Computer Graphics II
8: Skeletal Animation

Computer Graphics and
Multimedia Systems Group

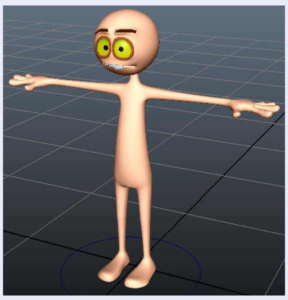
University of Siegen

Structure of Chapter

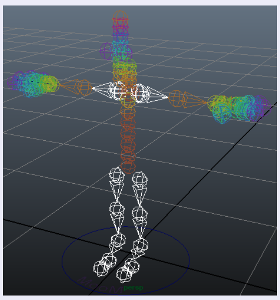
- *Subsection 1: Skeletons in 2D*
  - *Subsection 1: Hierarichal representation of skeletons*
  - *Subsection 2: Forward kinematics, i.e. from the state-parameters to the pose*
  - *Subsection 3: Inverse kinematics, i.e. from the pose to the state-parameters*
  - *Subsection 4: Numerical solutions of the inverse kinematics problem*
- *Subsection 2: Skeletons in 3D*
- *Subsection 3: Softbody animation, i.e. attaching skin to the skeleton*

Objective

Animation of characters by animating their skeletons.



Character "Moom"

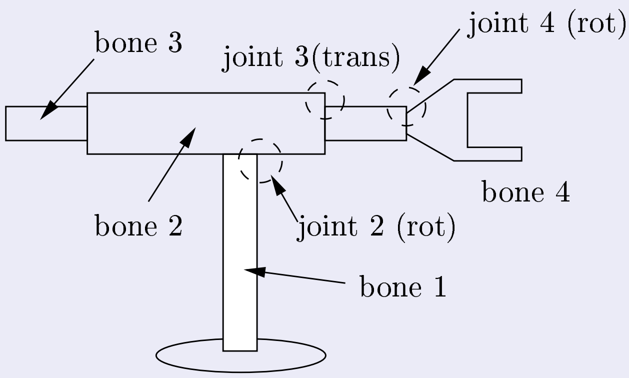


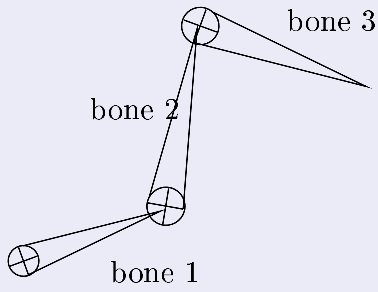
Skeleton of "Moom"

Notation

Skeletons: *Tree-like hierarchy of rigid members (bones), connected to groove joints or rotation joints.*

Effector: *Free ends of the skeleton (leaf nodes)*





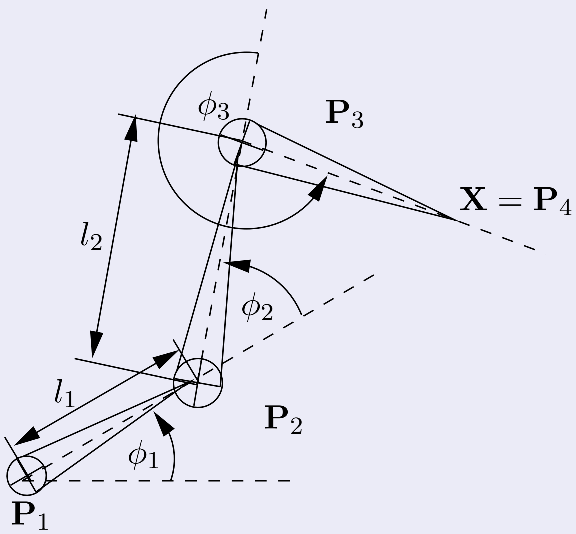
Examples: Robotics (rotation and groove joints) and animation (only rotation joints).

Definition (Description of Skeletons ($2D$))

Initial Position:
1. n-membered skeleton with fixed lengths $l_1, \ldots, l_n$
2. Pivot Point(dt: Drehpunkt) $\mathbf{P}_i$: Start point of member i
3. End-effector $\mathbf{X} = \mathbf{P}_{n+1}$: Ending point of the skeleton

Degrees of Freedom: $n + 2$ units:
- Pivot point $\mathbf{P}_1$ of the first member
- Angle ϕ_i to parent ($n > 1$) resp. orientation of first memeber ($n = 1$)

State Vector: Describes all free angles $\vec{\phi} = (\phi_1, \ldots, \phi_n)$



$2D$-skeleton without branching

Property

Successive calculation of $\mathbf{P}_{i+1}$ *for given* $\mathbf{P}_1$*:*

$$i = 1: \quad \mathbf{P}_2 = T_{1\to 2}(\mathbf{P}_1) = \mathbf{P}_1 + l_1 \begin{pmatrix} \cos\phi_1 \\ \sin\phi_1 \end{pmatrix}$$

$$i = 2: \quad \mathbf{P}_3 = T_{2\to 3}(\mathbf{P}_2) = \mathbf{P}_2 + l_2 \begin{pmatrix} \cos(\phi_1+\phi_2) \\ \sin(\phi_1+\phi_2) \end{pmatrix}$$

$$\textit{In general:} \quad \mathbf{P}_{i+1} = T_{i\to i+1}(\mathbf{P}_i) = \mathbf{P}_i + l_i \begin{pmatrix} \cos(\phi_1+\ldots+\phi_i) \\ \sin(\phi_1+\ldots+\phi_i) \end{pmatrix}$$

