Inverse Kinematik in der Computeranimation

Dr. Martin Hering-Bertram

Zusammenfassung

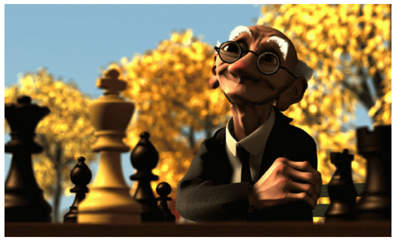
Die computergestützte Animation von Figuren und Charakteren erfordert hochentwickelte Techniken für die Modellierung realistischer Bewegungen. in Ergänzung zu physikalischen Modellen hat sich die inverse Kinematik als ein unverzichtbarer Baustein in Animationsprozessen etabliert. Neben den mathematischen und algorithmischen Grundlagen werden auch die Möglichkeiten und Grenzen dieser Methode aufgezeigt. Anwendungen liegen nicht nur in der Erzeugung von Spezialeffekten in Filmen oder Cartoons, sondern auch in der Simulation von Bewegungsabläufen, z.B. in der Robotik.

Überblick

1. Motivation

Das Problem:

Wie stellt man in der *Character-Animation* die Gelenkwinkel ein, so dass der Darsteller (hier: Geri) einen Gegenstand greift.



Quelle: Geri´s Game, Pixar Animation Studios [T. deRose M. Kass und T. Truong, *Subdivision Surfaces in Character Animation*, ACM Siggraph 1998, pp. 85-94]

1. Motivation

Die Lösung:

Mit Hilfe der **inversen Kinematik (IK)** kann man den **Endeffektor** einer Kette aus Gelenken und starren Bindungen in eine vorgegebene Position bringen. Dies ermöglicht dem Animateur:

- die Positionierung des Endeffektors zu bestimmten Zeitpunkten (*Key Frames*)
- Interpolation der Position und ggf. Orientierung nach der Zeit
- automatische Bewegung der Figur

Die Vorteile liegen in der geringeren Anzahl von Animationsparametern und der besseren Kontrolle (z.B. Vermeidung von Verschneidungen).

1. Motivation

Geschichte der inversen Kinematik

- die IK ist ein nichtlineares Inversionsproblem (numerische Lösungsverfahren)
- in der Computeranimation erstmalig in den 1980´ern eingesetzt, [M. Girard und A. A. Maciejewski, *Computational Modeling for the Computer Animation of Legged Figures*, Siggraph 1985]
- ursprünglich stammt die IK aus der Robotik, vgl. [D.E. Whitney, *The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators*, ASME Dynamic Systems, Measurement and Control, 1972]

1. Motivation

Anwendungen

Steuerung von Industrierobotern.

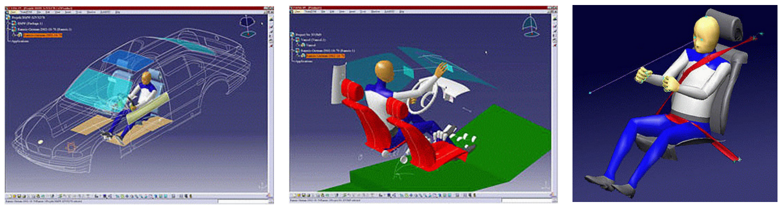
- **Kinematik** definiert die *geometrischen Bewegungsmöglichkeiten* in Abhängigkeit der Gelenkparameter (Konfigurationsraum).
- **Dynamik** liefert die *physikalischen Grundlagen* für die Simulation einer Bewegung, basierend auf Energie- und Impulserhaltung.

Quelle: http://de.wikipedia.org/wiki/Industrieroboter
KUKA Roboter GmbH, Deutschland

1. Motivation

Anwendungen

Im *CAD/CAM-Bereich* wird IK zum Beispiel für die Entwicklung von *Mensch-Maschine-Interfaces* eingesetzt (hier: Cockpit-Design für Automobilindustrie).

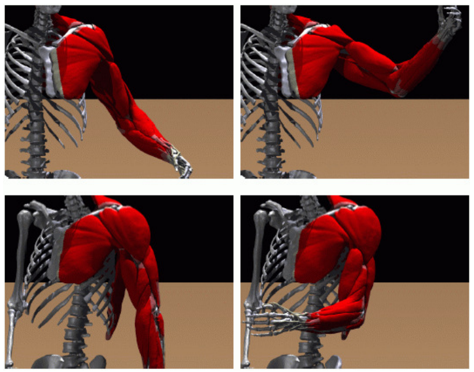


Quelle: RAMSIS-System, Human Solutions, http://www.ramsis.de

1. Motivation

Anwendungen

Virtuelle Menschmodelle



- Visualisierung medizinischer Daten
- computer-assistierte Chirurgie

Hier: virtuelles Muskelmodell des *Visible Human*.

[J. Teran et al., *Creating and Simulating Skeletal Muscle from the Visible Human Data Set*, IEEE Transactions on Visualization and Computer Graphics, Vol. 11, 2005, pp. 317-328]

1. Motivation

Anwendungen

Cartoons erfordern keine physikalisch korrekte Kinematik da die Bewegung eines „Charakters“ von dessen Denken getrieben ist.

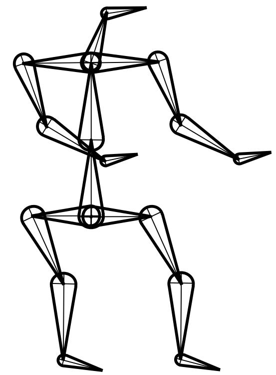
Dennoch wird die Kinematik häufig von menschlichen Darstellern gemessen (*Motion-Capture*) und auf die Proportionen der virtuellen Figur umgerechnet.

Spezialeffekte in Filmen basieren zwecks realistischer Darstellung zwar auf physikalischen Modellen, erlauben dem Animateur aber den Eingriff durch zahlreiche Parameter, z.B. externe Kräfte.

