschwindigkeit, aber Einhaltung von Drehmomentbeschränkungen ein Aufrichten des Roboters mit durchgestreckterem Fußgelenk unumgänglich. Während am realen Roboter das Knie bei Laufbewegungen sich innerhalb einer breiteren Winkelspanne beugte und bei Belastungsphasen nahezu durchgedrückt wurde, ist am Modell seine eher passiv federnde Rolle zu beobachten. Bei allen aufgeführten Bewegungsverläufen behält das Knie auch während Belastungsphasen eine gewisse Restbeugung und scheint so die Hüfte weicher zu lagern.

Aufgrund der vom Hüftoszillator vorbestimmten Phasenverschiebung konnte am Modell bis auf beinsynchrone Hüpfbewegungen viele verschiedene Laufmuster imitiert werden. Eine durch das menschliche Joggen motivierte, ergonomische Bewegung mit erstem Bodenkontakt an der Ferse und deutlichem Abrollen bis hin über den Fußballen konnte jedoch mit der derzeitigen Mechanik des Modells bislang nicht erzielt werden. Vielmehr beschränkte sich das Aufkommen der Ferse auf extrem kurze Zeitspannen, in denen die federnde Steifigkeit des Fußgelenks überwunden wurde und es so unbedingt nötig wird, mehr Gewicht abzufangen (z.B. beim Landen nach einer Flugphase).

Auch Abläufe, wie sie beim menschlichen Sprinten mit extrem vorgelagertem Oberkörper zu beobachten sind, werden höchstwahrscheinlich erst durch die Existenz desselbigen ermöglicht und konnten bisher nicht mit der nach wie vor aufrecht geführten Hüfte des Roboters realisiert werden.

Im Kontrast zu schonend langsamen Gehbewegungen (Startpunkt1, Optipunkt1), waren im Model die ersten Ergebnisse mit enormer Geschwindigkeitssteigerung Hopsperläufe (Testpunkt1-3, Optipunkt2) mit langen Flugphasen, die durch unrealistisch hohe Drehmomente erkaufte werden konnten, da an den Antrieben im Modell ja keine direkt physikalische Beschränkung aktiv sind. Aus diesem Grund ist die Implementierung solcher Laufmuster in der Realität entsprechend schwierig bzw. unmöglich. Im Folgenden beschränken wir uns deshalb auch die realistischen Gangarten.

4.4 Vergleichbare Laufmuster



Phase	Hintere Schwungphase	Vordere Schwungphase	Vordere Stützphase	Hintere Stützphase
Beginn	Lösen des Fußes vom Boden	Vertikalmoment	Fußaufsatz	Vertikalmoment
Ende	Vertikalmoment	Fußaufsatz	Vertikalmoment	Lösen des Fußes vom Boden
Funktion	Schwungfunktion – Entspannung – Vorbereitung eines effektiven Kniehubs	Schwungfunktion – Beitrag zur Sicherung der Schrittlänge – Vorbereitung der aktiven Landung	Stützfunktion – Amortisation des Landedrucks – Aufbau von Vorspannung	Stützfunktion – Entwicklung einer optimalen Abdruckkraft
Merkmale	– lockeres, entspanntes Auspendeln nach hinten-oben – Pendelverkürzung und Schwungmassenverlagerung durch Anfersen – maximales Anfersen (bis in Gesäßhöhe) im Moment, wenn Schwungbeinknie das Stützbein überholt	– aktiver schneller Kniehub mit möglichst spitzem Kniewinkel bis etwa 15° unterhalb der Waagerechten – lockeres, optimal weites Vorpendeln des Unterschenkels (bei gleichzeitigem Rücksenken des Oberschenkels) – Rückführen des Unterschenkels zum aktiv greifenden Fußaufsatz	– aktiver Fußaufsatz hoch auf dem Außenrist des Ballens, Fußspitzen nach vorn, – greifende Bewegung, Zugbewegung mit dem Fuß nach hinten, Hüfte schnell über Fußaufsatzpunkt – elastisches Abfangen des Landedrucks, geringes Senken der Ferse ohne Bodenberührung, leichtes Einbeugen im Kniegelenk – möglichst kurze Stützdauer im Vorderstütz	– optimale Streckung im Fuß-, Knie- und Hüftgelenk – Rumpf in leichter Vorlage bis aufrecht – aktive, wechselseitige Armarbeit, etwa rechtwinklig gebeugt, in Laufrichtung – entspannte Rumpf-, Schulter- und Halsmuskulatur – möglichst kurze Stützdauer

Abbildung 4.1: Übersicht über die Phasenstruktur beim Laufschrift ([2])

Beim menschlichen Joggen kann man zwischen vier Phasen unterscheiden: die **vordere** und **hintere Stützphase**, in der ein Fuß Bodenkontakt hat, sowie die **hintere** und **vordere Schwungphase** ohne Bodenkontakt. Man betrachte hierzu das im Schaubild stärker gezeichnete Bein (Abb. 4.1).

Weiter kann man unterscheiden zwischen:

Double-Support-Phase: Beide FüÙe haben Bodenkontakt.

Single-Support-Phase: Nur ein Fuß hat Bodenkontakt.

Flug-Phase: Kein Fuß hat Bodenkontakt.

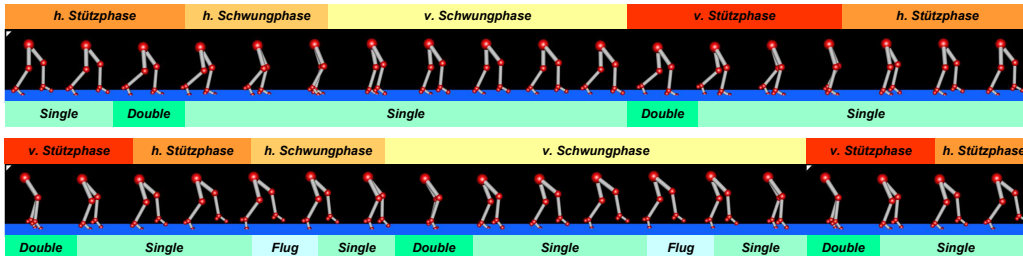


Abbildung 4.2: Phaseneinteilung der Konfiguration Optipunkt1 (oben) und Optipunkt9 (unten) (Referenzbein vorne).

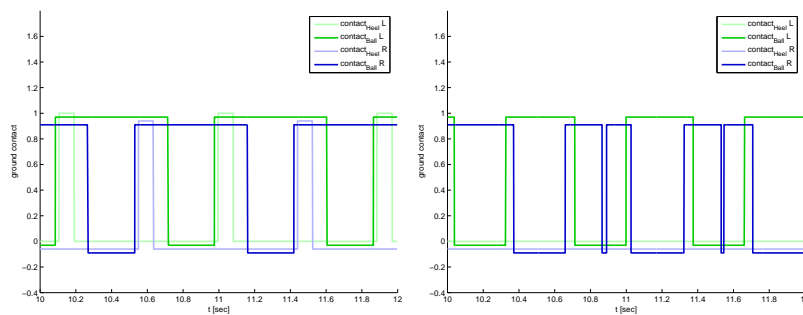


Abbildung 4.3: Bodenkontakt der Konfigurationen Optipunkt1 (links) und Optipunkt9 (rechts);

Im Folgenden werden drei signifikante Laufmuster des realen Roboters mit vergleichbar schnellen Laufbewegungen des Modell-Roboters verglichen, deren Bildersequenzen zur Analyse weiter hinten abgeführt sind. Jeweils sind hier ein bis zwei Perioden der bereits stabilisierten Laufbewegung herausgepickt, deren Beginn durch ein weißes Dreieck in der oberen Ecke eines Einzelbilds gekennzeichnet ist.

4.4.1 Langsames Gehen

Bei diesem Bewegungsmuster (Abb. 4.5) liegt die Geschwindigkeit unter $1 \frac{m}{s}$ (Startpunkt1). Die Schrittweite ist sehr gering und der Gang aufrecht, was für mehr Stabilität sorgt. An der Bildsequenz des realen Roboters kann man gut den ziehenden Fußaufsatz erkennen. Nachdem die Ferse zuerst Bodenkontakt hat, wird jetzt über den Mittelfuß sauber abgerollt und als letztes verlässt der Fußballen den Boden. Im Kontrast dazu fehlt beim Modell noch die explizite TA-

Muskelgruppe und somit auch das Hochziehen der Fußspitze vor dem Auftritt. Der erste Bodenkontakt kommt also mit dem Vorderfuß zustande und die Ferse tippt kurz auf, wenn sich der Schwerpunkt des Körpers über dem Auftrittspunkt befindet, wie Kontaktverläufe beider Fußpunkte zeigen. Danach verlässt der Vorderfuß, wie gewohnt, als letztes den Boden.

Während der Belastungsphasen ist zu beobachten, dass das Standbein des realen Roboters voll ausgestreckt ist, um das Gewicht tragen zu können. Dagegen hat das Modell eher die Tendenz, mit stets leicht angewinkelt Knie zu laufen und so liegt besonders in der vorderen Stützphase der Schwerpunkt vergleichbar tief. In der hinteren Stützphase ist das belastete Bein am meisten durchgestreckt, während das andere Bein sich noch in der vorderen Schwungphase befindet.