Definition (Types of Kinematics)

Forward Kinematics (FK): Variation of the degrees of freedom for movement of the end-effector $\mathbf{X}$

$$\mathbf{X} = f(\vec{\phi})$$

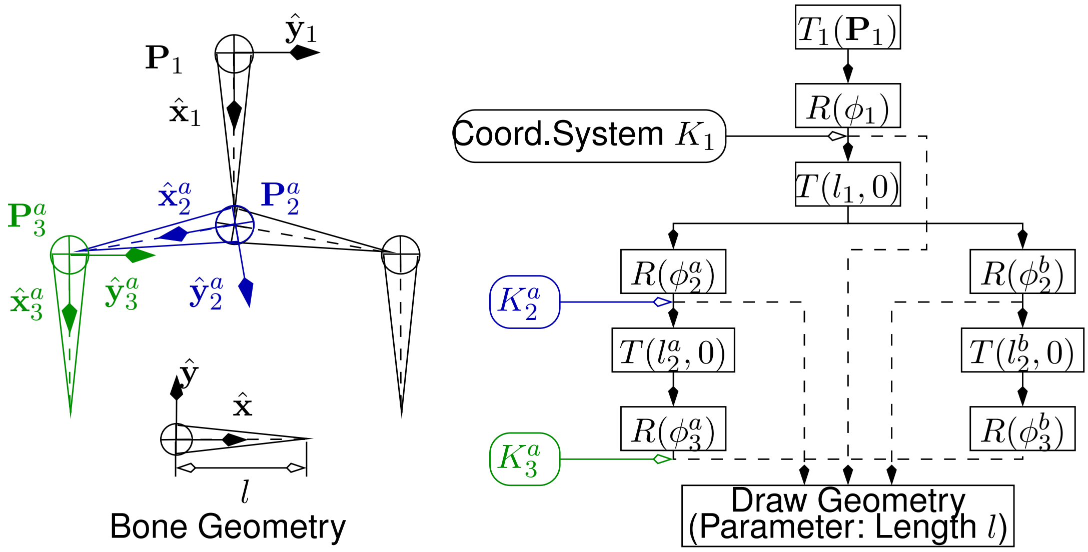
Inverse Kinematik (IK): Direct movement of the end-effector and determination of the degrees of freedom:

$$\vec{\phi} = f^{-1}(\mathbf{X})$$

Aim: Correct alignment of objects with regard to a member

Observation: The skeleton describes a *hierarchical coordinate system* $K_i = \{\mathbf{P}_i, \hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i\}$

Transformation $T_{i\to i+1} = T(l_i, 0) \cdot R(\phi_{i+1})$



Example

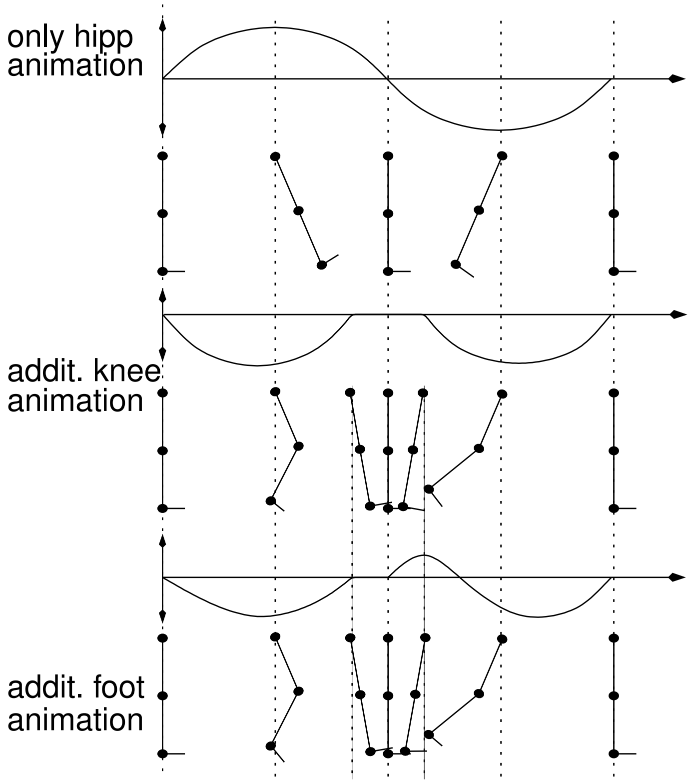
Hierarchical Models:
Hierarchical top-down animation

Example of a Leg:

1. Hip-animation
2. Knee-animation
3. Foot-animation

Problems with the above example

- Prevention of horizontal *sliding*
- Prevention of vertikal *penetration*



Remark (Essential Aspects of Forward Kinematics)