2. Kinematische Skelette

Kinematisches Skelett einer Figur

- Baum aus Gelenken (Knoten) und starren Bindungen (Kanten).
- Zykel sind nicht erlaubt
- Wurzelgelenk ist meist das Becken
- jede Bindung besitzt ein eigenes Koordinatensystem (*Frame*)

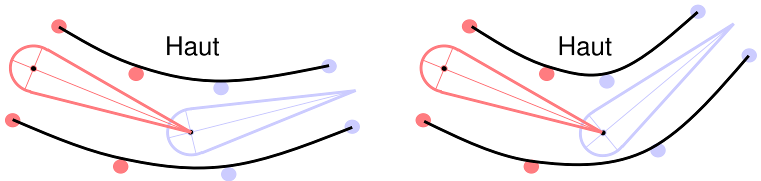


kinematisches Skelett

2. Kinematische Skelette

Äußere Schichten

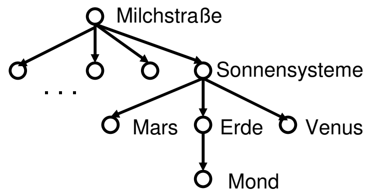
- die äußere Geometrie der Figur wird lokal an das kinematische Skelet gebunden.
- Beispiel: Kontrollpunkte einer Unterteilungsfläche
- automatische Verformung bei veränderter Gelenkstellung



2. Kinematische Skelette

Hierarchische Kinematik

- auch bei anderen Bewegungsvorgängen findet man hierarchische Strukturen
- Zykel kann man durch die Reduktion der Parameter auf die tatsächliche Anzahl von Freiheitsgraden eliminieren
- Gruppieren von Objekten (Clustering) vereinfacht die Kinematik komplexer Systeme

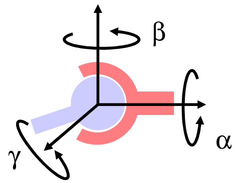


2. Kinematische Skelette

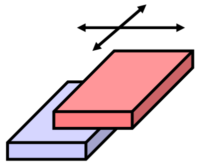
Gelenke

Gelenke mit mehreren Freiheitsgraden zerlegt man in einfache

- **Drehgelenke** und
- **prismatische** (translatierende) Gelenke.



Kugelgelenk mit Euler-Winkeln, bestehend aus drei Drehgelenken

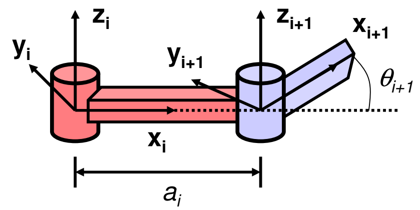


Planares Gelenk, bestehend aus zwei prismatischen Gelenken

2. Kinematische Skelette

Denavit-Hartenberg Notation (1955)

- kompakte Darstellung kinematischer Ketten (**Manipulatoren**)
- Ursprung in der Robotik
- Rotation um **z**-Achse
- Ausrichtung der Bindung nach **x**-Achse



Planarer Manipulator

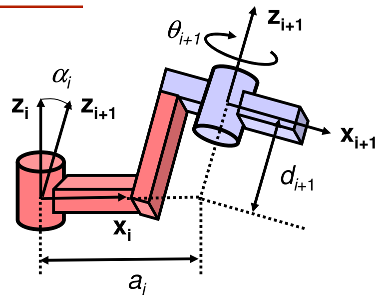
a_i: Länge der i-ten Bindung

θ_{i+1}: Rotationswinkel um $\mathbf{z_{i+1}}$

2. Kinematische Skelette

DH Notation

Im allgemeinsten Fall kommen noch zwei weitere Parameter hinzu.



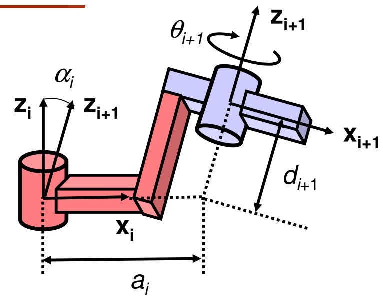
Parameter der DH Notation

Gelenkoffset in $\mathbf{z_{i+1}}$	d_{i+1}	(i+1)-te Gelenkparameter
Gelenkwinkel um $\mathbf{z_{i+1}}$	θ_{i+1}	
Bindungslänge in $\mathbf{x_i}$	a_i	i-te Bindungsparameter
Bindungstwist um $\mathbf{x_i}$	α_i	

2. Kinematische Skelette

DH Notation

Die Transformation des i-ten Frames in das i+1-te setzt sich aus zwei schraubenartigen Transformationen zusammen:



$$\mathbf{M}_\mathbf{i}^{\mathbf{i+1}} = \left(\mathbf{T_x}(a_i)\mathbf{R_x}(\alpha_i)\right)\left(\mathbf{T_z}(d_{i+1})\mathbf{R_z}(\theta_{i+1})\right)$$

2. Kinematische Skelette

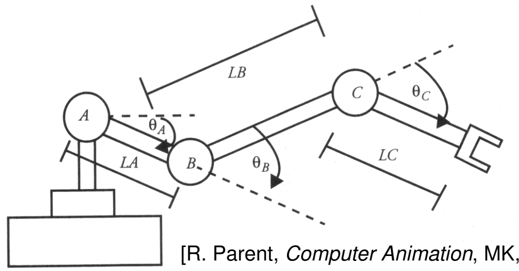
Transformation in homogenen Koordinaten

$$\mathbf{M}_i^{i+1} = \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 & 0 \\ \sin\theta_{i+1} & \cos\theta_{i+1} & 0 & 0 \\ 0 & 0 & 1 & d_{i+1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 & a_i \\ \cos\alpha_i \sin\theta_{i+1} & \cos\alpha_i \cos\theta_{i+1} & -\sin\alpha_i & -(\sin\alpha_i)d_{i+1} \\ \sin\alpha\sin\theta_{i+1} & \sin\alpha_i\cos\theta_{i+1} & \cos\alpha_i & (\cos\alpha_i)d_{i+1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Kinematische Skelette

Beispiel 1:



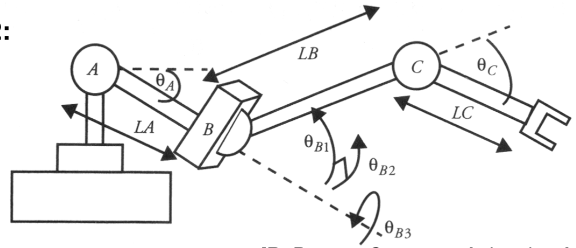
[R. Parent, *Computer Animation*, MK, 2002]

DH Parameter

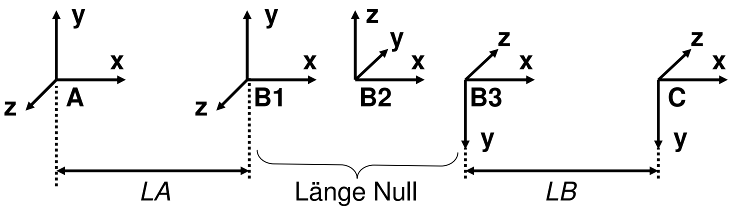
Gelenk	Offset d	Winkel θ	Länge a	Twist α
A	0	θ_A	LA	0
B	0	θ_B	LB	0
C	0	θ_C	LC	0

2. Kinematische Skelette

Beispiel 2:



[R. Parent, *Computer Animation*, MK, 2002]



2. Kinematische Skelette

Beispiel 2:

Das Kugelgelenk **B** ist durch drei Drehgelenke, die jeweils um 90° versetzt sind, realisiert. Um eine singuläre Ruhelage zu vermeiden, wird das mittlere Drehgelenk **B2** mit 90° initialisiert.