4.4.2 Schnelles Gehen

Bei diesen bereits optimierten Bewegungssequenzen (Abb. 4.6) ist die Schrittweite deutlich größer geworden und die Geschwindigkeit entsprechend höher (im Modell über $1.5 \frac{m}{s}$; Konfiguration Optipunkt1). Am Bildverlauf des realen Roboters ist zu erkennen, dass der Auftrittspunkt weit vor dem Schwerpunkt erfolgt. Auffallend ist die forciert kickende Bewegung des Schwungbeins und das anschließende tapsige Auftreten. Unmittelbar nach dem ersten Bodenkontakt der Ferse steht der gesamte Mittelfuß auf dem Grund und das immer noch weit vor dem Schwerpunkt. Die Laufbewegung des Modellroboters wirkt etwas behäbiger, aber runder. Wie bei langsamen Gehen auch, tippt im Modell der Hinterfuß nur kurz bei stärkerer Belastung auf (Abb. 4.3). Ansonsten wird diese Bewegung vom Auftreten mit dem Fußballen und dem Abfedern durch das Sprunggelenk bestimmt. Der Hauptunterschied wird gut in der Neigung des Unterschenkels deutlich. Von dem seitlichen Betrachtungspunkt (Sagittalebene) aus gesehen, bleibt der Unterschenkel des Modells durchgehend nachhängend rechtsschief, während der des realen Roboters im Auftrittspunkt stark linksschief ist. Das unterstreicht wieder die stark passive Arbeitsweise des Modellknies.

4.4.3 Elastisches, schnelles Laufen

Während am Modell sehr lange, teils ungewollte Flugphasen realisierbar waren, so sind die Flugphasen am realen Roboter aufgrund der Leistungsgrenze der Motoren noch relativ kurz. Bei den Bewegungssequenzen für das schnelle Laufen (Abb. 4.7) wurde bewusst eine realisierbare Konfiguration des Modells ausgewählt (Optipunkt9), die bezüglich der Energie minimiert wurde. So wurde die Geschwindigkeit des Modellroboters auf gute ca. $2.5 \frac{m}{s}$ gedrosselt. Bis auf die Tatsache, dass die Beine des realen Roboters, Schwungbein und Standbein, wieder komplett durchgestreckt werden, und im Modell stets die gewohnte Restkrümmung beibehalten wird, sind sich die Bewegungen erfreulicher Weise sehr ähnlich. Die

Kniegelenke müssen beim realen Roboter weitgehend durchgedrückt sein, um das Gewicht tragen zu können und die Muskelstränge am Knie nicht überzubelasten. Mit dieser Verlagerung auf die starren Elemente (bzw. Knochen) des Beins, wirken sich die Stöße auf die Hüfte bei schnellen Bewegungen verstärkt aus. Um diesen Nachteil wieder zu kompensieren, hat man bis jetzt das im Sprunggelenk verbleibende elastische Potential ausgenutzt, indem man, wie bereits erwähnt die GAS-Muskulatur spannt, um eine ausgestrecktere Fußhaltung zu erzielen. Positiv überraschend war: unabhängig von dieser Idee, verhielt sich das Modell gleich und der Optimierer nahm im Zuge der Drehmomentbeschränkungen entsprechende Konfigurationen vor. So sind beide Bewegungsverläufe sehr rücklastig und verdanken besonders der Steifigkeit des Fußgelenks ihren Schub nach vorne. Im Vergleich zu den behandelten Gehbewegungen sind die Stützphasen deutlich kürzer gegenüber der jetzt langen vorderen Schwungphasen geworden (Abb. 4.2) und beschränken sich auf federnde Impulse mit dem Vorderfuß. In Realität und Modell treten bei dieser Bewegung jeweils kurze, teils unregelmäßige Flugphasen auf (Abb. 4.3).

4.5 Tendenzanalyse

Nach den bisherigen Studien zeichnet sich im Modell deutlich die Tendenz ab, den Schwerpunkt möglichst hoch zu lagern und mit energisch federnden, kurzen Abstößen des Ballens, anstatt mit vollem Mittelfußzug, über den flachen Boden hinwegzugleiten (Abb. 4.4).



Abbildung 4.4: Mögliche Vergleiche? ([7], [3])

Was könnten die Gründe für dieses Verhalten sein? Wer die Abrollbewegungen des menschlichen Joggens am virtuellen JW2 erwartet hat, der wird in diesem Modellstadium noch enttäuscht. Aufgrund der fehlenden TA-Muskelgruppe (und deren evtl. explizite Aktuation) scheint kein ziehender Fußaufsatz und somit auch kein Auftritt mit der Ferse unter verkraftbaren Drehmomenten möglich zu sein. Der Fuß wird durch die hohe Drehsteifigkeit in seinem weiten Ruhewinkel stabilisiert, hängend hinterher gezogen. Es hätte also höchstwahrscheinlich keinen Sinn gehabt, mittels einer Schar von Nebenbedingungen an den Optimierer unter hohem Aufwand eine mit der aktuellen Mechanik vielleicht unrealisierbar unwirtschaftliche Bewegung, die dem menschlichen Abrollen ähnlich ist, zu erzwingen.

Weiter sitzt beim Menschen der Antrieb natürlich nicht nur in der Hüfte. Das längere Aufstellen auf den Fußballen ist nur mit anstrengendem Kraftaufwand möglich und führt zu einer, für den Menschen ungewohnten, Belastung des Sprunggelenks, die beim Joggen nicht empfehlenswert ist. Bei den bisherigen Optimierungsstudien wurden aber, wie mehrmals erwähnt, stets nur die Momente der Aktuatoren an den Hüftgelenken in das Kriterium eingebunden, während die deutlich sichtbare Erhöhung der Drehmomente in den Fußgelenken unbestraft blieb.

Da der Roboter in der Hüfte aufrecht gehalten wird und mittels eines z.B. verlagerbaren Oberkörpers noch keine richtige Vorlast erzeugen kann, wurden energische Bewegungen, wie sie beim Rennen oder Sprinten entstehen, noch nicht erreicht. Wenn ein Mensch wiederum versucht, sich vertikal/rücklastig möglichst schnell fortzubewegen, dann entstehen genau die auch bei der Optimierung beobachteten Laufmuster (z.B. Hopseläufe; rücklastiger Ballenlauf mit längeren, vorderen Schwungphasen). Die Optimierung war, bzgl. der ausgewählten Kriterien in Betracht auf das aktuelle Modellstadium, recht erfolgreich. Parallelen zu den signifikanten Änderungen an der Drehsteifigkeit des Sprunggelenks (ab Optipunkt5) sowie an den Wadenmuskelgruppen GAS (ab Optipunkt14) mit zunehmender Laufgeschwindigkeit waren auch bei den Experimenten des realen Roboters entsprechend nachvollziehbar.

