 

Kinematics of redundant characters

Advanced Computer Animation Techniques
Aug-Dec 2014
cesteves@cimat.mx

To Read ...

- K. Yamane and Y. Nakamura. "**Natural Motion Animation through Constraining and Deconstraining at Will**". *IEEE Transactions on Visualization and Computer Graphics*, 9(3). 2003.

Natural motion animation through constraining and deconstraining at will

- Generate a motion in which:
  - The link specified by the user (the dragged link) follows the indicated path,
  - Any number of links specified by the user (pinned links) stay at their reference positions,
  - Each joint angle stays in its motion range, and
  - Each joint angle stays as close as possible to the given reference angle.
- Difficulties:
  - Difficult (or virtually imposible) to derive an analytical method that can handle the general cases, and
  - The constraints often conflict with each other (e.g. when the user drags a link beyond the reachable space determined by the pinned links).

Differential kinematics with redundancy

- The Jacobian matrix of the position of a link with respect to the joint angles is defined as:

$$\mathbf{J}_i \triangleq \frac{\partial \mathbf{r}_i}{\partial \theta}$$

$\mathbf{r}_i$: position of link i.

θ: vector composed of all joint angles.

$\mathbf{J}_i$: Jacobian matrix of $\mathbf{r}_i$ with respect to θ

- The velocities of link i and joint angles are related by:

$$\dot{\mathbf{r}}_i = \mathbf{J}_i \dot{\theta}$$

- If the base link is not fixed to the inertial frame, its linear and angular velocities are also included in $\dot{\theta}$.

Differential kinematics with redundancy

- If $\mathbf{J}_i$ is square and nonsingular, it can be inverted to yield:

$$\dot{\theta} = \mathbf{J}_i^{-1}\dot{\mathbf{r}}_i$$

by which we can control the joints based on the reference trajectory of $\mathbf{r}_i$.

- Because $\mathbf{J}_i$ is not a square matrix, the pseudoinverse $\mathbf{J}_i^{\#}$ should be used:

$$\dot{\theta} = \mathbf{J}_i^{\#}\dot{\mathbf{r}}_i + (I - \mathbf{J}_i^{\#}\mathbf{J}_i)\mathbf{y}$$

where I is de identity matrix and $\mathbf{y}$ is an arbitrary vector.

- The second term shows the redundancy and reserves the degrees of freedom that we can use for other constraints.

Singularity-robust (SR) inverse or damped pseudoinverse

- Consider a linear equation **Ax = b**.
- If the coefficient matrix **A** is not square, we usually use its pseudoinverse $\mathbf{A}^{\#}$ to compute the least-squares solution with the minimal norm.
- The pseudoinverse solution tends to have singular points because it minimizes the norm of the error **|b - Ax|** first and then minimizes the norm of the solution **|x|**.
- The SR inverse avoids this problem by minimizing the sum of the norms of the error and the solution.
- For an **m**-by-**n** (**m<n**) matrix **A**, its pseudoinverse is computed by:

$$\mathbf{A}^{\#} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$$

- $\mathbf{A}^{\#}$ may have extremely large elements when $\mathbf{A}\mathbf{A}^{T}$ is nearly singular.
- The SR inverse uses the following equation:

$$\mathbf{A}^{*} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T + k\mathbf{I})^{-1}$$

$\mathbf{A}^{*}$: SR inverse of A

$\mathbf{I}$: identity matrix.

k: weighting between the norm of the solution and the error.

The Algorithm

(1) Compute the general solutions of joint velocities that move the dragged link toward the indicated position. (Section 3.1)

(2) Compute the desired velocities of the other constraint variables, taking account of their reference and current values. (Section 3.4)

(3) Compute the Jacobian matrix of the constraint variables with respect to the joint angles. (Section 3.3)

(4) Using the general solutions in Step 1, find a particular solution that closely satisfies the desired velocities and constraint variables. (Section 3.2)

(5) Numerically integrate the joint velocities to get the joint angles.