DH Parameter

Gelenk	Offset d	Winkel θ	Länge a	Twist α
A	0	θ_A	LA	0
B1	0	θ_{B1}	0	90°
B2	0	θ_{B2} + 90°	0	90°
B3	0	θ_{B3}	LB	0
C	0	θ_C	LC	0

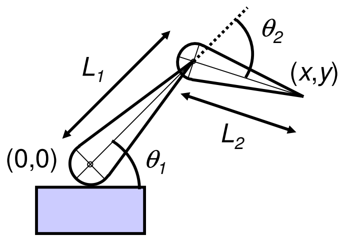
3. Inverse Kinematik

Inverse Kinematik

- Bedingungen an Endeffektor (z.B. Fuß am Boden; Hand am Lenkrad, etc.)
- Problem: Bestimmung der freien Parameter (Gelenkwinkel)

Beispiel: Analytische Berechnung

- nur in einfachen Fällen möglich
- hier: zwei Freiheitsgrade θ_1 und θ_2
- Vorgabe der Position (x, y)



3. Inverse Kinematik

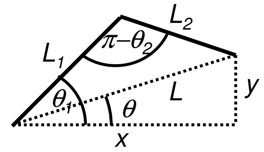
Analytisches Verfahren

- Gesamtlänge $L = \sqrt{x^2 + y^2}$.
- für $L_1 - L_2 < L < L_1 + L_2$ existieren zwei Lösungen (mit $\theta_2 > 0$ und $\theta_2 < 0$).
- θ_1 und θ_2 findet man mit dem Cosinus-Satz:

$$\cos\theta = x / L,$$

$$\cos(\theta_1 - \theta) = \frac{L_1^2 + L^2 - L_2^2}{2L_1L},$$

$$\cos(\pi - \theta_2) = \frac{L_1^2 + L_2^2 - L^2}{2L_1L_2}.$$

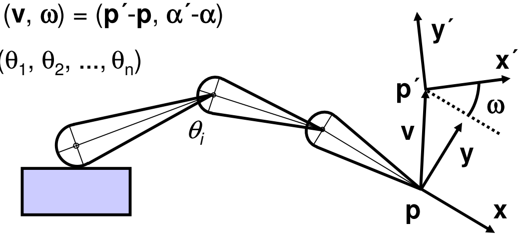


3. Inverse Kinematik

Iteratives Verfahren

In den meisten Fällen wird die IK iterativ mit folgenden Vorgaben für den Endeffektor gelöst:

- Startposition **p** und Orientierung $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$
- neue Position und ggf. Orientierung (**p´**, **α´**)
- Differenzvektor **V** = (**v**, **ω**) = (**p´**-**p**, **α´**-**α**)
- Freiheitsgrade $\Theta = (\theta_1, \theta_2, ..., \theta_n)$



3. Inverse Kinematik

Iteratives Verfahren

Die Lage des Endeffektors ist eine Funktion der Parameter Θ:

$$\begin{pmatrix} \mathbf{p} \\ \boldsymbol{\alpha} \end{pmatrix} = \mathbf{F}(\Theta), \qquad \begin{pmatrix} d\mathbf{p} \\ d\boldsymbol{\alpha} \end{pmatrix} = \frac{\partial \mathbf{F}}{\partial \Theta} d\Theta.$$

Linearisierung: Wäre **F** eine lineare Funktion, dann hätten wir

$$\mathbf{V} = \frac{\partial \mathbf{F}}{\partial \Theta} \Delta\Theta = \mathbf{J}(\Theta)\,\Delta\Theta,$$

wobei $\mathbf{J}(\Theta)$ die *Jacobi-Matrix* ist.
Für kleine Änderungen $\Delta\Theta$ ist dies korrekt.

3. Inverse Kinematik

Iteratives Verfahren

In Matrixform:

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ \omega_3 \end{pmatrix} = \begin{pmatrix} \frac{\partial p_1}{\partial \theta_1} & \frac{\partial p_1}{\partial \theta_2} & \cdots & \frac{\partial p_1}{\partial \theta_n} \\ \frac{\partial p_2}{\partial \theta_1} & \frac{\partial p_2}{\partial \theta_2} & & \frac{\partial p_2}{\partial \theta_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial \alpha_3}{\partial \theta_1} & \frac{\partial \alpha_3}{\partial \theta_2} & \cdots & \frac{\partial \alpha_3}{\partial \theta_n} \end{pmatrix} \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \vdots \\ \Delta\theta_n \end{pmatrix}$$

- im Idealfall ist n=6, **J** nicht-singulär und $\Delta\Theta = \mathbf{J}^{-1}\,\mathbf{V}$.
- wegen der Nichtlinearität von **F** muss man diese Lösung iterieren bis $\|\,\mathbf{V}\,\|$ hinreichend klein ist
- die Jacobi-Matrix muss in jedem Schritt neu berechnet werden

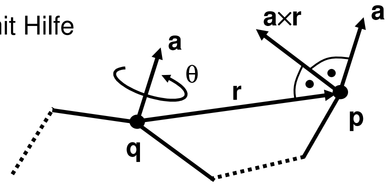
3. Inverse Kinematik

Einträge der Jacobi-Matrix

- die Größe $\partial\mathbf{p}/\partial\theta$ erhält man mit Hilfe einer Winkelgeschwindigkeit:

$$\dot{\mathbf{p}} = (\mathbf{a}\times\mathbf{r})\,\dot{\theta},$$

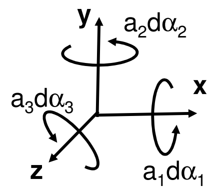
$$\frac{\partial\mathbf{p}}{\partial\theta} = \mathbf{a}\times(\mathbf{p}-\mathbf{q}).$$