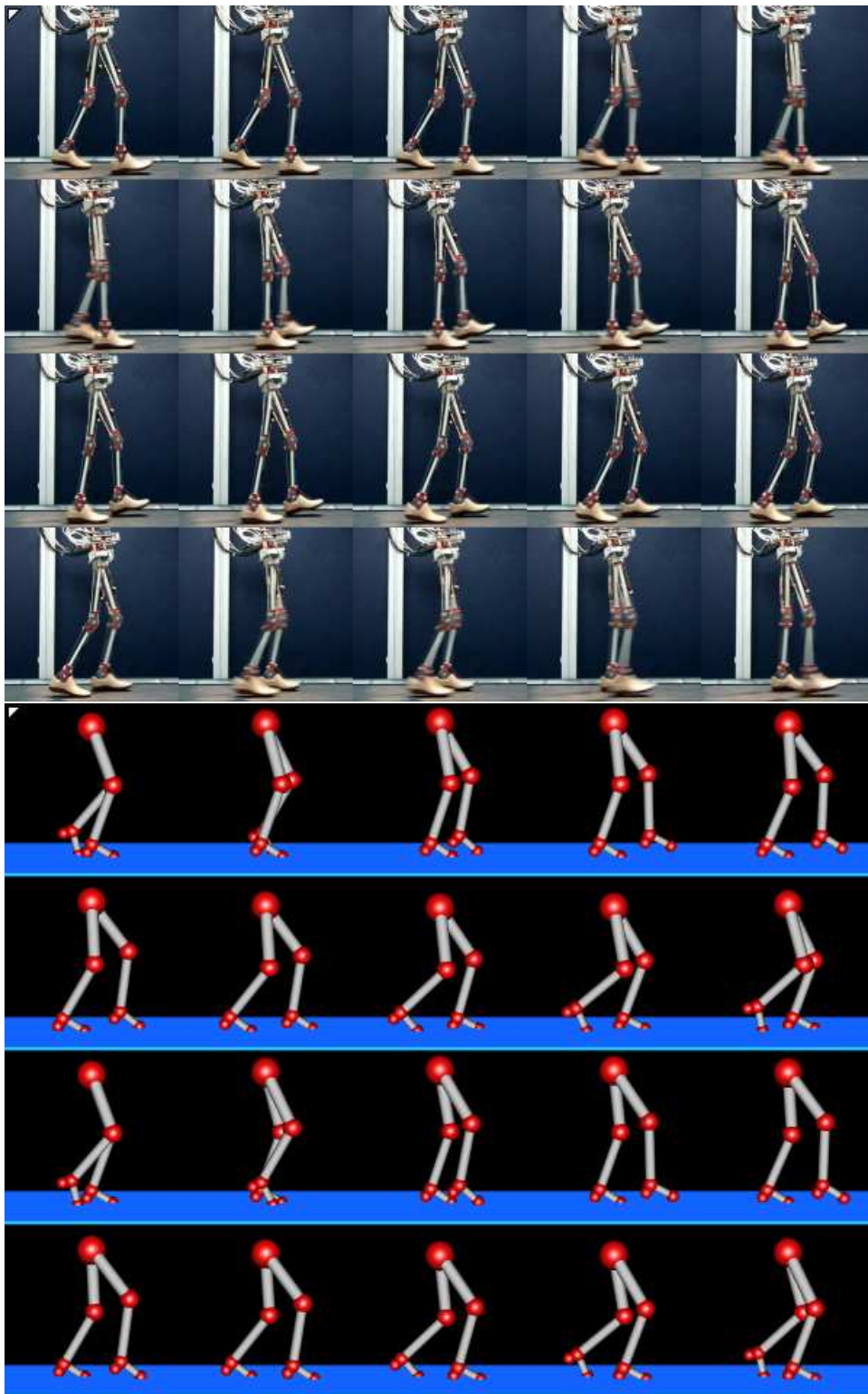


Abbildung 4.5: Langsames Laufmuster JW 2 im Vergleich zum Laufmuster des Modells zur Konfiguration startpunkt1

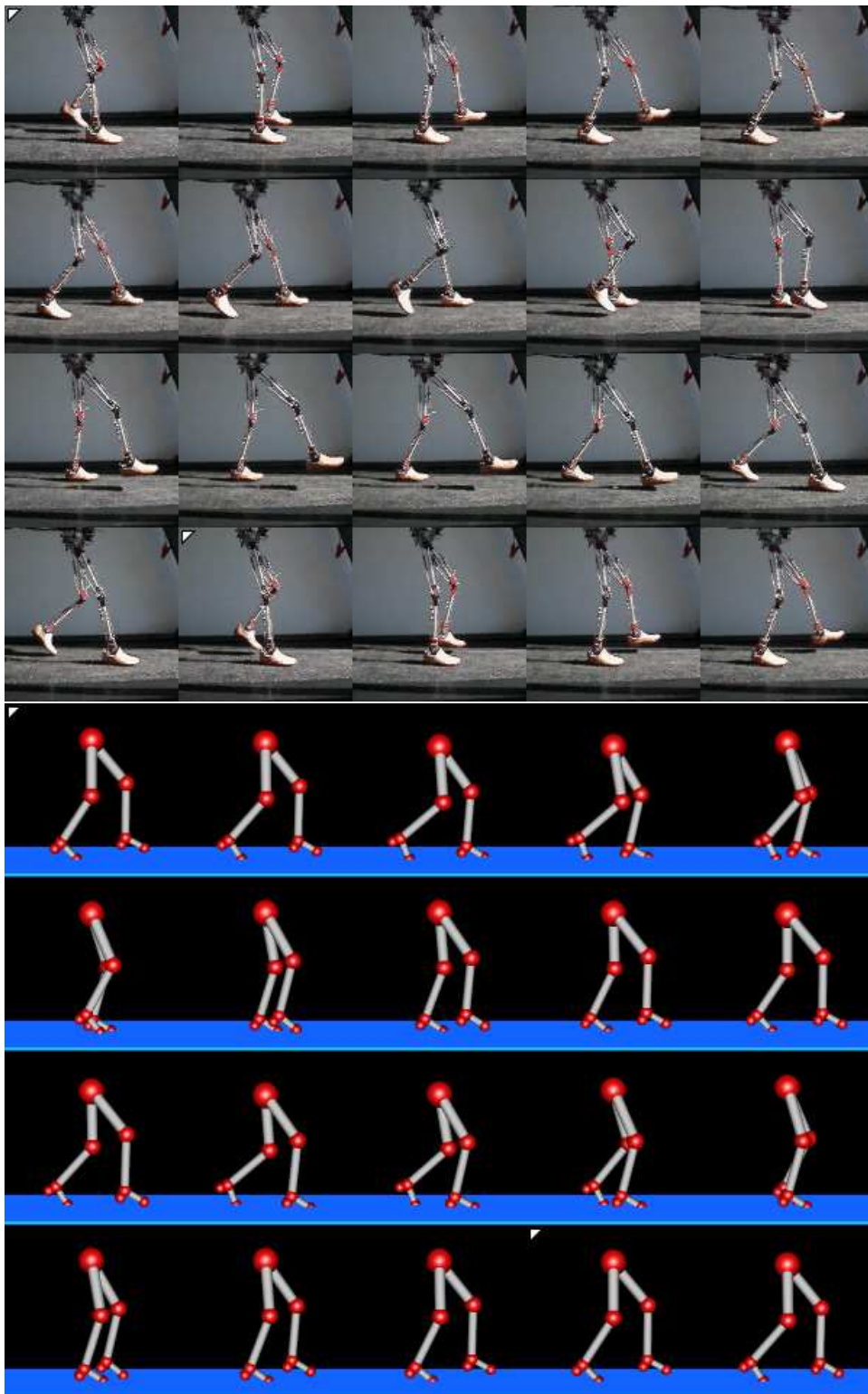


Abbildung 4.6: Gemäßigtes Laufmuster JW 2 im Vergleich zum Laufmuster des Modells zur Konfiguration optipunkt1

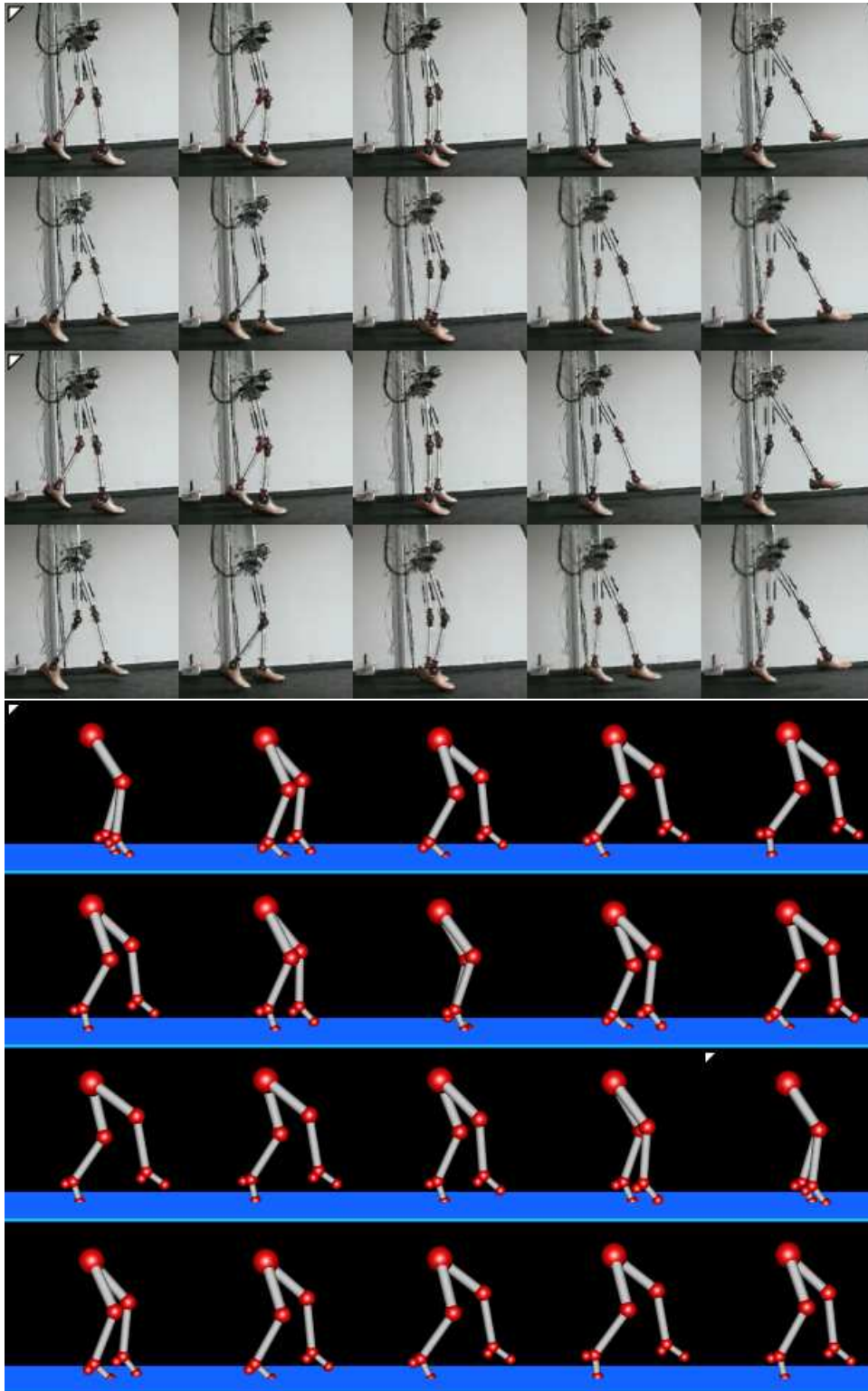


Abbildung 4.7: Schnelle Laufmuster JW 2 im Vergleich zum Laufmuster des Modells zur Konfiguration optipunkt9

Kapitel 5

Visualisierung des Kontrollraums

5.1 Einführung

Für die praktische Analyse am richtigen Roboter ist besonders der sogenannte Kontrollraum von Interesse. Wenn der Roboter mit fest installierten Motoren (feste max. Drehzahl ω) ausgestattet ist, und die aufwendigere Justierung der elastischen Eigenschaften p, c, d der Beingelenke und Muskeln bzw. Federn getätigt wurde, so beschränkt sich der Spielraum der Eingabeparameter auf die Frequenz f und den Offset-Winkel α . Hier möchte man jetzt erfahren, wieviel verschiedene Laufmuster ohne Komplikationen zu riskieren, wie z.B. Zusammenbrüche des Roboters und folgender Beschädigung oder ungewollter Verschiebung der Justierungen, allein durch das Variieren der beiden Eingabeparameter möglich sind.

