

Modellierung und Simulation biomechanischer Vorgänge am Beispiel Skisprung

Diplomarbeit
von
Syn Schmitt

31. Juli 2003

Hauptberichter: Prof. Dr. G. Wunner

Institut für Theoretische Physik I
Universität Stuttgart

Mitberichter: Prof. Dr. H. Ruder

Institut für Astronomie und Astrophysik
- Abteilung Theoretische Astrophysik -
Eberhard Karls Universität Tübingen

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
1 Einleitung	1
2 Grundlagen	2
2.1 Mechanische Grundlagen	2
2.2 Anatomische Grundlagen	4
2.3 Biomechanische Grundlagen	10
3 Methoden	13
3.1 Modellierung	13
3.2 Simulation	26
4 Ergebnisse und Diskussion	31
4.1 Überprüfung der Drehimpulserhaltung	31
4.2 Regelung	32
4.3 Luftwiderstand und Auftrieb	34
4.4 Momente in den Gelenken	39
4.5 Bodenreaktionskräfte	40
4.6 Position und Geschwindigkeit	41
4.7 Die Landung	42
4.8 Dämpfungswerte	42
5 Zusammenfassung und Ausblick	46
Literatur	49
A Die Rotationsfreiheitsgrade	53
B Die Modellkonfigurationsdatei <skijump.dys>	53

Abbildungsverzeichnis

1	Die verschiedenen Gelenktypen	6
2	Die Bezugsebenen des Körpers	7
3	Bestimmung des Bewegungsumfangs (SFTR-Methode)	9
4	Anatomische Stellung	9
5	Segmentierung des Menschen nach der NASA	14
6	Das erstellte Skispringermodell	15
7	Die Definition der Kontaktsituation beim Schanzenkontakt	18
8	Der Auftriebsparameter c_A als Funktion des Anstellwinkels β	22
9	Bildsequenz der Animation des Absprungs (Teil 1)	29
10	Bildsequenz der Animation des Absprungs (Teil 2)	30
11	Drehimpuls mit Aerodynamikkräften	31
12	Drehimpuls ohne Aerodynamikkräfte	32
13	Kraft-Zeit-Verlauf der Kontaktkraft zwischen Fuß und Ski	33
14	Momentverlauf des Sprunggelenks während der Anfahrt	34
15	Aktive Drehfeder im Sprunggelenk	35
16	Aktive Drehfeder im Kniegelenk	35
17	Aktive Drehfeder im Hüftgelenk	36
18	Aktive Drehfeder im Oberkörper	36
19	Aktive Drehfeder im Schulterbereich	37
20	Die aerodynamischen Kräfte am Unterarm	37
21	Die aerodynamischen Kräfte am Kopf	38
22	Luftwiderstands- und Auftriebskraft des gesamten Systems	38
23	Luftwiderstands- und Auftriebskraft des Thorax	39
24	Die Bodenreaktionskraft über alle acht Kontaktpunkte summiert	40
25	Der Positionsverlauf der Hüfte	41
26	Der Geschwindigkeitsverlauf der Hüfte	42
27	Der erste Teil der Landesequenz	43
28	Der zweite Teil der Landesequenz	44

Tabellenverzeichnis

1	Das anthropometrische Menschmodell als Baumstruktur	16
---	---	----

1 Einleitung

Das Studium sportlicher Bewegungen ist Teil einer bewegungswissenschaftlichen Betrachtungsweise des Menschen. Dabei sind Bewegungsmotivation, Bewegungsintension, Bewegungsdurchführung und Bewegungsergebnis die Hauptinteressensbereiche der Forschung. Mit biomechanischen Methoden können wichtige Erkenntnisse, vor allem in den Bereichen Bewegungsdurchführung und Bewegungsergebnis, beigetragen werden. Sehr viele Forschungseinrichtungen und Institute erzielen gute Ergebnisse durch Messungen direkt am Menschen sowohl im Labor als auch im Feld. Das benutzte Instrumentarium umfasst hierfür meist dreidimensionale Kraftmessplatten, hochauflösende Videoanalyseverfahren und Ableitung der Muskelaktivität durch Oberflächen Elektroden (EMG).

Daneben besteht die Möglichkeit durch Modellbildung und Simulation an Rechnern mit einer problemangepassten Vereinfachung der Bewegung weitere, in der Realität nicht messbare Details zu erforschen. Einige Arbeiten, die in der theoretischen Astrophysik am Institut für Astronomie und Astrophysik in Tübingen durchgeführt wurden sind dafür beeindruckende Beispiele. Die Fragestellung dieser Untersuchungen reichte dabei von gerichtsmedizinischen Aspekten [13, 22] über technische Entwicklungen [22] bis hin zu menschlichen/sportlichen Bewegungen [10, 2, 19].

Komplexe sportliche Bewegungen mit einem dreidimensionalen Ganzkörpermodell stellen hier eine große Herausforderung dar. Durch die große Zahl der Freiheitsgrade ist es besonders wichtig, sehr genau zwischen Detailliertheit und Vereinfachung des Modells abzuwägen. Denn in beiden Fällen droht wenig Aussagekraft durch entweder zu viele unbestimmte Parameter, die zur Modellierung aller Details notwendig sind, oder zu wenigen Freiheitsgraden, die wichtige Effekte systematisch ausschließen.

Diese Arbeit beschäftigt sich mit dem Skisprung und der Fragestellung, wie gut es möglich ist eine komplexe sportliche Bewegung dieser Art durch möglichst einfache Modellierung der Gelenke, der Umgebungsinteraktion und der aktiven Kontrolle der Bewegung vorwärtsdynamisch zu simulieren.

2 Grundlagen

2.1 Mechanische Grundlagen

Zur Erzeugung der Bewegungsgleichungen für die dynamische Analyse des mechanischen Systems wurde im Rahmen dieser Arbeit das Programm **DySim**, das an unserem Institut von HENZE [18] entwickelt wurde, verwendet. **DySim** basiert auf einem ableitungsfreien Verfahren zur Formulierung der LAGRANGESchen Gleichungen zweiter Art in generalisierten Koordinaten:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i, \quad i = 1, 2, \dots, n \quad . \quad (1)$$

Dieses Verfahren, erstmals vollständig von LEGNANI [24, 25] 1996 publiziert, benutzt den Ansatz mit homogenen 4×4 Matrizen, wie es für homogene Transformationen von DENAVIT und HARTENBERG [4, 3] vorgestellt wurde. Dabei werden zusätzlich zur 4×4 Lage-Matrix eines Körpers

$$\mathbf{R} = \begin{pmatrix} \mathbf{D} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad \begin{array}{l} \mathbf{t}: \text{Position} \\ \mathbf{D}: \text{Orientierung} \end{array} \quad ,$$

weitere Matrizen für die Geschwindigkeiten \mathbf{W} , die Beschleunigungen \mathbf{H} , die Impulse $\mathbf{\Gamma}$, die Trägheitsmomente \mathbf{J} und die Kräfte (und Drehmomente) $\mathbf{\Phi}$ der Körper eingeführt:

$$\mathbf{W} = \dot{\mathbf{R}} \mathbf{R}^{-1} = \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \mathbf{v}_0 \\ \mathbf{0} & 0 \end{pmatrix} \quad \begin{array}{l} \tilde{\boldsymbol{\omega}}: \text{Winkelgeschwindigkeit} \\ \mathbf{v}_0: \text{Geschwindigkeit} \end{array} \quad ,$$

$$\mathbf{H} = \dot{\mathbf{W}} + \mathbf{W}^2 = \begin{pmatrix} \dot{\tilde{\boldsymbol{\omega}}} + \tilde{\boldsymbol{\omega}}^2 & \mathbf{a}_0 \\ \mathbf{0} & 0 \end{pmatrix} \quad \mathbf{a}_0: \text{Beschleunigung} \quad ,$$

$$\mathbf{J} = \begin{pmatrix} \int \mathbf{x} \mathbf{x}^T \rho dV & m \mathbf{r}_{cg} \\ m \mathbf{r}_{cg}^T & m \end{pmatrix} \quad \begin{array}{l} m: \text{Masse} \\ \mathbf{r}_{cg}: \text{Schwerpunktlage} \\ \rho: \text{Massendichte} \end{array} \quad ,$$

$$\mathbf{\Phi} = \begin{pmatrix} \tilde{\mathbf{n}} & \mathbf{f} \\ -\mathbf{f}^T & 0 \end{pmatrix} \quad \begin{array}{l} \tilde{\mathbf{n}}: \text{Drehmoment} \\ \mathbf{f}: \text{Kraft} \end{array} \quad ,$$

$$\mathbf{\Gamma} = \begin{pmatrix} \tilde{\mathbf{l}} & \mathbf{p} \\ -\mathbf{p}^T & 0 \end{pmatrix} \quad \begin{array}{l} \tilde{\mathbf{l}}: \text{Drehimpuls} \\ \mathbf{p}: \text{Impuls} \end{array} \quad .$$

Durch diese Formulierung genügen einfache Matrix-Vektor-Multiplikationen mit dem Ortsvektor in körperfesten Koordinaten $\mathbf{x}' = [\mathbf{s}^T, 1]^T$ oder $\mathbf{x} = [\mathbf{s}^T, 1]^T$ in Inertialkoordinaten zur Beschreibung

◇ affiner Transformationen von körperfesten zu Inertialkoordinaten:

$$\mathbf{x} = \mathbf{R} \mathbf{x}' \Leftrightarrow \mathbf{r} = \mathbf{t} + \mathbf{D} \mathbf{s}' \quad , \quad (2)$$

◇ der Geschwindigkeiten und Beschleunigungen:

$$\dot{\mathbf{x}} = \mathbf{W} \mathbf{x} \Leftrightarrow \mathbf{v} = \mathbf{v}_0 + \boldsymbol{\omega} \times \mathbf{s} \quad , \quad (3)$$

$$\ddot{\mathbf{x}} = \mathbf{H} \mathbf{x} \Leftrightarrow \mathbf{a} = \mathbf{a}_0 + \dot{\boldsymbol{\omega}} \times \mathbf{s} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{s}) \quad . \quad (4)$$

Sowohl die kinematischen Matrizen \mathbf{W} und \mathbf{H} als auch die dynamischen Matrizen \mathbf{J} , $\mathbf{\Gamma}$ und $\mathbf{\Phi}$ lassen sich durch \mathbf{R} , definiert in kartesischen Inertialkoordinaten, vom körperfesten System ins Inertialsystem überführen:

$$\begin{aligned} \mathbf{W} &= \mathbf{R} \mathbf{W}' \mathbf{R}^{-1} \quad , \\ \mathbf{H} &= \mathbf{R} \mathbf{H}' \mathbf{R}^{-1} \quad , \\ \mathbf{J} &= \mathbf{R} \mathbf{J}' \mathbf{R}^T \quad , \\ \mathbf{\Gamma} &= \mathbf{R} \mathbf{\Gamma}' \mathbf{R}^T \quad \text{und} \\ \mathbf{\Phi} &= \mathbf{R} \mathbf{\Phi}' \mathbf{R}^T \quad . \end{aligned}$$

Mit der Erdbeschleunigungsmatrix \mathbf{H}_g lassen sich die kinetische und potentielle Energie in folgender Weise darstellen:

$$T = \frac{1}{2} \text{spur}(\mathbf{W} \mathbf{J} \mathbf{W}^T) \quad , \quad (5)$$

$$V = -\text{spur}(\mathbf{H}_g \mathbf{J}) \quad , \quad (6)$$

und in kompakter Form erhält man die NEWTON-EULER-Gleichungen zur Beschreibung der Dynamik des Körpers:

$$\mathbf{\Phi} = \mathbf{H} \mathbf{J} - \mathbf{J} \mathbf{H}^T \quad . \quad (7)$$

Wie aber eingangs erwähnt benützt DySim die LAGRANGE-Gleichungen zweiter Art. Um diese in generalisierten Koordinaten aufzustellen wird jeder Teilkörper mit einem Freiheitsgrad, relativ zu seinem Mutterkörper, versehen. Die Geschwindigkeitsmatrix \mathbf{W} , die zu jedem Zeitpunkt eine momentane Schraubenachse definiert, wird zerlegt in eine Rotationsbewegung um die Schraubenachse und eine Translationsbewegung entlang der Schraubenachse. Dadurch erhält man zwei Basisfreiheitsgrade:

$$\mathbf{L}_s = \begin{pmatrix} \tilde{\mathbf{e}} & -\tilde{\mathbf{e}} \mathbf{r}_e + p \mathbf{e} \\ \mathbf{0} & 0 \end{pmatrix} , \quad \mathbf{L}_t = \begin{pmatrix} \mathbf{0} & \mathbf{e} \\ \mathbf{0} & 0 \end{pmatrix} \quad \begin{array}{l} \mathbf{e}: \text{Schraubenachse} \\ \mathbf{r}_e: \text{Punkt auf der Achse} \\ p: \text{Vortrieb} \end{array} .$$

Für jeden Freiheitsgrad wird nun jeweils eine generalisierte Koordinate q eingeführt, so dass die Geschwindigkeitsmatrix eines Körpers mit einem Freiheitsgrad als Funktion der generalisierten Geschwindigkeit geschrieben werden kann,

$$\mathbf{W}_s(\dot{q}) = \mathbf{L}_s \dot{q} \quad \text{bzw.} \quad \mathbf{W}_t(\dot{q}) = \mathbf{L}_t \dot{q} \quad ,$$

aus der sich durch Integration die Komponenten der Lage-Matrix als Funktion von q ergeben:

$$\begin{aligned} \mathbf{D}_s(q) &= \mathbf{1} + \tilde{\mathbf{e}} \sin q + \tilde{\mathbf{e}}^2(1 - \cos q) & \text{bzw.} & & \mathbf{D}_t(q) &= \mathbf{1} \\ \mathbf{t}_s(q) &= (\mathbf{1} - \mathbf{D}_s(q)) \mathbf{r}_e + p \mathbf{e}_q & & & \mathbf{t}_t(q) &= \mathbf{e}_q \quad . \end{aligned}$$

So kann man mit den Ausdrücken der Energien (Gl. 5 und 6) die LAGRANGE-Funktion aufstellen:

$$L = \sum_{k=1}^n \frac{1}{2} \text{spur}(\mathbf{W}_{0,k} \mathbf{J}_k \mathbf{W}_{0,k}^T) + \text{spur}(\mathbf{H}_g \mathbf{J}_k) \quad n: \text{Anzahl der Teilkörper} \quad . \quad (8)$$

Sollte ein Starrkörper mehr als einen Freiheitsgrad besitzen, müssen masselose Hilfskörper dazwischengeschaltet werden, da jeder Körper nur einen Basisfreiheitsgrad besitzen kann.

Nun lassen sich die Lage- und Geschwindigkeitsmatrizen jedes Körpers als Funktion der generalisierten Koordinaten ausdrücken:

$$\mathbf{R}_{0,k} = \prod_{i=1}^{n_k} \mathbf{R}_{k_{i-1},k_i}(q_{k_i}) \quad , \quad \mathbf{W}_{0,k} = \sum_{i=1}^{n_k} \mathbf{L}_{k_{i-1},k_i} \dot{q}_{k_i} \quad ,$$

n_k : Anzahl aller benötigten Freiheitsgrade zur Beschreibung eines Körpers im Inertialsystem.

Somit ist es möglich die zur Aufstellung der LAGRANGESchen Bewegungsgleichung benötigten Ableitungen durch Summen und Produkte der Matrizen \mathbf{R} , \mathbf{L} , \mathbf{W} und \mathbf{J} darzustellen. Man erhält dann das folgende Gleichungssystem in generalisierten Koordinaten:

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad . \quad (9)$$

2.2 Anatomische Grundlagen

Die Anatomie des Menschen beschäftigt sich mit dem Bewegungsapparat, den inneren Organen, dem zentralen Nervensystem und den Sinnesorganen. Im Folgenden wird versucht, kurz einige grundlegende Strukturen und Definitionen, die direkt im Zusammenhang mit der Modellierung biomechanischer Vorgänge stehen, zu erwähnen. Im Vordergrund steht der passive Bewegungsapparat.

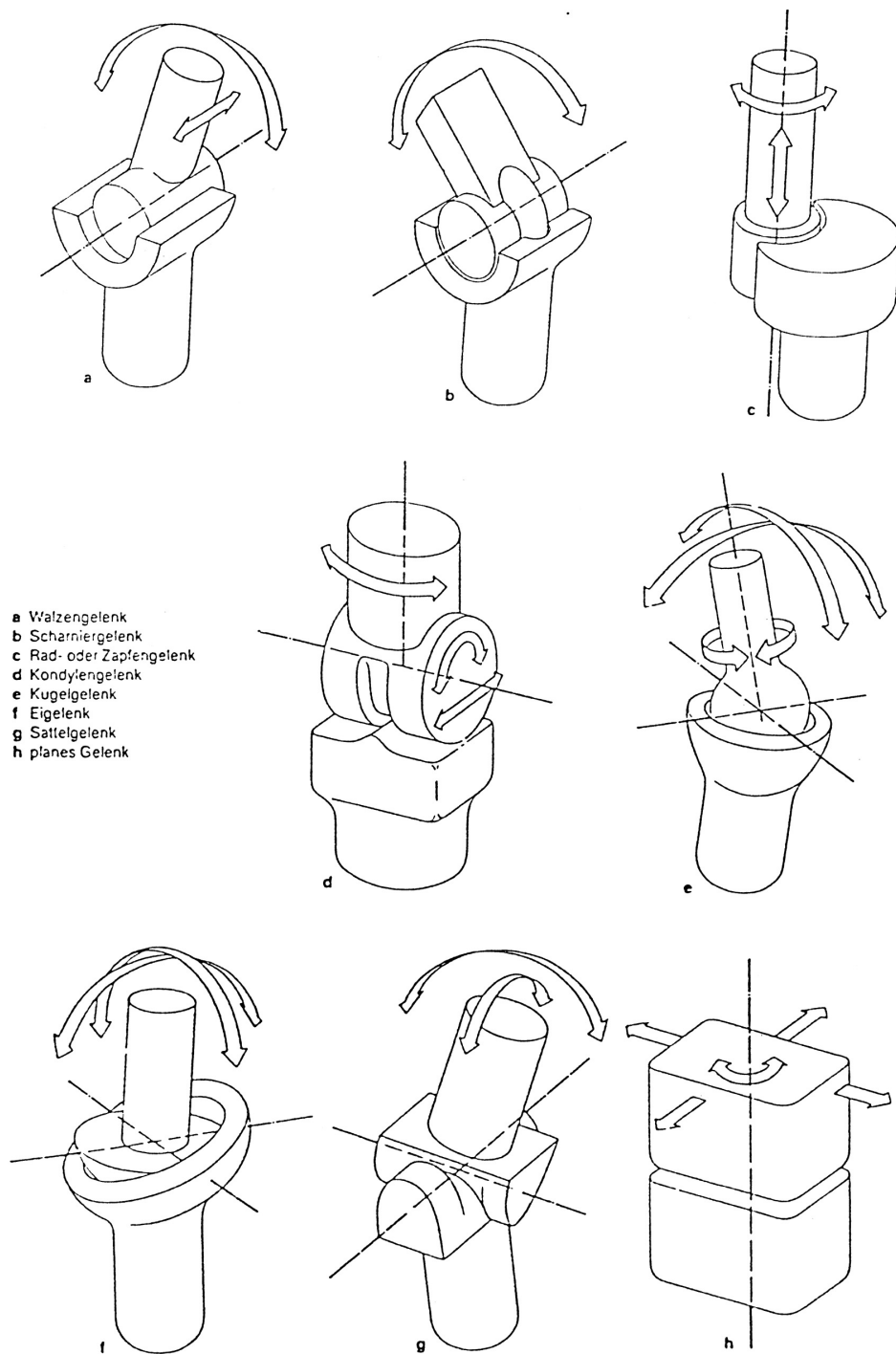
Der passive Bewegungsapparat besteht aus Knochen, Bändern, Knorpeln und Gewebe. Knochenverbindungen nennt man Gelenke, davon gibt es verschiedene Typen (siehe Abb. 1): Walzengelenk, Kugelgelenk, Ellipsoidgelenk, Sattelgelenk, ebenes Gelenk und straffes Gelenk.

Nach der Anzahl der in einem Gelenk miteinander artikulierenden Skelettelemente unterscheidet man *einfache* und *zusammengesetzte* Gelenke. Bei einem einfachen Gelenk stehen zwei Gelenkkörper miteinander in Kontakt, z.B. das Schulter- oder das Hüftgelenk. Ein zusammengesetztes Gelenk liegt vor, wenn mehr als zwei Skelettelemente gelenkig miteinander verbunden sind. Beispiele für zusammengesetzte Gelenke sind das Ellenbogen- oder das Sprunggelenk.

Die Unterscheidung der Gelenktypen findet nach der Form ihrer Gelenkkörper statt. Für die Gelenkmechanik sind folgende, bereits erwähnte Typen von Bedeutung (siehe Abb. 1):

- ◇ Das *Walzengelenk* ist ein einachsiges Gelenk in zwei verschiedenen Formen: Als *Scharniergelenk* z.B. im Oberarm-Ellen-Gelenk und als *Rad- oder Zapfengelenk* z.B. beim mittleren oberen Kopfgelenk.
- ◇ Das *Kugelgelenk* ist ein dreiachsiges Gelenk mit Flexion und Extension, Ab- und Adduktion und Innen- und Außenrotation. Typische Kugelgelenke sind das Schulter- und Hüftgelenk.
- ◇ Weitere zweiachsige Gelenke sind das *Ellipsoid- oder Eigelenk* z.B. das Handgelenk und das *Sattelgelenk* z.B. im Daumengelenk.
- ◇ Als *ebenes Gelenk* bezeichnet man ein Gelenk das translatorische und rotatorische Bewegungen zulässt und dessen Bewegungsmöglichkeiten abhängig von seiner Stellung im Körper sind. Typische ebene Gelenke sind die Wirbelgelenke der Wirbelsäule.
- ◇ Gelenke, deren Beweglichkeit durch die Form ihrer Gelenkkörper und durch kräftige Bandverstärkungen stark eingeschränkt ist, werden als *straffe Gelenke* bezeichnet, z.B. das Kreuzbein-Darmbeingelenk.

Die Gelenke erlauben dem passiven Bewegungsapparat einen Bewegungsumfang, der durch Bänder und Knorpel, aber auch durch aktive Komponenten, wie dem Muskel-Sehnen-Apparat, eingeschränkt sein kann. In der Anatomie wird der Bewegungsumfang relativ zu drei Bezugsebenen angegeben. Diese Bezugsebenen stehen paarweise senkrecht aufeinander und heißen *Sagittalebene*, *Frontalebene* und *Transversalebene* (siehe Abb. 2).



- a) Walzengelenk
- b) Scharniergelenk
- c) Rad- oder Zapfengelenk
- d) Kondylengelenk
- e) Kugelgelenk
- f) Eigelenk
- g) Sattelgelenk
- h) planes Gelenk

Abbildung 1: Die verschiedenen Gelenktypen: a) Walzengelenk, b) Scharniergelenk, c) Rad- oder Zapfengelenk, d) Kondylengelenk, e) Kugelgelenk, f) Eigelenk, g) Sattelgelenk, h) planes Gelenk (nach RAUBER UND KOPSCH [31]).

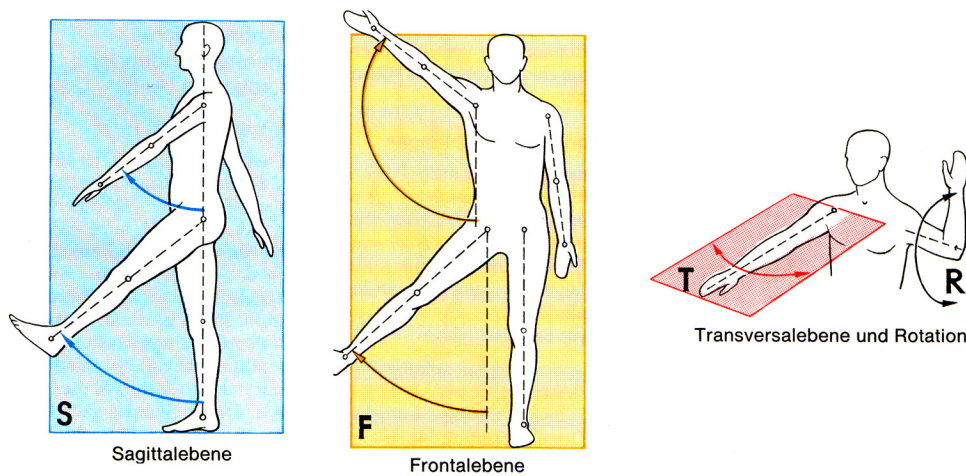


Abbildung 2: Der Körper eingeteilt in Ebenen ausgehend von den Raumdimensionen. Die drei Bezugsebenen stehen senkrecht aufeinander (nach DTV-ATLAS [29]).

- ◇ Eine *sagittale Ebene* ist im Stand vertikal und verläuft von bauchwärts nach rückenwärts. Der Name stammt vom Verlauf der Sagittalnaht des Schädels. Das Anheben des Arme und auch der Beine verläuft in einer Sagittalebene, jedoch nicht in der gleichen (siehe Abb. 2), weshalb nicht von *der* sondern von *einer* Ebene gesprochen wird. Die mittlere Sagittalebene heißt Median- oder Symmetrieebene und teilt den Körper in eine rechte und linke Hälfte.
- ◇ Eine *frontale Ebene* verläuft im Stand vertikal von der einen zur anderen Seite. Der Name stammt von der Kranznaht des Schädels. Die Ebene teilt den Körper in einen vorderen und hinteren Abschnitt.
- ◇ Eine *transversale Ebene* verläuft im Stand horizontal und teilt den Körper in einen oberen und unteren Abschnitt.