- für die Ableitung nach den Winkeln gilt

$$d\alpha_i = a_i\, d\theta_i,$$

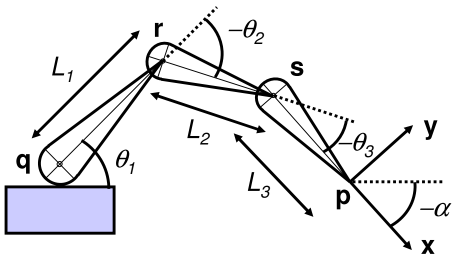
$$\frac{\partial\alpha}{\partial\theta} = \mathbf{a}\,.$$



3. Inverse Kinematik

Beispiel:

Gegeben ist ein planarer Manipulator mit drei Gelenken (gegen Uhrzeigersinn).
Mit $\mathbf{v} = \Delta\mathbf{p}$ und
$\omega = \Delta\alpha$
lautet das System:



$$\begin{pmatrix} v_1 \\ v_2 \\ \omega \end{pmatrix} = \begin{pmatrix} (\mathbf{z}\times(\mathbf{p}-\mathbf{q}))_x & (\mathbf{z}\times(\mathbf{p}-\mathbf{r}))_x & (\mathbf{z}\times(\mathbf{p}-\mathbf{s}))_x \\ (\mathbf{z}\times(\mathbf{p}-\mathbf{q}))_y & (\mathbf{z}\times(\mathbf{p}-\mathbf{r}))_y & (\mathbf{z}\times(\mathbf{p}-\mathbf{s}))_y \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{pmatrix}.$$

3. Inverse Kinematik

Überbestimmtes System

- in manchen Fällen gibt es zu wenig Freiheitsgrade
- approximative Lösung vermöge

$$\|\mathbf{J}\,\Delta\Theta - \mathbf{V}\|^2 \;\rightarrow\; \min.$$

- dies erhält man mittels der *pseudo-Inversen* $\mathbf{J}^+ := (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$:

$$\Delta\Theta = \mathbf{J}^+\,\mathbf{V}.$$

3. Inverse Kinematik

Unterbestimmtes System

- meistens liegen zu viele Freiheitsgrade vor
- zusätzliche Bedingungen, z.B. $\Delta\theta'_i$ zur „natürlichsten" Gelenk-stellung
- Optimierungsproblem

$$h := \sum_{i=1}^{n} w_i \left(\Delta\theta_i - \Delta\theta'_i\right)^2 \;\rightarrow\; \min,$$

unter der Nebenbedingung $\mathbf{J}\,\Delta\Theta = \mathbf{V}$

- Lösung mittels Lagrange-Multiplikatoren:

$$r \;:=\; h \;+\; \lambda^T(\mathbf{J}\,\Delta\Theta - \mathbf{V}) \;\rightarrow\; \min.$$

4. Optimierung

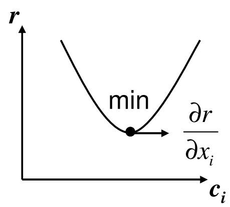
Minimierung der Fehlerquadrate (Gauß)

- überbestimmtes System $Ax=p$

$$\sum_{j=1}^{n} a_{ij} x_j = p_i \qquad (i = 1,\ldots,m > n)$$

- minimiere

$$r = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij} x_j - p_i\right)^2 \rightarrow \min$$



- (notwendige) Bedingung ist $\nabla r = 0$, d.h.

$$\frac{\partial r}{\partial x_i} \overset{!}{=} 0 \qquad i = 1,\ldots,n$$

4. Optimierung

Minimierung der Fehlerquadrate

$$\frac{\partial r}{\partial x_i} = \sum_{j=1}^{m} \frac{\partial}{\partial x_i}\left(\sum_{k=1}^{n} a_{jk} x_k - p_j\right)^2 \qquad \Big| \text{ Kettenregel}$$

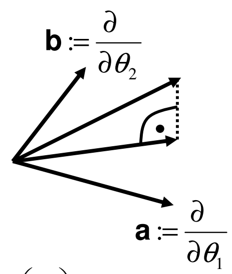
$$= \sum_{j=1}^{m} 2\left(\sum_{k=1}^{n} a_{jk} x_k - p_j\right)\frac{\partial}{\partial x_i}\sum_{k=1}^{n} a_{jk} x_k$$

$$= 2\sum_{j=1}^{m}\left(\sum_{k=1}^{n} a_{jk} x_k - p_j\right) a_{ji}$$

$$= 2\sum_{j=1}^{m}\sum_{k=1}^{n} a_{ji} a_{jk} x_k - 2\sum_{j=1}^{m} a_{ji} p_j \overset{!}{=} 0.$$

Das System lautet nun $A^T A x = A^T p$ oder $x = (A^T A)^{-1} A^T p$.

4. Optimierung



Beispiel:

- drei Bedingungen $\mathbf{V} = (v_1, v_2, v_3)^T$
- bestimme $\Delta\theta_1$ und $\Delta\theta_2$ mit

$$\left\|\Delta\theta_1\,\mathbf{a} + \Delta\theta_2\,\mathbf{b} - \mathbf{V}\right\|^2 \rightarrow \min$$

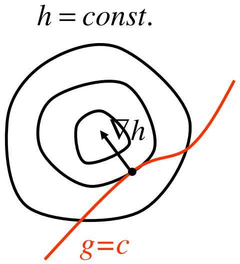
- überbestimmtes System $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix}\begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$
- approximative Lösung: $\begin{pmatrix} \mathbf{a}\cdot\mathbf{a} & \mathbf{a}\cdot\mathbf{b} \\ \mathbf{b}\cdot\mathbf{a} & \mathbf{b}\cdot\mathbf{b} \end{pmatrix}\begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{pmatrix} = \begin{pmatrix} \mathbf{a}\cdot\mathbf{v} \\ \mathbf{b}\cdot\mathbf{v} \end{pmatrix}$

4. Optimierung

Lagrange-Multiplikatoren

- Idee: Finde Extremwert einer Funktion $h(x_1,\ldots,x_n)$
  unter Nebenbedingung $g(x_1,\ldots,x_n) = c$

- Lösung:
  $g = c \quad \wedge \quad \nabla h = \lambda \nabla g$

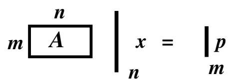


4. Optimierung

Lagrange-Multiplikatoren

Gegeben: m „harte“ Bedingungen (Endeffektor, vgl. $g=c$)
der Form

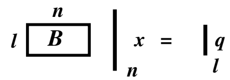
$Ax=p$



und

l „weiche“ Bedingungen („natürliche“ Haltung, $h \rightarrow$min)

$Bx=q$



mit den Unbekannten x_i $(i=1,\ldots,n)$, $n > m$.