(1) General Solutions of joint velocities to move the dragged link toward the indicated position

- First compute $\dot{\theta}$ with which the dragged link exactly follows its reference velocity $\dot{\mathbf{r}}_P^{ref}$ and position $\mathbf{r}_P^{ref}$. Let $\mathbf{r}_P$ denote the current position of the dragged link. Its desired velocity is computed by:

$$\dot{\mathbf{r}}_P^d = \dot{\mathbf{r}}_P^{ref} + \mathbf{K}_P(\mathbf{r}_P^{ref} - \mathbf{r}_P)$$

where $\mathbf{K}_P$ is a positive-definite gain matrix.

- The relationship between $\dot{\theta}$ and $\dot{\mathbf{r}}_P$ is given by:

$$\dot{\mathbf{r}}_P = \mathbf{J}_P\dot{\theta}$$

$\mathbf{J}_P$: Jacobian matrix of $\mathbf{r}_P$ with respect the joint angles.

- The general solution $\dot{\theta}$ for the desired velocity $\dot{\mathbf{r}}_P^d$ is computed by:

$$\dot{\theta} = \mathbf{J}_P^{\#}\dot{\mathbf{r}}_P^d + (\mathbf{I} - \mathbf{J}_P^{\#}\mathbf{J}_P)\mathbf{y}$$

(2) Desired velocities of the other constraint variables, taking account of their reference and current values.

- The desired velocity of each pinned link $\dot{\mathbf{r}}_{F_i}^d$ is computed by the following feedback law:

$\mathbf{r}_{F_i}^{ref}$: reference position.

$\mathbf{K}_{F_i}$: positive-definite gain matrix.

$$\dot{\mathbf{r}}_{F_i}^d = \mathbf{K}_{F_i}(\mathbf{r}_{F_i}^{ref} - \mathbf{r}_{F_i})$$

- The desired velocity of joints with their reference angles for 1-DOF joints:

θ_D^{ref}: reference joint angles.

$\mathbf{K}_D$: positive-definite gain matrix.

$$\dot{\theta}_D^d = \mathbf{K}_D(\theta_D^{ref} - \theta_D)$$

- The desired velocity of joints that exceed their motion ranges for 1-DOF joints:

$$\dot{\theta}_{L_i}^d = \begin{cases} \mathbf{K}_{L_i}(\theta_{L_i}^{max} - \theta_{L_i}) & \text{if } (\theta_{L_i} > \theta_{L_i}^{max}) \\ \mathbf{K}_{L_i}(\theta_{L_i}^{min} - \theta_{L_i}) & \text{if } (\theta_{L_i} < \theta_{L_i}^{min}) \end{cases}$$

$\theta_{L_i}^{max}$ and $\theta_{L_i}^{min}$: maximum and minimum joint angles.

K_{L_i}: positive scalar gain.

(3) Compute the Jacobian matrix (J_{aux}) of the constraint variables with respect to joint angles

- Let $\mathbf{J}_{F_i}$ $(i = 1 \dots N_F)$ be the Jacobian matrix of $\mathbf{r}_{F_i}$ with respect to the joint angles. Then for all pinned links, we have:

$$\dot{\mathbf{r}}_{F_i} = \mathbf{J}_{F_i}\dot{\theta}$$

- For the joints with reference angles, the relationship between their velocities $\dot{\theta}_D$ and $\dot{\theta}$ is described by:

$$\dot{\theta}_D = \mathbf{J}_D\dot{\theta}$$

  where J_D is the matrix whose *(i,j)*th element is **1** if the ith element of θ_D corresponds to the jth element of θ_D and **0** otherwise.

- The relationship between $\dot{\theta}$ and the velocity of θ_L is described as follows:

$$\dot{\theta}_L = \mathbf{J}_L\dot{\theta}$$

- Combining the above-defined matrices, J_{aux} is formed as follows:

$$\mathbf{J}_{aux} = \begin{pmatrix} \mathbf{J}_{F_1}^T & \dots & \mathbf{J}_{F_{N_F}}^T & \mathbf{J}_D^T & \mathbf{J}_L^T \end{pmatrix}^T$$

(4) Find a particular solution that closely satisfies the desired velocities and constraint variables.