Neben den Bezugsebenen für die Gelenkbewegungen werden Achsen definiert um die die Bewegungen stattfinden. Die Achsen werden benannt nach den zwei Ebenen, die sich in ihnen schneiden, z.B. findet das Strecken und Beugen des Fußes um eine sagittotransversale Achse statt. D.h. die Transversalebene “auf der der Fuß steht” wird gegen eine Sagittalebene “an der der Fuß angelehnt ist” verkippt. Man definiert folgende Achsen:

- ◇ Die *sagittotransversale Achse* bildet den Schnittpunkt der Sagittal- und Transversalebene und verläuft in bauchwärts-rückenwärtiger Richtung. Die Bewegung der Ab- (vom Körper weg) und Adduktion (zum Körper hin) finden um diese Achse in der Frontalebene statt.

- ◇ Die *frontotransversale Achse* bildet den Schnittpunkt der Frontal- und der Transversalebene und verläuft von der einen zur anderen Körperseite. Die Bewegungen der Flexion (Beugung) und Extension (Streckung) finden um diese Achse in der Sagittalebene statt.
- ◇ Die *sagittofrontale Achse* bildet den Schnittpunkt der Sagittal- und Frontalebene. Sie verläuft von oben nach unten. Die Bewegungen der Innen- und Außenrotation finden um diese Achse in der Transversalebene statt.
- ◇ Bewegungen, die ebenfalls in der Transversalebene um eine sagittofrontale Achse stattfinden, werden als “horizontales Rück- und Vorführen des Armes” oder als “Ab- und Adduktion des in der Hüfte 90 Grad gebeugten Beines” bezeichnet.

Die Bewegungen aller Gelenke werden zur Sagittalebene, Frontalebene und Transversalebene in Beziehung gesetzt. Das Messen der Bewegungen erfolgt entweder in einer dieser Ebenen oder parallel dazu. Entsprechend der Ebene, in der die Bewegung erfolgt, wird dem Bewegungsumfang der Buchstabe S, F oder T vorangesetzt. Eine Drehbewegung hingegen wird mit R (Rotation) bezeichnet (SFTR-Methode, siehe Abb. 3).

Die Ausgangsstellung für die Messung des Bewegungsumfangs ist die Neutral-0-Stellung, die auch *anatomische Stellung* genannt wird. Diese Stellung liegt vor, beim aufrechten Stand, das Gesicht nach vorne gerichtet, die Arme seitlich am Rumpf hängend und die Hand mit der Hohlhand nach vorne gedreht (siehe Abb. 4). Die anatomische Stellung entspricht nicht der “normalen” oder “physiologischen” Stellung, bei der die Hand mit der Handfläche zum Körper gerichtet steht.

Alle Bewegungen werden mit drei Zahlen notiert. Bewegungen der Extension - Bewegungen, die vom Körper wegführen - werden zuerst, Bewegungen der Flexion - Bewegungen, die zum Körper hinführen - werden zuletzt notiert. Die mittlere Zahl ist die neutrale Stellung, gewöhnlich -0- (Abb. 4).

Der *passive Bewegungsapparat* reicht zur biologischen Beschreibung des Menschen nicht aus und muss durch den *aktiven Bewegungsapparat* (die Muskeln), das *Nervensystem mit Sinnesorganen*, den *Regelungsorganen* und den *inneren Organen* erweitert werden. An dieser Stelle soll aber nicht näher darauf eingegangen werden, da im Rahmen dieser Arbeit weder Muskeln noch sonstige weitere Organe ins Modell implementiert wurden. Im Abschnitt *Modellierung der aktiven Elemente* dienen jedoch elementare Kenntnisse aus der Anatomie außerhalb des passiven Bewegungsapparats als *anatomische Repräsentanz* der verwendeten Algorithmen.

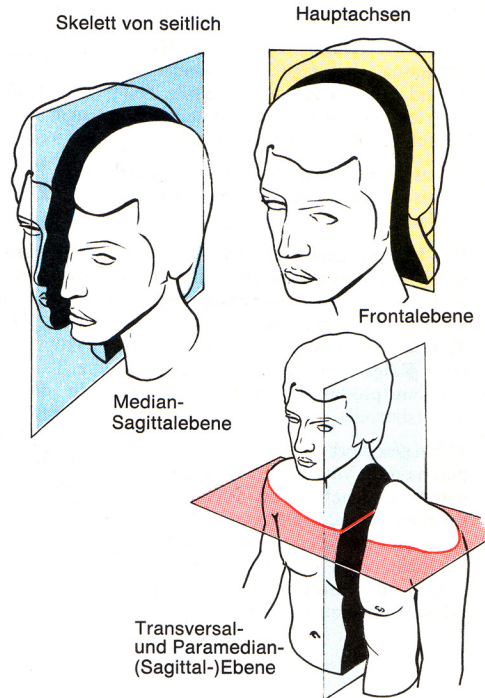


Abbildung 3: Basis der SFTR-Methode zur Bestimmung der Körpersegmentlagen sind die Nulllagen. Der Bewegungsumfang wird nun nach Sagittal-, Frontal-, Transversalebene und Rotation angegeben. Die erste Zahl bezeichnet die Extension, die zweite die Nulllage und die dritte Zahl die Flexion (nach DTV-ATLAS [29]).

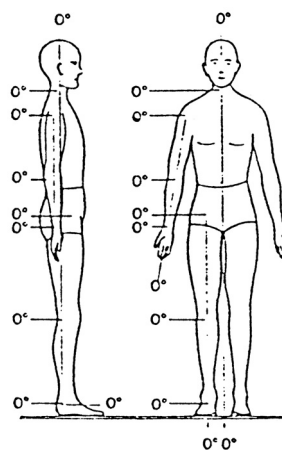


Abbildung 4: In dieser schematischen Darstellung sind die Nullwinkel eingezeichnet, ausgehend davon wird die Stellung der Körpersegmente angegeben.

2.3 Biomechanische Grundlagen

Im Sinne der Physik ist das Objekt *menschlicher Körper* eine Ansammlung von Materialien mit unterschiedlichsten Eigenschaften. Zusammengesetzt aus einer Vielzahl von Knochen, Sehnen, Bänder, Gelenke, Muskeln, Nervenfasern und weiteren Organen ist der Mensch ein komplexes Objekt mit viele Freiheitsgrade. Vereinfachende mathematische Beschreibungen des Objekts *Mensch* ergeben daher zwangsläufig Näherungen, deren Grad an Komplexität wiederum die Annäherung an die Realität widerspiegeln. Das erste Modell zu Anfang wird meistens ein relativ einfaches mathematisch/physikalischen Modell sein, welches im Laufe seiner Entwicklung verfeinert wird.

Die Methode der finiten Elemente (FEM) ist derzeit die Methode, die den menschlichen Körper am besten abbildet, vorausgesetzt die Parameter der benötigten Kontinuumseigenschaften der Körperteile sind bekannt. Am Institut für Astronomie und Astrophysik wurde hierzu eine Dissertation zur Berechnung von Kopfbelastungen und die Reproduktion von Hirnverletzungen bei Unfällen durchgeführt (GÖTZ [13]). Bei der Berechnung von Ganzkörpermodellen ist das sehr rechen- und datenintensive Verfahren aber noch unverhältnismäßig aufwändig und zuverlässige Parameter zu Kontinuumseigenschaften sind derzeit nicht immer vorhanden.

Das zur Anfertigung dieser Arbeit verwendete Modellierungs- und Simulationsverfahren ist das MKS-(Mehr-Körper-Simulations-) Verfahren. Hierbei werden die Körperteile durch starre Körper beschrieben, weshalb auch von der Starrkörpermethode gesprochen wird. Die einzelnen Körper sind über mechanische Gelenke miteinander verbunden (HATZE [16], YAMAGUCHI [39]).

Die direkt oder vorwärts dynamische biomechanische Modellierung des Bewegungsapparats - zur Unterscheidung "*direkt dynamisch* <-> *invers dynamisch*" siehe Dissertation von GÜNTHER [10] - mit dem Ziel der Simulation von Bewegungsabläufen lässt sich in drei Teilbereiche unterteilen:

Der Körperaufbau

Im Körperaufbau werden alle geometrischen und physikalischen Daten berücksichtigt, die bei der Modellierung benötigt werden, hier im Wesentlichen die Anatomie des passiven Bewegungsapparats. Das zugrundeliegende mechanische Menschmodell entscheidet über den detaillierten Körperaufbau. Das Modell der vorliegenden Arbeit enthält Teilkörper mit Längen, Schwerpunktlagen und Trägheitseigenschaften sowie die Verbindungen der Teilkörper, den Gelenken. Die Position und Orientierung der Gelenke an den Teilkörpern bestimmen die Anfangsbedingungen (siehe Abschnitt 3.1). Bei der Anatomie des Menschen wird der aktive Bewegungsapparat hauptsächlich durch die Muskeln bestimmt, die aber im Rahmen dieser Diplomarbeit

nicht modelliert wurden und anstatt denen Drehfedern als Generatoren der aktiven Kraft (Momentengeneratoren) verwendet wurden. Die Funktionsweise dieser Drehfedern wird im nächsten Kapitel im Abschnitt *Aktive Modellregelung* (3.1) detaillierter dargestellt.

Grundsätzlich sind vor allem die inneren Daten des Körperaufbaus am lebenden Menschen schwer zu ermitteln, weshalb hier Daten aus einer Datenbank der NASA [27] verwendet wurden - im Kapitel *Methoden* wird dies näher erläutert.

Die Mehrkörperdynamik

Der menschliche Körper als Objekt ist grundsätzlich der Physik unterworfen, trotz Bewegungssteuerung und -kontrolle. Seine Dynamik, die durch die NEWTONSchen Bewegungsgleichungen beschrieben wird, ist von Trägheit, Gravitation, kinematischen Kopplungen und daraus resultierenden Zwangskräften dominiert sowie von eingepägten, äußeren Kräften und Drehmomenten. In der Realität stellt die resultierende Bewegung selbst die Lösung der Bewegungsaufgabe dar, während die theoretische Simulation von komplexen biomechanischen Modellen nur durch die *numerische Lösung* der Bewegungsgleichungen erreichbar ist. Zur Lösung der Bewegungsgleichungen als Folge der wirkenden Kräfte und Drehmomente werden diese *numerisch integriert*. In Abschnitt 3.2 wird hierzu ein kurzer Überblick geboten.

Die Bewegungsregelung

Die Modellierung der Bewegungsregelung stellt eine ausgesprochen komplexe Aufgabe dar. Fein aufeinander abgestimmte Prozesse und Gegenprozesse bestimmen in der Realität die Bewegung des Menschen. Eine möglichst realistische Modellierung der Bewegungsregelung erfordert außer dem Vorhandensein von Muskulatur auch das Implementieren von Sinnesorganen und des zentralen Nervensystems.

Das Ziel dieser Arbeit aber ist es, in allen zur Durchführung des Skisprungs notwendigen Bereichen, eine möglichst einfache Bewegungsregelung zu verwenden oder zu finden. Sodass das Vorhandensein des zentralen Nervensystems (ZNS) als auch der Sensoren der Bewegungskoordination (Sinnesorgane, Muskelspindel, Golgi-Sehnen-Apparat u.a.) zwar bekannt sind, aber z bei der Modellierung nicht berücksichtigt werden. Trotzdem wurde bei der Suche nach einer einfachen Bewegungsregelung auf eine anatomische Begründbarkeit der gewählten Algorithmen geachtet (siehe Abschnitt 3.1).

Nicht alle der genannten Bereiche biomechanischer Bewegungserzeugung wurden bei der Modellierung in dieser Arbeit vollständig berücksichtigt. Es müssen Näherungen vorgenommen werden, um die Komplexität des Modells zu reduzieren und das Hauptaugenmerk auf das Modell eines Skispringers und die Konzeption eines biologisch/anatomisch motivierten Reglers für das Skispringen richten zu können. Im

folgenden Kapitel soll nun näher auf diejenigen biomechanischen Modellierungselemente, die wesentlich zur synthetischen Erzeugung des Skisprungs verantwortlich waren, eingegangen werden: Das Menschmodell an sich, den Kontakt Ski-Schanze, den Luftwiderstand mit Erweiterungen und die aktive Regelung bei der Anfahrt, zum Absprung, im Flug und bei der Landung.

3 Methoden

3.1 Modellierung

Das anthropometrische Menschmodell auf Basis statistischer Daten

HANAVAN [15] und HATZE [17] verwendeten bei ihrer Modellierung des menschlichen Körpers zur Abbildung der mechanischen Eigenschaften (Massen, Schwerpunktlagen und Segmentlängen) der Gliedmaßen des Menschen umfangreiche und aufwändige Vermessungen am lebenden Menschen. Indem sie geometrische Näherungen und geometrische Annahmen benutzten, konnten sie die im Rahmen ihrer Ansätze benötigten Daten erheben. Zu einer allgemeinen Beschreibung einer Bewegungsaufgabe im Rahmen der biomechanischen Modellierung, z.B. der Bewegungsaufgabe Skisprung, ist aber die exakte Abbildung einzelner Personen noch von untergeordneter Bedeutung. Zu diesem Zweck ist es nötig mit wenigen Vorgaben einen guten anthropometrischen Datensatz zu ermitteln.

In der Arbeitsgruppe Biomechanik am Institut für Astronomie und Astrophysik wurde hierzu der *Calcman* entwickelt. Er ist ein biomechanischer Präprozessor, der auf Grundlage statistischer Daten der NASA [27] und vom Deutschen Institut für Normung e.V. [5] ein anthropometrisches Menschmodell errechnet. Zur Erzeugung eines Datensatzes für ein Menschmodell benötigt *Calcman* die Vorgaben *Geschlecht*, *Masse* und *Körpergröße*, um aus den statistischen Daten und aus weiteren individuellen Angaben, z.B. Ausgangsorientierungen, ein individuelles Modell erstellen zu können. Diese individuellen Angaben müssen direkt im Programmcode vorgenommen werden. An unserem Institut benutzten bereits HAHN [14] und GÜNTHER [10] für ihre zweidimensionalen Modelle anstelle von geometrischen Annahmen und Modellen vorwiegend statistische Daten der NASA. Im Laufe der langen Entwicklung wurde der *Calcman* auf drei Dimensionen erweitert und enthält mittlerweile Ausgaberroutinen für verschiedene MKS-Pakete, zuletzt wurde nun die Ausgabe der Modellkonfigurationsdatei für das Simulationsprogramm *DySim* hinzugefügt. Zum theoretischen Hintergrund des *Calcman*-Verfahrens sei auf die Dissertation von HENZE [19] verwiesen.

Der *Calcman* berechnet folgende Daten:

- ◇ Die Segmentlängen,
- ◇ die Segmentmassen,
- ◇ die Segmentbreiten, -tiefen und -umfänge,
- ◇ die Trägheitstensoren der Segmente in körperfesten Koordinaten,

- ◇ die Position und Orientierung der Segmente in Inertialkoordinaten,
- ◇ die Position und Orientierung der Koordinatensysteme der Gelenke und
- ◇ die Position der Kontaktpunkte.

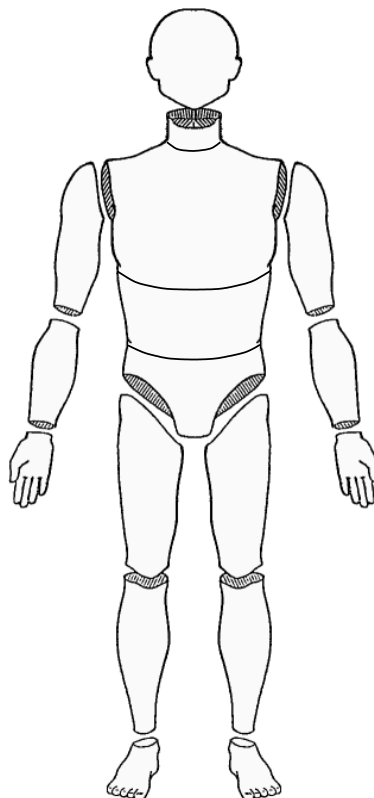


Abbildung 5: Die Abbildung zeigt die Segmentierung des Menschen nach der NASA (nach NASA [27]).

Das erstellte Menschmodell umfasst zum aktuellen Zeitpunkt folgende Körpersegmente: Fuß, Unterschenkel, Oberschenkel, Hüfte, Bauch/Lenden, Brust, Oberarm, Unterarm, Hand, Hals und Kopf. Bei der Erstellung dieses Modell liegt die Annahme zugrunde, dass der Körper symmetrisch zu seiner Medianebene ist, d.h. linke und rechte Seite sind anthropometrisch identisch.

Im Allgemeinen ist dieses Verfahren sehr gut dazu geeignet, um dreidimensionale Ganzkörpermodelle für biomechanische Fragestellungen zu entwickeln ohne auf individuelle Daten einzelner Versuchspersonen, sofern sie denn erhoben werden können, zurückgreifen zu müssen. Mit folgenden Vorgaben wurde aus der NASA-Datenbank das vorliegende Starrkörpermodell errechnet: Geschlecht *männlich*, Körpermasse *60kg* und Körpergröße *178cm*.

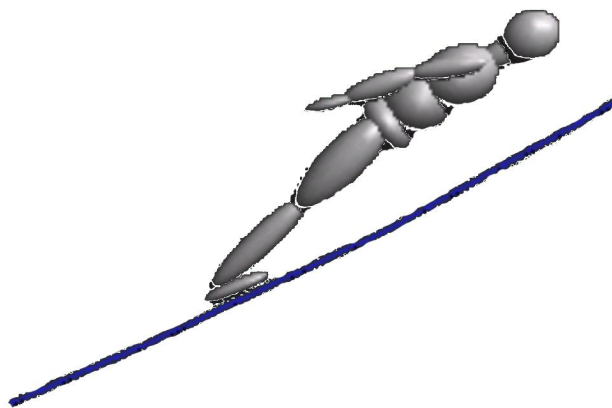


Abbildung 6: Das erstellte Skispringermodell: In der Flugphase lässt sich die Segmentierung deutlich erkennen, man kann sogar das durch den Luftwiderstand abgeklappte vordere Skiteil sehen. Die Trägheitseigenschaften lassen sich allerdings in dieser Darstellung nicht zeigen, da die Körpersegmente durch einfache Grafikprimitive repräsentiert werden.

Da bei der simulierten Bewegung Skisprung außer bei der Landung, die jedoch nicht angestrebt wurde, keine hochdynamischen Bewegungen auftreten sollten, wurde die Weichteildynamik bei der Modellierung nicht berücksichtigt. Daher sind keine Schwabbelmassen an das Starrkörpermodell angekoppelt.

Im Folgenden wird das erstellte Starrkörpermodell detaillierter beschrieben, wobei als Darstellung der Segmentanordnung eine Baumstruktur gewählt wurde (siehe Tabelle 3.1). Alle Körpersegmente sind mit ihren englischen Bezeichnungen versehen und fett gedruckt. Die Teilkörper sind durch Gelenke verbunden. Im vorliegenden Modell sind alle Gelenke vom Typ Scharniergelenk (engl.: revolute) außer dem ersten Gelenk zum Inertialsystem (**world**), das ein Gelenk mit sechs Freiheitsgraden ist, ein sogenanntes freies Gelenk (**Free**). Der dargestellte Baum enthält ausschließlich die Extremitäten einer Körperhälfte, da ein median-symmetrisches Modell angenommen wurde. Weitere Modellparameter, wie z.B. die Trägheitsmomente oder die Parameter der Drehfedern (Steifigkeit, Dämpfung), werden im Anhang aufgelistet [B].

Das Inertialsystem **world** ist so definiert, dass die Bewegung in der x-z-Ebene stattfindet, der Skispringer fährt und fliegt also zu größeren x-Werten. Die körperfesten Bezugssysteme sind so orientiert, dass die x-Achse in der Sagittalebene, die y-Achse in der Frontalebene und die z-Achse in der Transversalebene liegt.

An manchen Stellen verzweigt der Baum zu einem Element 'ContactSimple'. Dieses Element ist ein Kraftelement, das eingeschaltet wird sobald sich die zwei kontaktierenden Körper durchdringen. Dieses Kontaktelement ist in DySim schon vorhanden, es ist ein einfaches Punkt-Flächen-Kontaktelement. Für dieses Element müssen zwei Koordinatensysteme definiert werden, an jedem der beiden kontaktierenden Körper eines. Das eine Koordinatensystem definiert den Kontaktpunkt des Elements und die z-Richtung des anderen Koordinatensystems definiert die Flächennormale der Kontaktfläche (siehe DySim-Theorie [18]).

Im vorliegenden Modell kontaktieren der Kopf mit dem Skivorderteil, der Unterschenkel (Knie) mit dem Oberkörper und die Ferse mit der Skimitte. Die Kontaktpunkte werden jeweils so an den Segmenten positioniert und orientiert, dass sie der realen Kontaktsituation entsprechen, z.B. das Fersenkontakt dreibein ist am Fuß in x-Richtung zum Fußende und in z-Richtung zur Fußunterseite verschoben.

Außer den Kontaktkräften werden weitere Kräfte benutzt, um die Aktionen und Interaktionen des Modells realistisch abzubilden. So ist an jedem Scharniergelenk eine Drehfeder modelliert, die ausgehend von einer Nulllage bei Auslenkung eine rücktreibende Kraft produziert. Die Kräfte (und Drehmomente) sind abhängig von weiteren Drehfedereigenschaften, wie Federsteifigkeit und -dämpfung, die ebenfalls angegeben werden. Eine so modellierte Drehfeder ist ein passives Element, da sie nur auf Änderungen reagiert. Im vorliegenden Modell werden aber weitere Drehfedern definiert. Indem deren Nulllagen während der Simulation verändert werden, agieren diese als aktive Elemente (siehe Abschnitt *Aktive Modellregelung*).

Der Sprungski wurde in drei Segmente unterteilt: Skiende, Skimitte und Skifront. Zwischen den Segmenten sind Scharniergelenke definiert an denen Drehfedern die rücktreibende Kraft zur Erhaltung der Skiform liefern. Der Ski wurde in seiner Eigenschaft streng nach den Spezifikationen der Wettkampfausrüstung 2002/2003 der Fédération Internationale de Ski (FIS) [9] modelliert, in der sowohl der Befestigungspunkt der Bindung als auch Gewicht, Länge, Breite und Höhe festgelegt sind.

Der Schanzenkontakt

Für die Kontaktmodellierung Ski - Schanze musste ein neues Kontakt-Kraft-Element in DySim formuliert werden. Die Grundlage hierfür war das einfache Kontaktelement, das schon vorhanden war. Dieses Element benützt jedoch eine unendlich ausgedehnte Ebene in Richtung der x- und y- Achse des ersten Kontaktkoordinatensystems - im Folgenden wird anstatt Koordinatensystem von Triade gesprochen - als Kontaktfläche. Um das Schanzenprofil als Kontaktfläche zu modellieren, ist aber minde-

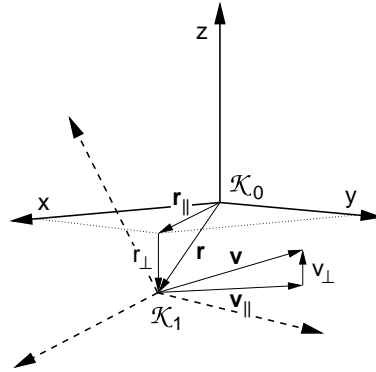


Abbildung 7: Die Definitionen in der Kontaktsituation beim Schanzkontakt. \mathcal{K}_0 legt den Punkt auf der Kontaktfläche fest und der Ursprung von \mathcal{K}_1 den Kontaktpunkt.

stens eine zweidimensional gekrümmte Fläche nötig. Das nun implementierte dritte Kontakt-Kraft-Element in DySim wird folgendermaßen definiert:

```

ContactSpline "NAME" {
  TriadFrom = NAME-CONTACTSPLINE-TRIAD;           →  $\mathcal{K}_{00}$ 
  TriadTo = NAME-CONTACTPOINT-TRIAD;              →  $\mathcal{K}_1$ 
  RangeX1 = FIRST-X-RANGE-VALUE-OF-SPLINE;
  RangeX2 = LAST-X-RANGE-VALUE-OF-SPLINE;        →  $\kappa_{\perp}$ 
  ElasticLinearZ = 100000.0;                       →  $\rho_{\perp}$ 
  DampingLinearZ = 1000.0;                          →  $\rho_{\parallel}$ 
  FrictionXY = 0.1;
}

```

Mit diesem Kontaktelement können auf einfache Weise Kontakte zwischen Punkten und zweidimensional gekrümmten Flächen beschrieben werden (vgl. Abbildung 7). Dabei wird die Kontaktfläche durch die Triade \mathcal{K}_{00} definiert: Die Triade legt den Bezugspunkt der Koordinatenangaben, welche durch eine separate Datei *Splinedaten* gegeben sind, im Modellraum fest. Die Datei *Splinedaten* trägt denselben Namen, wie die Triade `TriadFrom = NAME-CONTACTSPLINE-TRIAD` und muss im gleichen Verzeichnis wie die Konfigurationsdatei liegen. Sie enthält zwei Spalten: Eine Spalte für die x-Koordinaten und eine für die z-Koordinaten, relativ zum Bezugspunkt Triade \mathcal{K}_{00} . Zwischen den gegebenen Werten wird mit einem Spline interpoliert. Bei der Kontaktdetektion wird die Triade \mathcal{K}_0 berechnet aus der Position der Triade \mathcal{K}_1 in Inertialkoordinaten, der Position und Orientierung der Triade \mathcal{K}_{00} in Inertialkoordinaten und der Splinefunktion. Zum Kontakt kann es kommen, wenn der senkrechte relative Abstand r_{\perp} des Ursprungs von \mathcal{K}_1 zur Normale der interpolierten Funktion bezüglich von \mathcal{K}_0 negativ wird. In y -Richtung ist die Fläche unendlich ausgedehnt. Die Kontaktdissipation ist proportional dem Produkt zur senkrechten Komponente v_{\perp} der Relativgeschwindigkeit und des senkrechten relativen Abstandes

\mathbf{r}_\perp . Tangential zur Kontaktfläche wirkt lediglich eine Gleitreibungskraft, die proportional zur tangentialen Komponente \mathbf{v}_\parallel der Relativgeschwindigkeit der Triaden ist. In Koordinaten der Triade \mathcal{K}_0 lautet die Kontaktkraft auf den Körper an dem \mathcal{K}_{00} befestigt ist:

$$\mathbf{f} = \begin{pmatrix} f_\perp \\ \mathbf{f}_\parallel \end{pmatrix} = \begin{cases} \begin{pmatrix} \kappa_\perp r_\perp - \rho_\perp v_\perp r_\perp \\ \rho_\parallel \mathbf{v}_\parallel \\ \mathbf{0} \end{pmatrix} & : \text{solange } f_\perp < 0 \\ & : \text{sonst} \end{cases} .$$

Sobald die Vertikalkomponente der Kontaktkraft auf den \mathcal{K}_0 -Körper nicht mehr negativ ist, besteht kein Kontakt mehr. Dies kann auch der Fall sein, wenn r_\perp zwar negativ ist, sich aufgrund des Reibungsterms jedoch eine positive, d.h. anziehende Komponente f_\perp ergibt.