4. Optimierung

Lagrange-Multiplikatoren

- minimiere

$$r = \sum_{k=1}^{l}\left(\sum_{j=1}^{n} b_{kj}x_j - q_k\right)^2 + \sum_{k=1}^{m}\lambda_k\left(\sum_{j=1}^{n} a_{kj}x_j - p_k\right) \rightarrow \min$$

- r hängt von x_i $(i=1,\ldots,n)$ und λ_k $(k=1,\ldots,m)$ ab.

$$\frac{\partial r}{\partial x_i} = 2\sum_{k=1}^{l}\left(\sum_{j=1}^{n} b_{kj}x_j - q_k\right)b_{ki} + \sum_{k=1}^{m}\lambda_k a_{ki} \overset{!}{=} 0$$

(weiche Bedingungen, $\nabla h = \lambda \cdot \nabla g$)

$$\frac{\partial r}{\partial \lambda_i} = \sum_{j=1}^{n} a_{ij}x_j - p_i \overset{!}{=} 0$$

(harte Bedingungen, $g=c$)

4. Optimierung

Lagrange-Multiplikatoren

in Matrixform:

$$\begin{pmatrix} 2B^TB & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} 2B^Tq \\ p \end{pmatrix}$$

Bemerkung:

Die Lösung besteht nur aus den Werten x_i.

Die Lagrange-Multiplikatoren λ_k dienen zum Auffinden der Lösung.

4. Optimierung

Algorithmus

(1) Startwerte für Θ und Translation **V**

(2) bestimme $J(\Theta)$

(3) eliminiere ggf. redundante Parameter (Singularität)

(4) löse nach $\Delta\Theta$ (Optimierung ggf. mit Nebenbedingung)

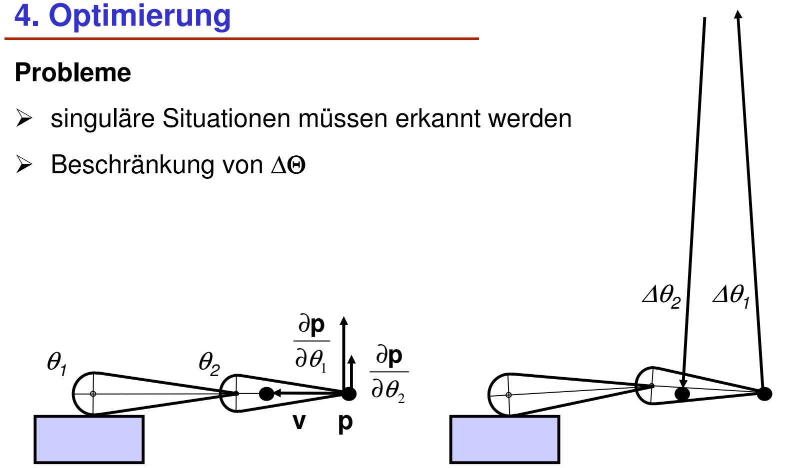
(5) aktualisiere Θ und **V**

(6) falls Lösung noch nicht erreicht, gehe nach (2)

4. Optimierung

Probleme

- singuläre Situationen müssen erkannt werden
- Beschränkung von $\Delta\Theta$



blockierter Manipulator

Umgebung der Singularität

5. Literatur

Bücher

Alan Watt und Mark Watt, *Advanced Animation and Rendering - Theory and Practice*, ACM Press, 1992.

Rick Parent, *Computer Animation - Algorithms and Techniques*, Morgan Kaufmann, 2002.

Michael Bender und Manfred Brill, *Computergraphik - Ein anwendungsorientiertes Lehrbuch*, 2. Auflage, Hanser, 2006.

Jorge Angeles, *Fundamentals of Robotic Mechanical Systems - Theory, Methods, and Algorithms*, 2. Auflage, Springer New York, 2002.

Stephan H. Crandall, Dean C. Karnopp, Eduard F. Kurtz Jr. und David C. Pridmore-Brown, *Dynamics of Mechanical and Electro-mechanical Systems*, Krieger Publishing, 1968.

5. Literatur

Kinematik

M. Girard und A.A. Maciejewski, *Computational Modeling for the Computer Animation of Legged Figures*, ACM Siggraph 1985, pp. 263-270.

D.E. Whitney, *The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators*, ASME Transactions, Dynamic Systems, Measurement and Control, Vol. 122, 1972, pp. 303-309.

Dynamik

D. Baraff, *Linear-Time Dynamics using Lagrange Multipliers*, ACM Siggraph 1996, pp. 137-146.

C. Karen Liu, Aaron Hertzmann, Zoran Popovic, *Learning Physics-based Motion Style with Inverse Optimization*, ACM Siggraph 2005, pp. 1071-1081.

5. Literatur

R. Weinstein, J. Teran, und R. Fedkiw, *Dynamic Simulation of Articulated Rigid Bodies with Contact and Collision*, IEEE TVCG 12, 365-374 (2006).

Modellbildung

J. Teran, E. Sifakis, S. Blemker, V. Ng Thow Hing, C. Lau und R. Fedkiw, *Creating and Simulating Skeletal Muscle from the Visible Human Data Set*, IEEE TVCG, Vol.11, 2005, pp. 317-328.

T. deRose M. Kass und T. Truong, *Subdivision Surfaces in Character Animation*, ACM Siggraph 1998, pp. 85-94.

6. Ausblick

Ausblick

Erweiterungen und Verbesserungen der IK sind möglich:

- **Quaternionen** statt Euler-Winkel (kein Blockieren)
- simulierte Dynamik mittels **Euler-Lagrange-Gleichungen**

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_j}\right) - \frac{\partial L}{\partial \theta_j} = \mathbf{F}_j \qquad (j = 1,\dots,n),$$

$$L = L(\dot{\theta}_1,\dots,\dot{\theta}_n,\theta_1,\dots,\theta_n,t) := E_k - E_p.$$

- unsere wissenschaftlichen Methoden sind nur ein kleiner Beitrag zur Computeranimation, denn ...