Es wurden in diesem Zusammenhang mehrere Konfiguration untersucht. Im Folgenden werden die Ergebnisse zu Startpunkt1 und Optipunkt9, deren verschiedene Bewegungsabläufe im letzten Kapitel untersucht wurden, präsentiert und miteinander verglichen.

5.2 Kontrollraum der Konfiguration Startpunkt1

Wie auch im Kapitel zur Optimierung eine rechenaufwändige Auswertungsreihe der Geschwindigkeit v in Abhängigkeit von Frequenz f und Drehzahl ω angefertigt wurde, so sei es auch an dieser Stelle getan, mit dem einzigen Unterschied, dass die Frequenz hier von dem Offset-Winkel α als zweite Eingabe verdrängt wurde.

Im Schaubild (Abb. 5.1) sind die wichtigsten Kriterien dargestellt: die Geschwindigkeit, sowie die verbrauchte Energie pro Strecke. Man könnte annehmen, dass diese beiden Kriterien gegenspielerische Wirkung aufeinander haben, nach dem Motto: "Je schneller ich sein will, desto mehr Energie verbrauche ich". Aber

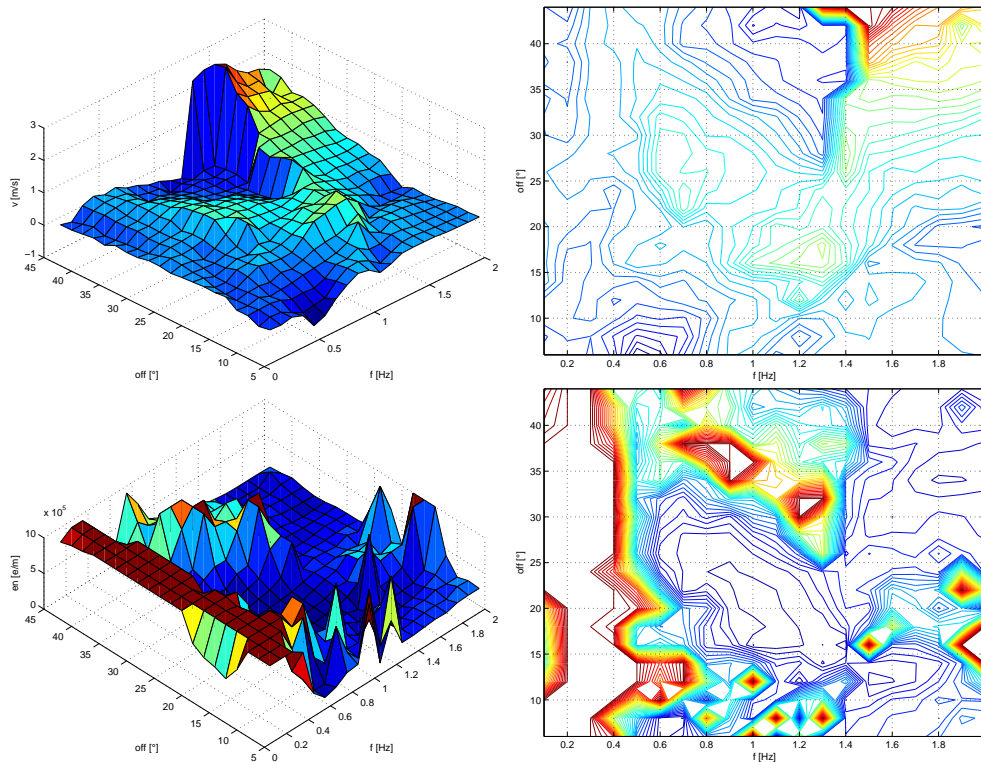


Abbildung 5.1: Entwicklung des f - α -Kontrollraums zu Startpunkt1: Geschwindigkeit v (obere Reihe) und verbrauchte Energie pro Strecke epm (untere Reihe) jeweils in Abhängigkeit von der Frequenz f und dem Offset-Winkel α .

wie die Schaubilder nahebringen, liegen beide Kriterien eng beieinander und ziehen sogar zusammen in die selbe Richtung am einem Strang.

Die Visualisierung der Geschwindigkeit zu Konfiguration Startpunkt1 zeigt eine bergige, aber nicht zerklüftete Landschaft. Folgt man dem grünen Pfad nach oben, so erreicht man auf der roten Spitze nach 10 Sekunden bereits Geschwindigkeiten über $2.5 \frac{m}{s}$. Der Roboter findet hierbei selbständig neue Laufmuster. Das ganze ist jedoch eine Gratwanderung, wie man unschwer erkennen kann. Bei ausreichend hoher der Frequenz (ca. 1.5[Hz]) kann durch sukzessives Erhöhen des Offset-Winkels proportional zunehmende Geschwindigkeit beobachtet werden. Fällt die Frequenz f aber bei hohem Offset-Winkel α , also extremer Schiefelage des Roboters, unter eine gewissen Mindestwert (hier: ca. 1.4[Hz]) so kommt es zu einem Kollaps, wie der steile Abhang im Schaubild zeigt.

Betrachtet man dazu die verbrauchte Energie pro Wegstrecke, so fallen in der relativ stark zerklüfteten Landschaft flache, tiefe Ebenen auf, und zwar erfreulicher Weise genau an den Stellen, wo der Roboter ein stabiles Laufmuster mit höherer Geschwindigkeit findet. Durch die einfache Tatsache, dass der Roboter bei diesen schnelleren Laufbewegungen in kurzer Zeit viel Weg hinter sich bringt,

und nur eine niedrige Erhaltungsenergie braucht, um das Muster zu stabilisieren, fällt der Quotient $\frac{\text{Energie}}{\text{Weg}}$ so gering aus. Umgekehrt, an den Stellen, an denen der Roboter sich langsam fortbewegt und hohe Reibung verursacht, z.B. wenn er kollabiert ist und mit hohen Drehmomenten weiterkriecht, entstehen generell viele vereinzelt hohe Peaks an verbrauchter Energie.

(Energiewerte über $10 \cdot 10^5 [\frac{e}{m}]$, wie sie bei Frequenzen unter $0.4[\text{Hz}]$ entstehen, wurden bewusst abgeschnitten (dunkelrote, hohe Ebene) um eine günstigere Skalierung beizubehalten. Man kann davon ausgehen, dass das Model in diesem uninteressanten Bereich nicht realistisch arbeitet und sollte für Optimierungen mittels Nebenbedingungen den Spielraum der Frequenz entsprechend einschränken.)

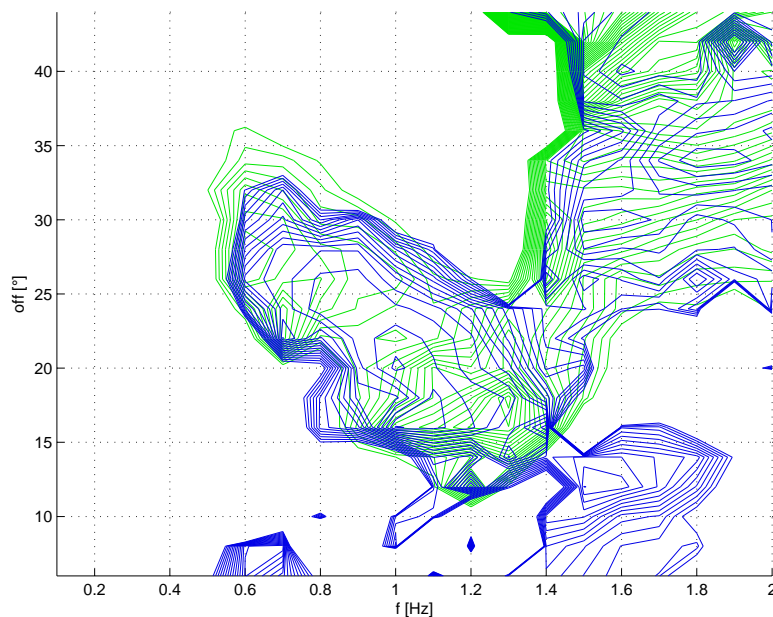


Abbildung 5.2: f - α -Kontrollraum zu Startpunkt1

Legt man so die Höhenlinien in die gewollten Bereiche, hier z.B. $v \in [0.4 \frac{m}{s}, 4 \frac{m}{s}]$ und $epm \in [0.0 \cdot 10^5 \frac{e}{m}, 2.0 \cdot 10^5 \frac{e}{m}]$, so wird der entsprechende Kontrollraum (Abb. 5.2) sichtbar. Alle f - α -Werte innerhalb des blau-grün schraffierten Bereichs garantieren jetzt, zusammen mit den voreingestellten Werten der Konfiguration Startpunkt1, eine sichere Simulation mit guten, vorhersehbaren Werten.