In der obigen Gleichung wird die parallele Kraft $f_\parallel = \rho_\parallel v_\parallel$ und nicht wie erwartet $f_\parallel = \rho_\parallel f_{Normal}$ benutzt. Dies ist bei den anderen bereits vorhandenen Kontaktkraft-Elementen genauso.

Mit den Angaben RANGEX1 und RANGEX2 wird der Gültigkeitsbereich der Kontaktfläche in x-Richtung des körperfesten Systems angegeben.

Zudem erhält das Element CONTACTSPLINE ein neues Grafikprimitiv für die spätere Animation des Modells (siehe Kapitel 3.2).

Die in dieser Arbeit verwendeten Schanzendaten stammen aus Aufzeichnungen der 90m-Schanze in Oberstdorf, könnten aber ohne Probleme durch andere Daten ausgetauscht werden.

Der Luftwiderstand mit Erweiterungen

Im vorangehenden Abschnitt wurde die Modellierung der Interaktion des Skispringers bzw. der Ski mit der Schanze dargestellt. Eine weitere sehr wichtige Interaktion des Skispringers mit seiner Umwelt soll nun Thema dieses Abschnitts sein: Die Interaktion des Springers mit der ihn umgebenden Luft.

Der Luftwiderstand F_W berechnet sich mit

$$F_W = 0.5 \cdot c_W \cdot \rho_{Luft} \cdot A_W \cdot v^2 \quad . \quad (10)$$

Mit der Dichte von Luft $\rho_{Luft} = 1.29 \text{ kg/m}^3$, dem Luftwiderstandsbeiwert c_W , der Widerstandsfläche A_W und dem Betragsquadrat der Relativgeschwindigkeit (Körper, Umgebungsluft) v^2 wird die Widerstandskraft berechnet, die entgegen der Richtung der Relativbewegung wirkt. Dabei ist c_W abhängig von der Form des Widerstandskörpers und die Widerstandsfläche A_W ist die Fläche, die effektiv in Richtung

der Bewegung erscheint. Genau so wurde die Widerstandskraft in das Simulationsprogramm `DySim` implementiert. Es müssen zwei Triaden definiert werden: Die eine an dem Körper dessen Luftwiderstand berücksichtigt werden soll und die andere am Bezugssystem `world` gegenüber der die Bewegungsrichtung berechnet wird. Die Orientierung der Triade am Körper definiert zusätzlich die Flächennormale der Luftwiderstandsfläche, die der Körper haben soll. Ein Beispiel: Vom Teilkörper `THORAX` soll der Luftwiderstand bei der Simulation berücksichtigt werden. Also definiert man zwei Triaden, die eine am Körper `THORAX` bestmöglich in seinem Schwerpunkt, die andere in `WORLD`, damit die Relativgeschwindigkeit der Körper gegenüber dem Inertialsystem ausgerechnet wird. In diesem Beispiel soll der Oberkörper nur eine Widerstandsfläche bekommen, die in der Frontalebene liegt. Deshalb wird die Triade am Körper um den Winkel $\pi/2$ um die y -Achse gedreht, damit die Flächennormale in der Sagittalebene aus dem Körper herauszeigt.

Mit diesen Angaben wird das Element in der Konfigurationsdatei definiert:

```
AerodynSimple "NAME" {  
  TriadFrom = Triad-to-calculate-the-relative-velocity;  
  TriadTo = Triad-on-body;  
  Plane = 0.0;  
}
```

Das Programm `DySim` berechnet abhängig von der momentanen Bewegungsrichtung des Teilkörperschwerpunkts die effektive Widerstandsfläche. Mit der effektiven Widerstandsfläche, seiner Geschwindigkeit und dem c_W -Wert von 0.5 für einen Zylinder wird dann die Widerstandskraft berechnet, die im Teilkörperschwerpunkt entgegen seiner Bewegungsrichtung wirkt.

Körper, die in Luft bewegt werden, erfahren, je nach Form und Orientierung, zusätzlich zum Bewegungswiderstand einen Auftrieb senkrecht zur Bewegungsrichtung. Dieser Auftrieb muss in der Modellierung des Skispringers ebenfalls berücksichtigt werden. Ein erster Ansatz führt zu einer Energiebetrachtung des Luftwiderstandes.

Der vorhergehende Abschnitt erläutert die Berechnung des Luftwiderstand, es wird aber keine Aussage über die Energiedissipation getroffen. Angenommen - Gedankenexperiment - nicht 100 Prozent der Widerstandsenergie dissipiert durch Wirbelbildung und Wärmeentwicklung, sondern ein Teil $alpha$ dissipiert durch Auftrieb, dann könnte man die Auftriebskraft als Teil der Widerstandskraft berechnen:

$$F_A = alpha \cdot F_W$$

Gedankenexperiment: Ein Teil der Energie, die dem System durch Luftwiderstand entzogen wird, nutzt das System um Auftrieb zu erzeugen. Das bedeutet, dass der Betrag der Auftriebskraft durch die Widerstandskraft berechnet werden kann. Die Richtung der Auftriebskraft muss senkrecht zur Bewegungsrichtung und in Richtung des Anstellwinkels zeigen (schräggestellte Fläche). Mit dieser Art der Berechnung des Auftriebs ist man physikalisch nicht exakt, da weder BERNOULLIS Gesetze noch die Winkelposition der Fläche zur Relativgeschwindigkeit des Körpers auftriebswirksam berücksichtigt werden und zusätzlich Widerstand und Auftrieb linear gekoppelt sind, d.h. bei großem Widerstand erhält man großen Auftrieb. Einzige Ausnahme: Aufgrund der Berechnung der Richtung des Auftriebs über die orthogonale Projektion der Flächennormale auf die Widerstandskraft erhält man im Fall maximalen Auftrieb für die Projektion *Null* und somit keinen Auftrieb. Eine Berechtigung für diese Art der Berechnung des Auftriebs liefern zwei Argumente: *Erstens* lässt sich in der Simulation ein Gefühl für die Aerodynamik entwickeln und *zweitens* befindet sich der Skispringer im Flug an allen auftriebswirksamen Körpersegmenten in einer mittleren Winkelposition, in denen ein linearer Zusammenhang zwischen Luftwiderstand und Auftrieb in erster Näherung betrachtet werden darf.

In einem nächsten Ansatz wird die Strategie verfolgt, wie sie im Beitrag “Ikarus’ Traum und die aerodynamische Wirklichkeit” in Praxis der Naturwissenschaften-Physik [33] diskutiert wird. Der Artikel beschreibt die Herleitung des Auftriebs aus allgemeinen Erfahrungen z.B. die Hand im Luftstrom und aus einigen Proportionalitätsexperimenten z.B. Verdopplung der Strömungsgeschwindigkeit. Basierend auf NEWTONS drittem Axiom *actio = reactio* ergibt sich dann der dynamische Auftrieb zu:

$$F_A = 0.5 \cdot c_A \cdot \rho_{Luft} \cdot A_F \cdot v^2 \quad . \quad (11)$$

Die Ähnlichkeit der Struktur zur Formel für die Berechnung des Luftwiderstands ist offensichtlich. Es wird hierbei ebenso die Dichte der Luft ρ_{Luft} und das Betragsquadrat der Relativgeschwindigkeit v^2 berücksichtigt. Als Auftriebsfläche A_F wird die gesamte Körperfläche betrachtet. Während bei der Berechnung der Widerstandskraft der Beiwert c_W , der den Einfluss der Körperform beschreibt, in erster Näherung als konstant betrachtet wird, ist

$$c_A = c_A(\text{Winkel}(\text{Körperfläche}, \text{Bewegungsrichtung}), \text{Form der Auftriebsfläche}) \quad .$$

Im oben angesprochenen Beitrag wird allerdings keine Funktion für c_A vorgeschlagen, sodass diese in einem Gedankenversuch gefunden werden musste.

Gedankenversuch: Betrachten wir ein Blatt Papier, das horizontal fortbewegt wird und variieren dabei den Anstellwinkel zwischen 0 und 90 Grad relativ zur Horizontalgeschwindigkeit, so erhalten wir folgende Aussage für den Auftrieb:

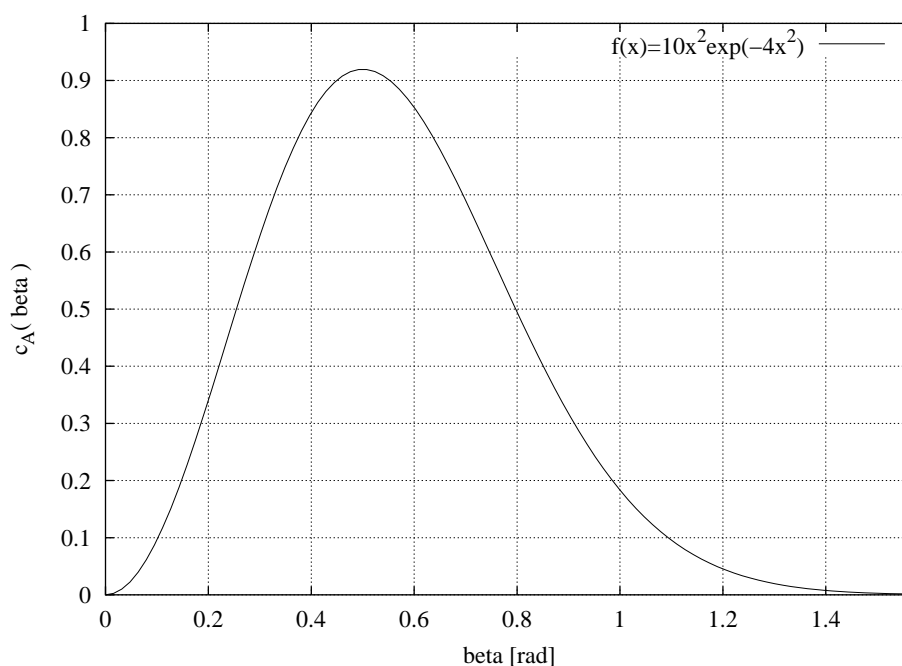


Abbildung 8: Der Auftriebsparameter c_A als Funktion des Anstellwinkels β . Eine der MAXWELLSchen Geschwindigkeitsverteilung ähnliche Funktion $f(x) = 10 \cdot x^2 \cdot \exp(-4 \cdot x^2)$. Das Maximum der Funktion zwischen 0 und $\pi/2$ legt den maximalen c_A -Wert $c_{A_{max}}$ fest und der x-Wert des Maximums den optimalen Auftriebswinkel.

- ◇ Bei 0 und 90 Grad gibt es keinen Auftrieb.
- ◇ Zwischen 0 und 90 Grad hat der Auftrieb ein Maximum.
- ◇ Zu Beginn der Winkeländerung steigt der Auftrieb stark an und fällt danach wieder ab.

Um diesen Sachverhalt in Form einer Funktion $c_A(\beta)$ auszudrücken, wurde die Funktion

$$f(x) = a \cdot x^2 \cdot \exp(-b \cdot x^2) \quad (12)$$

gewählt, ähnlich der MAXWELLSchen Geschwindigkeitsverteilung.

Die Punkte $c_{A_{max}}$ und $\beta_{c_{A_{max}}}$ können mit den Parametern a und b festgelegt werden. Die Abhängigkeit von c_A von der Form der Auftriebsfläche wird als konstant betrachtet und ist im Parameter a enthalten.

Mit dieser Form der Beschreibung des dynamischen Auftriebs ist es möglich den Auftrieb gegenüber dem Luftwiderstand zu optimieren um einen möglichst günstigen Anstellwinkel für maximalen Auftrieb bei minimalem Widerstand zu erhalten (siehe dazu Kapitel 5). Die Ergebnisse der Berechnung des Luftwiderstandes und zur Berechnung des dynamischen Auftriebs werden im Kapitel *Ergebnisse* diskutiert.

Die aktive Modellregelung

Die bis zu diesem Zeitpunkt angesprochenen Elemente der Modellierung ermöglichen es dem Skispringermodell, eine beliebige Schanze hinunterzugleiten, wobei das Modell durch den Luftwiderstand abgebremst würde. Manche Teilsegmente mit “gutem” Anstellwinkel würden dabei Auftrieb produzieren. Alle Elemente haben aber eines gemeinsam, sie wirken von außen auf das Modell ein, d.h. das Modell ist passiv. Der folgende, letzte Abschnitt in diesem Teil des Methodenkapitels beschreibt wie im Modell die aktiven Komponenten umgesetzt wurden, um innere Kräfte zu produzieren, damit

- ◇ bei der Anfahrt Positionsregelung stattfindet,
- ◇ der Absprung durchgeführt wird und
- ◇ zur Fluglage übergegangen werden kann sowie
- ◇ die Flugphase reguliert wird und
- ◇ der Landevorgang eingeleitet werden kann.

Zwei Vorbemerkungen zur Modellregelung: Eine Möglichkeit der Regulation der Anfahrtsposition ohne eine aktive Komponente wäre, die Steifigkeitswerte in den Drehfedern der Gelenke sehr hoch zu wählen. Dazu ist, um die Federn aus ihren Nulllagen zu bringen, ein hohes Drehmoment notwendig, ansonsten verharrt das Modell in der vorgegebenen Position. Dieses Vorgehen entspricht einem mechanistischen Ansatz, der im Rahmen dieser Arbeit nicht weiter verfolgt werden soll (siehe Einleitung, Kap. 1). Wie erwähnt können bei biomechanischen Messungen die Steifigkeitswerte in den menschlichen Gelenken bestimmt werden und sollen im Rahmen eines anatomisch/physiologischen Ansatzes, wie ihn diese Arbeit verfolgt, Verwendung finden.

Eine weitere Vorbemerkung zur Regelung des Modells beschäftigt sich mit Regelungen, die in der Literatur zu finden sind [10]. Für diese Regelungen bedarf es einiger Modellinformationen, z.B. der Position des Körperschwerpunktes, der Unterstützungsfläche und der Geschwindigkeit des Körperschwerpunktes, aus denen die Antwort des Systems berechnet wird. Diese Regelungsantwort ist vielschichtig, z.B. bei kleinen Störungen findet eine Positionsregelung nur im Sprunggelenk statt, während

bei großen Störungen die Positionsregelung bei allen Körpersegmenten durchgeführt wird. Die Idee der vorliegenden Arbeit besteht in der Realisierung eines reduktionistischen Ansatzes. Hierbei soll mit möglichst wenigen Parametern und auf möglichst einfache Weise eine komplexe Bewegung zustande gebracht werden, dies gilt insbesondere auch für die Komplexität der Regelungsprozesse.

Zu diesem Zweck wurde die Idee der Ziellängen (λ -Modell nach Feldman [7]) aufgegriffen und stark vereinfacht auf die aktive Regelung angewandt. Dazu musste die Gesamtbewegung "Skisprung" in vier Phasen eingeteilt werden, sie lauten:

- ◇ Anfahrtsphase,
- ◇ Sprungphase,
- ◇ Flugphase und
- ◇ Landephase.

Jede Phase zeichnet sich dabei durch einen speziellen Satz von Gelenkwinkeln aus, den Zielkonfigurationen - auch Nulllagen genannt.

Die Regelung funktioniert so, dass in der Simulation zwischen diesen Zielkonfigurationen zustandsabhängig umgeschaltet wird, damit das Modell von einer Zielkonfiguration in eine nächste übergehen kann. Hierzu wurde ein neues Kraftelement für DySim programmiert, das **ARSDA**. Dieses Kraftelement hat dieselben Eigenschaften wie das bereits vorhandene Kraftelement **RSDA**¹ (siehe DySim-Theorie [18]), jedoch können zusätzlich verschiedene Zielkonfigurationen definiert werden. Durch die Angabe mehrerer Zielkonfigurationen besteht die Möglichkeit, während der Simulation die Zielkonfigurationen der Drehfedern zu verändern. Das Umschalten der Zielkonfiguration bewirkt, dass die Drehfeder plötzlich um die Auslenkung $\phi = (\text{Nulllage1} - \text{Nulllage2})$ gespannt ist. Diese Vorspannung führt zu einem Drehmoment, das eine Bewegung in Richtung der neuen Nulllage produziert. Das erzeugte Drehmoment hängt von der Differenz der Nulllagen und der Steifigkeit der Drehfeder ab. Die spezielle Verwendung dieses neuen Elements in den einzelnen Phasen des Skisprungs wird im Folgenden besprochen.

Wie in allen Phasen der Bewegung muss das Modell auch in der Anfahrtsphase eine bestimmte Zielkonfiguration einnehmen. Diese Zielkonfiguration wird allerdings durch verschiedene Einflüsse, z.B. den Luftwiderstand, die Reibung der Ski an der Schanze und der Zentrifugalkraft in der Kurvenbahn kurz vor dem Schanzentisch

¹Bei diesem Kraftelement handelt es sich um eine eindimensionale Drehfeder, die an einem Scharniergelenk angebracht ist und abhängig von der Nulllage und der momentanen Auslenkung ein Drehmoment erzeugt, um die Nulllage einzunehmen.

(Kompression) gestört. Deshalb ist es dort nötig zwei weitere Zielkonfigurationen zu definieren, eine Konfiguration um aus der aktuellen Position, die möglicherweise zu weit vor der Zielkonfiguration *Anfahrt* liegt, *zurück* zu kommen und eine, die das Gegenteil bewirkt, also die Bewegung nach *vorne* einleitet. Ähnlich einer Temperaturregelung, die zwischen zwei Schranken pendelt um eine gewünschte Zieltemperatur zu erreichen, pendelt dabei das Modell bei der *Anfahrt* um die Zielkonfiguration *Anfahrt* indem in der Simulation zwischen *vor* und *zurück* umgeschaltet wird. Dieses Umschalten kommt dadurch zustande, dass ständig die Fersenkraft kontrolliert wird. Übersteigt diese den Wert $F_{Ferse} = -400N$, an jeder Ferse $F_{Ferse,links,Ferse,rechts} = -200N$, in negativer Richtung, wird als neue Zielkonfiguration die *Vor*-Konfiguration angestrebt, andernfalls umgekehrt.

Im Bereich des Schanzentischs, genau dann, wenn der Schwerpunkt des Fußsegments sich bei $x_{1,inertial} = 61,5 m$ befindet, schaltet die Simulation auf Sprung. Bei dieser Konfiguration wird nicht zwischen Schranken geregelt, sondern das Modell strebt möglichst schnell die Endlage an. Sobald sich der Skispringer vom Schanzentisch gelöst hat, bei Fußgelenkschwerpunkt $x_{2,inertial} = 65,5 m$, wird in die Flugkonfiguration geschaltet. Diese Konfiguration stellt die angenommene ideale Fluglage dar. Die Werte für x_1 und x_2 sind frei gewählt, decken sich aber mit Beobachtungen in der Praxis, wonach ungefähr in den letzten 4 – 6 m der *Anfahrt* der Sprung eingeleitet wird. Außerdem zeigen Beobachtung der Simulation, dass das Skispringermodell bei Absprüngen in diesem Bereich die Beine bis zum Verlassen des Schanzentischs durchstrecken kann.

Während der Flugphase werden ähnlich wie bei der *Anfahrt* Störungen, wie z.B. eine Rotation vorwärts bzw. rückwärts durch einen schlecht abgestimmten Absprung oder Luftwiderstandseinflüsse, zwischen zwei Schranken geregelt. Genau wie in der *Anfahrtsregelung* existieren zwei Konfigurationen zwischen denen abhängig von einem Regelkriterium umgeschaltet wird. Die Konfiguration *zurück* vergrößert den Luftwiderstand der oberen Extremitäten und führt zu einem Drehmoment das eine Rückwärtsrotation des Körperrumpfes zur Folge hat, die Konfiguration *vor* bewirkt das Gegenteil, der Körperrumpf rotiert nach vorne. Während bei der *Anfahrt* die Kontaktkraft der Ferse die Winkelkonfiguration der aktiven Gelenke bestimmt, wird in der Flugphase die Orientierung des Oberkörpersegments THORAX berechnet. Die Zielorientierung ist $\phi_{Thorax} = 65$ Grad um die y -Achse gemessen von der Vertikalen. Ist die Oberkörperorientierung größer als die Zielorientierung wird die Konfiguration *zurück* eingeschaltet, ist die Oberkörperorientierung kleiner als die Zielkonfiguration wird die Konfiguration *vor* eingeschaltet.

Die Landekonfiguration wird zustandsabhängig eingenommen. Wenn sich der Fußsegmentenschwerpunkt unterhalb von $z_{Diff} = 1,5 m$ über dem Aufsprunghügel befindet, schaltet die Regelung in die Landekonfiguration. Die Landung erfordert eine gut abgestimmte Bewegungsregelung, welche bei dieser Arbeit nicht erreicht werden soll-

te. Deshalb benutzt die vorhandene Zielkonfiguration nur das Ziel, einen möglichst tiefen Schwerpunkt des Modell zum Landezeitpunkt zu erreichen. Erfreulicherweise zeigte die Simulation, dass damit durchaus eine realistische Landung gelingen konnte, siehe Ergebnisse 4.7. Die genauen Winkelkonfigurationen lassen sich im Anhang B nachschlagen, dort sind bei den Elementen *ARSDA* unter *l0X* die Werte der Gelenkwinkel in Grad notiert. *X* steht dabei für die verschiedenen Konfigurationen:

- ◇ l0: Zielkonfiguration *Anfahrt*
- ◇ l01: Konfiguration *Anfahrt zurück*
- ◇ l02: Konfiguration *Anfahrt vor*
- ◇ l03: Zielkonfiguration *Sprung*
- ◇ l04: Zielkonfiguration *Flugphase*
- ◇ l05: Konfiguration *Flug zurück*
- ◇ l06: Konfiguration *Flug vor*
- ◇ l07: Zielkonfiguration *Landung*

3.2 Simulation

Die Numerik

Ziel der Simulation ist es, die bei der Modellierung aufgestellten Bewegungsgleichungen für bestimmte Zeitintervalle zu lösen. Daraus erhält man die Zeitentwicklung des aufgestellten Systems. Eine analytische Lösung der Bewegungsgleichungen ist aber schon bei einem Schwerependel nicht mehr möglich. Bei mehrgliedrigen Körpern mit einigen Zwangsbedingungen und eventuellen nichtlinearen Kräften lässt sich dieses Ziel dann ganz sicher nicht mehr analytisch erreichen. Die Bewegungsgleichungen müssen numerisch integriert werden. In der Literatur gibt es zahlreiche Verfahren zur numerischen Lösung von ODEs (Ordinary Differential Equations), z.B. der Integrator *de* von SHAMPINE und GORDON [35]. Dieser Integrator wird unter anderem auch in DySim benutzt und wurde im Rahmen dieser Arbeit eingesetzt.

In der Konfigurationsdatei kann festgelegt werden, welcher der in DySim eingebauten Integratoren verwendet werden soll. Im sogenannten *Header*-Block erlaubt eine Zeile dessen Angabe, für den eingesetzten *de*-Integrator ist 'de' anzugeben. Neben der Wahl des Integrators, lassen sich aber auch dessen Eigenschaften über die Konfigurationsdatei steuern: Die Angabe des relativen und absoluten Fehlers erlaubt eine

Festlegung der numerischen Fehlergrenzen, es lässt sich der Start- und Endzeitpunkt der Integration festlegen und die Ausgabeschrittweite vom einen zum nächsten Integrations-schritt.

Der absolute Fehler beschreibt für den Fall eines Zustandes nahe Null die höchstzulässige Abweichung des Integrationsergebnisses von der absoluten Null. Der relative Fehler gibt die maximal erlaubte Differenz zweier verschiedener Integrationslösungen an. Diese Fehlergrenzen werden von einigen Integratoren, auch vom *de* dadurch eingehalten, dass die Integrationsschrittweite automatisch angepasst wird. Mit der Schrittweite *Tstep* wird nur die Häufigkeit der Datenausgabe festgelegt, was zur späteren Datenbearbeitung, z.B. bei einer Fouriertransformation, von Interesse ist.

Die Anfangsbedingungen

Zur numerischen Lösung der Bewegungsgleichung werden Anfangsbedingungen benötigt, welche das System eindeutig definieren. Die Anfangsbedingungen in dieser Simulation sind Lagen, also Position und Orientierung, und Geschwindigkeiten, also Translations- und Winkelgeschwindigkeit. Im Abschnitt Modellierung wurde bei der Menschmodellierung schon die kinematische Kette erwähnt. Diese kinematische Kette bildet die hierarchische Struktur des Modells und deren Verbindungsglieder sind die Gelenke. Anfangsbedingungen in Gelenken bestimmen daher das Modellsystem eindeutig. Die Wahl der Anfangsbedingungen für das Skispringermodell wurde im *Calcman* durchgeführt und ist in der Modellkonfigurationsdatei *skijump.dys* (siehe Anhang B) abgespeichert.

Ein Beispiel: Um Integrationszeit zu sparen, startet der Skispringer nicht aus der Ruhe oben an der Schanze, sondern bei $x_{0,inertial} = (30, 0, 80.2)^T m$ mit einer Anfangstranslationsgeschwindigkeit $v_{0,inertial} = (25, 0, -20)^T m/s$.

Die Animation

Die Animation der Daten dient zur visuellen Überprüfung der Simulationsergebnisse. Sie kann direkt mit eigenen Beobachtungen oder Videomaterial verglichen werden. Somit ist einerseits schnell ein qualitativer Eindruck über die Richtigkeit der simulierten Bewegung gewonnen und andererseits lassen sich damit Details mehrfach visuell studieren.