6. Ausblick

"Story is King"

- Martin Reddy, Pixar

Quelle: Geri´s Game,
Pixar Animation Studios
http://www.pixar.com

A. Quaternionen

Quaternionen

Die Darstellung von Orientierungen mit Matrizen oder Euler Winkeln ist meist kompliziert und hat gravierende Nachteile bei der Interpolation. Eine bessere Darstellung bieten die **Quaternionen** (Quaternions, W. Hamilton, 1843).

Ein Quaternion ist ein 4-Tupel [s,x,y,z] oder [s,**v**]; $\mathbf{v} = (x,y,z)^T$. Die Rotation $\mathbf{R}_\mathbf{a}(\theta)$ um einen (normierten) Vektor **a** um den Winkel θ entspricht dem Quaternion

$$\mathbf{R}_\mathbf{a}(\theta) = [\cos(\theta/2), \sin(\theta/2)\,\mathbf{a}].$$

A. Quaternionen

Bevor wir mit Quaternionen arbeiten können, müssen wir noch einige Rechengesetze, wie z.B. Addition und Multiplikation, einführen:

$$[s_1, \mathbf{v}_1] + [s_2, \mathbf{v}_2] = [s_1+s_2, \mathbf{v}_1+\mathbf{v}_2],$$

$$[s_1, \mathbf{v}_1] \cdot [s_2, \mathbf{v}_2] = [s_1 s_2 - \mathbf{v}_1 \bullet \mathbf{v}_2,\ s_1\mathbf{v}_2 + s_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2]$$

Die Multiplikation ist assoziativ, aber **nicht** kommutativ. Das neutrale Element ist [1,0,0,0], also Q · [1,0,0,0] = Q.

Das inverse Quaternion zu Q = [s,**v**] ist

$$Q^{-1} = \frac{1}{\|Q\|^2}[s, -\mathbf{v}], \qquad \|Q\| = \sqrt{s^2 + \mathbf{v} \bullet \mathbf{v}}$$

A. Quaternionen

Normierte Quaternionen Q / ||Q|| bezeichnen wir als **Einheits-quaternionen**. Ein Punkt oder Vektor **a** lässt sich ebenfalls als Quaternion darstellen:

a = [0, **a**].

Die Rotation von **a** mittels eines Quaternions Q ist nun gegeben durch

$$\mathbf{a}' = \mathbf{R}_Q\, \mathbf{a} = Q \cdot a \cdot Q^{-1}.$$

Wegen der Multiplikation mit dem inversen Quaternion ist keine Normierung erforderlich. Bei mehreren Rotationen gilt:

$$\mathbf{R}_Q\, \mathbf{R}_P\, \mathbf{a} = Q\, P\, a\, P^{-1} Q^{-1} = (Q\, P)\, a\, (Q\, P)^{-1} = \mathbf{R}_{PQ}\, \mathbf{a}.$$

A. Quaternionen

Umwandeln von Quaternionen in Matrizen

Die äquivalente Rotationsmatrix zu einem Einheitsquaternion Q = [s, x, y, z] (mit $s^2 + x^2 + y^2 + z^2 = 1$) lautet

$$\mathbf{M} = \begin{pmatrix} 1-2(y^2+z^2) & 2(xy-sz) & 2(xz+sy) \\ 2(xy+sz) & 1-2(x^2+z^2) & 2(yz-sx) \\ 2(xz-sy) & 2(yz+sx) & 1-2(x^2+y^2) \end{pmatrix}.$$

Bei der umgekehrten Konstruktion kann man ausnutzen, dass sich die Elemente auf der Diagonale zu $4s^2$-1 summieren:

A. Quaternionen

Umwandeln von Matrizen in Quaternionen

Ist **M** eine orthonormale Matrix, so erhalten wir das äquivalente Quaternion aus den Elementen der Diagonale:

$$m_{11} + m_{22} + m_{33} = 4s^2 - 1 \quad \text{liefert} \quad s = \frac{1}{2}\sqrt{m_{11} + m_{22} + m_{33} + 1}$$

und

$$m_{11} = 1 - 2(y^2 + z^2) = 1 - 2(1 - x^2 - s^2) = 2x^2 + 2s^2 - 1$$

(analog für m_{22} und m_{33}) liefert

$$x = \pm\sqrt{\frac{m_{11} + 1 - 2s^2}{2}}; \quad y = \pm\sqrt{\frac{m_{22} + 1 - 2s^2}{2}}; \quad z = \pm\sqrt{\frac{m_{33} + 1 - 2s^2}{2}}.$$

A. Quaternionen

Beispiel 1: $\mathbf{R_x}(\alpha)$ lautet als Quaternion:

$$[\cos(\alpha/2), \sin(\alpha/2)\,\mathbf{x}] \;=\; [s, (t,0,0)^T], \qquad s = \cos(\alpha/2),\quad t = \sin(\alpha/2).$$

Beim Umwandeln in eine Matrix ergibt sich:

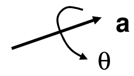
$$\mathbf{M} = \begin{pmatrix} 1-2(y^2+z^2) & 2(xy-sz) & 2(xz+sy) \\ 2(xy+sz) & 1-2(x^2+z^2) & 2(yz-sx) \\ 2(xz-sy) & 2(yz+sx) & 1-2(x^2+y^2) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1-2t^2 & -2st \\ 0 & 2st & 1-2t^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

mit $1-2\sin^2(\alpha/2) = \cos(\alpha)$, $\quad 2\sin(\alpha/2)\cos(\alpha/2) = \sin(\alpha)$.

A. Quaternionen

Beispiel 2:



Umkehren der Rotationsachse und Negieren des Winkels liefert dasselbe Quaternion:

$\mathbf{R_a}(\theta) = [\cos(\theta/2), \sin(\theta/2)\,\mathbf{a}] = [\cos(-\theta/2), \sin(-\theta/2)\,(-\mathbf{a})] = \mathbf{R_{-a}}(-\theta)$.

Addiert man jedoch 2π zum Rotationswinkel θ, so erhält man ein entgegengesetztes Quaternion für dieselbe Rotation:

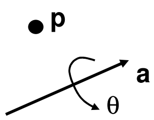
$\mathbf{R_a}(\theta + 2\pi) = [\cos(\theta/2 + \pi), \sin(\theta/2 + \pi)\,\mathbf{a}]$

$= [-\cos(\theta/2), -\sin(\theta/2)\,\mathbf{a}]$

Die Quaternionen $[s,\mathbf{v}]$ und $[-s, -\mathbf{v}]$ haben also die selbe Wirkung. Eine Rotation um 360° liefert [-1, 0,0,0]; um 720° liefert [1, 0,0,0].