To Read ...

- K. Yamane, J.J. Kuffner and J.K. Hodgins. "**Synthesizing Animations of Human Manipulation Tasks**". *ACM Transactions on Graphics* (*SIGGRAPH 2004*). 23(3). 2004.

The Algorithm

- The planning phase generates a collision-free path for the object while taking into account the naturalness of the poses the character must use to position the object in a given location and the task constraints (balance and collision avoidance).

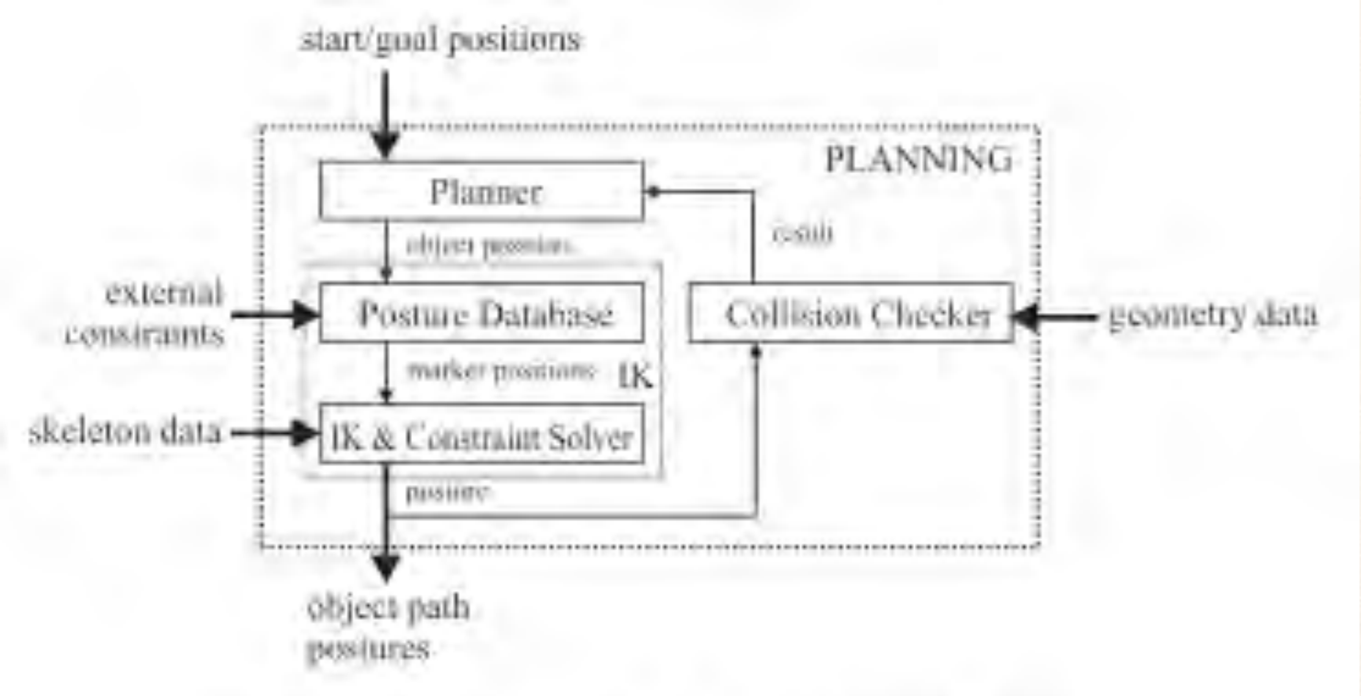

Figure 2: Overview of the planning phase.

To Read ...

- K. Grochow, S.L. Martin, A. Hertzmann and Z. Popovic. "**Style-based Inverse Kinematics**". *ACM Transactions on Graphics* (SIGGRAPH 2004). 23(3). 2004.