Um dies zu erreichen, müssen bei der Modellierung Grafikprimitive für alle anzuzeigenden Elemente vergeben werden. Meistens werden die Teilkörper und verschiedene Triaden an den Körpern visualisiert. Bereits im Abschnitt *Modellierung* (Kap. 3.1) wurde auf ein neu programmiertes Grafikprimitiv hingewiesen. Die Neuformulierung war nötig, um eine Darstellung der interpolierten Datenpunkte der Skischanze zu ermöglichen. Das Grafikprimitiv *Spline2D* berechnet aus der interpolierten Funktion 10000 Werte und stellt für jeden Wert eine Linie mit der Länge 2 m in Richtung der y-Achse des körperfesten Koordinatensystems dar.

Alle Grafikprimitive werden ebenfalls in der Konfigurationsdatei angegeben und abgespeichert. Bei der Integration der Bewegungsgleichung werden diese Grafikprimitive zwar nicht berücksichtigt, es wird jedoch eine Animationsdatei *skijump.ani* angelegt, in der die zeitliche Lageentwicklung der Teilkörper abgespeichert wird. Bei der Animation des Modells mit dem Programm AniDySim können so mithilfe der Konfigurationsdatei *skijump.dys* und der Animationsdatei *skijump.ani* die integrierten Bewegungsgleichungen als Bildfolge dargestellt werden.

In der Animation ist die Beobachterposition frei wählbar, sodass es dem Modellierer möglich ist, die Simulation aus allen Perspektiven zu betrachten. In Abbildung 9 und 10 ist eine Bildsequenz des Absprungs dargestellt.

Daten und Diagramme

Zur Ausgabe der Integrationsergebnisse zu den jeweiligen Zeitschritten verwendet DySim drei Dateien:

- ◇ *skijump.frc* zur Ausgabe aller Kräfte und Drehmomente der Körper und Gelenke.
- ◇ *skijump.pva* zur Ausgabe der Lage, Geschwindigkeit und Beschleunigung des Gesamtsystems und der Teilkörper.
- ◇ *skijump.epl* zur Ausgabe der Energie-, Impuls- und Drehimpulsdaten des Gesamtsystems und der Teilkörper.

Die Ausgaberroutine für die Datei *skijump.frc* musste erweitert werden, damit auch die neu programmierten Krafterelemente mit ausgegeben werden. Hierzu wurde die Ausgabe der Kraft des einfachen Kontaktelements, des Schanzenkontakts, der Aerodynamikkkräfte und der aktiven Drehfedern hinzugefügt.

Mit einem Grafikplotprogramm, wie z.B. *Gnuplot* lassen sich die in Tabellen abgespeicherten Daten anschaulich in Diagramme drucken und bearbeiten.

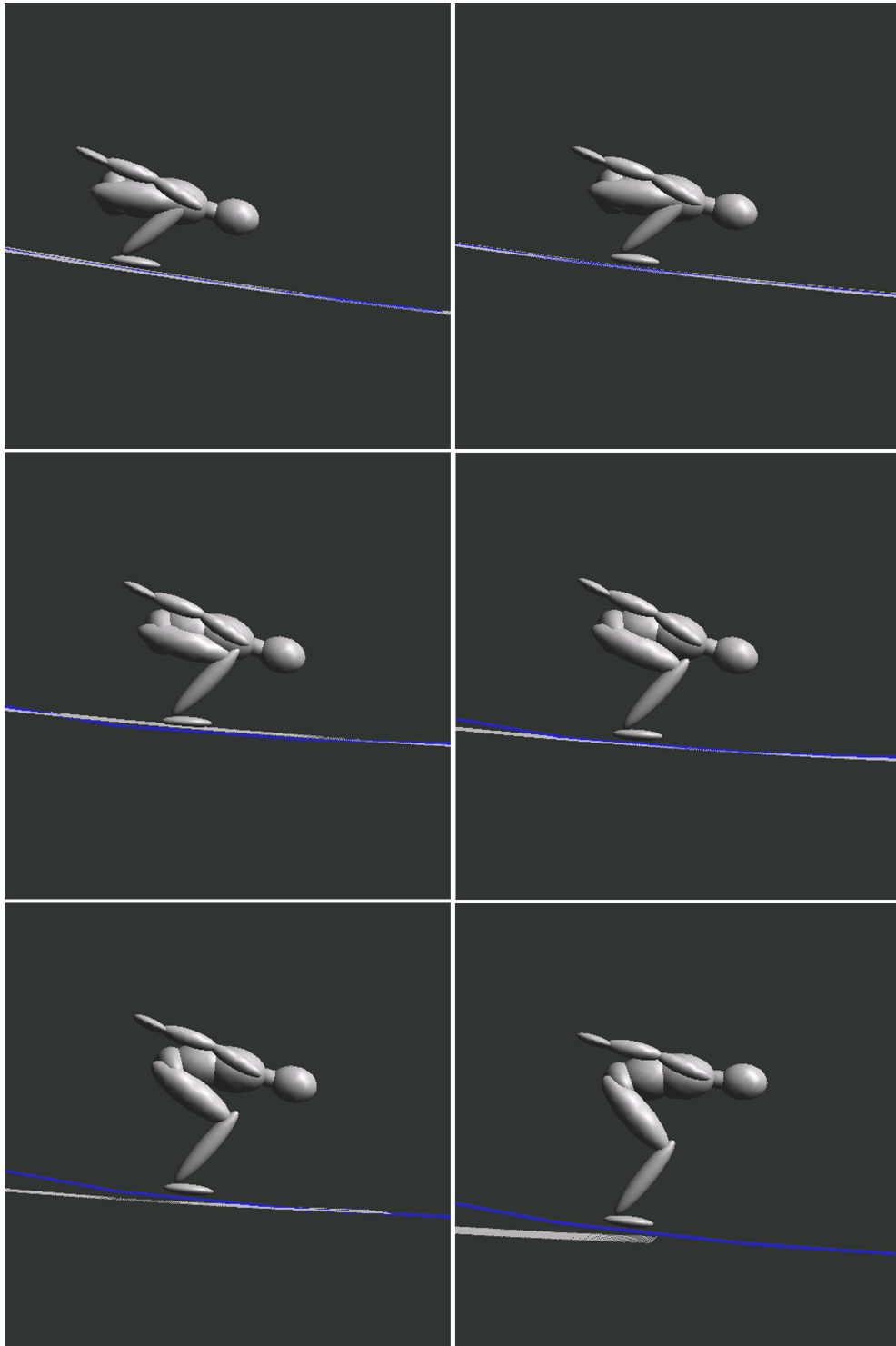


Abbildung 9: Hier ist eine Bildsequenz dargestellt, die den Absprung zeigt (Teil 1).

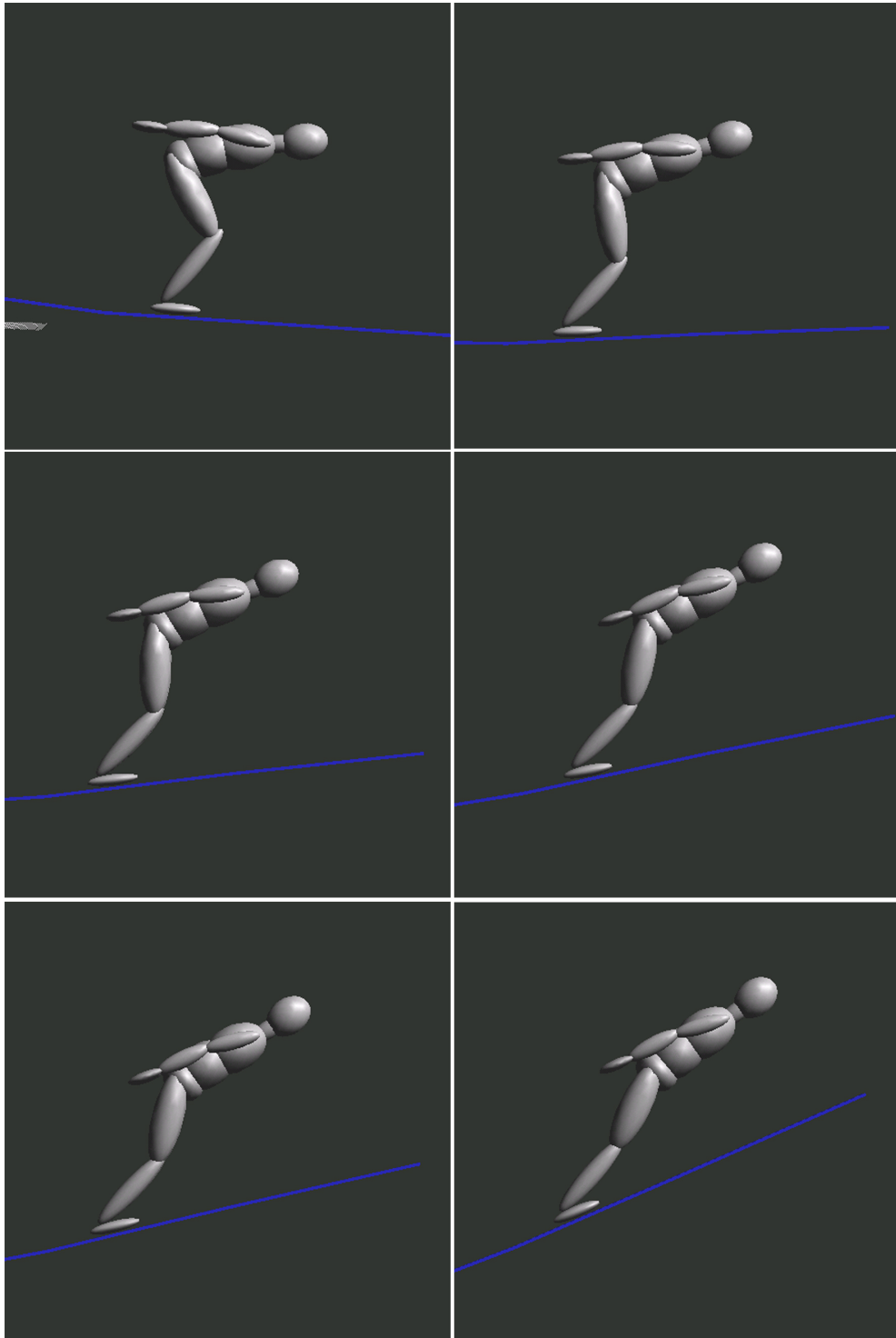


Abbildung 10: Hier ist eine Bildsequenz dargestellt, die den Absprung zeigt (Teil 2).

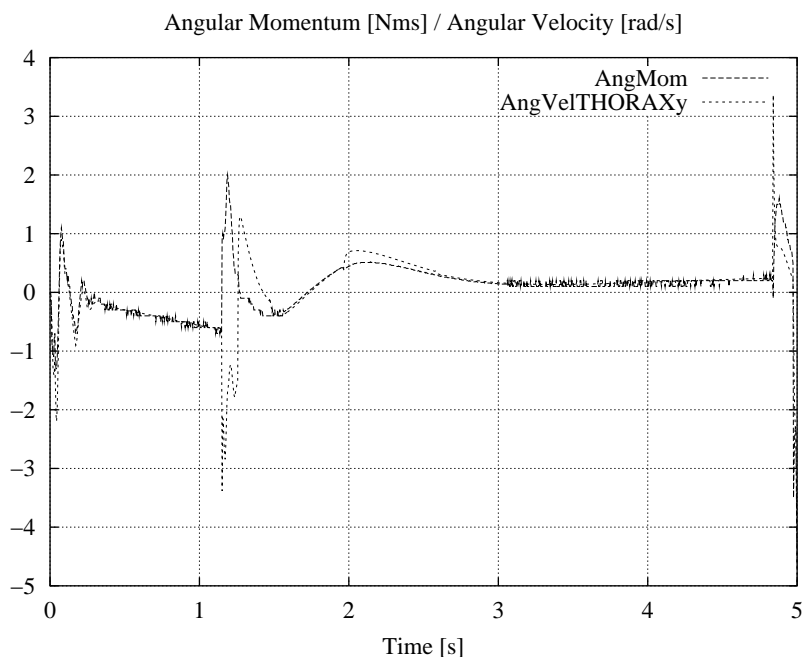


Abbildung 11: Drehimpuls mit Aerodynamikkräften. Die Flugphase beginnt bei $t = 1,7$ s. Deutlich ist die Drehimpulsänderung zu erkennen, die durch die Interaktion des Modells mit dem Umgebungsmedium resultiert.

4 Ergebnisse und Diskussion

4.1 Überprüfung der Drehimpulserhaltung

Zur aktiven Regelung des Modells werden, wie in Kapitel 3.1 bereits dargestellt, Drehfedern verwendet, deren Ruhelagen verändert werden. Dadurch wird ein Drehmoment erzeugt, welches die Energie und den Drehimpuls des Gesamtsystems verändern kann. Die physikalisch richtige Modellierung kann durch die Überprüfung der Drehimpulserhaltung durchgeführt werden. D.h. ohne Interaktion mit der Umgebungsluft muss der Drehimpuls in der Flugphase auch dann erhalten bleiben, wenn die Regelung aktiv ist. Die beiden Abbildungen (Abb. 11 und Abb. 12) zeigen die Differenz zwischen dem Drehimpuls des Gesamtsystems und dem Bahndrehimpuls des Modells und die Winkelgeschwindigkeit des Körpers THORAX um seine y-Achse.

In Abbildung 11 sind die Ergebnisse dargestellt für den Fall mit Luftreibung und Auftrieb und in Abbildung 12 ohne Interaktion mit dem Umgebungsmedium. Die Flugphase beginnt bei $t_1 = 1,7$ s für den Fall mit Luftreibung (Abb. 11) und bei $t_2 = 1,5$ s für den Fall ohne Luftreibung (Abb. 12), davor kann das Skispringermodell mit der Umgebung Skischanze Drehimpuls austauschen, danach nur im Falle

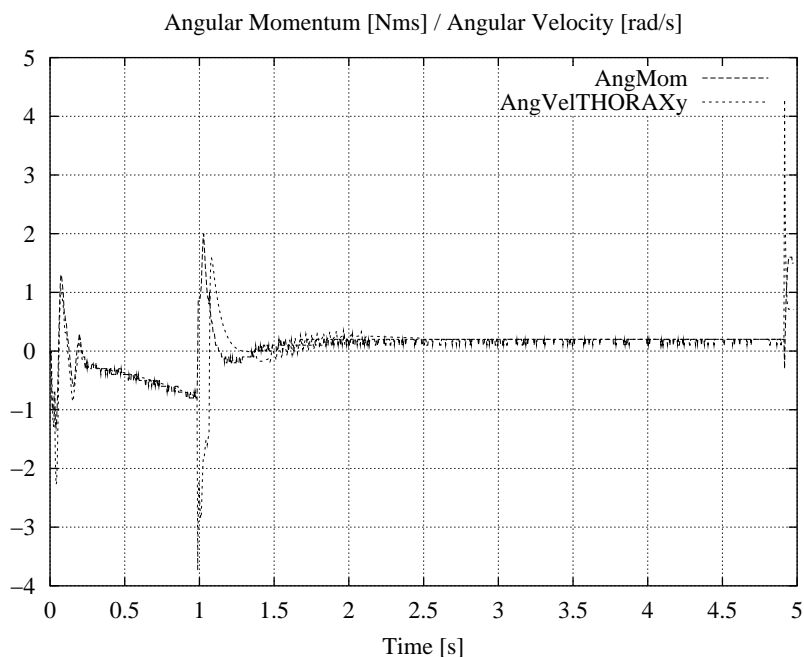


Abbildung 12: Drehimpuls ohne Aerodynamikkräfte. Nach Beginn der Flugphase bei $t = 1,5$ s findet kein Drehimpulsaustausch des Skispringermodells statt. Der Drehimpuls bleibt konstant knapp über 0.

des eingeschalteten Kraftelements Luftreibung (siehe Abb. 11). Dabei wird die Folgebewegung der Winkelgeschwindigkeit zum Drehimpuls gut sichtbar. Dagegen muss ohne Luftreibung die Drehimpulserhaltung in der Flugphase gezeigt werden, was in Abbildung 12 durch einen konstanten Wert bestätigt wird. Die durchgeführten Änderungen an den Kraftelementen von DySim gehen also konform mit der Physik, der Drehimpuls des Gesamtsystems bleibt erhalten.

4.2 Regelung

Während der Bearbeitung der Anfahrtsphase wurden zwei Regelungsstrategien verfolgt (Kap. 3.1), wobei sich eine aufgrund der Einfachheit und einer direkteren anatomisch/physiologischen Begründbarkeit durchgesetzt hat. Die Regelung über die Kontaktkraft zwischen der Ferse und dem Ski erweist sich als die bessere Lösung, da ein einfacher Parameter, die Fersenkraft, ausreicht, um zwischen den Konfigurationen *vor* und *zurück* umzuschalten und die Zielkonfiguration *Anfahrt* einhalten zu können. Abbildung 14 zeigt den Momentverlauf im Sprunggelenk bei der Anfahrt. Zusätzlich ist der Kraft-Zeit-Verlauf des Fersenkontakts am linken Bein dargestellt (Abb. 13). Zur Regelung der Anfahrtslage wurden die Winkel in den Gelenken zwi-

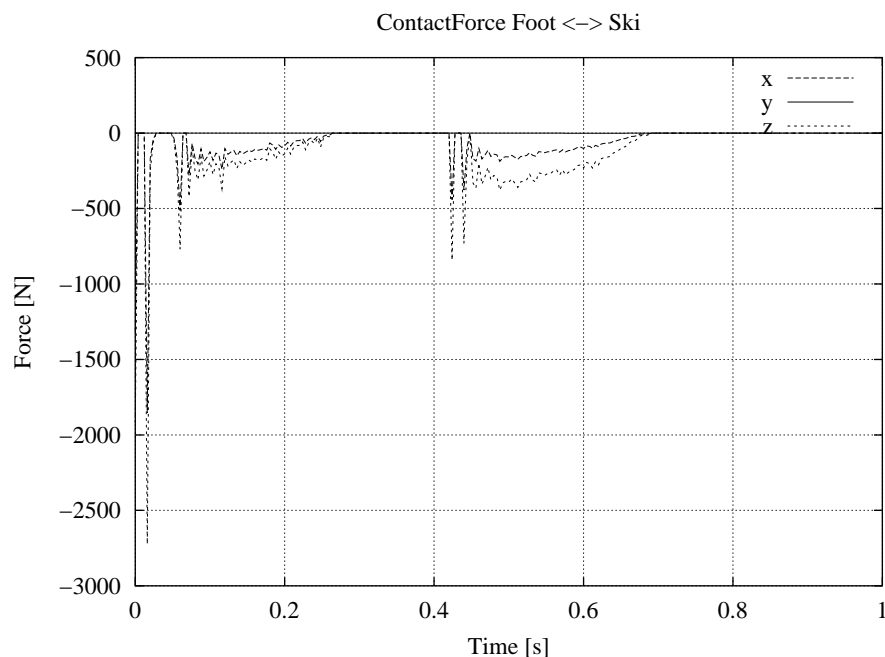


Abbildung 13: Wie bereits im Text beschrieben wird während der Simulation die Kontaktkraft zwischen der Ferse und dem Ski gemessen und abhängig von ihrer Größe das Modell nach *vorne* oder *zurück* geregelt. Dargestellt ist die Kontaktkraft des linken Fußes bei der Anfahrt in der ersten Sekunde. Im Vergleich mit Abb. 14 ist deutlich zu sehen, wie das Modell auf die Veränderungen in der Kontaktkraft reagiert.

schen Fuß/Unterschenkel, Unterschenkel/Oberschenkel, Oberschenkel/Hüfte verändert und zusätzlich Luftwiderstandseffekte des Kopfes und der Arme ausgenützt.

Die bisher in diesem Kapitel vorgestellten Ergebnisse sind mit Dämpfungswerten in den Gelenken erstellt worden, wie sie GÜNTHER [10] verwendet. Diese geringen Werte führten allerdings nach der Anfahrtsphase, vor allem beim Absprung zum Kollaps des Modells, weshalb zur weiteren Betrachtung der Ergebnisse ein Skispringermodell mit höheren Werten für die Gelenkdämpfung gewählt wurde (siehe hierzu die Diskussion in Kapitel 4.8 und die Modellkonfigurationsdatei im Anhang B).

Die Anlaufphase geht direkt in die Absprungphase über, zustandsabhängig bei $x = 61,5 \text{ m}$ in Inertialkoordinaten (der Schanzentisch endet bei $x = 65 \text{ m}$). Nach dem Absprung geht der Skispringer bei $x = 65,5 \text{ m}$ in die Flugphase über, die wieder eine aktive Lageregelung beinhaltet.

Die Regelung der Fluglage wird über die Orientierung des Oberkörpers durchgeführt. Es wird erneut zwischen zwei Konfigurationen umgeschaltet, wobei dazu die Stellung der Arme, des Kopfes und des Oberkörpers so variiert wird, dass sich der Widerstand erhöht oder verringert und somit eine Kraft außerhalb des Körperschwerpunktes

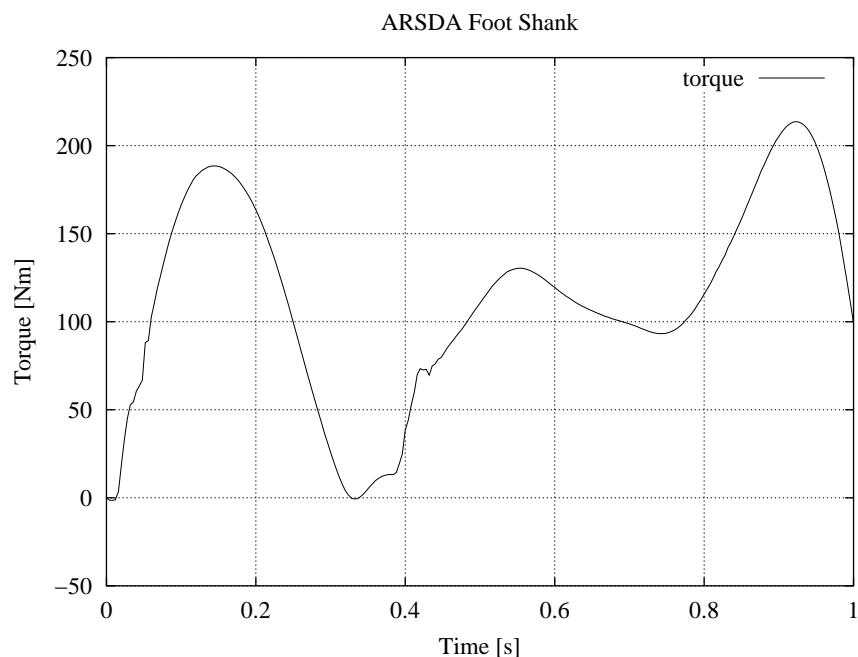


Abbildung 14: In dieser Abbildung ist der Momentverlauf im Sprunggelenk in der ersten Sekunde der Anfahrt dargestellt.

variiert wird, damit die Fluglage verändert werden kann.

Sobald sich der Fuß unter $1,5\text{ m}$ über der Schanze befindet, wird in die Landephase umgeschaltet.

Die Diagramme [15 - 19] zeigen die Momentverläufe einer Simulation. Ebenso sind die Luftreibungskräfte mit den Auftriebskräften addiert dargestellt [20 und 21].

4.3 Luftwiderstand und Auftrieb

Die Darstellung der aerodynamischen Kräfte, hier Luftwiderstand und Auftrieb, in Abbildung 22 verdeutlichen den Anteil des Luftwiderstandes (x -Anteil) und den Anteil an Auftrieb (z -Anteil). Die Koordinaten sind in Inertialdarstellung gewählt. Insgesamt sind die Kräfte entgegen der Bewegungsrichtung allerdings zu groß, vor allem im Bereich des Absprungs ab $t = 1,2\text{ s}$. Dies zeigt auch die Einzeldarstellung des Körpersegments THORAX in Abbildung 23. Es wurde versucht, diese Daten mit vorhandenen FEM-Rechnungen zu vergleichen. Dies scheiterte vorerst daran, dass die Daten eine Druckverteilung einer bestimmten Fluglage angeben und zusätzliche Parameter, wie Bewegungsgeschwindigkeit, Körpergröße und Gewicht fehlten. Die

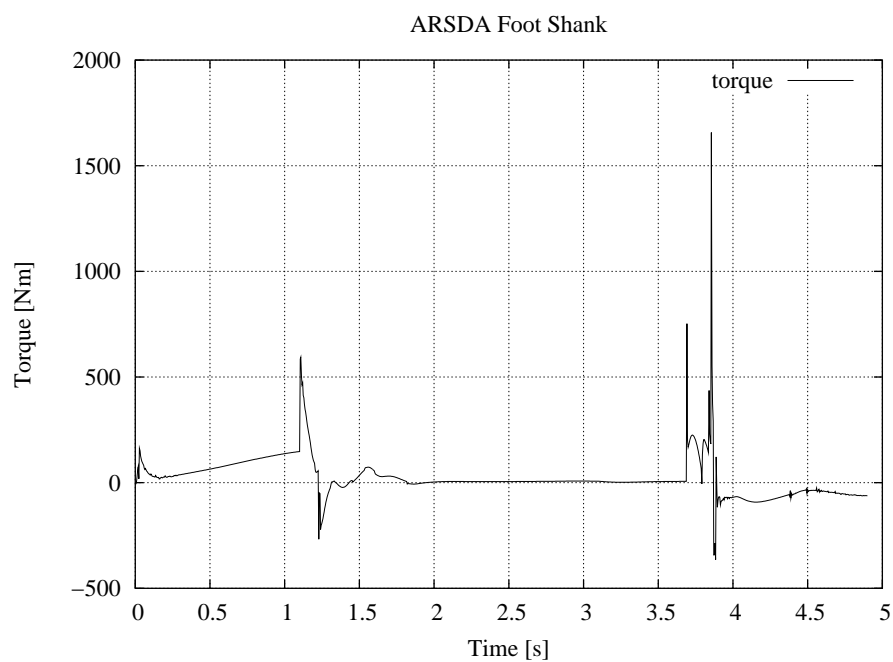


Abbildung 15: Aktive Drehfeder im Sprunggelenk. Bei $t_1 = 1,2 \text{ s}$ beginnt der Absprung, danach werden die Ski zur Fluglage angestellt.

