Inverse Kinematics

Inverse kinematics (IK) is the process of determining the parameters of a jointed flexible object (a kinematic chain) in order to achieve a desired pose. IK is an alternative to traditional Visual3D models (Six Degrees of Freedom (6 DoF) models) in which the user defines joints (e.g. explicitly state which segments are connected by a joint) and specifies the properties of all joints.

Because the targets used to track the segments are often subject to measurement error and soft tissue artifact, motion about some of a joint's degrees of freedom may be much larger than the motion that would be realistically possible. Lu and O'Connor (1999) described a global optimization process where physically realistic joint constraints can be added to the model to minimize the effect of the soft tissue and measurement error. Lu and O'Connor termed this process **Global Optimization** while others inside the biomechanics community prefer the term **Inverse Kinematics**. Inverse Kinematics, or IK, is the termed used by Visual3D.

Theory of Inverse Kinematics

Traditionally in Visual3D the motion of segments are fully described by six degrees of freedom (three rotational parameters and three translational parameters); thus the segments are free to move as if there are no constraints at the joints.

Photogrammetric procedures for obtaining measurements of six degree-of-freedom (DOF) segmental motion require that a system of three or more non-collinear points be fixed to each segment. These non-collinear points are used to define orthogonal segment coordinate systems (SCSs) located independently within each of the segments. In addition, an orthogonal laboratory coordinate system (LCS), which is assumed to be stationary, is defined during system calibration.

Measurement of the position and orientation of a local segment coordinate system (SCS) with respect to the laboratory coordinate system (LCS) can be used to completely describe the segment's motion.

A least squares procedure is used by Visual3D to determine the position and orientation. To understand how this procedure works, consider a point located on a segment at position $\overline{A}$ in the SCS. The location of the point in the LCS ($\overline{P}$) is given by:

$$\overline{P} = T\overline{A} + \overline{O}$$

where $\mathbf{T}$ is the rotation matrix from the SCS to the LCS and $\overline{O}$ is the translation between coordinate systems.

If the segment undergoes motion, the new orientation matrix $\mathbf{T}$ and translation vector $\overline{O}$ may be computed at any instant, provided that the noncolinear points $\overline{A}$ are predetermined and P is measured. The matrix $\mathbf{T}$ and origin vector $\overline{O}$ are found by minimizing the sum of squares error expression:

$$\sum_{i=1}^{m}\left((\overline{P}_i - T\overline{A}_i) - \overline{O}\right)^2$$

under the orthonormal constraint:

$$T^tT = I$$

where m is equal to the number of targets on the segment (m > 2). (This solution is adapted from the solution outlined by Spoor & Veldpaus in the Journal of Biomechanics, pp. 391- 393, 1980.).

However, because the targets used to track the segments are often subject to measurement error and soft tissue artifact, the measured motion about some of the degrees of freedom maybe much larger than the motion that would be realistically possible. Lu and O'Connor (1999) described a global optimization process which applies physically realistic joint constraints to the model to minimize the effect of the soft tissue and measurement error.

Mathematically Van Den Bogert and Su (2008) described this approach which specifies the configuration of the total body based on a set coordinates q. In this case T and O of equation 1 become a function of all the generalized coordinates:

$$\overline{P}_i = T(q)\overline{A_i} + \overline{O}(q)$$

and the expression that is minimized becomes:

$$\sum_{i=1}^{m_t}\left((\overline{P}_i - T(q)\overline{A}_i) - \overline{O}(q)\right)^2$$

where now mt is the total number of targets on all the segments in the Inverse Kinematics chain.

Inverse Kinematics as a Global Optimization Problem

In Visual3D the Inverse Kinematics problem is solved as a global optimization problem, which

computes the pose of a model that best matches the motion capture data in terms of a global criterion.

Defining IK Constraints

IK chains are created in Visual3D using the IK Constraints tab in the Model Builder portion of Visual3D. Prior to creating an IK chain all of this segments that are to be included in the chain must already be defined by in the Segments tab of the Visual3D Model builder.

Segment Weight

When creating an IK chain the user may want to make sure that certain segments follow the tracking targets with a higher degree of accuracy then other segments. For example, the user may want to assure that the distance between the foot (RFT or LFT) and the floor (or force platform if available) remains similar to the values that would be obtained using the tradition Visual3D 6 DoF method. To help with this situation Visual3D lets the user add a weight factor for the mobilizers associated with each segment. (The default weighting factor for all segments is 1.0). Adding weight factors effectively changes equation 3 to:

$$\sum_1^n k_n \left[\sum_i^m \left((\bar{\boldsymbol{P}}_i - \boldsymbol{T}(\boldsymbol{q}) \bar{\boldsymbol{A}}_i) - \bar{\boldsymbol{O}}(\boldsymbol{q}) \right)^2 \right]$$

where:

1. n is the number of segments in the IK chain;
2. k_n Is the weight factor for the mobilizers associated with the segment; and
3. m is the number of targets used to track that segment.

Boundary Conditions

To use the LBFGSB optimizer you first have to setup a traditional IK model, then:

1. Change the optimizer to LBFGSB
2. Select the segment for which you want to add boundary conditions in the IK Segment List Box. (Not you can only add boundary conditions for degrees of freedom where the mobilizer is checked, If the mobilizer is not check that component of the joint is fixed and not part of the IK solution,)
3. Select the “Properties” Button to bring up the “IK Degrees of Freedom Property Dialog”
4. Change the Low and High Range boundary condition for the degree of freedom(s) of interest.
5. Click OK to accept the changes and close the Dialog.
6. Build (or Recalc) the Model.

This will now solve the IK problem within the limits of any boundary conditions you specify.

Comparing 6DOF and Inverse Kinematics

Visual3D models are based on a linked set of rigid segments. Traditional Visual3D models (6DOF) assumed that segments were implicitly linked by the motion capture data (e.g. segments didn't come apart because the subject didn't come apart) and the joints were modeled with 6 degrees of freedom (e.g. all segments were treated as if they were independent). The mapping of motion capture markers to 6 DOF segments is a matter of tracking a set of markers that are linked rigidly to the segment. This least squares solution requires specifying the Segment Coordinate System and tracking the pose (position and orientation) of that segment. Essentially this is a straightforward pattern recognition; the pattern (configuration) of the tracking markers are specified in a standing trial, and this pattern is fit to the marker configuration in each frame of motion capture data.

The difference between the traditional Visual3D 6DOF model and the IK model is that the latter allows constraints to be added between segments that restrict the relative motion between the segments. This is accomplished by creating one or more IK chains. An Inverse Kinematics solution is dependent on the choice of hierarchical model (the specified IK chains) because the task is to identify an articulated figure consisting of a set of rigid segments connected with joints. Varying angles of the joints yields an indefinite number of configurations, so in the general case there is no analytic solution. This necessitates solving for pose with an IK model as an optimization problem.

Examples

Certain common scenarios occur when building Visual3D models where we recommend considering an IK model.

Kinematic Foot

When using 6 DOF, users commonly create a second kinematic-only foot which represents neutral (0 degrees flexion/extension) as when the foot is aligned with the floor. The reason for this is explained further here.

With an IK model, it is important that the proximal end of the segment be defined as the point of rotation. For this reason, the kinematic only foot segments described in the Foot and Ankle Tutorial may not be appropriate.

The IK Kinematic Foot Tutorial describes kinematic only foot segments which are tracked using the IK POSE estimation.