5.3 Kontrollraum der Konfiguration Optipunkt9

Analog zu den vorherigen Untersuchungen sei jetzt die Konfiguration Optipunkt9 behandelt. Betrachtet man die fixierten Parameter, d.h. die feste Drehzahl und die voreingestellten, elastischen Justierungen an den Beingelenken, so sind diese bereits mit dem Nomad-Verfahren optimiert worden. Auskunft über die Beschaf-

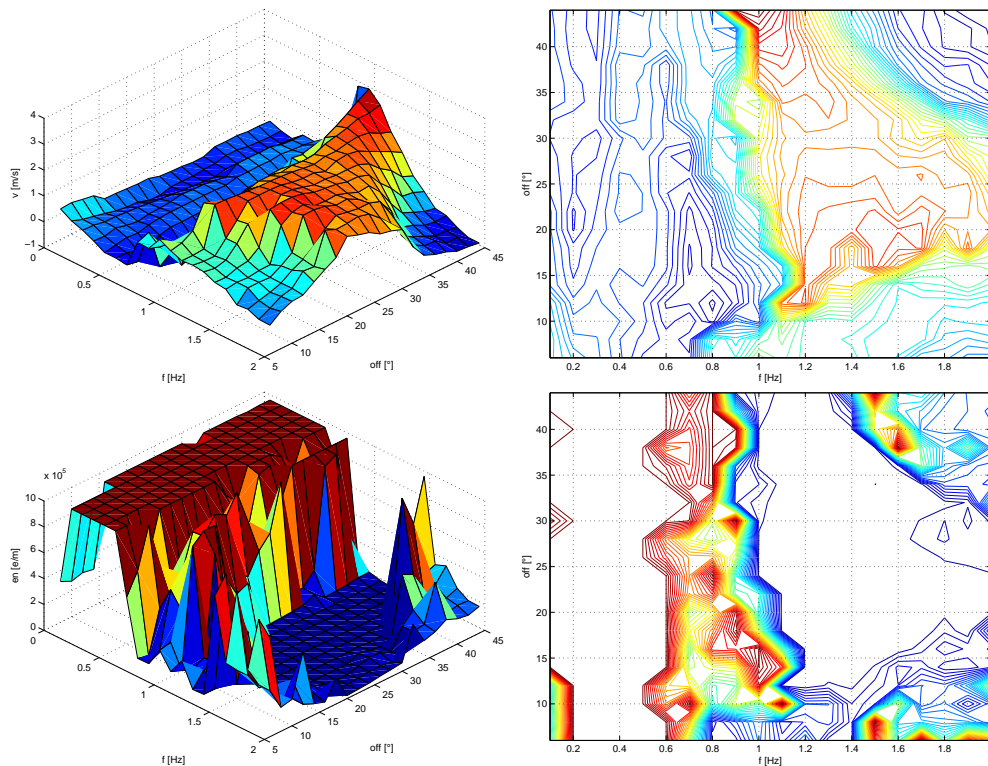


Abbildung 5.3: Entwicklung des f - α -Kontrollraums zu Optipunkt9: Geschwindigkeit v (obere Reihe) und verbrauchte Energie pro Strecke epm (untere Reihe) jeweils in Abhängigkeit von der Frequenz f und dem Offset-Winkel α .

fenheit dieses Optimums und seiner unmittelbaren Umgebung liefert der f - α -Kontrollraum.

Betrachtet man das Schaubild zur Geschwindigkeit (Abb. 5.3), so fällt sofort die wesentlich einfachere Form auf (wobei auch der andere Blickwinkel zu beachten ist, der besonders aufgrund der Beschaffenheit des Energieraums so gewählt wurde). Man erkennt deutlich die beiden Ebenen (blau $v \in [0.0 \frac{m}{s}, 0.4 \frac{m}{s}]$ und rot $v \in [2.0 \frac{m}{s}, 3.0 \frac{m}{s}]$), die wie zwei Schichten übereinander liegen. Bei dem Schaubild zur verbrauchten Energie ist neben der durch Filterung unrealistischer Werte künstlich entstandenen Hochebene die durchgängig tiefblaue, charakteristische Färbung des Tals auffällig. Diese Talebene ist im Vergleich zu Startpunkt1 sogar noch flacher und tiefer.

Es scheint mit der Konfiguration Optipunkt9 also ein relativ ausgeglichenes, breites Optimum gefunden zu sein, das neben dem Aspekt seiner Wirtschaftlichkeit auch eine Verbesserung der Stabilität mit sich bringt. So kann man z.B. bei gleichbleibender Justierung der Bein elastizität und Dämpfung zu einer Frequenz von ca. 1.2[Hz] hier den Offset-Winkel im vollen Bereich 10° - 50° ändern, ohne dabei ein Umfallen des Roboters zu verursachen.

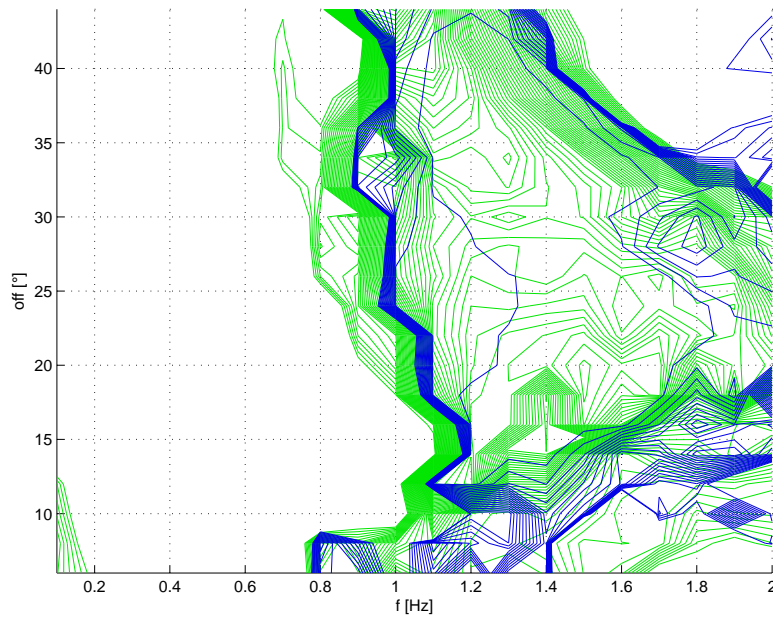


Abbildung 5.4: f - α -Kontrollraum zu Optipunkt9

Verschiebt man auch hier die Höhenlinien in die gewollten Bereiche, z.B. $v \in [0.4 \frac{m}{s}, 4 \frac{m}{s}]$ und $epm \in [0.0 \cdot 10^5 \frac{e}{m}, 2.0 \cdot 10^5 \frac{e}{m}]$, so wird der entsprechende Kontrollraum (Abb. 5.4) sichtbar. Dieser ist im Vergleich zum vorherigen Beispiel zusammenhängender und strenger begrenzt. Mit den elastischen Beineinstellungen aus Optipunkt9 ist innerhalb dieses Kontrollraums eine relativ enge Spanne von vorbestimmt höheren Geschwindigkeiten realisierbar. Auch was die verbrauchten Energien hierbei angeht, so unterscheiden diese sich nur unmerklich von einander. Um die Mitte des Kontrollraums ($f = 1.4[\text{Hz}], \alpha = 25^\circ$) ist die Laufbewegung sehr robust und konstant, verlässt man aber den flachen Kontrollraum, so riskiert man offensichtlich umgehend einen Zusammenbruch des Laufapparats.

5.4 Ausblick

Diese Erkenntnisse legen nahe, dass die Geschwindigkeit und die verbrauchte Energie pro Strecke in brauchbarer Weise zusammenhängen. D.h. auf der Suche nach dem wirtschaftlichsten Laufmuster könnte sogar ein unbeschränktes Optimierungskriterium durchaus gute Ergebnisse liefern. Innerhalb des erforschten Kontrollraums könnte man so alleinig die verbrauchte Energie pro Strecke minimieren. Aufgrund der glatten Beschaffenheit würden sich hierzu sogar weniger robuste, aber schnellere Optimierungsverfahren eignen (Gradientenapproximation, Ersatzfunktion). Der Suchraum eines auf die beiden Kriterien (v, epm) angelegten Optimierers kann dann auch effektiv eingeschränkt werden. So könnten

auch professionelle, umfangreiche Optimierungsläufe direkt am realen Roboter erfolgen, ohne zu viele Gütefunktionsauswertungen zu riskieren.

Kapitel 6

Analyse zur Störungsanfälligkeit

6.1 Einführung

Zum Abschluss der Arbeit ist es angebracht, die Robustheit der optimierten Laufmuster zu demonstrieren.