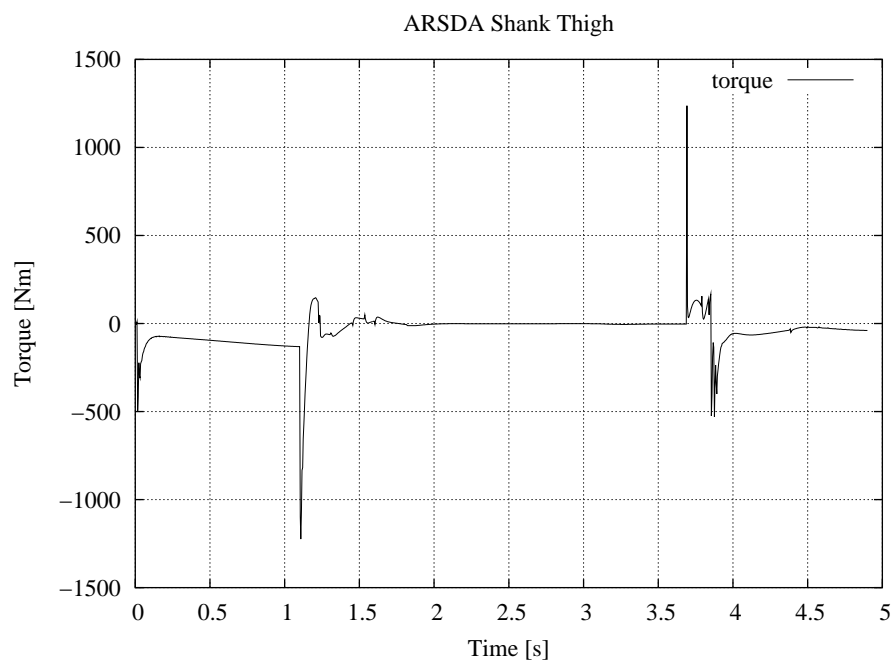


Abbildung 16: Aktive Drehfeder im Kniegelenk. Hier sieht man deutlich, dass das Moment zu plötzlich auftritt und die Spitzen einen physiologisch vertretbaren Wert übersteigen.

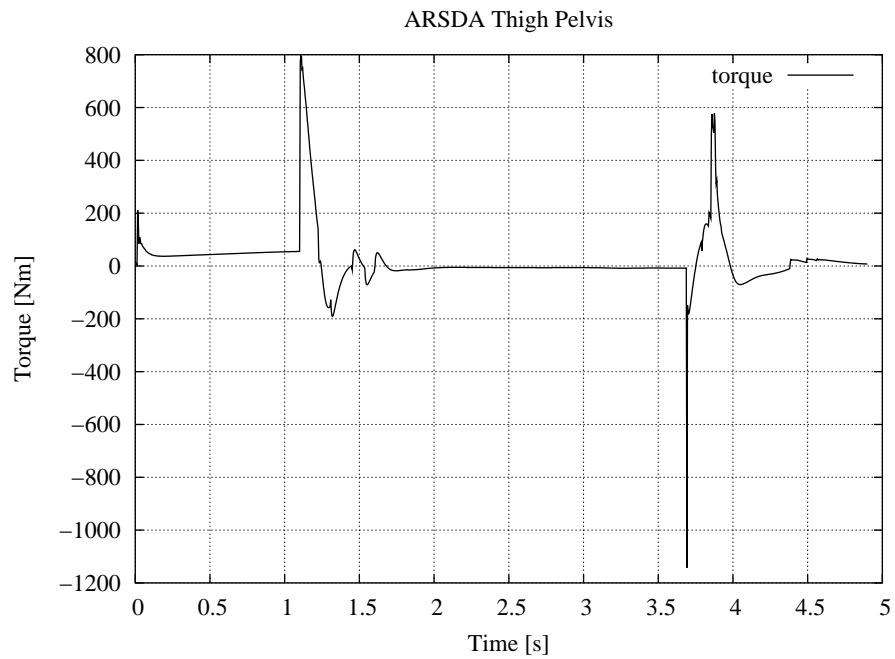


Abbildung 17: Aktive Drehfeder im Hüftgelenk. Nach dem Absprung überträgt sich das durch die Fluglageregelung der oberen Extremität produzierte Moment auf die Hüfte (um $t = 1,5$ s).

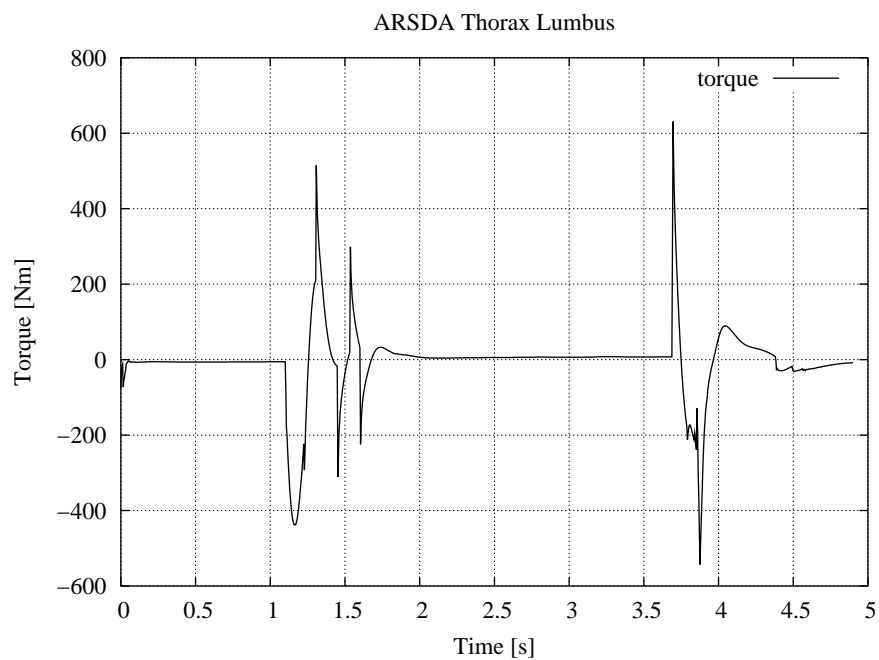


Abbildung 18: Der Oberkörper trägt entscheidend zur Fluglageregelung bei. Es ist deutlich zu erkennen, dass zu Beginn häufig zwischen *vor* und *zurück* umgeschaltet wird.

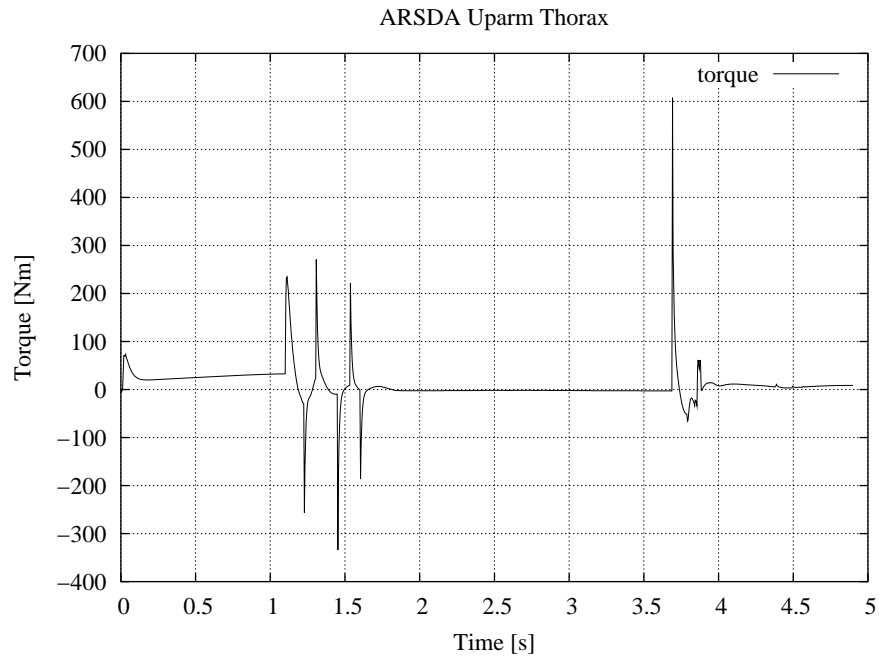


Abbildung 19: Aktive Drehfeder im Schulterbereich. Diese Darstellung korrespondiert sehr gut mit Abbildung 18, ebenso ist die Fluglageregelung deutlich zu erkennen.

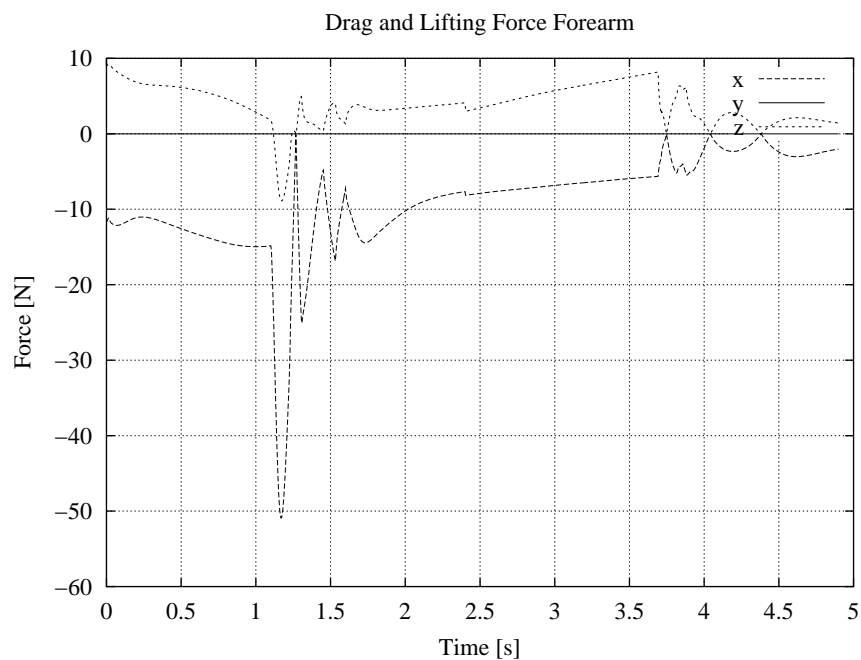


Abbildung 20: Die aerodynamischen Kräfte am Unterarm.

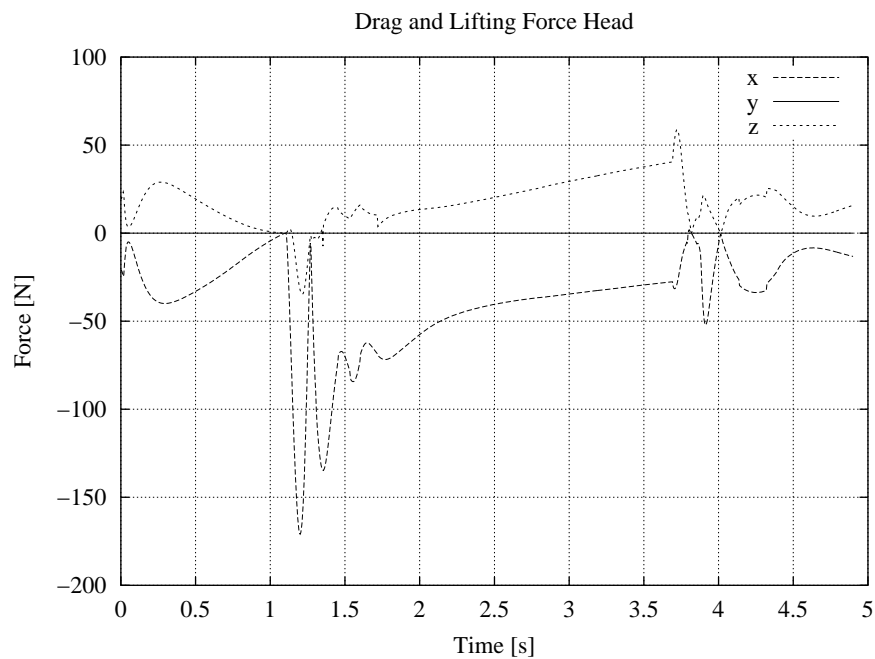


Abbildung 21: Wie bereits angesprochen, werden durch eine veränderliche Stellung des Kopfes zur Bewegungsrichtung Kräfte produziert, die ein Drehmoment auf das Skispringermodell erzeugen.

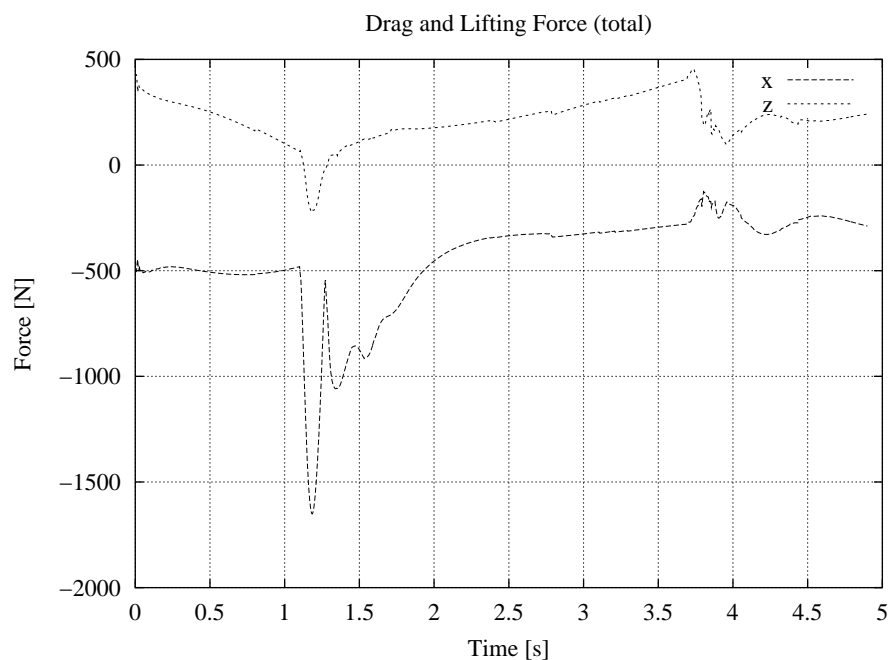


Abbildung 22: In dieser Abbildung sind die Kräfte, die durch Luftwiderstand und Auftrieb der einzelnen Teilsegmente entstehen, in x - und z -Richtung addiert dargestellt.

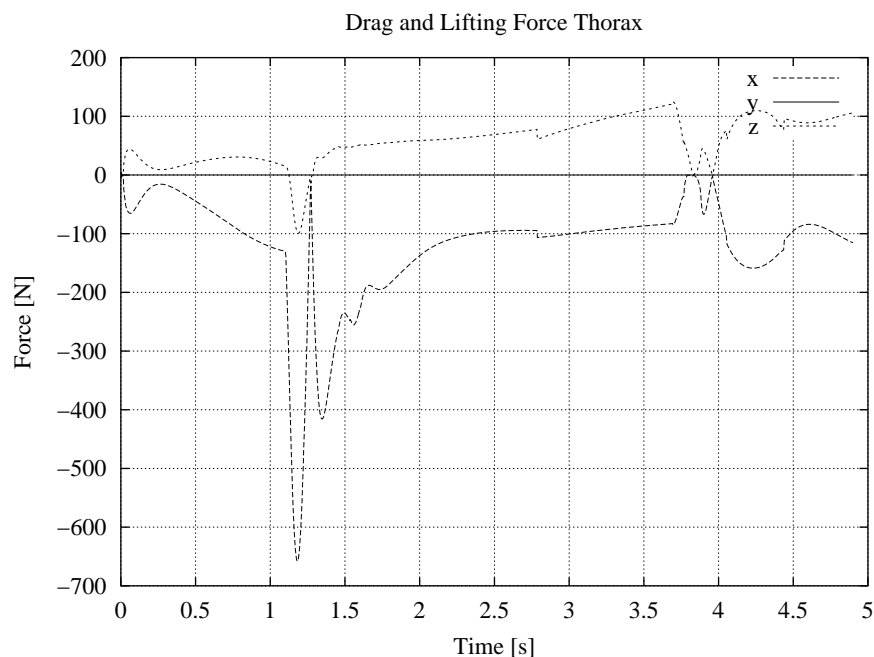


Abbildung 23: Den größten Beitrag zur gesamten Luftkraft leistet sicherlich der Oberkörper. In dieser Abbildung ist die hohe Kraftspitze kurz nach dem Absprung zu erkennen.

in den Ergebnissen sichtbar große Kraft entgegen der Bewegungsrichtung beim Absprung ist sehr wahrscheinlich der bedeutendste Grund, weshalb die Sprungweite nur zirka 45 m beträgt. Da, wie in Kapitel 3.1 erläutert, Luftwiderstand und Auftrieb voneinander unabhängig berechnet werden, ist hier die Möglichkeit gegeben, eine Optimierungsrechnung durchzuführen. Diese Rechnung sollte eine aus aerodynamischer Sicht optimale Sprungsequenz ermitteln.

4.4 Momente in den Gelenken

In Kapitel 3.1 wurde der Regelungsmechanismus erklärt, der auf verschiedenen Zielkonfigurationen basiert, zwischen welchen umgeschaltet wird. Dieses Umschalten findet instantan statt, d.h. es werden plötzlich große Momente erzeugt. In der Simulation zeigte sich, dass durch eine Anstiegsbegrenzung der Momente realistischere Momentenverläufe produziert werden können, allerdings sind die erzielten Momente immer noch sehr hoch. RUDER ET AL. [32] schlugen eine Berechnung dieser Begrenzung vor. Dazu wird für jedes Moment eine Differentialgleichung mitintegriert, in

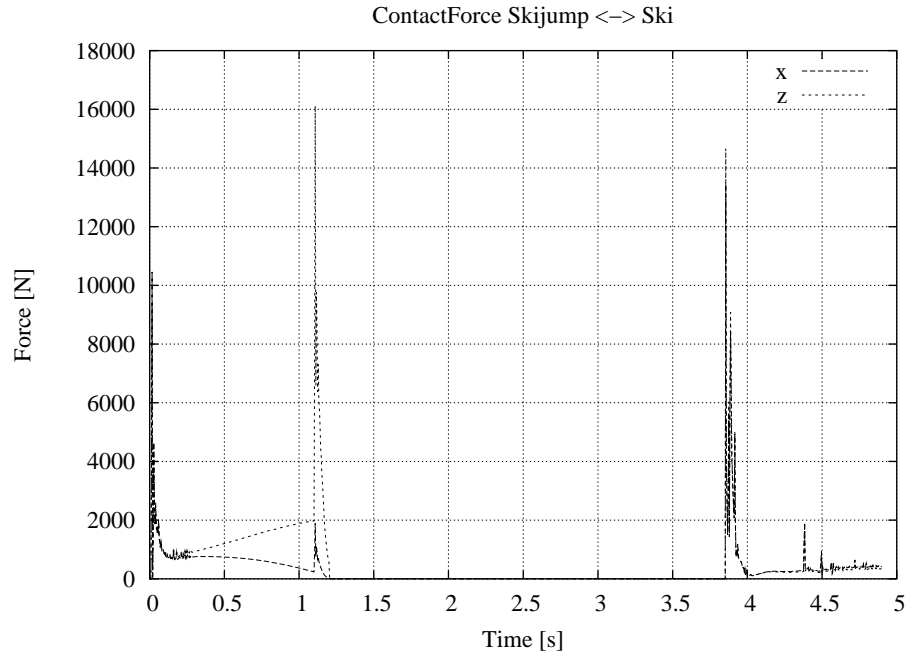


Abbildung 24: Die Bodenreaktionskraft zwischen Ski und Schanze über alle acht Kontaktpunkte summiert. Hiermit lässt sich der Beginn und das Ende der Flugphase eindeutig bestimmen.

der eine Anstiegsrate und ein Maximalmoment definiert sind.

$$M(t + dt) = M(t) + \int dt c \dot{\phi} \frac{M_{max} - M(t)}{M_{max}} \quad (13)$$

Diese Gleichung wird derzeit noch nicht mitintegriert, da dazu erhebliche Änderungen am Simulationsprogramm DySim vorgenommen werden müssen. Diese Änderungen werden durchgeführt, allerdings sind noch keine Ergebnisse einer Simulationsrechnung vorhanden. Es wird aber erwartet, dass mit dieser Differentialgleichung einerseits die Momentverläufe auf ein physiologisch vertretbares Maß reduziert werden können und andererseits das System auch mit geringeren Dämpfungswerten in den Gelenken mit der einfachen Regelungsstrategie funktioniert.

4.5 Bodenreaktionskräfte

Als Folge der in Kapitel 4.4 angesprochenen, schnell ansteigenden und sehr hohen Momente in den aktiven Gelenken wird zwischen Ski und Schanze eine hohe Kontaktkraft erzeugt, die in ihren Spitzen von realistischen Werten weit entfernt ist. Auch hier bringt die Implementierung der Momentverlaufsteuerung, wie in Kapitel

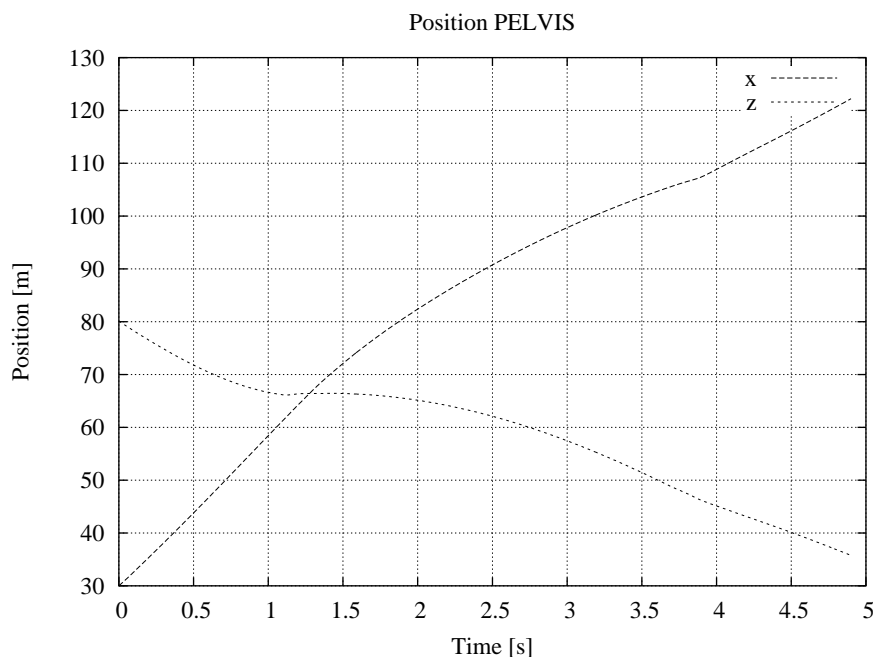


Abbildung 25: Der Positionsverlauf der Hüfte. Aus dieser Darstellung kann die Sprungweite zwischen $t_1 = 1,2 \text{ s}$ und $t_2 = 3,9 \text{ s}$ zu 45 m abgeschätzt werden.

4.4 angesprochen, eine erhebliche Verbesserung. Abbildung 24 zeigt die auftretenden Bodenreaktionskräfte, die summiert über alle acht Kontaktpunkte zwischen Ski und Schanze dargestellt sind. Über die Bodenreaktionskraft lässt sich der Beginn und das Ende der Flugphase eindeutig festlegen. In dieser Simulation beginnt die Flugphase bei $t_B = 1,2 \text{ s}$ und endet bei $t_E = 3,9 \text{ s}$.

4.6 Position und Geschwindigkeit

Als Ergebnisse aus einer Simulationsrechnung lassen sich die Positionen und die Geschwindigkeiten aller Teilstimente darstellen, ein erheblicher Vorteil gegenüber Messungen in der Realität. Hier wurden exemplarisch die Position (Abb. 25) und die Geschwindigkeit (Abb. 26) des Teilstiments PELVIS (Hüfte) ausgewählt. Aus den Positionsdaten lässt sich die Sprungweite näherungsweise berechnen, die in vorliegendem Fall bei $W = 45 \text{ m}$ liegt. Die Darstellung der Geschwindigkeit erlaubt eine Aussage darüber, wie gut die Simulationsbedingungen gewählt wurden und mit der Realität übereinstimmen. Im vorliegenden Fall trifft dies zumindest für die Anfangsbedingungen zu, danach nimmt die Geschwindigkeit als Folge des großen Luftwiderstandes zu stark ab.

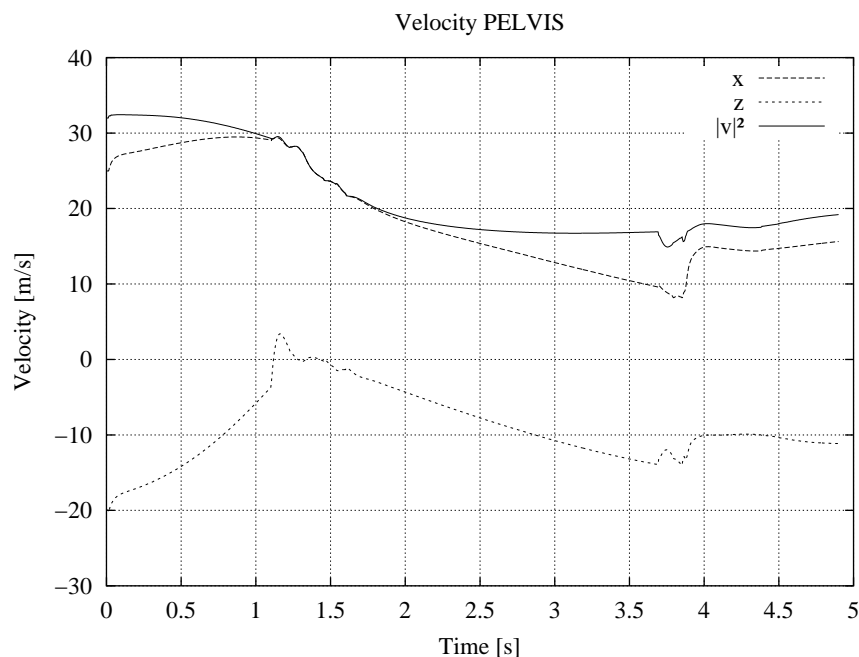


Abbildung 26: Der Geschwindigkeitsverlauf der Hüfte. Es zeigt sich, dass die Anfangsbedingungen gut gewählt wurden und mit realistischen Werten übereinstimmen.