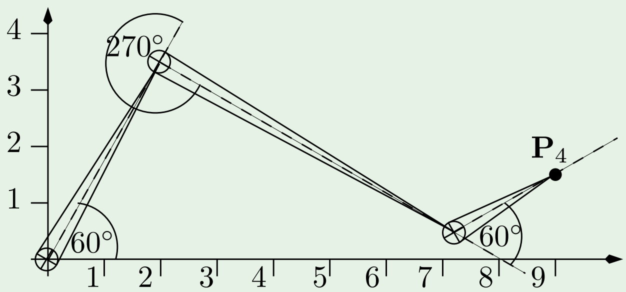
- *Rotations are inherited via a hierarchy*
- *Animation results from changing of angles over time* $\phi_i(t)$
- *Low-level of animation via angles becomes confusing for larger skeletons*

Example (Three Bones with no Braching)

$\phi_1 = 60^\circ,\ \phi_2 = 270^\circ,\ \phi_3 = 60^\circ$

$l_1 = 4,\ l_2 = 6,\ l_3 = 2,\ \mathbf{P}_1 = (0,0)$

$$\mathbf{P}_2 = P_1 + l_1 \begin{pmatrix}\cos\phi_1\\ \sin\phi_1\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} + 4\cdot\begin{pmatrix}\cos 60^\circ\\ \sin 60^\circ\end{pmatrix} = \begin{pmatrix}2\\ 2\sqrt{3}\end{pmatrix}$$

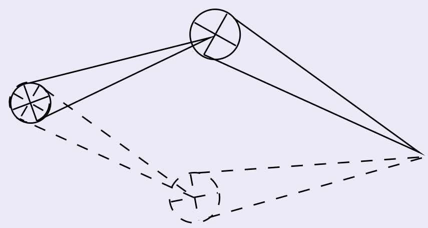


$$\mathbf{P}_3 = P_2 + l_2 \begin{pmatrix}\cos(\phi_1+\phi_2)\\ \sin(\phi_1+\phi_2)\end{pmatrix} = \begin{pmatrix}2\\ 2\sqrt{3}\end{pmatrix} + 6\cdot\begin{pmatrix}\cos 330^\circ\\ \sin 330^\circ\end{pmatrix} = \begin{pmatrix}2+3\sqrt{3}\\ 2\sqrt{3}-3\end{pmatrix}$$

$$\mathbf{P}_4 = P_3 + l_3 \begin{pmatrix}\cos(\phi_1+\phi_2+\phi_3)\\ \sin(\phi_1+\phi_2+\phi_3)\end{pmatrix} = \begin{pmatrix}2+3\sqrt{3}\\ 2\sqrt{3}-3\end{pmatrix} + 2\cdot\begin{pmatrix}\cos 30^\circ\\ \sin 30^\circ\end{pmatrix} = \begin{pmatrix}2+4\sqrt{3}\\ 2\sqrt{3}-2\end{pmatrix}$$

Problem (Inverse Kinematics)

1. *f^{-1} can, in general, not be determined analytically $\Rightarrow$ numerical solution of inverse kinematics*
2. *Inverse kinematics is generally not uniquely solvable:*



3. *Automatism of inverse kinematics does not easily allow for typical movement patterns, e.g. limping*(dt: hinken)

Notation

Under-determined System: *“Number of degrees of freedom” > “number of conditions”*

Accessible Workspace: *Space of positions obtainable by the end effector*

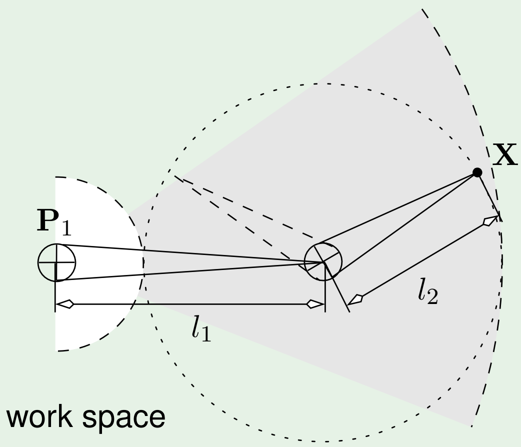
Example (Analytical solution of inverse kinematics)

Given: Skelton with two bones

- positions $\mathbf{P}_1 = \binom{x_1}{y_1}$
- end effector $\mathbf{X}\binom{x}{y}$
- lengths l_1, l_2

Obtainable workspace:

$\{\mathbf{Q} : \|\mathbf{Q} - \mathbf{P}_1\| \in [|l_1 - l_2|, l_1 + l_2]\}$



Analytical solution of inverse kinematics:

$$\phi_2 = \cos^{-1}\left(\frac{(x-x_1)^2 + (y-y_1)^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

$$\phi_1 = \tan^{-1}\left(\frac{(y-y_1)(l_1 + l_2\cos\phi_2) - (x-x_1) l_2 \sin\phi_2}{(x-x_1)(l_1 + l_2\cos\phi_2) + (y-y_1) l_2 \sin\phi_2}\right)$$