A. Quaternionen

Beispiel 3:



Wir zeigen nun, dass ein Punkt **p** mit Hilfe des Quaternions

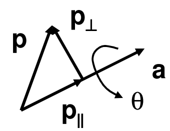
$Q = [\cos(\theta/2), \sin(\theta/2)\,\mathbf{a}]$

tatsächlich um die Achse **a** ($||\mathbf{a}||=1$) rotiert wird:

$$\begin{aligned} QpQ^{-1} &= [s,\mathbf{v}]\cdot[0,\mathbf{p}]\cdot[s,-\mathbf{v}] \\ &= [0, (s^2 - \mathbf{v}\bullet\mathbf{v})\mathbf{p} + 2(\mathbf{v}\bullet\mathbf{p})\mathbf{v} + 2s\mathbf{v}\times\mathbf{p}] \\ &= [0, (\cos^2(\tfrac{\theta}{2}) - \sin^2(\tfrac{\theta}{2}))\mathbf{p} + 2\sin^2(\tfrac{\theta}{2})(\mathbf{a}\bullet\mathbf{p})\mathbf{a} \\ &\qquad + 2\cos(\tfrac{\theta}{2})\sin(\tfrac{\theta}{2})\,\mathbf{a}\times\mathbf{p}] \\ &= [0, (\cos\theta)\mathbf{p} + (1-\cos\theta)(\mathbf{a}\bullet\mathbf{p})\mathbf{a} + (\sin\theta)\mathbf{a}\times\mathbf{p}] \end{aligned}$$

A. Quaternionen

Beispiel 3:



Das selbe Resultat erhält man auch, indem man **p** in zwei Komponenten $\mathbf{p}_{||}$, $\mathbf{p}_{\perp}$ zerlegt und die Rotation auf $\mathbf{p}_{\perp}$ anwendet ($\mathbf{a}\times\mathbf{p}$ ist identisch mit $\mathbf{p}_{\perp}$ um 90° rotiert):

$$\begin{aligned} R\mathbf{p} &= \mathbf{p}_{||} + R\mathbf{p}_{\perp} \\ &= \mathbf{p}_{||} + (\cos\theta)\mathbf{p}_{\perp} + (\sin\theta)\mathbf{a}\times\mathbf{p} \\ &= (\mathbf{a}\bullet\mathbf{p})\mathbf{a} + (\cos\theta)(\mathbf{p} - (\mathbf{a}\bullet\mathbf{p})\mathbf{a}) + (\sin\theta)\mathbf{a}\times\mathbf{p} \\ &= (\cos\theta)\mathbf{p} + (1-\cos\theta)(\mathbf{a}\bullet\mathbf{p})\mathbf{a} + (\sin\theta)\mathbf{a}\times\mathbf{p} \end{aligned}$$

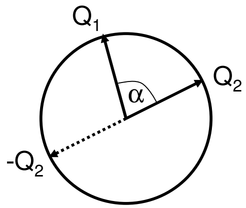
Dies ist mit der vorherigen Rechnung identisch. □

A. Quaternionen

Sphärische lineare Interpolation

Jedes Einheitsquaternion Q = [s, x, y, z] entspricht genau einem Punkt auf der vierdimensionalen Einheitssphäre.

Lineare Interpolation zweier Quaternionen Q_1 und Q_2 ist auf dieser 4D Einheitssphäre durchzuführen.



Der Winkel zwischen Q_1 und Q_2 kann mit dem Skalarprodukt (im $I\!R^4$) berechnet werden:

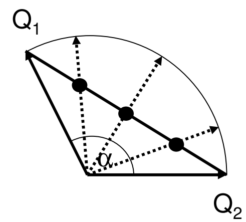
$$\cos\alpha = Q_1 \bullet Q_2 = s_1 s_2 + x_1 x_2 + y_1 y_2 + z_1 z_2$$

Ist dieses Skalarprodukt negativ, empfiehlt es sich, Q_2 durch $-Q_2$ zu ersetzen, um Umwege auf der Einheitssphäre zu vermeiden.

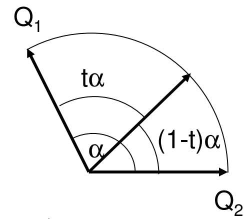
A. Quaternionen

Sphärische lineare Interpolation

Es genügt **nicht**, die Quaternionen Q_1 und Q_2 linear zu interpolieren und das Ergebnis zu normieren, da dies zu einer ungleichmässigen Parametrisierung führt.



Statt dessen ist der Winkel linear zu interpolieren:



$$slerp(Q_1, Q_2, t) = \frac{\sin((1-t)\alpha)}{\sin\alpha} Q_1 + \frac{\sin(t\alpha)}{\sin\alpha} Q_2$$

(slerp = spherical linear interpolation; liefert Geodäten).

A. Quaternionen

Quaternionen-Kurven

Die Sphärische lineare Interpolation kann man nun direkt in den de Casteljau- oder de Boor-Algorithmus einsetzen, um mit Hilfe von Kontroll-Quaternionen Q_i eine glatte Orientierungskurve Q(t) zu generieren. [K. Shoemake, 85]

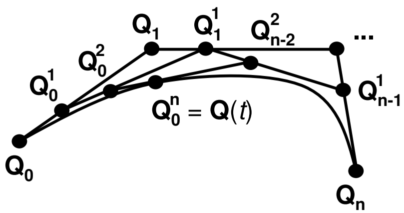
Beispiel: de Casteljau-Algorithmus

$$Q_i^0 = Q_i \quad (i = 0, \cdots, n)$$

$$Q_i^j = slerp(Q_i^{j-1}, Q_{i+1}^{j-1}, t)$$

$$(i = 0, \cdots, n-j;\ j = 1, \cdots, n)$$

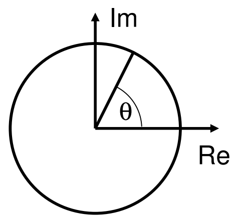
$$Q(t) = Q_0^n$$



A. Quaternionen

Exponential- und Logarithmusfunktion

in Analogie zu den komplexen Zahlen definieren wir Exponential- und Logarithmusfunktionen für Einheitsquaternionen:



komplexe Zahlen:	**Quaternionen:**
$s + ix := [s, x],\ i^2 = -1$	$s + ix + jy + kz := [s, x,y,z]$, $i^2 = j^2 = k^2 = -1,\ ij = k,\ ji = -k$.
$e^{i\theta} = \exp[0, \theta] = [\cos\theta, \sin\theta]$	$\exp[0, \theta\,\mathbf{a}] = [\cos\theta, \sin\theta\,\mathbf{a}]$
$\log[\cos\theta, \sin\theta] = [0, \theta]$	$\log[\cos\theta, \sin\theta\,\mathbf{a}] = [0, \theta\,\mathbf{a}]$.