4.7 Die Landung

Diese Arbeit hat nicht das Ziel, die Landung in der Simulation durchführen zu können, weshalb es umso überraschender und erfreulicher ist, dass dies gelungen ist. Es wurden hierbei keine feindifferenzierten Regelungsstrategien verwendet, sondern lediglich bei der Wahl der Winkelkonfiguration beachtet, dass der Schwerpunkt des Modells möglichst tief und stabil über der Unterstützungsfläche liegt. In [Abbildung 27](#) und [28](#) ist die Landung als Bildfolge dargestellt.

4.8 Dämpfungswerte

Es wurde bereits mehrfach im Text erwähnt, dass die Parameter für Steifigkeit und Dämpfung einen entscheidenden Einfluss auf die Simulation haben. Außerdem muss die Modellierung einer Regelung auf diese Werte abgestimmt sein. Große Steifigkeiten führen zu großen Kräften, die bei aktivem Gelenkeinsatz produziert werden können oder müssen, um das Gelenk aus seiner Ruhelage zu bewegen. Die Dämpfung nimmt ebenfalls Einfluss auf die Gelenkeigenschaften, wobei eine hohe Dämpfung ein System gegen Störungen stabil macht, während bei geringer Dämpfung

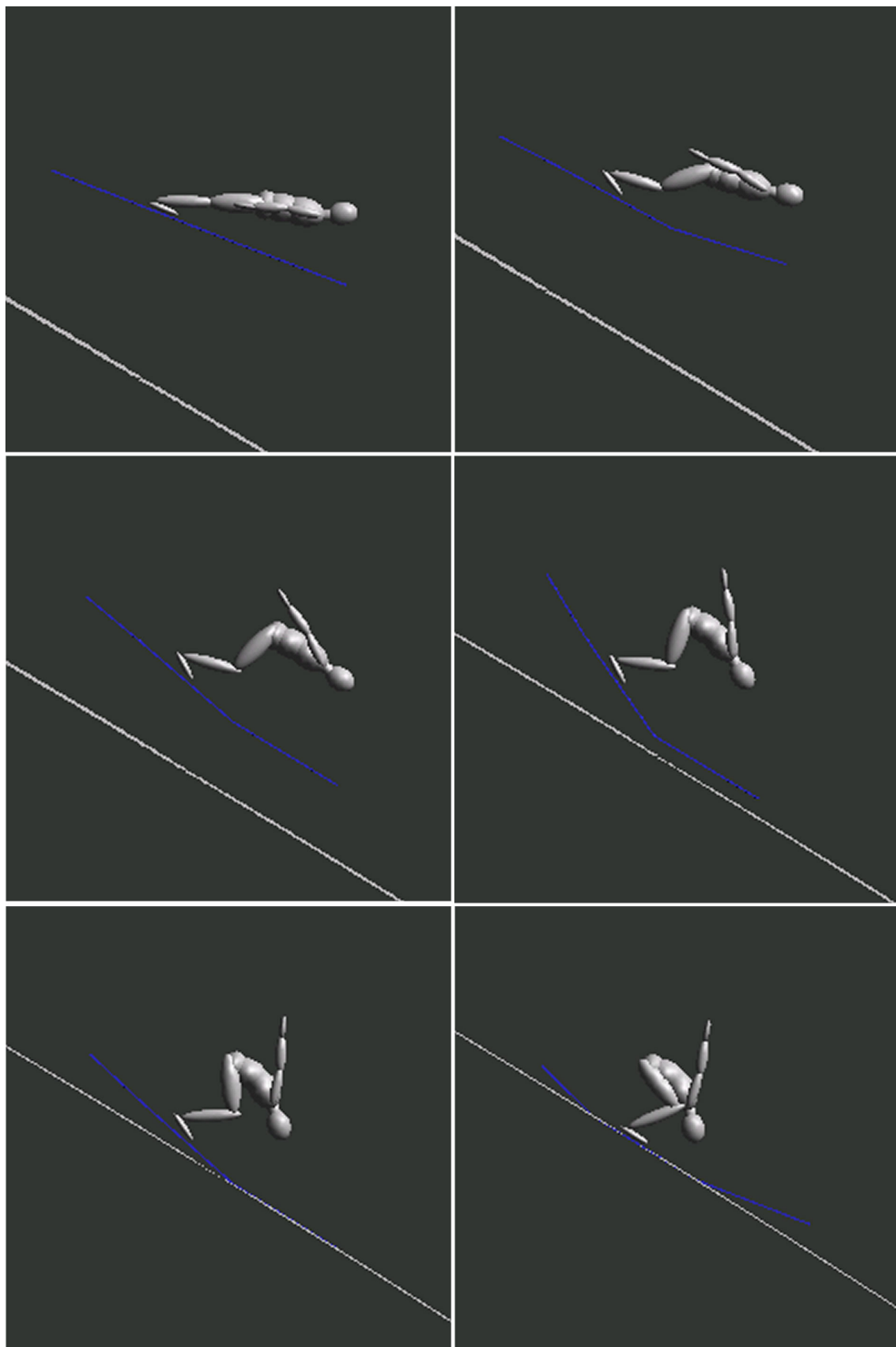


Abbildung 27: Darstellung der Landesequenz (Erster Teil).

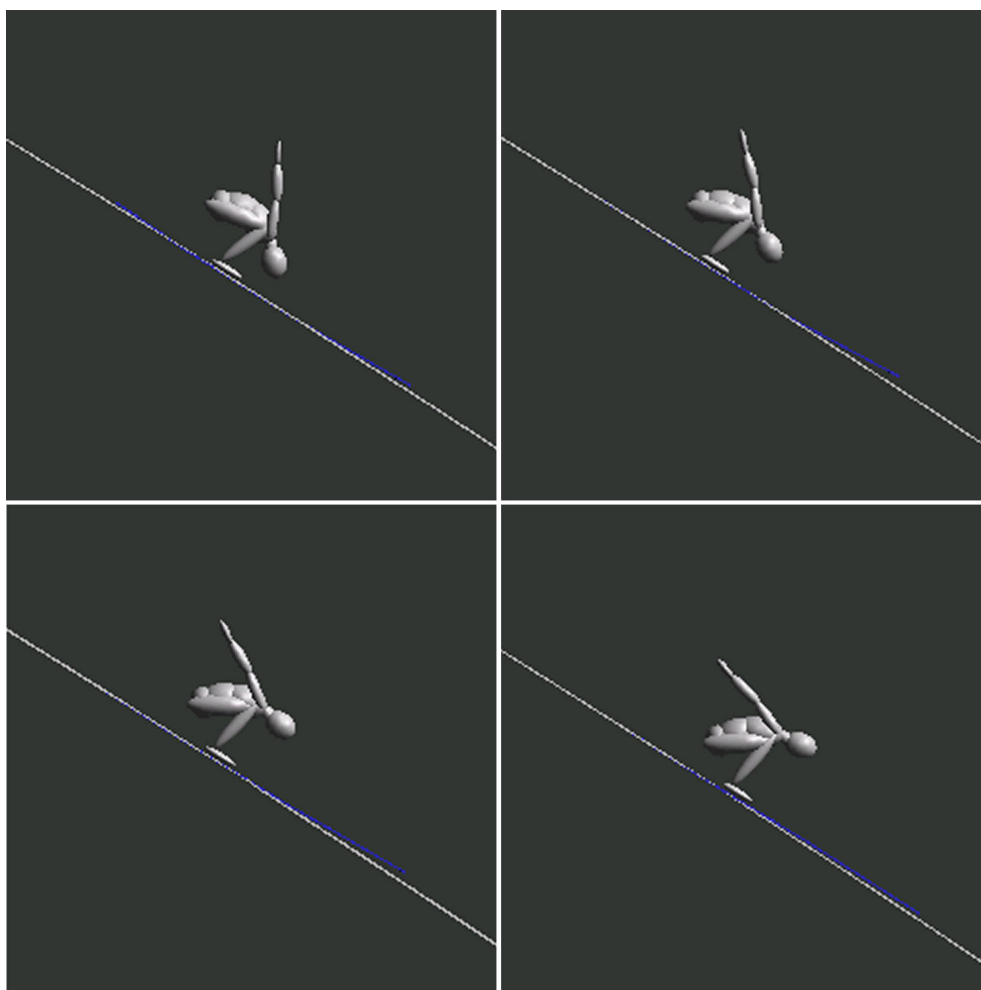


Abbildung 28: Darstellung der Landesequenz (Zweiter Teil). Ausgehend von der Fluglage ist es in diesem Fall möglich, nur mit einer neuen Winkelkonfiguration für alle aktiven Gelenke eine Landung zu simulieren.

sehr schnell Schwingungen auftreten können. Im Rahmen dieser Arbeit wurde zunächst mit eigenen Werten für die Dämpfung² gearbeitet, die sich, wie später der Vergleich mit Werten aus der Literatur [10] ergab, als sehr hoch herausstellten. Der in dieser Arbeit verwendete Regelungsmechanismus ist auf die höheren Werte abgestimmt und bedarf einiger Änderungen, die bereits angesprochen wurden oder noch angesprochen werden, damit auch mit niedrigeren Werten für die Gelenkdämpfung die Bewegungsregelung in der vorgestellten Weise durchgeführt werden kann. Der Unterschied zwischen den Parametern für die Steifigkeit und die Dämpfung besteht

²Hierzu wurde eigens ein Testmodell programmiert, das nur aus einem Bein bestand. Damit konnte die aktive Regelung sehr gut getestet werden und anhand dieses Testmodells konnte das Regelverhalten studiert werden.

darin, dass die Werte für die Steifigkeit in biomechanischen Messungen mit der Genauigkeit innerhalb einer Größenordnung erhoben werden können. Messungen, die die Dämpfung eines menschlichen Gelenks bestimmen, sind dagegen sehr ungenau, weshalb auch hier noch Handlungsbedarf besteht, um das vorliegende Modell zu validieren.

Zurück zur vorliegenden Arbeit. Hierbei wurden alle Ergebnisse mit hohen Werten für die Dämpfung ($d = 1 \text{ Nms/grad}$) erzielt, einzige Ausnahme, ebenfalls bereits erwähnt, ist die Darstellung der Anfahrtsregelung. Dort wurden die Ergebnisse mit sehr niedrigen Dämpfungswerten ($d = 0,01 \text{ Nms/grad}$) berechnet und dennoch der angesprochene Regelungsmechanismus verwendet. Allerdings kollabiert das Modell beim Absprung, da dort ein zu großer Sprung in den Nulllagen durchgeführt werden muss. Dieses Problem ist nur dadurch zu lösen, in dem Differentialgleichungen, die den Anstieg des Moments begrenzen, mitintegriert werden. Erste Versuche wurden in diese Richtung unternommen, leider sind dabei noch keine brauchbaren Ergebnisse entstanden. Dies bleibt als Aufgabe bestehen.

5 Zusammenfassung und Ausblick

Die vorigen Kapitel haben gezeigt, dass es möglich ist durch einfache Strategien und Modelllösungen eine komplexe sportliche Bewegung, wie es der Skisprung darstellt, vorwärtsdynamisch zu simulieren. An einigen Stellen ergaben sich Schwierigkeiten, die eine komplexere Herangehensweise erfordern, jedoch an anderen Stellen konnten bereits gute Ergebnisse erzielt werden.

Was wurde erreicht?

Es wurde im Text mehrfach das MKS-Paket `DySim` erwähnt. Dieses von HENZE [18] geschriebene Programm zeigte große Stärken: Die implementierte Bibliothek `Spacelib` von LEGNANI [24, 25] erleichterte den Formalismus zum Aufstellen der Bewegungsgleichungen. Alle Angaben, wie z.B. Positionen und Orientierungen, werden in körperfesten Koordinaten gemacht, wodurch intuitiv die Matrizen benutzt werden konnten, z.B. um neue Modellierungselemente zu entwerfen. Außerdem wäre es ohne das Vorhandensein des Quelltextes des Programms nicht möglich gewesen, neue Krafterelemente, wie den Schanzenkontakt, die Aerodynamik oder die aktive Modellregelung hinzuzufügen. Dieses Programmpaket wurde im Rahmen dieser Arbeit erweitert und steht somit dem Institut für weitere Simulationen zur Verfügung.

Die programmtechnische Implementierung von aktiven Modellelementen, die ARS-DA als Momentengeneratoren allein über eine veränderbare Federnulllage, konnte physikalisch korrekt durchgeführt werden. So können im Modell innere Momente aufgebracht werden, um beispielsweise Positionsregelungen vorzunehmen. Ebenso zeigen die Ergebnisse der Regelungsstrategien, dass es möglich ist, mit wenigen Parametern, wie z.B. der Kontaktkraft der Ferse, eine Positionsregelung durchzuführen. Schlussendlich war der Denkansatz, eine Bewegungslösung nur über Zielkonfigurationen zu erzielen, erfolgreich. Der Skisprung konnte von der Anfahrt über den Absprung und der Flugphase bis hin zur Landung simuliert werden. In diesem Sinne sollte es möglich sein, durch eine einfache Analyse einer sportlichen Bewegung, deren Zwischenpositionen zu ermitteln, mit denen dann eine vorwärtsdynamische Modellierung und Simulation durchgeführt werden kann. Diese Aussage muss durch andere Simulationen komplexer sportlicher Bewegungen überprüft werden.

Was wurde (noch) nicht erreicht?

Die Frage der Momentengenerierung durch Umschalten der Nulllagen der Drehfedern in den Gelenken konnte mit den benutzten Strategien nicht realistisch und anatomisch/physiologisch richtig geklärt werden. Wie im Kapitel 4.4 angesprochen muss dazu eine Differentialgleichung mitberechnet werden, die Auf- und Abbau der Gelenkmomente steuert. Das kann einerseits bedeuten, dass die durch das plötzliche

Umschalten auftretenden un stetigen Momente verändert werden oder andererseits das Umschalten an sich über eine “Rampe” geschieht, bei der die Nulllagensprünge der Drehfedern begrenzt werden. Durch eine Verbesserung der Momentgenerierung wird erwartet, dass dadurch die Momente realistisch werden und als Folge daraus die Gelenkdämpfung reduziert werden kann, ohne dass das Modell beim Absprung kollabiert. Als weitere Folge aus der Verbesserung der Momentgeneration ist zu erwarten, dass die Bodenreaktionskräfte in ihrer Höhe reduziert werden und auch hier ein realistischeres Verhalten gezeigt werden kann.

Die Modellierung der Landung gehörte nicht zum planmäßigen Umfang dieser Arbeit. Die eingesetzte Näherungslösung enthält noch zu hohe Gelenkmomente und riesige Spitzen in den Bodenreaktionskräften der Kontaktpunkte. Die Landung ist ein hochdynamischer Vorgang, weshalb die Modellierung der Körpersegmente ohne Schwabbelmassen systematisch nicht richtig ist. Die Effekte, die sich in den Momentendiagrammen (15-17) zum Zeitpunkt der Landung zeigen, sind denen, die durch GRUBER [11] beobachtet wurden ähnlich. Daraus lässt sich schließen, dass eine Implementierung von Schwabbelmassen in das Skispringermodell für eine realistische Simulation der Landung unumgänglich ist.

Wie geht es weiter?

Im vorhergehenden Abschnitt wurden bereits Verbesserungen angesprochen, die am Programm DySim und am Skispringermodell eingearbeitet werden müssen um eine sehr realistische Skisprungsimulation zu erreichen. Diese Verbesserungen müssen einhergehen mit weiteren Überlegungen zum Modell und den verwendeten Näherungen um eine höhere Aussagekraft des Modells zu erzielen.

Die im Modell verwendeten Aerodynamikkräfte Luftwiderstand und Auftrieb müssen durch Windkanalexperimente validiert werden. Einerseits können vorhandene Messungen des Skiverbandes berücksichtigt werden, andererseits müssen spezielle Messungen im Windkanal in Kooperation mit entweder dem Skiverband oder Leistungsdiagnostikern der Sportwissenschaft oder den Windkanalbetreibern der Universität Stuttgart durchgeführt werden. In diesen Messungen müssen die Parameter c_W und c_A genauer bestimmt und die Größe der Aerodynamikkraft validiert werden.

Im Rahmen der vorliegenden Menschmodellierung wurde auf statistische Daten der NASA zurückgegriffen (siehe Kap. 3.1). Dieses erstellte Skispringermodell könnte durch eine Individualisierung der Segmenteigenschaften, wie z.B. Länge, Masse, besser an einzelne Skispringer angepasst werden. Zudem ist eine Weiterentwicklung des Modells in den Gelenkstrukturen hin zu komplexen Gelenkstrukturen wünschenswert, damit z.B. die Gelenkstrukturbelastung bei einer Landung realistischer abgeschätzt werden könnte. Außerdem könnte durch komplexere Gelenkstrukturen, z.B.

Kugelgelenke in den Hüften, die Flugkonfiguration realitätsnaher angepasst werden (typischer V-Stil).

Entscheidende Verbesserungen der Aussagekraft des Modells im Bereich der Bewegungssteuerung könnte durch die Modellierung von Muskulatur erreicht werden. Dieser Schritt ist allerdings sowohl in der Programmierung als auch in der Validierung immens aufwändig. In einem Schritt zuvor, durch die Verwendung eines komplexeren Regelungsalgorithmus, liegen aber bereits gute Modellverbesserungen. Denn im existierenden Modell produzieren P-D-Regler anhand von Längen und Längenänderungen der Drehfedern in den Gelenken das Gelenkmoment, welches direkt in der Mechanik des Modells, also am Gelenk aufgebracht wird. In der Realität erzeugen aber die Muskeln Kräfte, die über Sehnen an die Knochen weitergeleitet werden. Das resultierende Drehmoment ist abhängig vom Sehnenansatz (Hebelarm) und einigen weiteren Sehnen- und Muskeleigenschaften, wie z.B. einer Zeitdilatation der Kraftentfaltung, möglicherweise einer Frequenzfilterung und auch einer gewissen Muskelredundanz. Diese Besonderheit der Drehmomentgeneration muss durch den Regelungsmechanismus berücksichtigt werden, z.B. in Form von Momentanstiegsraten oder ähnlichem. Dann könnten auch ohne den Einsatz von Muskeln im Modell gute Aussagen zur Bewegungsregelung getroffen werden.

Zusätzlich könnten Optimierungen des Programmquelltextes die Schnelligkeit der numerischen Integration erhöhen. Vor allem bessere Kontaktdetektionsalgorithmen könnten eine entscheidende Verbesserung erzielen. Denn Simulationen die nahe an der Echtzeit durchgeführt werden könnten, würden weitere Anwendungsfelder erschließen, beispielsweise als Trainingsanalyse direkt nach einem absolvierten realen Sprung oder um Sprungstrategien direkt an Umgebungseinflüsse, wie z.B. Seitenwind, vor einem Sprung anzupassen.

Mit diesen Verbesserungen ist es denkbar, in mehrere Richtungen den Skisprung zu studieren. Das könnten Untersuchungen sein, die verschiedene Schanzenprofile bei identischen Umgebungsbedingung vergleichen oder Fragestellungen, die sich eher auf eine Belastungsanalyse der Springer bei Landungen jenseits der Schanzenrekorde beziehen. Es wären Optimierungrechnungen denkbar, die eine optimale Absprungsstrategie ergeben könnten mit dem Ziel einer maximalen Sprungweite angepasst an bestimmte Umgebungsbedingungen. Ein entscheidender Vorteil könnte auch sein, Schanzenprofile im Hinblick auf die Sicherheit und Gesundheit der Springer vor der Umsetzung in der Realität im Modell am Rechner zu entwickeln und zu studieren.

Literatur

- [1] BÖHM, H.: *Direkte Dynamik einer Riesenfelge im Rahmen von Mehrkörpersystemen*. Diplomarbeit, Universität Tübingen, 1997.
- [2] BÖHM, H.: *Computer Simulation of Muscle Series Elastic Element Function in Drop Jumping*. Doktorarbeit, 2001. [1](#)
- [3] DENAVIT, R.S. und HARTENBERG ET AL.: *Velocity, Acceleration and Static-Forces Analyses of Spatial Linkages*. Journal of Applied Mechanics, Seite 903, 12 1965. [2.1](#)
- [4] DENAVIT, R.S. und R. HARTENBERG: *A Kinematic Notation for Lower Pairs Mechanisms Based on Matrices*. Journal of Applied Mechanics, 22:215–221, 06 1955. [2.1](#)
- [5] DIN: *DIN 33402-1, Körpermasse des Menschen*. Technischer Bericht, DIN, 1986. [3.1](#)
- [6] FELDMAN, A.G.: *Control of the length of the muscle*. Biophysics, 19:766–771, 1974.
- [7] FELDMAN, A.G.: *Once more on the equilibrium-point hypothesis (λ model) for motor control*. Journal of Motor Behavior, 18:17–54, 1986. [3.1](#)
- [8] FELDMAN, A.G., D.J. OSTRY, M.F. LEVIN, P.L. GRIBBLE und A.B. MITNITSKI: *Recent tests of the equilibrium-point hypothesis (λ model)*. Motor Control, 2(3):189–205, 1998.
- [9] FIS: *Spezifikationen der Wettkampfausrüstung 2002/2003*. <http://www.fis-ski.com/>. [3.1](#)
- [10] GÜNTHER, M.: *Computersimulationen zur Synthetisierung des muskulär gesteuerten menschlichen Gehens unter Verwendung eines biomechanischen Mehrkörpermodells*. Doktorarbeit, Universität Tübingen, 1997. [1](#), [2.3](#), [3.1](#), [3.1](#), [4.2](#), [2](#)
- [11] GRUBER, K.: *Entwicklung eines Modells zur Berechnung der Kräfte im Knie- und Hüftgelenk bei sportlichen Bewegungsabläufen mit hohen Beschleunigungen*. Doktorarbeit, Universität Tübingen and ETH Zürich, 1987. [5](#)
- [12] GRUBER, K., H. RUDER, J. DENOTH und K. SCHNEIDER: *A comparative study of impact dynamics: wobbling mass modell versus rigid body models*. Journal of Biomechanics, 1998.

- [13] GÖTZ, CH.: *Entwicklung eines Finite-Elemente-Modells des menschlichen Schädels zur Simulation von Stößen*. Doktorarbeit, Universität Tübingen, 1998. [1](#), [2.3](#)
- [14] HAHN, U.: *Entwicklung mehrgliedriger Modelle zur realistischen Simulation dynamischer Prozesse in biologischen Systemen*. Diplomarbeit, Universität Tübingen, 1993. [3.1](#)
- [15] HANAVAN, E. P.: *A mathematical model of the human body*. Technischer Bericht, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, OH, 1964. [3.1](#)
- [16] HATZE, H.: *A complete set of control equations for the human musculo-skeletal system*. Journal of Biomechanics, 10:799–805, 1977. [2.3](#)
- [17] HATZE, H.: *A mathematical model for the computational determination of parameter values of anthropomorphic segments*. Journal of Biomechanics, 13:833–844, 1980. [3.1](#)
- [18] HENZE, A.: *DySim - Ein Werkzeug zur Simulation mechanischer Starrkörpersysteme*. Institut für Astronomie und Astrophysik Tübingen, 2001. [2.1](#), [3.1](#), [1](#), [5](#), [A](#)
- [19] HENZE, A.: *Dreidimensionale biomechanische Modellierung und die Entwicklung eines Reglers zur Simulation zweibeinigen Gehens*. Doktorarbeit, Universität Tübingen, 2002. [1](#), [3.1](#)
- [20] HOSPACH, F.: *Computersimulationen von Abstürzen mittels biomechanischer Ganzkörpermodelle*. Doktorarbeit, Universität Tübingen, 1993.
- [21] KEPPLER, V.: *Neuronale Oszillatoren als Steuerungskonzept in der Biomechanik*. Diplomarbeit, Universität Tübingen, 1997.
- [22] KEPPLER, V.: *Modellbildung und Simulation biomechanischer Menschmodelle in Interaktion mit einer virtuellen Umgebung*. Doktorarbeit, Universität Tübingen, 2003. [1](#)
- [23] KREBS, T.: *Biomechanische Simulationen an mehrgliedrigen Modellen zur Untersuchung von Bewegungsabläufen der Maus*. Diplomarbeit, Universität Tübingen, 1992.
- [24] LEGNANI, G., P. RIGHETTINI, B. ZAPPA und F. CASOLO: *A homogenous matrix approach to 3D kinematics and dynamics. Part I: Theory*. Mechanisms and Machine Theory (the scientific journal of IFToMM), 31:573–587, 1996. [2.1](#), [5](#)

- [25] LEGNANI, G., P. RIGHETTINI, B. ZAPPA und F. CASOLO: *A homogenous matrix approach to 3D kinematics and dynamics. Part II: Applications*. Mechanisms and Machine Theory (the scientific journal of IFToMM), 31:589–605, 1996. [2.1](#), [5](#)
- [26] MUTSCHLER, H.: *Biomechanische Simulation zielgerichteter Bewegungen mit neuronalen Netzen*. Diplomarbeit, Universität Tübingen, 1997.
- [27] NASA: *The internal properties of the body and it's segments*. Technischer Bericht, NASA Scientific and Technical Information Office, Johnson Space Center, Houston, TX, 1978. Anthropometric Source Book. [2.3](#), [3.1](#), [5](#)
- [28] PATLA, A.E.: *The lambda-Model — can it walk?* Behavioral and Brain Sciences, 18(4):775–776, 1995.
- [29] PLATZER, W.: *dtv-Atlas der Anatomie*, Band 5. Georg Thieme Taschenbuchverlag, 1986. ISBN-3134920050. [2](#), [3](#)
- [30] PRESS, W.H., S.A. TEUKOLSKY, V.T. VETTERLING und FLANNERY B.P.: *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge, 1994.
- [31] RAUBER, A. und KOPSCH: *Anatomie des Menschen, Lehrbuch und Atlas*. Georg Thieme Verlag Stuttgart und New York, 1987. [1](#)
- [32] RUDER, H., K. GRUBER, M. GÜNTHER, M. RUDER, J. SUBKE und K. WIDMAYER: *Kinematic and dynamics for computer animation*. In: COQUILLART, W., W. STRASSER und P. STUCKI (Herausgeber): *From Object Modeling to Advanced Visual Communication*, Seiten 76–117. Springer, Berlin, 1994. [4.4](#)
- [33] SCHLICHTLING, H.J. und B. RODEWALD: *Ikarus' Traum und die aerodynamische Wirklichkeit*. Praxis der Naturwissenschaften - Physik, (35):5–7, 1986. [3.1](#)
- [34] SCHÜSZLER, O.: *Computersimulationen von realen Kraftfahrzeug-Fußgänger-Unfällen*. Doktorarbeit, Universität Tübingen, 1998.
- [35] SHAMPINE, L.F. and M.K. GORDON: *Computer-lösung gewöhnlicher differentialgleichungen. das anfangswertproblem*, 1984. [3.2](#)
- [36] WELTNER, K. and M. INGELMAN-SUNDBERG: *Misinterpretations of bernoulli's law*. Internet, 06 2003. submitted to European Journal of Physics.
- [37] WELTNER, K. and M. INGELMAN-SUNDBERG: *Physics of flight - reviewed*. Internet, 06 2003. submitted to European Journal of Physics.