With $\Delta x = x - x_1,\ \Delta y = y - y_1,\ s_i = \sin\phi_i,\ c_i = \cos\phi_i,\ s_{1+2} = \sin(\phi_1+\phi_2),\ c_{1+2} = \cos(\phi_1+\phi_2)$, it follows from the forward calculation:

$$\begin{aligned}
\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} &= \begin{pmatrix}l_1c_1 + l_2c_{1+2}\\ l_1s_1 + l_2s_{1+2}\end{pmatrix}\\
\Rightarrow \Delta x^2 + \Delta y^2 &= l_1^2\underbrace{(c_1^2+s_1^2)}_{=1} + l_2^2\underbrace{(c_{1+2}^2+s_{1+2}^2)}_{=1} + 2l_1l_2(\underbrace{c_1c_{1+2}+s_1s_{1+2}}_{=\cos(\phi_1+\phi_2-\phi_1)=\cos\phi_2})\\
&= l_1^2 + l_2^2 + 2l_1l_2\cos\phi_2 \Rightarrow \phi_2 = \arccos\left(\frac{\Delta x^2+\Delta y^2-l_1^2-l_2^2}{2l_1l_2}\right)
\end{aligned}$$

and for ϕ_1 with $t_i = \tan(\phi_i)$:

$$\begin{aligned}
\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} &= \begin{pmatrix}l_1c_1 + l_2(c_1c_2 - s_1s_2)\\ l_1s_1 + l_2(s_1c_2 + c_1s_2)\end{pmatrix}\\
\Rightarrow \Delta x\,(l_1s_1 + l_2(s_1c_2 + c_1s_2)) &= \Delta y\,(l_1c_1 + l_2(c_1c_2 - s_1s_2)) \quad |:\cos(\phi_1)\\
\Rightarrow \Delta x\,(l_1t_1 + l_2(t_1c_2 + s_2)) &= \Delta y\,(l_1 + l_2(c_2 - t_1s_2))\\
\Leftrightarrow t_1(\Delta x(l_1+l_2c_2) + \Delta y l_2s_2) &= (\Delta y(l_1+l_2c_2) - \Delta x l_2s_2)
\end{aligned}$$

Example (Inverse Kinematics for the Two-Bone Skeleton)

Given: $\mathbf{P}_1 = (0,0),\ \mathbf{X} = (10,0),\ l_1 = l_2 = 10$

Calculation of ϕ_2:

$$\phi_2 = \cos^{-1}\left(\frac{(x-x_1)^2+(y-y_1)^2-l_1^2-l_2^2}{2l_1l_2}\right) = \cos^{-1}\left(\frac{-10^2}{2\cdot 10^2}\right) \in \{-120°, 120°\}$$

Calculation of ϕ_1:

Gen.: $\phi_1 = \tan^{-1}\left(\frac{(y-y_1)(l_1+l_2\cos\phi_2) - (x-x_1)l_2\sin\phi_2}{(x-x_1)(l_1+l_2\cos\phi_2) + (y-y_1)l_2\sin\phi_2}\right)$

For $\phi_2 = 120°$:

$\phi_1 = \tan^{-1}\left(\frac{0-10\cdot 10\sin(120°)}{10(10+10\cos(120°))+0}\right) = \tan^{-1}(-\sqrt{3}) \Rightarrow \phi_1 \in \{-60°, 120°\}$

For $\phi_2 = -120°$:

$\phi_1 = \tan^{-1}\left(\frac{0-10\cdot 10\sin(-120°)}{10(10+10\cos(-120°))+0}\right) = \tan^{-1}(\sqrt{3}) \Rightarrow \phi_1 \in \{-120°, 60°\}$

Approach

Problem:
- *In general the IK-solution cannot be determined analytically ($\rightarrow$ numerical solution)*
- *there is either none, one or an infinite number of solutions*

Given: $\vec{\phi}^{old}, \mathbf{X}^{old}$ *with* $f(\vec{\phi}^{old}) = \mathbf{X}^{old}$ *and the new end effector* $\mathbf{X}^{new}$

Wanted: $\vec{\phi}^{new}$ *with* $f(\vec{\phi}^{new}) = \mathbf{X}^{new}$*, i.e.* $g(\vec{\phi}) := f(\vec{\phi}) - \mathbf{X}^{new} = 0$

Reminder: *One dimensional Newton's method*

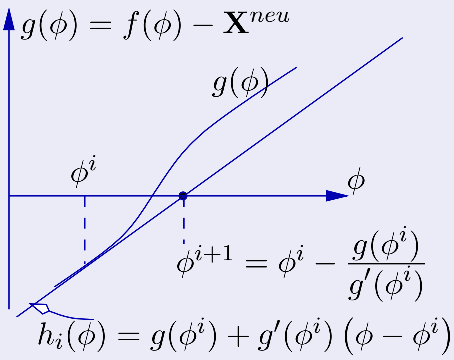
Initial: $\phi^0 = \phi^{old}$

Iteration: ϕ^{i+1} = *Root of the tangent at* g *in* ϕ^i

Tangent: $h_i(\phi) = g(\phi^i) + g'(\phi^i)(\phi - \phi^i)$

Root: $\phi^{i+1} = \phi^i - \frac{g(\phi^i)}{g'(\phi^i)} = \phi^i - \frac{f(\phi^i) - \mathbf{X}^{neu}}{f'(\phi^i)}$

Derivative: $f'(\phi^i) \approx \dfrac{f(\phi^i + \Delta) - f(\phi^i)}{\Delta}$



Objective

Is an iterative problem in animation:

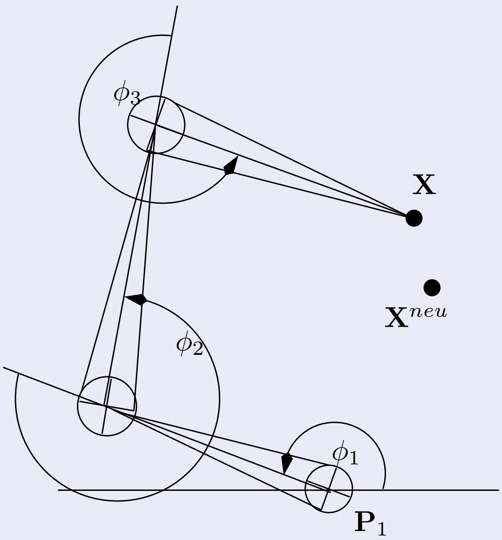
Known: $\vec{\phi}^{old}, \mathbf{X}^{old} = f(\vec{\phi}^{old}), \mathbf{X}^{new}$*, as well as a fixed starting point* $\mathbf{P}_1$

Wanted: $\vec{\phi}^{new}$ *with* $\mathbf{X}^{new} = f(\vec{\phi}^{new})$

$$\Leftrightarrow \quad g(\vec{\phi}^{new}) := f(\vec{\phi}^{new}) - \mathbf{X}^{new} = 0 \text{ with}$$

$$\left\|\vec{\phi}^{new} - \vec{\phi}^{old}\right\| = \min\{\left\|\vec{\phi} - \vec{\phi}^{old}\right\| \mid g(\vec{\phi}) = 0\}$$

Note: f *and* g *are functions with* n *variables and* m $(m = 2, 3)$ *components, in most cases* $n \geq m$



$$f : \mathbb{R}^n \mapsto \mathbb{R}^m, \quad f(\vec{\phi}) = f(\phi_1, \ldots, \phi_n) = \begin{pmatrix} f_1(\phi_1, \ldots, \phi_n) \\ \vdots \\ f_m(\phi_1, \ldots, \phi_n) \end{pmatrix}$$

Approach

Proceedure *analog to the one-dimensional case*

Adjustments: $\vec{\phi}, \mathbf{X}$ *is a multi-dimensional vektor i.e. point*

FK-Function: $f : \mathbb{R}^n \mapsto \mathbb{R}^n$

Deduction: $D_f(\vec{\phi}) = \begin{pmatrix} \frac{\partial f_1}{\partial \phi_1} & \cdots & \frac{\partial f_1}{\partial \phi_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \phi_1} & \cdots & \frac{\partial f_n}{\partial \phi_n} \end{pmatrix} \in \mathbb{R}^{n \times n}$ *(Jacobian) replaces* f'

Newton-Iteration: $\vec{\phi}^{i+1}$ *is root of the linear function at* $g(\phi)$ *in* $\vec{\phi}^i$

Lin. funct.: $h_i(\vec{\phi}) = g(\vec{\phi}^i) + D_g(\vec{\phi}^i)(\vec{\phi} - \vec{\phi}^i), \quad D_g = D_f$

Root: $\vec{\phi}^{i+1} = \vec{\phi}^i - \left(D_g(\vec{\phi}^i)\right)^{-1} g(\vec{\phi}^i) = \vec{\phi}^i - \left(D_f(\vec{\phi}^i)\right)^{-1} \left(f(\vec{\phi}^i) - \mathbf{X}^{new}\right)$

if $\left(D_f(\vec{\phi}^i)\right)^{-1}$ *exists*

Proposition (Pseudoinverse)

Given: *Underdetermined system* $A\vec{\mathbf{x}} = \vec{\mathbf{b}}$ *with* $A \in \mathbb{R}^{m \times n}, m > n, \vec{\mathbf{b}} \in \mathbb{R}^n$

Wanted: $\vec{\mathbf{x}}^+ \in \mathbb{R}^n$ *with* $A\vec{\mathbf{x}}^+ = \vec{\mathbf{b}}$ *and* $\|\vec{\mathbf{x}}^+\|$ *minimal among all solutions.*

Solution: $\vec{\mathbf{x}}^+ = A^+\vec{\mathbf{b}}$ *with* $A^+ = A^T(AA^T)^{-1}$ *(Pseudoinverse)*

Since: 1. $A\vec{\mathbf{x}}^+ = AA^T(AA^T)^{-1}\vec{\mathbf{b}} = \vec{\mathbf{b}}$

2. *Let* $\vec{\mathbf{x}}'$ *be another solution, then* $(\vec{\mathbf{x}}^+ - \vec{\mathbf{x}}') \perp \vec{\mathbf{x}}^+$ *holds true, since*

$$\begin{aligned}(\vec{\mathbf{x}}^+)^T(\vec{\mathbf{x}}^+ - \vec{\mathbf{x}}') &= (A^T(AA^T)^{-1}\vec{\mathbf{b}})^T \left(A^T(AA^T)^{-1}\vec{\mathbf{b}} - \vec{\mathbf{x}}'\right) \\ &= \vec{\mathbf{b}}^T(AA^T)^{-T}A\left(A^T(AA^T)^{-1}\vec{\mathbf{b}} - \vec{\mathbf{x}}'\right) \\ &= \vec{\mathbf{b}}^T(AA^T)^{-T}\vec{\mathbf{b}} - \vec{\mathbf{b}}^T(AA^T)^{-T} = 0 \text{ since } A\vec{\mathbf{x}}' = \vec{\mathbf{b}}\end{aligned}$$