A. Quaternionen

Exponential- und Logarithmusfunktion

SLERP lässt sich ebenfalls mit Hilfe der Exponentialfunktionen beschreiben:

$$\begin{aligned} slerp(Q_1, Q_2, t) &= Q_1 \exp\big(t \log(Q_1^{-1} Q_2)\big) \\ &= Q_1 \big(Q_1^{-1} Q_2\big)^t . \end{aligned}$$

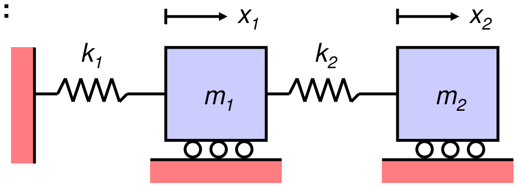
Achtung: im allgemeinen ist

exp(P) exp(Q) ≠ exp(P+Q),

denn sonst müsste ja gelten: exp(P) exp(Q) = exp(Q) exp(P), was aber wegen der fehlenden Kommutativität **nicht der Fall** ist.

B. Mechanische Systeme

Beispiel 1:



Das System bestehend aus zwei Massen m_1, m_2 und zwei Federn k_1, k_2 befindet sich für $x_1 = x_2 = 0$ in Ruhelage. Es gilt

$$\begin{aligned} E_k^* &= \tfrac{1}{2} m_1 \dot{x}_1^2 + \tfrac{1}{2} m_2 \dot{x}_2^2 , \\ E_p &= \tfrac{1}{2} k_1 x_1^2 + \tfrac{1}{2} k_2 (x_2 - x_1)^2 , \end{aligned}$$

B. Mechanische Systeme

Beispiel 1:

$$L = \tfrac{1}{2} m_1 \dot{x}_1^2 + \tfrac{1}{2} m_2 \dot{x}_2^2 - \tfrac{1}{2} k_1 x_1^2 - \tfrac{1}{2} k_2 (x_2 - x_1)^2 ,$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_1}\right) = \frac{d}{dt}(m_1 \dot{x}_1) = m_1 \ddot{x}_1 , \qquad \frac{\partial L}{\partial x_1} = -k_1 x_1 + k_2 (x_2 - x_1),$$

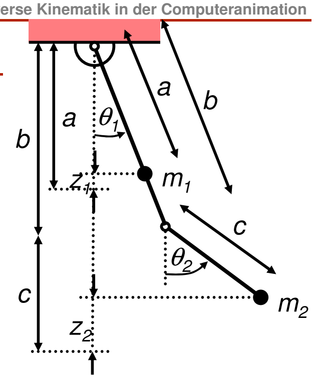
$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_2}\right) = \frac{d}{dt}(m_2 \dot{x}_2) = m_2 \ddot{x}_2 , \qquad \frac{\partial L}{\partial x_2} = -k_2 (x_2 - x_1).$$

Liegen keine äußeren Kräfte vor, so ist die Dynamik des Systems durch folgende zwei Differentialgleichungen beschrieben:

$$m_1 \ddot{x}_1 + k_1 x_1 - k_2 (x_2 - x_1) = 0 \quad \text{und} \quad m_2 \ddot{x}_2 + k_2 (x_2 - x_1) = 0.$$

B. Mechanische Systeme

Beispiel 2:



Gegeben ist ein doppeltes Pendel bestehend aus zwei Massen m_1, m_2 und zwei Freiheitsgraden θ_1, θ_2.
Für Höhe und Geschwindigkeit gilt:

$$\begin{aligned} z_1 &= a\,(1 - \cos\theta_1), \\ z_2 &= b\,(1 - \cos\theta_1) + c\,(1 - \cos\theta_2), \\ v_1^2 &= a^2 \dot{\theta}_1^2 , \\ v_2^2 &= b^2 \dot{\theta}_1^2 + c^2 \dot{\theta}_2^2 + 2bc\,\dot{\theta}_1 \dot{\theta}_2 \cos(\theta_2 - \theta_1). \end{aligned}$$

B. Mechanische Systeme

Beispiel 2:

Für die kinetische Koenergie und die potentielle Energie ergibt sich unter Berücksichtigung der Erdbeschleunigung g:

$$E_k^* = \tfrac{1}{2}(m_1 a^2 + m_2 b^2)\dot{\theta}_1^2 + m_2 bc\,\dot{\theta}_1\dot{\theta}_2\cos(\theta_2 - \theta_1) + \tfrac{1}{2} m_2 c^2 \dot{\theta}_2^2,$$

$$E_p = (m_1 a + m_2 b)\,g\,(1 - \cos\theta_1) + m_2 c\, g\,(1 - \cos\theta_2).$$

Durch Einsetzen in die Lagrange-Gleichungen

$$\frac{d}{dt}\left(\frac{\partial E_k^*}{\partial \dot{\theta}_j}\right) - \frac{\partial E_k^*}{\partial \theta_j} + \frac{\partial E_p}{\partial \theta_j} = 0 \qquad (j = 1, 2).$$

erhalten wir zwei Differentialgleichungen $(\ddot{\theta}_1, \ddot{\theta}_2) = f(\dot{\theta}_1, \dot{\theta}_2, \theta_1, \theta_2, t)$.

B. Mechanische Systeme

Beispiel 2:

$$\frac{d}{dt}\Big((m_1 a^2 + m_2 b^2)\dot{\theta}_1 + m_2 bc\,\dot{\theta}_2\cos(\theta_2 - \theta_1)\Big)$$
$$- m_2 bc\,\dot{\theta}_1\dot{\theta}_2\sin(\theta_2 - \theta_1) + (m_1 a + m_2 b)\,g\sin\theta_1 = 0,$$

$$\frac{d}{dt}\Big(m_2 bc\,\dot{\theta}_1\cos(\theta_2 - \theta_1) + m_2 c^2 \dot{\theta}_2\Big)$$
$$+ m_2 bc\,\dot{\theta}_1\dot{\theta}_2\sin(\theta_2 - \theta_1) + m_2 c\, g\sin\theta_2 = 0.$$