- [38] WIDMAYER, K.: *Simulation von bewegungsabläufen beim menschen mit hilfe von mehrkörpersystemen*. Master's thesis, Universität Tübingen, 1990.
- [39] YAMAGUCHI, G.T.: *Performing whole-body simulations of gait with 3-d, dynamic musculoskeletal models*. In WINTERS, J.M. and S.L.Y. WOO (editors): *Multiple Muscle Systems*, chapter 43, pages 663–679. Springer, New York, 1990.
[2.3](#)
- [40] YAMAGUCHI, G.T., D.W. MORAN, and J. SI: *A computationally efficient method for solving the redundant problem in biomechanics*. *Mathematical Biosciences*, 28(8):999–1005, 1995.

A Die Rotationsfreiheitsgrade

Eine Zwischenbemerkung zu den Rotationsfreiheitsgraden, aus DySim-Theorie [18] :

Bei der Verwendung von generalisierten Koordinaten ist man zur Beschreibung der Körperorientierung – beispielsweise in Gelenken mit mehreren Rotationsfreiheitsgraden – prinzipiell auf Winkelkoordinaten angewiesen, d.h. auf die Drehwinkel aufeinander folgender Rotationen um unterschiedliche Achsen. Da diese Achsen im Laufe der Bewegung linear abhängig werden können, führt dies zu Koordinatensingularitäten, wenn nicht zusätzlich entsprechende Vorkehrungen getroffen werden, um in jedem Moment eine eindeutige Systembeschreibung zu gewährleisten. In DySim wird dieses Problem dadurch gelöst, dass vor dem Erreichen einer Singularität, deren Lage für jeden Satz Winkelkoordinaten a priori bekannt ist, frühzeitig eine andere Rotationsreihenfolge umgeschaltet wird. Die generalisierten Koordinaten q_i erhalten durch diese Vorgehensweise zwar Unstetigkeiten – die entscheidenden kinematischen und dynamischen Matrizen, vor allem die Lage-, die Geschwindigkeits- und die Beschleunigungsmatrizen, bleiben davon jedoch unbeeinflusst. Dieses Verfahren hat sich bei sämtlichen mit DySim durchgeführten Simulationen als sehr zuverlässig erwiesen, wobei es ausreichend ist, lediglich zwischen EULER- und KARDAN-Winkeln hin- und herzuschalten, um die Singularitäten wirkungsvoll zu umgehen.

B Die Modellkonfigurationsdatei <skijump.dys>

```
Header "header.data" {
  ModelName = skijump;
  OutputDataAnimaton = true;
}
Solver "solver.data" {
  Type = dynamic;
  Solver = derf;
  AbsErr = 1.e-5;
  RelErr = 1.e-5;
  Tstart = 0.0;
  Tend = 7.0;
  Tstep = 0.004;
}
System "system.data" {
  GravAccel = 0.0, 0.0, -9.8065;
  ScaleGrav = 1.0;
}
Triad "Skijump" {
  Parent = world;
  Position = 0.0, 0.0, 0.0;
  Orientation = 0.0, 0.0, 0.0;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Skijump_Joint_Pelvis" {
  Parent = world;
  Position = 0.0, 0.0, 0.0;
  Orientation = 0.0, 0.0, 0.0;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Skijump_Joint_Snow" {
  Parent = world;
  Position = 0.0, 0.0, 0.0;
  Orientation = 0.0, 0.0, 0.0;
}
```

```

AngleType = cardan;
Degrees = deg;
}
Triad "Skijump_ContactSpline_01" {
Parent = world;
Position = 0.0, -1.0, 0.0;
Orientation = 0.0, 0.0, 0.0;
AngleType = cardan;
Degrees = deg;
}
Triad "Skijump_ContactSpline_02" {
Parent = world;
Position = 0.0, -1.0, 0.0;
Orientation = 0.0, 0.0, 0.0;
AngleType = cardan;
Degrees = deg;
}
Triad "world_aerodynamicsimple" {
Parent = world;
Position = 0.0, 0.0, 0.0;
Orientation = 0.0, 0.0, 0.0;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Skijump_01" {
Parent = world;
Position = 0.0, 0.0, 0.0;
Cardan = 0.0, 0.0, 0.0;
Degrees = deg;
Color = 0.7, 0.7, 0.7, 0.7;
Type = Spline2D;
Range = 0, 65;
Width = 2.0;
Height = 0.2;
NumberSplineVertexes = 10000;
SplineName = Skijump_ContactSpline_01;
}
Primitive "Primitive_Skijump_02" {
Parent = world;
Position = 0.0, 0.0, 0.0;
Cardan = 0.0, 0.0, 0.0;
Degrees = deg;
Color = 0.7, 0.7, 0.7, 0.7;
Type = Spline2D;
Range = 65, 200;
Width = 2.0;
Height = 0.2;
NumberSplineVertexes = 10000;
SplineName = Skijump_ContactSpline_02;
}
Body "Pelvis" {
Parent = world;
Joint = Joint_Skijump->Pelvis;
Mass = 7.662959;
Inertia = 0.063881, 0.000000, 0.000000,
0.028735, 0.000000,
0.079456;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Pelvis_Joint_Skijump" {
Parent = Pelvis;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_Joint_Lumbus" {
Parent = Pelvis;
Position = 0.0000, 0.0000, 0.0508;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_Drehfeder3D_Lumbus" {
Parent = Pelvis;
Position = 0.0000, 0.0000, 0.0508;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_Joint_Thigh" {

```

```
Parent = Pelvis;
Position = 0.0000, 0.1171, -0.0508;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_ARSDA_Thigh" {
Parent = Pelvis;
Position = 0.0000, 0.1171, -0.0508;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_Joint_Thigh_re" {
Parent = Pelvis;
Position = 0.0000, -0.1171, -0.0508;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_ARSDA_Thigh_re" {
Parent = Pelvis;
Position = 0.0000, -0.1171, -0.0508;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Pelvis_AerodynSimple_Skijump_7" {
Parent = Pelvis;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Pelvis_01" {
Parent = Pelvis;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.0000;
Type = Sphere;
RadiusX = 0.1224;
RadiusY = 0.1729;
RadiusZ = 0.1015;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Lumbus" {
Parent = Pelvis;
Joint = Joint_Pelvis->Lumbus;
Mass = 8.406381;
Inertia = 0.083335, 0.000000, 0.000000,
0.051952, 0.000000,
0.094336;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Lumbus_Joint_Thorax" {
Parent = Lumbus;
Position = 0.0000, 0.0000, 0.0855;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Lumbus_ARSDA_Thorax" {
Parent = Lumbus;
Position = 0.0000, 0.0000, 0.0855;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Lumbus_Joint_Pelvis" {
Parent = Lumbus;
Position = 0.0000, 0.0000, -0.0855;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Lumbus_Drehfeder3D_Pelvis" {
Parent = Lumbus;
Position = 0.0000, 0.0000, -0.0855;
Orientation = 0.0000, 0.0000, 0.0000;
```

```

AngleType = cardan;
Degrees = deg;
}
Triad "Lumbus_AerodynSimple_Skijump_4" {
Parent = Lumbus;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Lumbus_01" {
Parent = Lumbus;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.0000;
Type = Sphere;
RadiusX = 0.1224;
RadiusY = 0.1729;
RadiusZ = 0.1710;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Thorax" {
Parent = Lumbus;
Joint = Joint_Lumbus->Thorax;
Mass = 12.523793;
Inertia = 0.165172, 0.000000, 0.000000,
0.118417, 0.000000,
0.140542;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Thorax_Joint_Neck" {
Parent = Thorax;
Position = 0.0000, 0.0000, 0.1122;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thorax_ARSDA_Neck" {
Parent = Thorax;
Position = 0.0000, 0.0000, 0.1122;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thorax_Joint_Lumbus" {
Parent = Thorax;
Position = 0.0000, 0.0000, -0.1309;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thorax_ARSDA_Lumbus" {
Parent = Thorax;
Position = 0.0000, 0.0000, -0.1309;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thorax_Joint_Uparm" {
Parent = Thorax;
Position = 0.0000, 0.1854, 0.1089;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thorax_ARSDA_Uparm" {
Parent = Thorax;
Position = 0.0000, 0.1854, 0.1089;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thorax_Joint_Uparm_re" {
Parent = Thorax;
Position = 0.0000, -0.1854, 0.1089;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
}

```

```

Triad "Thorax_ARSDA_Uparm_re" {
  Parent = Thorax;
  Position = 0.0000, -0.1854, 0.1089;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Thorax_Kontakt_Shank" {
  Parent = Thorax;
  Position = 0.0000, 0.0000, 0.1224;
  Orientation = 0.0000, 90.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Thorax_Kontakt_Shank_re" {
  Parent = Thorax;
  Position = 0.0000, 0.0000, 0.1224;
  Orientation = 0.0000, 90.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Thorax_AerodynSimple_Skijump_10" {
  Parent = Thorax;
  Position = 0.0000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Primitive "Primitive_Thorax_01" {
  Parent = Thorax;
  Position = 0.0000, 0.0000, -0.0393;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.7000, 0.7000, 0.7000, 0.0000;
  Type = Sphere;
  RadiusX = 0.1224;
  RadiusY = 0.1729;
  RadiusZ = 0.1833;
  GridLong = 12.0000;
  GridLati = 12.0000;
}
Body "Neck" {
  Parent = Thorax;
  Joint = Joint_Thorax->Neck;
  Mass = 1.498484;
  Inertia = 0.003168, 0.000000, 0.000000,
0.003168, 0.000000,
0.002795;
  CGFramePosition = 0.0, 0.0, 0.0;
  CGFrameOrientation = 0.0, 0.0, 0.0;
  AngleType = euler;
  Degrees = deg;
}
Triad "Neck_Joint_Head" {
  Parent = Neck;
  Position = 0.0000, 0.0000, 0.0595;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Neck_Drehfeder3D_Head" {
  Parent = Neck;
  Position = 0.0000, 0.0000, 0.0595;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Neck_Joint_Thorax" {
  Parent = Neck;
  Position = 0.0000, 0.0000, -0.0595;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Neck_ARSDA_Thorax" {
  Parent = Neck;
  Position = 0.0000, 0.0000, -0.0595;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Primitive "Primitive_Neck_01" {
  Parent = Neck;
  Position = 0.0000, 0.0000, 0.0000;
}

```

```

Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0489;
RadiusY = 0.0489;
RadiusZ = 0.1191;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Head" {
Parent = Neck;
Joint = Joint_Neck->Head;
Mass = 4.343123;
Inertia = 0.019255, 0.000000, 0.000000,
0.018161, 0.000000,
0.023518;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Head_Joint_Neck" {
Parent = Head;
Position = 0.0000, 0.0000, -0.0844;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Head_Drehfeder3D_Neck" {
Parent = Head;
Position = 0.0000, 0.0000, -0.0844;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Head_AerodynSimple_Skijump_2" {
Parent = Head;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Head_ContactSpline_Skijump_3" {
Parent = Head;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Head_ContactSpline_Ski_front_4" {
Parent = Head;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Head_01" {
Parent = Head;
Position = 0.0000, 0.0000, -0.0238;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0972;
RadiusY = 0.0925;
RadiusZ = 0.1259;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Uparm" {
Parent = Thorax;
Joint = Joint_Thorax->Uparm;
Mass = 1.676195;
Inertia = 0.012807, 0.000000, 0.000000,
0.012549, 0.000000,
0.001750;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Uparm_Joint_Thorax" {
Parent = Uparm;

```

```

Position = 0.0000, 0.0000, 0.1471;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_ARSDA_Thorax" {
Parent = Uparm;
Position = 0.0000, 0.0000, 0.1471;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_Joint_Forearm" {
Parent = Uparm;
Position = 0.0000, 0.0000, -0.1594;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_Drehfeder3D_Forearm" {
Parent = Uparm;
Position = 0.0000, 0.0000, -0.1594;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_AerodynSimple_Skijump_4" {
Parent = Uparm;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Uparm_01" {
Parent = Uparm;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0432;
RadiusY = 0.0432;
RadiusZ = 0.1532;
GridLong = 12.0000;
GridLat1 = 12.0000;
}
Body "Uparm_re" {
Parent = Thorax;
Joint = Joint_Thorax->Uparm_re;
Mass = 1.676195;
Inertia = 0.012807, 0.000000, 0.000000,
0.012549, 0.000000,
0.001750;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Uparm_re_Joint_Thorax" {
Parent = Uparm_re;
Position = 0.0000, 0.0000, 0.1471;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_re_ARSDA_Thorax" {
Parent = Uparm_re;
Position = 0.0000, 0.0000, 0.1471;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_re_Joint_Forearm_re" {
Parent = Uparm_re;
Position = 0.0000, 0.0000, -0.1594;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Uparm_re_Drehfeder3D_Forearm_re" {
Parent = Uparm_re;
Position = 0.0000, 0.0000, -0.1594;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
}

```

```

Degrees = deg;
}
Triad "Uparm_re_AerodynSimple_Skijump_4" {
Parent = Uparm_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Uparm_re_01" {
Parent = Uparm_re;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0432;
RadiusY = 0.0432;
RadiusZ = 0.1532;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Forearm" {
Parent = Uparm;
Joint = Joint_Uparm->Forearm;
Mass = 1.003106;
Inertia = 0.005961, 0.000000, 0.000000,
0.005785, 0.000000,
0.000713;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Forearm_Joint_Uparm" {
Parent = Forearm;
Position = 0.0000, 0.0000, 0.1117;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_Drehfeder3D_Uparm" {
Parent = Forearm;
Position = 0.0000, 0.0000, 0.1117;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_Joint_Hand" {
Parent = Forearm;
Position = 0.0000, 0.0000, -0.1608;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_Drehfeder3D_Hand" {
Parent = Forearm;
Position = 0.0000, 0.0000, -0.1608;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_AerodynSimple_Skijump_4" {
Parent = Forearm;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Forearm_01" {
Parent = Forearm;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0453;
RadiusY = 0.0453;
RadiusZ = 0.1532;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Forearm_re" {
Parent = Uparm_re;

```



```

Joint = Joint_Uparm_re->Forearm_re;
Mass = 1.003106;
Inertia = 0.005961, 0.000000, 0.000000,
0.005785, 0.000000,
0.000713;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Forearm_re_Joint_Uparm_re" {
Parent = Forearm_re;
Position = 0.0000, 0.0000, 0.1117;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_re_Drehfeder3D_Uparm_re" {
Parent = Forearm_re;
Position = 0.0000, 0.0000, 0.1117;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_re_Joint_Hand_re" {
Parent = Forearm_re;
Position = 0.0000, 0.0000, -0.1608;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_re_Drehfeder3D_Hand_re" {
Parent = Forearm_re;
Position = 0.0000, 0.0000, -0.1608;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Forearm_re_AerodynSimple_Skijump_4" {
Parent = Forearm_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Forearm_re_01" {
Parent = Forearm_re;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0453;
RadiusY = 0.0453;
RadiusZ = 0.1532;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Hand" {
Parent = Forearm;
Joint = Joint_Forearm->Hand;
Mass = 0.408470;
Inertia = 0.000767, 0.000000, 0.000000,
0.000633, 0.000000,
0.000222;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Hand_Joint_Forearm" {
Parent = Hand;
Position = 0.0000, 0.0000, 0.0571;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Hand_Drehfeder3D_Forearm" {
Parent = Hand;
Position = 0.0000, 0.0000, 0.0571;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
}

```

```

Triad "Hand_AerodynSimple_Skijump_2" {
  Parent = Hand;
  Position = 0.0000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Primitive "Primitive_Hand_01" {
  Parent = Hand;
  Position = 0.0000, 0.0000, 0.0000;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.7000, 0.7000, 0.7000, 0.6000;
  Type = Sphere;
  RadiusX = 0.0270;
  RadiusY = 0.0875;
  RadiusZ = 0.0957;
  GridLong = 12.0000;
  GridLati = 12.0000;
}
Body "Hand_re" {
  Parent = Forearm_re;
  Joint = Joint_Forearm_re->Hand_re;
  Mass = 0.408470;
  Inertia = 0.000767, 0.000000, 0.000000,
0.000633, 0.000000,
0.000222;
  CGFramePosition = 0.0, 0.0, 0.0;
  CGFrameOrientation = 0.0, 0.0, 0.0;
  AngleType = euler;
  Degrees = deg;
}
Triad "Hand_re_Joint_Forearm_re" {
  Parent = Hand_re;
  Position = 0.0000, 0.0000, 0.0571;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Hand_re_Drehfeder3D_Forearm_re" {
  Parent = Hand_re;
  Position = 0.0000, 0.0000, 0.0571;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Hand_re_AerodynSimple_Skijump_2" {
  Parent = Hand_re;
  Position = 0.0000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Primitive "Primitive_Hand_re_01" {
  Parent = Hand_re;
  Position = 0.0000, 0.0000, 0.0000;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.7000, 0.7000, 0.7000, 0.6000;
  Type = Sphere;
  RadiusX = 0.0270;
  RadiusY = 0.0875;
  RadiusZ = 0.0957;
  GridLong = 12.0000;
  GridLati = 12.0000;
}
Body "Thigh" {
  Parent = Pelvis;
  Joint = Joint_Pelvis->Thigh;
  Mass = 6.112484;
  Inertia = 0.105055, 0.000000, 0.000000,
0.109353, 0.000000,
0.018737;
  CGFramePosition = 0.0, 0.0, 0.0;
  CGFrameOrientation = 0.0, 0.0, 0.0;
  AngleType = euler;
  Degrees = deg;
}
Triad "Thigh_Joint_Pelvis" {
  Parent = Thigh;
  Position = 0.0000, 0.0000, 0.1744;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}

```

```

}
Triad "Thigh_ARSDA_Pelvis" {
Parent = Thigh;
Position = 0.0000, 0.0000, 0.1744;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_Joint_Shank" {
Parent = Thigh;
Position = 0.0000, 0.0000, -0.2509;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_ARSDA_Shank" {
Parent = Thigh;
Position = 0.0000, 0.0000, -0.2509;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_AerodynSimple_Skijump_4" {
Parent = Thigh;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Thigh_01" {
Parent = Thigh;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0828;
RadiusY = 0.0828;
RadiusZ = 0.2552;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Thigh_re" {
Parent = Pelvis;
Joint = Joint_Pelvis->Thigh_re;
Mass = 6.112484;
Inertia = 0.105055, 0.000000, 0.000000,
0.109353, 0.000000,
0.018737;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Thigh_re_Joint_Pelvis" {
Parent = Thigh_re;
Position = 0.0000, 0.0000, 0.1744;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_re_ARSDA_Pelvis" {
Parent = Thigh_re;
Position = 0.0000, 0.0000, 0.1744;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_re_Joint_Shank_re" {
Parent = Thigh_re;
Position = 0.0000, 0.0000, -0.2509;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_re_ARSDA_Shank_re" {
Parent = Thigh_re;
Position = 0.0000, 0.0000, -0.2509;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Thigh_re_AerodynSimple_Skijump_4" {
Parent = Thigh_re;

```

```
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Thigh_re_01" {
  Parent = Thigh_re;
  Position = 0.0000, 0.0000, 0.0000;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.7000, 0.7000, 0.7000, 0.6000;
  Type = Sphere;
  RadiusX = 0.0828;
  RadiusY = 0.0828;
  RadiusZ = 0.2552;
  GridLong = 12.0000;
  GridLati = 12.0000;
}
Body "Shank" {
  Parent = Thigh;
  Joint = Joint_Thigh->Shank;
  Mass = 2.687749;
  Inertia = 0.039429, 0.000000, 0.000000,
0.039235, 0.000000,
0.002903;
  CGFramePosition = 0.0, 0.0, 0.0;
  CGFrameOrientation = 0.0, 0.0, 0.0;
  AngleType = euler;
  Degrees = deg;
}
Triad "Shank_Joint_Thigh" {
  Parent = Shank;
  Position = 0.0000, 0.0000, 0.1825;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Shank_ARSDA_Thigh" {
  Parent = Shank;
  Position = 0.0000, 0.0000, 0.1825;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Shank_Joint_Foot" {
  Parent = Shank;
  Position = 0.0000, 0.0000, -0.2322;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Shank_ARSDA_Foot" {
  Parent = Shank;
  Position = 0.0000, 0.0000, -0.2322;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Shank_Kontakt_Thorax" {
  Parent = Shank;
  Position = 0.0000, 0.0000, 0.2073;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Shank_AerodynSimple_Skijump_5" {
  Parent = Shank;
  Position = 0.0000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Primitive "Primitive_Shank_01" {
  Parent = Shank;
  Position = 0.0000, 0.0000, 0.0000;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.7000, 0.7000, 0.7000, 0.6000;
  Type = Sphere;
  RadiusX = 0.0522;
  RadiusY = 0.0522;
  RadiusZ = 0.2488;
  GridLong = 12.0000;
  GridLati = 12.0000;
}
```

```

}
Body "Shank_re" {
Parent = Thigh_re;
Joint = Joint_Thigh_re->Shank_re;
Mass = 2.687749;
Inertia = 0.039429, 0.000000, 0.000000,
0.039235, 0.000000,
0.002903;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Shank_re_Joint_Thigh_re" {
Parent = Shank_re;
Position = 0.0000, 0.0000, 0.1825;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Shank_re_ARSDA_Thigh_re" {
Parent = Shank_re;
Position = 0.0000, 0.0000, 0.1825;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Shank_re_Joint_Foot_re" {
Parent = Shank_re;
Position = 0.0000, 0.0000, -0.2322;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Shank_re_ARSDA_Foot_re" {
Parent = Shank_re;
Position = 0.0000, 0.0000, -0.2322;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Shank_re_Kontakt_Thorax" {
Parent = Shank_re;
Position = 0.0000, 0.0000, 0.2073;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Shank_re_AerodynSimple_Skijump_5" {
Parent = Shank_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Shank_re_01" {
Parent = Shank_re;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.0522;
RadiusY = 0.0522;
RadiusZ = 0.2488;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Foot" {
Parent = Shank;
Joint = Joint_Shank->Foot;
Mass = 0.894626;
Inertia = 0.003615, 0.000000, 0.000000,
0.003353, 0.000000,
0.000822;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Foot_Joint_Shank" {
Parent = Foot;
Position = -0.0640, 0.0000, 0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
}

```

```

AngleType = cardan;
Degrees = deg;
}
Triad "Foot_ARSDA_Shank" {
Parent = Foot;
Position = -0.0640, 0.0000, 0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Foot_Joint_Ski_mid" {
Parent = Foot;
Position = 0.1334, 0.0000, -0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Foot_Drehfeder3D_Ski_mid" {
Parent = Foot;
Position = 0.1334, 0.0000, -0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Foot_Kontakt_Ski_mid_4" {
Parent = Foot;
Position = -0.1334, 0.0000, -0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Foot_01" {
Parent = Foot;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.1334;
RadiusY = 0.0521;
RadiusZ = 0.0294;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Foot_re" {
Parent = Shank_re;
Joint = Joint_Shank_re->Foot_re;
Mass = 0.894626;
Inertia = 0.003615, 0.000000, 0.000000,
0.003353, 0.000000,
0.000822;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Foot_re_Joint_Shank_re" {
Parent = Foot_re;
Position = -0.0640, 0.0000, 0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Foot_re_ARSDA_Shank_re" {
Parent = Foot_re;
Position = -0.0640, 0.0000, 0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Foot_re_Joint_Ski_mid_re" {
Parent = Foot_re;
Position = 0.1334, 0.0000, -0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Foot_re_Drehfeder3D_Ski_mid_re" {
Parent = Foot_re;
Position = 0.1334, 0.0000, -0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
}