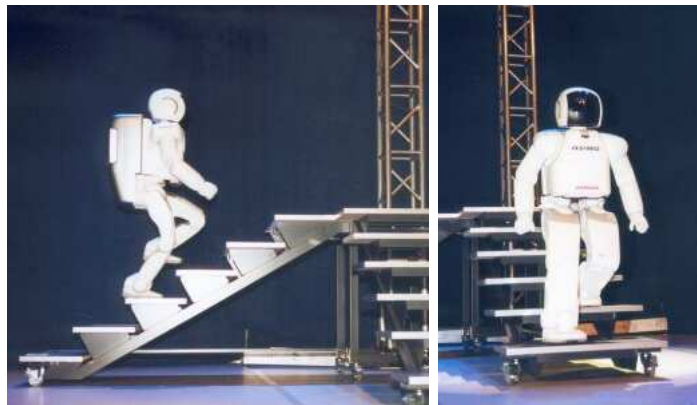


Abbildung 6.1: Asimo von Honda beim Treppensteigen ([8]).

Der Asimo Roboter kann durch an den Untergrund angepasste Kontrollalgorithmen Treppen steigen (Abb. 6.1). Das gilt zweifellos als ein schwieriges Können in Roboterkreisen. Die Stufen haben aber fest vorbestimmte Ausmaße und werden in der Regel von der Firma Honda zu jeder Präsentation mitgenommen. So gehört, wie zum Menschen das passende Paar Schuhe, die präzise abgemessene Treppe zum Roboter. Doch die eigentliche Kunst sollte doch darin bestehen, sich flexibel an seine Umgebung anpassen zu können. Das erfordert von vorne herein das Einplanen von Toleranzen. Je komplexer aber die Bauteile, Freiheitsgrade der Gelenke und die speziellen Feedback-Kontrollalgorithmen werden, desto schwieri-

ger ist diese Aufgabe. Deswegen wird in vielen Fällen alles getan, um Toleranzen weitgehend zu eliminieren und auf Präzision zu setzen.

Während elastische Bein-Designs bei präzisen Positionsberechnungen ihre Nachteile offenbaren, so weisen sie ohne aktives Einwirken von außen schon eine gewisse Flexibilität auf, die es ihnen ermöglicht, kleinere Störungen spielerisch zu kompensieren. Wenn z.B. die Treppenstufe für den unternommenen Schritt etwas zu hoch ist, so stößt der Roboter mit den starren Strukturen hart dagegen, schiebt sich vielleicht noch zurück und fällt im schlimmsten Fall um, während der Roboter mit dem elastischen Design vielleicht in der Lage ist, dank eines flexiblen Sprunggelenks den Fuß schnappend über die Stufenkante hinwegzuziehen.

Im Folgenden werden deshalb entsprechende Szenarien am Modell des Jena Walker 2 untersucht, die den Vorteil der Elastizität in Bezug auf die Störungsanfälligkeit demonstrieren sollen.

6.2 Szenarien

Mögliche Störungen des Laufmusters von außen können über plötzlich direkt an den Strukturen des Roboters erzeugte Kräfte erfolgen, z.B. ein Stoß auf die Hüfte, der den Roboter aus dem Gleichgewicht bringen soll. Im nächsten Abschnitt werden Störungen in Form von Bodenunebenheiten erzeugt, d.h. die Stöße von außen beschränken sich also auf die Kontaktpunkte der Füße des Roboters. Mittels einer Funktion, welche die Bodenstruktur verkörpert, wird der Auftrittspunkt im Kontaktmodell modifiziert, je nach dem wird schon früher oder später eine der Bodenkonsistenz entsprechende Gegenkraft auf den Fuß erzeugt. Hierzu wurde auch eine effiziente Visualisierung gebaut, die fähig ist, Bodenfunktionen jeder Art darzustellen und sich wesentlich auf die 2-dimensionale Ebene des seitlich mitgehenden Betrachters beschränkt.

Experimente, die eine Reihe von Zufallsszenarien unterschiedlicher Bodenbeschaffenheiten beinhalteten, zeigten überraschend, dass nicht der feste Stand auf beiden Fußpunkten und der anschließende, volle Mittelfußzug robust ist, sondern eher ein kurzes, intensives Auftreten des Vorderfußes, das den Impuls zum Schub in die richtige Richtung gibt. So rutschten Laufmuster der Konfigurationen Startpunkt1 oder Optipunkt1 bei Steigung regelmäßig ab. Beim Fußball sollte man während des Schusses nicht in Rücklage geraten, sonst zieht man Bein und Fuß zu hoch und der Ball fliegt über das Tor. Bei der kickenden Laufbewegung zu Optipunkt9 wird die Rücklage durch den hohen Offset-Winkel α verursacht, aber anders als beim Fußball hat das hier einen positiven Effekt. Beim Schreiten werden die Füße sehr hoch angehoben und somit das Erklimmen von höheren Ebenen erst möglich gemacht.

Im Folgenden sei deshalb hauptsächlich das robuste Laufmuster der Konfiguration Optipunkt9 untersucht.

6.2.1 Treppensteigen

Mit geeigneten Parametereinstellungen ist es dem ca. 1m großen Modell-Roboter möglich, Treppenstufen der Höhe 10cm spielend zu bewältigen (Abb. 6.2). Die Elastizität spielt hierbei eine große Rolle und verhindert mit, dass der Roboter nach hinten umkippt. Denn nach dem Auftreten auf einer höheren Stufe, tritt der Laufapparat kurzzeitig auf der Stelle, bzw. die Schrittweite wird automatisch verkleinert, bis sich die Beinhaltung wieder entspannt, an das Muster gewöhnt und wieder Schub aufgenommen werden kann. Somit ist also das Erklimmen von höheren Ebenen ohne spezielle Modifikation der Roboterkontrollen möglich. Allerdings sollten die Stufen nicht zu kurz aufeinander folgen, sonst ist die Rhythmusstörung zu stark bzw. zu langanhaltend, um kompensiert zu werden.

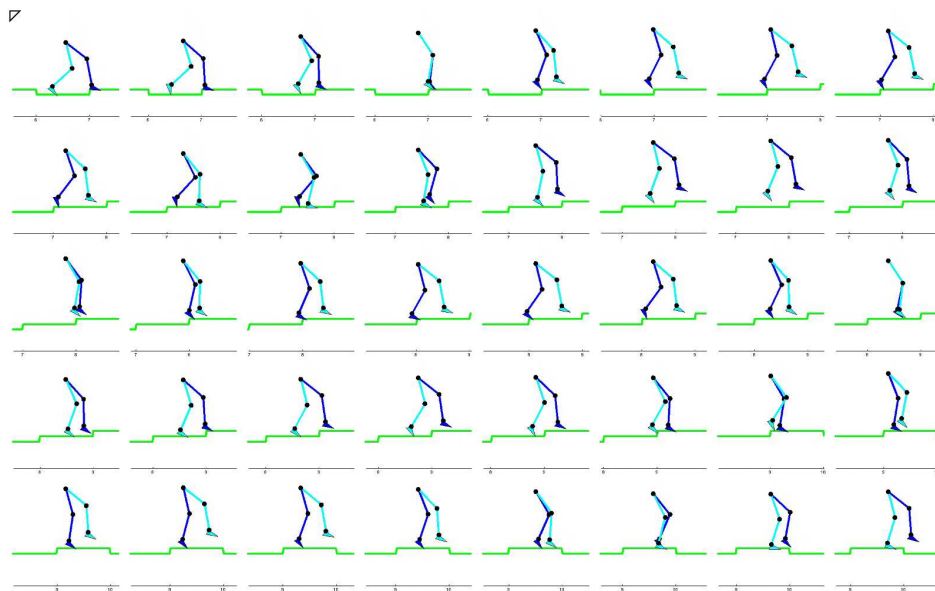


Abbildung 6.2: Mit dem Laufmuster der Konfiguration Optipunkt9 können 10cm hohe Treppenstufen überwunden werden.

6.2.2 Hindernisblock

Selbst 20cm hohe Hindernisblöcke können unter großen Drehmomenten überwunden werden (Abb. 6.3). Die Blockposition sollte allerdings ungefähr an den Schrittrhythmus der Konfiguration angepasst werden, denn die Elastizität, die beim Bewältigen der Stufenkante gebraucht wird, hat auch ihre Grenzen. Der Roboter kann nur den Vorderfuß nach einer eventuellen Kollision mit der Kante schnell genug nachziehen, aber natürlich nicht den gesamten Unterschenkel.