Using Pythagoras we obtain:

$$\|\vec{\mathbf{x}}'\|^2 = \|(\vec{\mathbf{x}}' - \vec{\mathbf{x}}^+) + \vec{\mathbf{x}}^+\|^2 = \|(\vec{\mathbf{x}}' - \vec{\mathbf{x}}^+)\|^2 + \|\vec{\mathbf{x}}^+\|^2 \geq \|\vec{\mathbf{x}}^+\|^2$$

Multi-Dimensional Newton's Method (case $n > m$)

Approach

Problem: *Jacobian matrix no longer quadratic, i.e.*

$$g(\phi^i) + D_g(\vec{\phi}^i)(\vec{\phi} - \vec{\phi}^i) = 0 \Leftrightarrow D_g(\vec{\phi}^i)(\vec{\phi} - \vec{\phi}^i) = -g(\phi^i) \qquad (1)$$

has $0, 1$ or ∞ many solutions.

Pseudosinverse: *If D_f has full rank, then the* Pseudo-Inverse *can be determined:*

$$(D_f)^+ = D_f^T \cdot (D_f D_f^T)^{-1}, \qquad D_f D_f^T \in \mathbb{R}^{m \times m} \text{ invertible}$$

The pseudoinverse delivers one solution *from the equation* (1)*:*

$$\vec{\phi}^{i+1} - \vec{\phi}^i = -(D_f)^+ g(\phi^i) \text{ whereby } \left\| (\vec{\phi}^{i+1} - \vec{\phi}^i) \right\| \text{ minimal among all solutions}$$

Interpretation: $\left\| (\vec{\phi}^{i+1} - \vec{\phi}^i) \right\|$ *produces minimal deviation of the new angles as compared to the old angles*

Skeletal Animation

Algorithm

Status: *Initial state vector $\vec{\phi}_0 = \vec{\phi}(t_0)$ and end effector $\mathbf{X}_0$*

Algorithm *:*

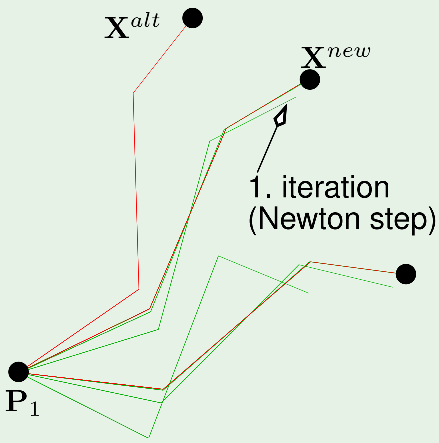
`while ( true ) { // top-level animation`
  $\mathbf{X}_{i+1}$ *by external movement/dynamics*
  `j=0;`
  $\vec{\phi}^0_{i+1} = \vec{\phi}_i$; `// initial state in time-step i+1`
  `do {  // iterative Newton solution for time-step i+1`
    $\vec{\phi}^{j+1}_{i+1} = \vec{\phi}^j_{i+1} - (D_f(\phi^j_{i+1}))^+ \cdot g(\phi^j_{i+1})$; `// Newton iteration j`
    `j++;`
  `} while (`$\left\| \vec{\phi}^j_{i+1} - \vec{\phi}^{j-1}_{i+1} \right\| < \varepsilon$`);`
  `... // further processing of the solution` $\vec{\phi}_{i+1}$
  `i++;`
`}`

Example

Initial: $\mathbf{P}_1 = (50, 400),\ l_1 = 150,\ l_2 = 200,\ l_3 = 100,$

$\vec{\phi}^{old} = \vec{\phi}_0 = (0.6, 1.0, -0.7),\ \mathbf{X}^{old} = (230, 763), \mathbf{X}^{new} = (350, 700)$

1. Newton-step:



$$\frac{\partial f}{\partial \phi_1} = \begin{pmatrix} -363 \\ 180 \end{pmatrix},\ \frac{\partial f}{\partial \phi_2} = \begin{pmatrix} -278 \\ 56 \end{pmatrix},\ \frac{\partial f}{\partial \phi_3} = \begin{pmatrix} -78 \\ 62 \end{pmatrix}$$

$$\Rightarrow\ D_f = \begin{pmatrix} -363 & -278 & -78 \\ 180 & 56 & 62 \end{pmatrix}$$

und $D_f^+ = \begin{pmatrix} 0.001025 & 0.006793 \\ -0.005497 & -0.010536 \\ 0.002010 & 0.00595 \end{pmatrix}$

$$\vec{\phi}^1 - \vec{\phi}^0 = -D_f^+ g(\vec{\phi}^0) = -D_f^+(\mathbf{X}^{old} - \mathbf{X}^{new})$$

$$\Rightarrow \vec{\phi}^1 = \begin{pmatrix} 0.6 \\ 1.0 \\ -0.7 \end{pmatrix} - \begin{pmatrix} 0.001025 & 0.006793 \\ -0.005497 & -0.010536 \\ 0.002010 & 0.00595 \end{pmatrix} \begin{pmatrix} -120 \\ 63 \end{pmatrix} = \begin{pmatrix} 0.295 \\ 1. \\ -0.833 \end{pmatrix},\ f(\phi^1) = \begin{pmatrix} 336.44 \\ 681.27 \end{pmatrix}$$

In highend-animations the disadvantages of IK mostly outweigh its advantages.