```

```

Triad "Foot_re_Kontakt_Ski_mid_re_4" {
Parent = Foot_re;
Position = -0.1334, 0.0000, -0.0442;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Foot_re_01" {
Parent = Foot_re;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.7000, 0.7000, 0.7000, 0.6000;
Type = Sphere;
RadiusX = 0.1334;
RadiusY = 0.0521;
RadiusZ = 0.0294;
GridLong = 12.0000;
GridLati = 12.0000;
}
Body "Ski_mid" {
Parent = Foot;
Joint = Joint_Foot->Ski_mid;
Mass = 1.000000;
Inertia = 0.000852, 0.000000, 0.000000,
0.083352, 0.000000,
0.084167;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Ski_mid_Joint_Foot" {
Parent = Ski_mid;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_Drehfeder3D_Foot" {
Parent = Ski_mid;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_Joint_Ski_front" {
Parent = Ski_mid;
Position = 0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_Drehfeder3D_Ski_front" {
Parent = Ski_mid;
Position = 0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_Joint_Ski_end" {
Parent = Ski_mid;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_Drehfeder3D_Ski_end" {
Parent = Ski_mid;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_ContactSpline_Skijump_6" {
Parent = Ski_mid;
Position = 0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_ContactSpline_Skijump_7" {
Parent = Ski_mid;
Position = -0.5000, 0.0000, 0.0000;
}

```

```

Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_AerodynSimple_Skijump_8" {
Parent = Ski_mid;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 90.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_Kontakt_Foot_4" {
Parent = Ski_mid;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Ski_mid_01" {
Parent = Ski_mid;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.2000, 0.2000, 0.9000, 0.5500;
Type = Box;
LengthX = 1.0000;
LengthY = 0.1000;
LengthZ = 0.0150;
}
Body "Ski_mid_re" {
Parent = Foot_re;
Joint = Joint_Foot_re->Ski_mid_re;
Mass = 1.000000;
Inertia = 0.000852, 0.000000, 0.000000,
0.083352, 0.000000,
0.084167;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Ski_mid_re_Joint_Foot_re" {
Parent = Ski_mid_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_Drehfeder3D_Foot_re" {
Parent = Ski_mid_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_Joint_Ski_front_re" {
Parent = Ski_mid_re;
Position = 0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_Drehfeder3D_Ski_front_re" {
Parent = Ski_mid_re;
Position = 0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_Joint_Ski_end_re" {
Parent = Ski_mid_re;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_Drehfeder3D_Ski_end_re" {
Parent = Ski_mid_re;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_ContactSpline_Skijump_6" {

```



```

Parent = Ski_mid_re;
Position = 0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_ContactSpline_Skijump_7" {
Parent = Ski_mid_re;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_AerodynSimple_Skijump_8" {
Parent = Ski_mid_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 90.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_mid_re_Kontakt_Foot_re_4" {
Parent = Ski_mid_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Ski_mid_re_01" {
Parent = Ski_mid_re;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.2000, 0.2000, 0.9000, 0.5500;
Type = Box;
LengthX = 1.0000;
LengthY = 0.1000;
LengthZ = 0.0150;
}
Body "Ski_front" {
Parent = Ski_mid;
Joint = Joint_Ski_mid->Ski_front;
Mass = 1.000000;
Inertia = 0.000852, 0.000000, 0.000000,
0.083352, 0.000000,
0.084167;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Ski_front_Joint_Ski_mid" {
Parent = Ski_front;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_front_Drehfeder3D_Ski_mid" {
Parent = Ski_front;
Position = -0.5000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_front_ContactSpline_Skijump_2" {
Parent = Ski_front;
Position = 0.4000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_front_AerodynSimple_Skijump_3" {
Parent = Ski_front;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 90.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_front_ContactSpline_Head_4" {
Parent = Ski_front;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}

```

```

}
Primitive "Primitive_Ski_front_01" {
  Parent = Ski_front;
  Position = 0.0000, 0.0000, 0.0000;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.2000, 0.2000, 0.9000, 0.5500;
  Type = Box;
  LengthX = 1.0000;
  LengthY = 0.1000;
  LengthZ = 0.0150;
}
Body "Ski_front_re" {
  Parent = Ski_mid_re;
  Joint = Joint_Ski_mid_re->Ski_front_re;
  Mass = 1.000000;
  Inertia = 0.000852, 0.000000, 0.000000,
0.083352, 0.000000,
0.084167;
  CGFramePosition = 0.0, 0.0, 0.0;
  CGFrameOrientation = 0.0, 0.0, 0.0;
  AngleType = euler;
  Degrees = deg;
}
Triad "Ski_front_re_Joint_Ski_mid_re" {
  Parent = Ski_front_re;
  Position = -0.5000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Ski_front_re_Drehfeder3D_Ski_mid_re" {
  Parent = Ski_front_re;
  Position = -0.5000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Ski_front_re_ContactSpline_Skijump_2" {
  Parent = Ski_front_re;
  Position = 0.4000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Ski_front_re_AerodynSimple_Skijump_3" {
  Parent = Ski_front_re;
  Position = 0.0000, 0.0000, 0.0000;
  Orientation = 0.0000, 90.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Primitive "Primitive_Ski_front_re_01" {
  Parent = Ski_front_re;
  Position = 0.0000, 0.0000, 0.0000;
  Cardan = 0.0000, 0.0000, 0.0000;
  Degrees = deg;
  Color = 0.2000, 0.2000, 0.9000, 0.5500;
  Type = Box;
  LengthX = 1.0000;
  LengthY = 0.1000;
  LengthZ = 0.0150;
}
Body "Ski_end" {
  Parent = Ski_mid;
  Joint = Joint_Ski_mid->Ski_end;
  Mass = 0.600000;
  Inertia = 0.000511, 0.000000, 0.000000,
0.018011, 0.000000,
0.018500;
  CGFramePosition = 0.0, 0.0, 0.0;
  CGFrameOrientation = 0.0, 0.0, 0.0;
  AngleType = euler;
  Degrees = deg;
}
Triad "Ski_end_Joint_Ski_mid" {
  Parent = Ski_end;
  Position = 0.3000, 0.0000, 0.0000;
  Orientation = 0.0000, 0.0000, 0.0000;
  AngleType = cardan;
  Degrees = deg;
}
Triad "Ski_end_Drehfeder3D_Ski_mid" {
  Parent = Ski_end;

```

```

Position = 0.3000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_end>ContactSpline_Skijump_2" {
Parent = Ski_end;
Position = -0.2000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_end_AerodynSimple_Skijump_3" {
Parent = Ski_end;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 90.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Ski_end_01" {
Parent = Ski_end;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.2000, 0.2000, 0.9000, 0.5500;
Type = Box;
LengthX = 0.6000;
LengthY = 0.1000;
LengthZ = 0.0150;
}
}
Body "Ski_end_re" {
Parent = Ski_mid_re;
Joint = Joint_Ski_mid_re->Ski_end_re;
Mass = 0.600000;
Inertia = 0.000511, 0.000000, 0.000000,
0.018011, 0.000000,
0.018500;
CGFramePosition = 0.0, 0.0, 0.0;
CGFrameOrientation = 0.0, 0.0, 0.0;
AngleType = euler;
Degrees = deg;
}
Triad "Ski_end_re_Joint_Ski_mid_re" {
Parent = Ski_end_re;
Position = 0.3000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_end_re_Drehfeder3D_Ski_mid_re" {
Parent = Ski_end_re;
Position = 0.3000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_end_re>ContactSpline_Skijump_2" {
Parent = Ski_end_re;
Position = -0.2000, 0.0000, 0.0000;
Orientation = 0.0000, 0.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Triad "Ski_end_re_AerodynSimple_Skijump_3" {
Parent = Ski_end_re;
Position = 0.0000, 0.0000, 0.0000;
Orientation = 0.0000, 90.0000, 0.0000;
AngleType = cardan;
Degrees = deg;
}
Primitive "Primitive_Ski_end_re_01" {
Parent = Ski_end_re;
Position = 0.0000, 0.0000, 0.0000;
Cardan = 0.0000, 0.0000, 0.0000;
Degrees = deg;
Color = 0.2000, 0.2000, 0.9000, 0.5500;
Type = Box;
LengthX = 0.6000;
LengthY = 0.1000;
LengthZ = 0.0150;
}
}
Free "Joint_Skijump->Pelvis" {
Parent = Pelvis;
TriadFrom = Skijump_Joint_Pelvis;
}

```

```

TriadTo = Pelvis_Joint_Skijump;
Qinit = 30.000000, 0.000000, 80.150000, 0.000000, 114.000000, 0.000000;
QDinit = 25.000000, 0.000000, -20.000000, 0.000000, 0.000000, 0.000000;
AngleType = cardan;
Degrees = deg;
}
}
Revolute "Joint_Pelvis->Lumbus" {
Parent = Lumbus;
TriadFrom = Pelvis_Joint_Lumbus;
TriadTo = Lumbus_Joint_Pelvis;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Lumbus->Thorax" {
Parent = Thorax;
TriadFrom = Lumbus_Joint_Thorax;
TriadTo = Thorax_Joint_Lumbus;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 15.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Thorax->Neck" {
Parent = Neck;
TriadFrom = Thorax_Joint_Neck;
TriadTo = Neck_Joint_Thorax;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Neck->Head" {
Parent = Head;
TriadFrom = Neck_Joint_Head;
TriadTo = Head_Joint_Neck;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -5.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Thorax->Uparm" {
Parent = Uparm;
TriadFrom = Thorax_Joint_Uparm;
TriadTo = Uparm_Joint_Thorax;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 20.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Thorax->Uparm_re" {
Parent = Uparm_re;
TriadFrom = Thorax_Joint_Uparm_re;
TriadTo = Uparm_re_Joint_Thorax;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 20.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Uparm->Forearm" {
Parent = Forearm;
TriadFrom = Uparm_Joint_Forearm;
TriadTo = Forearm_Joint_Uparm;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -10.000000;
QDinit = 0.0;
Degrees = deg;
}
}
Revolute "Joint_Uparm_re->Forearm_re" {
Parent = Forearm_re;
TriadFrom = Uparm_re_Joint_Forearm_re;
TriadTo = Forearm_re_Joint_Uparm_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -10.000000;
QDinit = 0.0;
}
}

```

```
Degrees = deg;
}
Revolute "Joint_Forearm->Hand" {
Parent = Hand;
TriadFrom = Forearm_Joint_Hand;
TriadTo = Hand_Joint_Forearm;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 5.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Forearm_re->Hand_re" {
Parent = Hand_re;
TriadFrom = Forearm_re_Joint_Hand_re;
TriadTo = Hand_re_Joint_Forearm_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 5.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Pelvis->Thigh" {
Parent = Thigh;
TriadFrom = Pelvis_Joint_Thigh;
TriadTo = Thigh_Joint_Pelvis;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -170.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Pelvis->Thigh_re" {
Parent = Thigh_re;
TriadFrom = Pelvis_Joint_Thigh_re;
TriadTo = Thigh_re_Joint_Pelvis;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -170.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Thigh->Shank" {
Parent = Shank;
TriadFrom = Thigh_Joint_Shank;
TriadTo = Shank_Joint_Thigh;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 130.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Thigh_re->Shank_re" {
Parent = Shank_re;
TriadFrom = Thigh_re_Joint_Shank_re;
TriadTo = Shank_re_Joint_Thigh_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 130.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Shank->Foot" {
Parent = Foot;
TriadFrom = Shank_Joint_Foot;
TriadTo = Foot_Joint_Shank;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -40.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Shank_re->Foot_re" {
Parent = Foot_re;
TriadFrom = Shank_re_Joint_Foot_re;
TriadTo = Foot_re_Joint_Shank_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = -40.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Foot->Ski_mid" {
Parent = Ski_mid;
```

```

TriadFrom = Foot_Joint_Ski_mid;
TriadTo = Ski_mid_Joint_Foot;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.500000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Foot_re->Ski_mid_re" {
Parent = Ski_mid_re;
TriadFrom = Foot_re_Joint_Ski_mid_re;
TriadTo = Ski_mid_re_Joint_Foot_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.500000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Ski_mid->Ski_front" {
Parent = Ski_front;
TriadFrom = Ski_mid_Joint_Ski_front;
TriadTo = Ski_front_Joint_Ski_mid;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Ski_mid_re->Ski_front_re" {
Parent = Ski_front_re;
TriadFrom = Ski_mid_re_Joint_Ski_front_re;
TriadTo = Ski_front_re_Joint_Ski_mid_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Ski_mid->Ski_end" {
Parent = Ski_end;
TriadFrom = Ski_mid_Joint_Ski_end;
TriadTo = Ski_end_Joint_Ski_mid;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.000000;
QDinit = 0.0;
Degrees = deg;
}
Revolute "Joint_Ski_mid_re->Ski_end_re" {
Parent = Ski_end_re;
TriadFrom = Ski_mid_re_Joint_Ski_end_re;
TriadTo = Ski_end_re_Joint_Ski_mid_re;
JointAxis = 0.0, 1.0, 0.0;
JointPoint = 0.0, 0.0, 0.0;
Qinit = 0.000000;
QDinit = 0.0;
Degrees = deg;
}
ContactSimple "Kontakt_Shank->Thorax" {
TriadFrom = Thorax_Kontakt_Shank;
TriadTo = Shank_Kontakt_Thorax;
ElasticLinearZ = 10000.0000;
DampingLinearZ = 1000.0000;
FrictionXY = 0.0010;
}
ContactSimple "Kontakt_Shank_re->Thorax" {
TriadFrom = Thorax_Kontakt_Shank_re;
TriadTo = Shank_re_Kontakt_Thorax;
ElasticLinearZ = 10000.0000;
DampingLinearZ = 1000.0000;
FrictionXY = 0.0010;
}
ContactSimple "Kontakt_Foot->Ski_mid" {
TriadFrom = Ski_mid_Kontakt_Foot_4;
TriadTo = Foot_Kontakt_Ski_mid_4;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 10000.0000;
FrictionXY = 0.0010;
}
ContactSimple "Kontakt_Foot_re->Ski_mid_re" {
TriadFrom = Ski_mid_re_Kontakt_Foot_re_4;
TriadTo = Foot_re_Kontakt_Ski_mid_re_4;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 10000.0000;
}

```

```
FrictionXY = 0.0010;
}
}
RSDA "Drehfeder3D_Lumbus->Pelvis" {
  Joint1D = Joint_Pelvis->Lumbus;
  ElasticLinear = 30.0000;
  DampingLinear = 1.0000;
  ZeroAngle = 0.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Head->Neck" {
  Joint1D = Joint_Neck->Head;
  ElasticLinear = 25.0000;
  DampingLinear = 1.0000;
  ZeroAngle = -5.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Forearm->Uparm" {
  Joint1D = Joint_Uparm->Forearm;
  ElasticLinear = 20.0000;
  DampingLinear = 1.0000;
  ZeroAngle = -10.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Forearm_re->Uparm_re" {
  Joint1D = Joint_Uparm_re->Forearm_re;
  ElasticLinear = 20.0000;
  DampingLinear = 1.0000;
  ZeroAngle = -10.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Hand->Forearm" {
  Joint1D = Joint_Forearm->Hand;
  ElasticLinear = 15.0000;
  DampingLinear = 1.0000;
  ZeroAngle = 5.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Hand_re->Forearm_re" {
  Joint1D = Joint_Forearm_re->Hand_re;
  ElasticLinear = 15.0000;
  DampingLinear = 1.0000;
  ZeroAngle = 5.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Ski_mid->Foot" {
  Joint1D = Joint_Foot->Ski_mid;
  ElasticLinear = 10.0000;
  DampingLinear = 0.1000;
  ZeroAngle = 0.5000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Ski_mid_re->Foot_re" {
  Joint1D = Joint_Foot_re->Ski_mid_re;
  ElasticLinear = 10.0000;
  DampingLinear = 0.1000;
  ZeroAngle = 0.5000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Ski_front->Ski_mid" {
  Joint1D = Joint_Ski_mid->Ski_front;
  ElasticLinear = 10.0000;
  DampingLinear = 0.1000;
  ZeroAngle = 0.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Ski_end->Ski_mid" {
  Joint1D = Joint_Ski_mid->Ski_end;
  ElasticLinear = 10.0000;
  DampingLinear = 0.1000;
  ZeroAngle = 0.0000;
  OneSide = bi;
  Degrees = deg;
}
RSDA "Drehfeder3D_Ski_front_re->Ski_mid_re" {
  Joint1D = Joint_Ski_mid_re->Ski_front_re;
```

```

ElasticLinear = 10.0000;
DampingLinear = 0.1000;
ZeroAngle = 0.0000;
OneSide = bi;
Degrees = deg;
}
RSDA "Drehfeder3D_Ski_end_re->Ski_mid_re" {
Joint1D = Joint_Ski_mid_re->Ski_end_re;
ElasticLinear = 10.0000;
DampingLinear = 0.1000;
ZeroAngle = 0.0000;
OneSide = bi;
Degrees = deg;
}
ContactSpline "ContactSpline_Head->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Head_ContactSpline_Skijump_3;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Head->Skijump" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Head_ContactSpline_Skijump_3;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Head->Ski_front" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Head_ContactSpline_Ski_front_4;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Head->Ski_front" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Head_ContactSpline_Ski_front_4;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_mid_ContactSpline_Skijump_6;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid->Skijump" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_mid_ContactSpline_Skijump_6;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid->Skijump 00" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_mid_ContactSpline_Skijump_7;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid->Skijump 00" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_mid_ContactSpline_Skijump_7;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
}

```



```
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid_re->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_mid_re_ContactSpline_Skijump_6;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid_re->Skijump" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_mid_re_ContactSpline_Skijump_6;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid_re->Skijump 01" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_mid_re_ContactSpline_Skijump_7;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_mid_re->Skijump 01" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_mid_re_ContactSpline_Skijump_7;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_front->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_front_ContactSpline_Skijump_2;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_front->Skijump" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_front_ContactSpline_Skijump_2;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_front_re->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_front_re_ContactSpline_Skijump_2;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_front_re->Skijump" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_front_re_ContactSpline_Skijump_2;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_end->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_end_ContactSpline_Skijump_2;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_end->Skijump" {
```

```

TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_end_ContactSpline_Skijump_2;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_end_re->Skijump" {
TriadFrom = Skijump_ContactSpline_01;
TriadTo = Ski_end_re_ContactSpline_Skijump_2;
RangeX1 = 0;
RangeX2 = 65;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
ContactSpline "ContactSpline_Ski_end_re->Skijump" {
TriadFrom = Skijump_ContactSpline_02;
TriadTo = Ski_end_re_ContactSpline_Skijump_2;
RangeX1 = 65;
RangeX2 = 200;
ElasticLinearZ = 1000000.0000;
DampingLinearZ = 1.0000;
FrictionXY = 0.0010;
}
AerodynSimple "AerodynSimple_Pelvis->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Pelvis_AerodynSimple_Skijump_7;
Plane = 0.009140;
}
AerodynSimple "AerodynSimple_Lumbus->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Lumbus_AerodynSimple_Skijump_4;
Plane = 0.014784;
}
AerodynSimple "AerodynSimple_Thorax->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Thorax_AerodynSimple_Skijump_10;
Plane = 0.134158;
}
AerodynSimple "AerodynSimple_Head->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Head_AerodynSimple_Skijump_2;
Plane = 0.034256;
}
AerodynSimple "AerodynSimple_Uparm->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Uparm_AerodynSimple_Skijump_4;
Plane = 0.022588;
}
AerodynSimple "AerodynSimple_Uparm_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Uparm_re_AerodynSimple_Skijump_4;
Plane = 0.022588;
}
AerodynSimple "AerodynSimple_Forearm->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Forearm_AerodynSimple_Skijump_4;
Plane = 0.007904;
}
AerodynSimple "AerodynSimple_Forearm_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Forearm_re_AerodynSimple_Skijump_4;
Plane = 0.007904;
}
AerodynSimple "AerodynSimple_Hand->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Hand_AerodynSimple_Skijump_2;
Plane = 0.004189;
}
AerodynSimple "AerodynSimple_Hand_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Hand_re_AerodynSimple_Skijump_2;
Plane = 0.004189;
}
AerodynSimple "AerodynSimple_Thigh->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Thigh_AerodynSimple_Skijump_4;
Plane = 0.020036;
}
AerodynSimple "AerodynSimple_Thigh_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Thigh_re_AerodynSimple_Skijump_4;
}

```

```

Plane = 0.020036;
}
AerodynSimple "AerodynSimple_Shank->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Shank_AerodynSimple_Skijump_5;
Plane = 0.012326;
}
AerodynSimple "AerodynSimple_Shank_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Shank_re_AerodynSimple_Skijump_5;
Plane = 0.012326;
}
AerodynSimple "AerodynSimple_Ski_mid->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Ski_mid_AerodynSimple_Skijump_8;
Plane = 0.025000;
}
AerodynSimple "AerodynSimple_Ski_mid_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Ski_mid_re_AerodynSimple_Skijump_8;
Plane = 0.025000;
}
AerodynSimple "AerodynSimple_Ski_front->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Ski_front_AerodynSimple_Skijump_3;
Plane = 0.025000;
}
AerodynSimple "AerodynSimple_Ski_front_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Ski_front_re_AerodynSimple_Skijump_3;
Plane = 0.025000;
}
AerodynSimple "AerodynSimple_Ski_end->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Ski_end_AerodynSimple_Skijump_3;
Plane = 0.015000;
}
AerodynSimple "AerodynSimple_Ski_end_re->Skijump" {
TriadFrom = world_aerodynamicsimple;
TriadTo = Ski_end_re_AerodynSimple_Skijump_3;
Plane = 0.015000;
}
ARSDA "ARSDA_Thigh->Pelvis" {
Joint1D = Joint_Pelvis->Thigh;
ElasticLinear = 17.0000;
DampingLinear = 1.0000;
ZeroAngle = -170.0000;
ZeroAngle1 = -170.0000;
ZeroAngle2 = -190.0000;
ZeroAngle3 = -10.0000;
ZeroAngle4 = -10.0000;
ZeroAngle5 = -10.0000;
ZeroAngle6 = -10.0000;
ZeroAngle7 = -160.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Thigh_re->Pelvis" {
Joint1D = Joint_Pelvis->Thigh_re;
ElasticLinear = 17.0000;
DampingLinear = 1.0000;
ZeroAngle = -170.0000;
ZeroAngle1 = -170.0000;
ZeroAngle2 = -190.0000;
ZeroAngle3 = -10.0000;
ZeroAngle4 = -10.0000;
ZeroAngle5 = -10.0000;
ZeroAngle6 = -10.0000;
ZeroAngle7 = -160.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Thorax->Lumbus" {
Joint1D = Joint_Lumbus->Thorax;
ElasticLinear = 30.0000;
DampingLinear = 1.0000;
ZeroAngle = 15.0000;
ZeroAngle1 = 10.0000;
ZeroAngle2 = 25.0000;
ZeroAngle3 = -5.0000;
ZeroAngle4 = 15.0000;
ZeroAngle5 = 2.0000;
ZeroAngle6 = 20.0000;
ZeroAngle7 = 25.0000;
}

```

```
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Neck->Thorax" {
Joint1D = Joint_Thorax->Neck;
ElasticLinear = 25.0000;
DampingLinear = 1.0000;
ZeroAngle = 0.0000;
ZeroAngle1 = -5.0000;
ZeroAngle2 = 5.0000;
ZeroAngle3 = -5.0000;
ZeroAngle4 = -5.0000;
ZeroAngle5 = -15.0000;
ZeroAngle6 = 10.0000;
ZeroAngle7 = 20.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Uparm->Thorax" {
Joint1D = Joint_Thorax->Uparm;
ElasticLinear = 25.0000;
DampingLinear = 1.0000;
ZeroAngle = 20.0000;
ZeroAngle1 = 20.0000;
ZeroAngle2 = 20.0000;
ZeroAngle3 = 30.0000;
ZeroAngle4 = 20.0000;
ZeroAngle5 = 0.0000;
ZeroAngle6 = 30.0000;
ZeroAngle7 = 50.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Uparm_re->Thorax" {
Joint1D = Joint_Thorax->Uparm_re;
ElasticLinear = 25.0000;
DampingLinear = 1.0000;
ZeroAngle = 20.0000;
ZeroAngle1 = 20.0000;
ZeroAngle2 = 20.0000;
ZeroAngle3 = 30.0000;
ZeroAngle4 = 20.0000;
ZeroAngle5 = 0.0000;
ZeroAngle6 = 30.0000;
ZeroAngle7 = 50.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Shank->Thigh" {
Joint1D = Joint_Thigh->Shank;
ElasticLinear = 20.0000;
DampingLinear = 1.0000;
ZeroAngle = 130.0000;
ZeroAngle1 = 130.0000;
ZeroAngle2 = 230.0000;
ZeroAngle3 = 10.0000;
ZeroAngle4 = 0.0000;
ZeroAngle5 = 0.0000;
ZeroAngle6 = 0.0000;
ZeroAngle7 = 130.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Shank_re->Thigh_re" {
Joint1D = Joint_Thigh_re->Shank_re;
ElasticLinear = 20.0000;
DampingLinear = 1.0000;
ZeroAngle = 130.0000;
ZeroAngle1 = 130.0000;
ZeroAngle2 = 230.0000;
ZeroAngle3 = 10.0000;
ZeroAngle4 = 0.0000;
ZeroAngle5 = 0.0000;
ZeroAngle6 = 0.0000;
ZeroAngle7 = 130.0000;
OneSide = bi;
Degrees = deg;
}
ARSDA "ARSDA_Foot->Shank" {
Joint1D = Joint_Shank->Foot;
ElasticLinear = 25.0000;
DampingLinear = 1.0000;
ZeroAngle = -40.0000;
ZeroAngle1 = -40.0000;
```

```
ZeroAngle2 = -70.0000;  
ZeroAngle3 = 0.0000;  
ZeroAngle4 = -70.0000;  
ZeroAngle5 = -70.0000;  
ZeroAngle6 = -70.0000;  
ZeroAngle7 = -10.0000;  
OneSide = bi;  
Degrees = deg;  
}  
ARSDA "ARSDA_Foot_re->Shank_re" {  
Joint1D = Joint_Shank_re->Foot_re;  
ElasticLinear = 25.0000;  
DampingLinear = 1.0000;  
ZeroAngle = -40.0000;  
ZeroAngle1 = -40.0000;  
ZeroAngle2 = -70.0000;  
ZeroAngle3 = 0.0000;  
ZeroAngle4 = -70.0000;  
ZeroAngle5 = -70.0000;  
ZeroAngle6 = -70.0000;  
ZeroAngle7 = -10.0000;  
OneSide = bi;  
Degrees = deg;  
}
```