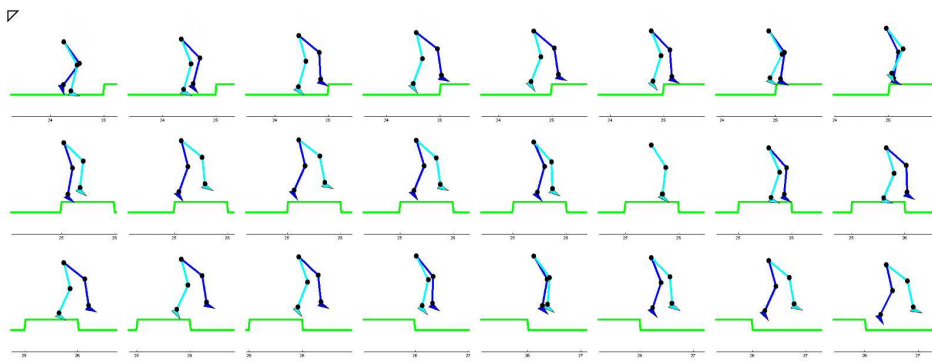


Abbildung 6.3: Ein 20cm hoher Block wird überwunden (Optipunkt9)

6.2.3 Sinuslaufen

Anders als zu den vorherigen Versuchen, in denen es der Roboter nur mit unterschiedlich hohen waagrechten Ebenen zu tun hatte, kommen jetzt abwechselnd Steigung und Gefälle ins Spiel. Insbesondere bedeutet das, dass die Auftrittshöhe von Schritt zu Schritt immer unterschiedlich ausfällt und der Roboter somit keine Gelegenheit bekommt, sich auf das für die ursprünglich waagrechte Ebene optimierte Laufmuster einzupendeln.

Die noch ungedrosselt schnelle und flache Laufbewegung zur Konfiguration Optipunkt5 überwindet zwar den ersten Hügel, aber nach dem Tiefpunkt im Tal, bei erneutem Anstieg, dort wo die Kräfte auf die Beine am stärksten werden, sackt die Laufmaschine zusammen (Abb. 6.4). Der Roboter kann aufgrund der Steigung hierbei nicht schnell genug die Füße nach vorne ziehen, und so wird ihm seine hohe Geschwindigkeit zum Verhängnis.

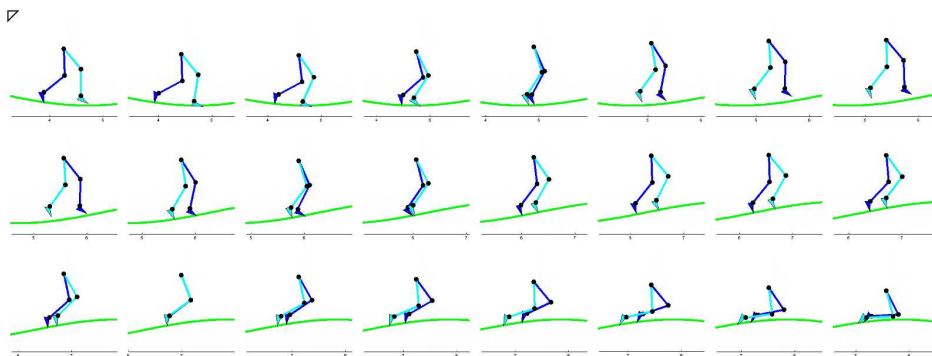


Abbildung 6.4: Eine abgeflachte, sinusförmige Bodenfunktion $y = \frac{1}{5} \cdot \sin(x)$ (Einheit: Meter) wird vom Roboter mit Konfiguration Optipunkt5 abgelaufen.

Im Vergleich dazu bewältigt das Laufmuster zur Konfiguration Optipunkt9 die noch relativ flachen Hügel mit fast spielender Leichtigkeit (Abb. 6.5). Es ist erkennbar, dass der Modell-Roboter bei seinem Weg ins Tal kurze Flugphasen un-

ternimmt. Nach Überwinden des Tiefpunkts hat die Ferse sogar Bodenkontakt, um das Gewicht abzufangen. Wegen der hohen Drehsteifigkeit im daher runtergedrückten Sprunggelenk folgen anschließend explosive Schritte bergaufwärts. Hier kommt dem Roboter sein Rückenlage zugute und er schafft es rechtzeitig, einen Fuß vor den anderen zu setzen.

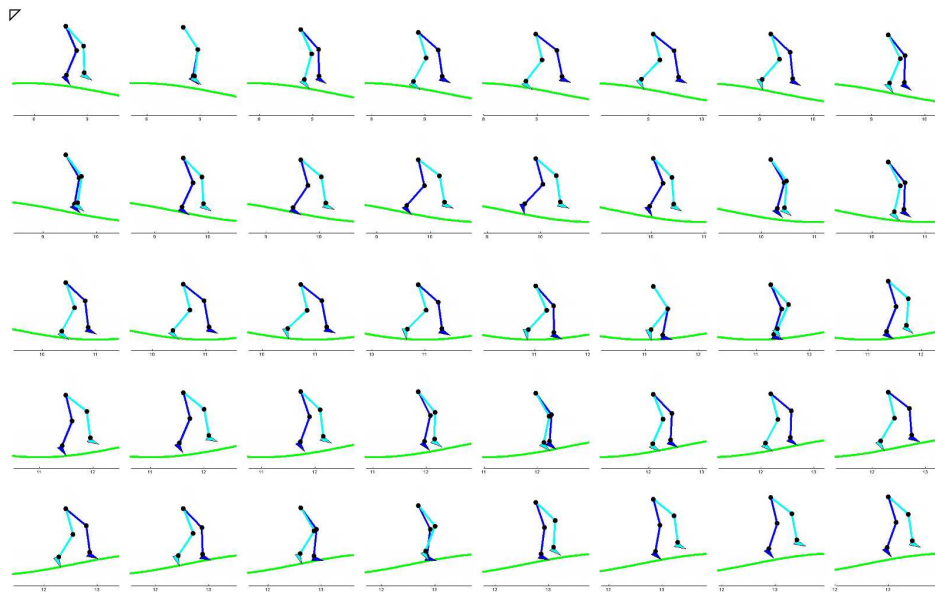


Abbildung 6.5: Eine abgeflachte, sinusförmige Bodenfunktion $y = \frac{1}{5} \cdot \sin(x)$ (Einheit: Meter) wird vom Roboter mit Konfiguration Optipunkt9 abgelaufen.

Erhöht man die Amplitude der sinusförmigen Bodenfunktion, so kommt auch dieses robuste Laufmuster an seine Grenzen (Abb. 6.6). Nachdem innerhalb der ersten 13 Meter bereits Hügel bewältigt worden sind, kommt der Modell-Roboter bei dem visualisierten Anstieg kurzzeitig in gefährliche Rücklage. Das geschieht als der Roboter die vermeintlich schwerste Phase schon hinter sich hat, und der Hügel sich schon wieder zu seinem Gipfel hin abflacht. Der Roboter schafft es noch, am steilen Abhang einen Fuß vor den anderen zu setzen, aber fast oben angekommen, unternimmt er aufgrund des flacheren Bodens einen zu großen Schritt (dunkelblaues Bein, 15. Bild der Sequenz) und stößt sich wieder leicht zurück, in Richtung Tal ab. Es folgt ein längerer Balanceakt mit teils ungewohnt ganz durchgedrückten Knien und sogar automatischen Abfangschritten nach hinten. Dieses Verhalten zeigt sehr gut die Fähigkeit des elastischen Bein-Designs, sich dynamisch an seine Umgebung anzupassen, indem die Schrittweite variiert werden kann.

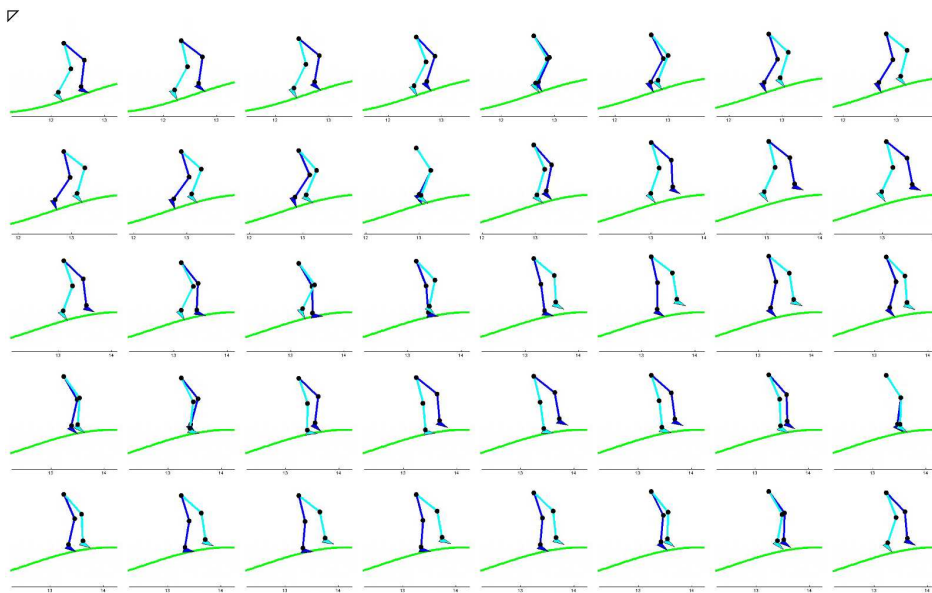


Abbildung 6.6: Eine etwas steilere, sinusförmige Bodenfunktion $y = \frac{1}{3} \cdot \sin(x)$ (Einheit: Meter) wird vom Roboter mit Konfiguration Optipunkt9 abgelaufen.