Motion-Capturing: Determining the motion path by motion capturing of charakters

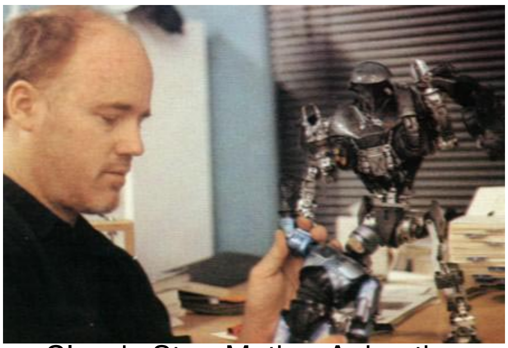
Stop-Motion-Animation:

- Classical approach: Framewise motion capturing (“stop-motion”)
- Keyframe-wise: Tapping der joint-angles and keyframe-animation

Post-editing: Smoothing, overlay etc. of motion paths

Parallel application of FK and IK


Motion-Capturing


Classic Stop-Motion-Animation

Approach (Hierarcical coordination systems in 3D)

Approach: *Description of the limb hierarchy as in 2D*

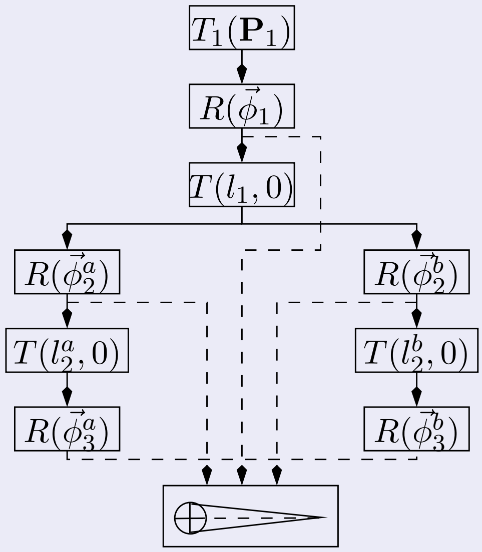
Transformation $T_{i\to i+1} = T(l_i, 0) \cdot R(\vec{\phi}_{i+1})$

whereby $R(\vec{\phi}_{i+1}) = R(\phi^x_{i+1}, \phi^y_{i+1}, \phi^z_{i+1})$ *(rotation matrices with Euler angles)*

Forward kinematics *can be used directly with this approach*

Inverse kinematics *requires optimization of the rotation parameters:*

- *Avoid ambiguities by using quaternions*
- *Problem: Optimization of quaternions enforces adherence to the condition of unit quaternions:* $\|\underline{\mathbf{q}}\| = 1$



Approach (Denavit-Hardenberg)

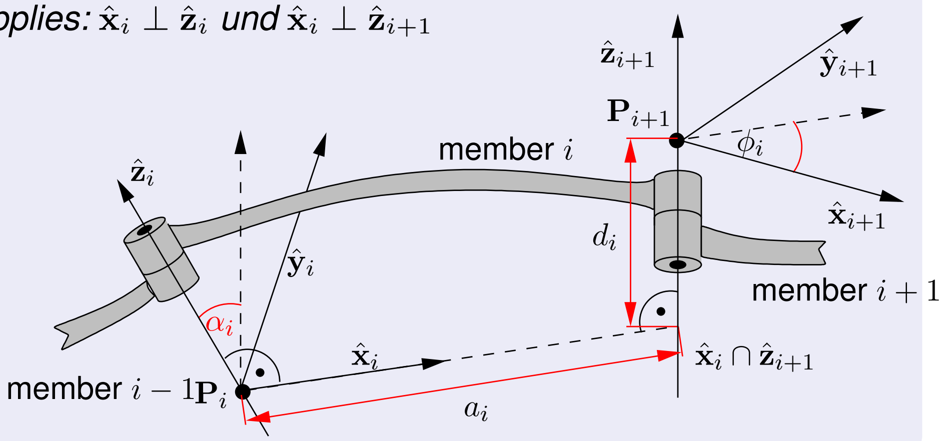
Alternative Approach *with simple angle parameters*

Initial position:

- n*-limbed skeleton without branching*
- *each limb has a rotation axis*

Coordination systems *for* i*-th member* $K_i = \{\mathbf{P}_i, \hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i\}$*, wereby*

- $\hat{\mathbf{z}}_i$ *corresponds to the axis of rotation*
- $\hat{\mathbf{x}}_i$ *and* $\mathbf{P}_i$ *is determined by the shortest connection between* $\hat{\mathbf{z}}_i, \hat{\mathbf{z}}_{i+1}$*:*
  - *exists always: skew*(dt: windschief) $(\hat{\mathbf{z}}_i \nparallel \hat{\mathbf{z}}_{i+1})$ *or parallel*
  - *automatically applies:* $\hat{\mathbf{x}}_i \perp \hat{\mathbf{z}}_i$ *und* $\hat{\mathbf{x}}_i \perp \hat{\mathbf{z}}_{i+1}$



Approach (Denavit-Hardenberg (ctd.))