6.3 Fazit der Arbeit und Ausblick

In dieser Arbeit wurden die Bewegungsmuster einer bionisch inspirierten Laufmaschine mit nachgiebig elastischen Beinen anhand eines Software-Modells optimiert. Dabei kamen im Zuge verschiedener Gütekriterien robuste Optimierungsverfahren in teils sehr zerklüfteten Suchräumen mit bis zu 12 Dimensionen zum Einsatz. Neben stabilen, langsamen Gehmustern wurden auch extrem schnelle Laufmuster gefunden, die sich als sehr effizient in Bezug auf die verbrauchte Energie herausstellten. Die passiv elastischen Fähigkeiten der Beine wurden hierbei optimal genutzt, so dass die Antriebsmotoren mit minimalem Aufwand das Muster stabil halten konnten. Es waren deutliche Tendenzen bzgl. der Modifikation der Beineigenschaften in den Optimierungsergebnissen erkennbar, die sich auch mit den Experimenten am realen Roboter decken. Weiter wurden in Betracht auf zukünftige Optimierungsstudien direkt am richtigen Roboter die Beschaffenheit der Kontrollräume ergründet. Neben dem starken Zusammenhang zwischen der verbrauchten Energie pro Strecke und der Geschwindigkeit konnte eine deutliche Glättung des Kontrollraums durch die Optimierung beobachtet werden. Laufmuster mit glatten, breiten Kontrollräumen waren in der Lage, ganz ohne zusätzliche Feedback-Kontroll-Algorithmen, größere Störungen zu kompensieren. Hierbei fand der Roboter durch selbstständiges Regeln seiner Schrittweite immer wieder zu stabilen Mustern zurück.

Diese Aspekte motivieren zu weiterer Forschung. Jedoch sollten im Zuge weiterer Optimierungen zuerst anstehende Änderungen an dem Software-Modell voll-

zogen werden. Neue Muskeln (TA) mit präzisen Ansteuerungen könnten ein Abrollen des Fußes ermöglichen. Die noch aktive künstliche Fixierung des Offset-Winkels in der Hüfte könnte durch einen ausgleichend gesteuerten Oberkörper ersetzt werden. Weiter ist zu überlegen, ob man die Momente nicht direkt an den Aktuatoren entsprechend den Leistungsgrenzen der verwendeten realen Motoren beschränkt, um so unrealistisches Verhalten des Modell-Roboters auszuschließen. Mit weiteren Analysen zur Störungsanfälligkeit sollte zuerst auch das Kontaktmodell, das nach wie vor nur für die waagrechte Ebene korrekt arbeitet, entsprechend erweitert werden. Mit zusätzlicher Normalenvektordinformation zur Bodenfunktion können die an einer schiefen Ebene wirkenden Kräfte genau berechnet werden.

In dieser Arbeit erwies sich das elastische Design als äußerst praktisch und der Roboter wirkte viel agiler als seine herkömmlich starren Verwandten. Man ist also generell auf dem richtigen Weg und das Nachfolgermodell JW3 steht bereits schon in den Startlöchern.

Literaturverzeichnis

- [1] M. A. Abramson. *Pattern Search Filter Algorithms for Mixed Variable General Constrained Optimization Problems*. PhD thesis, Rice University, 2002. [cited at p. 14]
- [2] K.-H. Bauersfeld and G. Schröter. *Grundlagen der Leichtathletik*. Sportverlag Berlin, 1992. [cited at p. 51, 76]
- [3] Ballettschule Prokop Bildergalerie. *Der sterbende Schwan*. <http://www.ballett-prokop.at/>, 2006. [Online; accessed 20-Sep-2006]. [cited at p. 54, 76]
- [4] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22:1217–1227, 1989. [cited at p. 2]
- [5] G. A. Cavagna, F. P. Saibene, and R. Margaria. Mechanical work in running. *Journal of Applied Physiology*, 19:249–256, 1964. [cited at p. 2]
- [6] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive dynamic walkers. *Science Magazine*, 307:1082–1085, 2005. [cited at p. 1]
- [7] Kapstadt Fotowettbewerb. *Die besten Bilder aus Südafrika und Kapstadt*. <http://www.kapstadt.org/>, 2006. [Online; accessed 20-Sep-2006]. [cited at p. 54, 76]
- [8] Only-Robots.com Gallery. *Read the latest news on robotics and upload your own Robot pictures*. <http://onlyrobots.com/>, 2006. [Online; accessed 20-Sep-2006]. [cited at p. 4, 65, 76]
- [9] H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. *Proc. Roy. Soc. Lond. B*, DOI 10.1098:rspb.2006.3637, 2006. [cited at p. 2]
- [10] P. Gilmore and C. Kelley. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM Journal on Optimization*, 5:269–285, 1995. [cited at p. 14]
- [11] F. Iida, Y. Minekawa, J. Rummel, and A. Seyfarth. Toward a human-like biped robot with compliant legs. *Intelligent Autonomous Systems 9*, T. Arai, R. Pfeifer, T. Balch, H. Yokoi (eds.). IOS Press:820–827, 2006. [cited at p. 4]

- [12] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2), 1990. [cited at p. 1, 2]
- [13] M. H. Raibert. Legged robots that balance. *MIT Press, Cambridge, MA*, 1986. [cited at p. 2]
- [14] A. Seyfarth, R. Tausch, M. Stelzer, F. Iida, A. Karguth, and O. von Stryk. Towards bipedal running as a natural result of optimizing walking speed for passively compliant three-segmented legs. In *CLAWAR 2006: 9th International Conference on Climbing and Walking Robots*, pages 396–401, Brussels, Belgium, September 12-14 2006. [cited at p. 4, 5, 8]
- [15] Wikipedia. *Matlab - The MathWorks, Inc.* <http://de.wikipedia.org/wiki/Matlab>, 2006. [Online; accessed 20-Sep-2006]. [cited at p. 7]

Abbildungsverzeichnis

1.1	McGeers passiv dynamische Laufmaschine	2
1.2	Raiberts selbst-balancierender einbeiniger Hüpfroboter	2
1.3	Laufendes Feder-Masse-Modell	3
1.4	Der erste Jena Walker	3
1.5	Asimo, Johnnie, Qrio, HRP-2	4
1.6	Jena Walker 2 Aufbau	5
2.1	Simulink-Simmechanics-Modell des Jena Walker 2	8
2.2	Der Parameterblock des Simulink-Modells	9
2.3	Die kinematische Kette des linken Beins	11
2.4	JW2 Aufbau und Visualisierung	12
3.1	Weber-Problem, Nelder-Mead, Newton	15
3.2	Der Optimierungskreislauf	16
3.3	Architektur der Optimierung	16
3.4	Sequenzdiagramm zur Optimierung	18
3.5	Auswertungsreihe f-om-v	19
3.6	Koordinaten möglicher Startlösungen - Testpunkte	21
3.7	Geschwindigkeit zu Startpunkt1 und Testpunkt1-3	23
3.8	Drehmomente in Knie- und Fußgelenk zu Startpunkt1, Testpunkt1-3	24
3.9	Laufmuster: Testpunkt2	24
3.10	Drehmomente in den Hüftgelenken zu Startpunkt1, Testpunkt1-3	25
3.11	Industrieroboter (KR 6 ARC, KUKA Roboter GmbH)	28
3.12	Optipunkt1: Konvergenz, Geschwindigkeit, Drehmomente	29
3.13	Optipunkt2: Konvergenz, Geschwindigkeit, Drehmomente	32
3.14	Laufmuster: Optipunkt2	33
3.15	Optipunkt5: Konvergenz, Geschwindigkeit, Drehmomente	34
3.16	Nomad-Verfahren, Variieren	36
3.17	Optipunkt9: Konvergenz, Geschwindigkeit, Drehmomente	37
3.18	Laufmuster: Optipunkt5, Optipunkt9	39

3.19	Optipunkt14: Konvergenz, Geschwindigkeit, Drehmomente	41
3.20	Optipunkt15: Konvergenz, Geschwindigkeit, Drehmomente	43
3.21	Laufmuster: Optipunkt14, Optipunkt15	45
3.22	Bodenkontakt: Optipunkt14, Optipunkt15	45
4.1	Übersicht über die Phasenstruktur beim Laufschrift ([2])	51
4.2	Phaseneinteilung	52
4.3	Bodenkontakt: Optipunkt1, Optipunkt9	52
4.4	Mögliche Vergleiche? ([7], [3])	54
4.5	Langsames Laufmuster JW 2	56
4.6	Gemäßigtes Laufmuster JW 2	57
4.7	Schnelle Laufmuster JW 2	58
5.1	Entwicklung des f - α -Kontrollraums zu Startpunkt1	60
5.2	f - α -Kontrollraum zu Startpunkt1	61
5.3	Entwicklung des f - α -Kontrollraums zu Optipunkt9	62
5.4	f - α -Kontrollraum zu Optipunkt9	63
6.1	Asimo von Honda beim Treppensteigen ([8]).	65
6.2	10cm hohe Treppenstufen, Optipunkt9	67
6.3	Ein 20cm hoher Block wird überwunden, Optipunkt9	68
6.4	Sinusförmige Bodenfunktion $y = \frac{1}{5} \cdot \sin(x)$, Optipunkt5	68
6.5	Sinusförmige Bodenfunktion $y = \frac{1}{5} \cdot \sin(x)$, Optipunkt9	69
6.6	Sinusförmige Bodenfunktion $y = \frac{1}{3} \cdot \sin(x)$, Optipunkt9	70