Control Parameter:

- a_i *distance between axes $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_{i+1}$*
- α_i *angle between $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_{i+1}$ (both z-axes $\perp \hat{\mathbf{x}}_i$)*
- d_i *distance of the intersection $\hat{\mathbf{x}}_i \cap \hat{\mathbf{z}}_{i+1}$ to origin $\mathbf{P}_{i+1}$ from K_{i+1}*
- ϕ_i *angle between $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i+1}$ (both $\perp \hat{\mathbf{z}}_{i+1}$)*

With starting point $\mathbf{P}_1$: $4 \times n + 3$ degrees of freedom

Transformation *from K_i to K_{i+1} in local coordinates:*

$R(x, \alpha_i)$: *Map $\hat{\mathbf{z}}_i$ to $\hat{\mathbf{z}}_{i+1}$ (current x-axis is $\hat{\mathbf{x}}_i$).*
$T(a_i, 0, d_i)$: *Slide $\mathbf{P}_i$ to $\mathbf{P}_{i+1}$*
$R(z, \phi_i)$: *Map $\hat{\mathbf{x}}_i$ to $\hat{\mathbf{x}}_{i+1}$.*
Combined:

$$\mathbf{Q}^{i+1} = T_{i\to i+1}(\mathbf{Q}^i), \quad T_{i\to i+1} = R(x, \alpha_i) \circ T(a_i, 0, d_i) \circ R(z, \phi_i)$$

Objective

Goal: *Animation of bodies with muscles and skin on a skeleton basis*

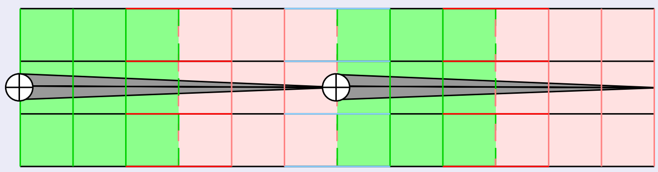
Simple Layer-Model: *Control of the skeleton with* linkage *to the skin*

- **Skeleton-level:** *Lowest level; corresponds to the multi limbed model*
- **Skin-level:** *geometry from NURBS-surface or another geometric representation*
- **Muscle-level:** *Connects skeleton and skin via FFD's*

Approach (Chadwick '89 (special case: arm or leg))

Initial position: *Two members with zero rotation*

Definition of FFD-volumes: *Each member is furnished with tri-cubic FFD Bézier-volumes.*



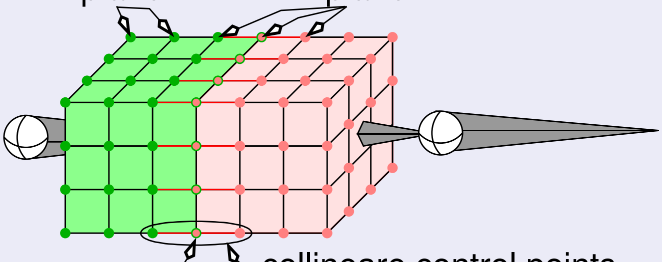
initial state

Approach (Chadwick '89 (ctd.))

Edge conditions *for C^1-transitions* between *FFD-volumes:*

1. *KPs of the border planes are identical*
2. *KPs of the neighboring planes lie collinearly (ratio 1:1)*

Seven control point planes *per member,* three inner planes, *per each two* connection planes

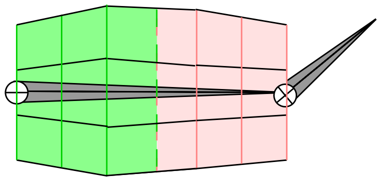


Bending of the joint: Fulfills C^1-transitions

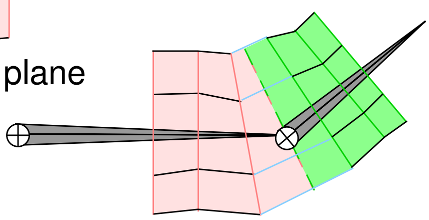
1. Skaling of the two free inner planes: Muscle control
2. Rotation of the two free connection planes: Alignment of the members for joint rotation

Binding: Geometry is distorted according to it's position in the FFD volume

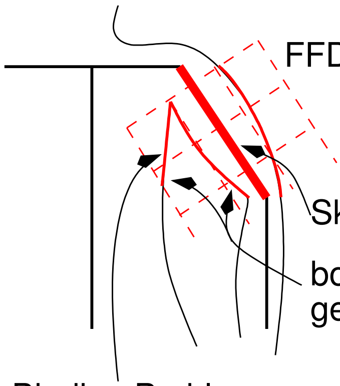
Problem: Automatic association of geoemtry is not always correct $\rightarrow$ interactive binding of geometry to skeleton



adoption of inner plane



adoption of connecting planes



Binding Problem