

visual3d models are based on a linked set of rigid segments. traditional visual3d models (6 dof) 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 mocap 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 the [specification of the segment coordinate system](#) and the [tracking of the pose](#) (position and orientation) of a 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 mocap data.

an alternative to the 6 dof solution is to define joints (e.g. explicitly state which segments are connected by a joint) and to specify 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 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 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 (ik is the termed used by visual3d).

the difference between the traditional visual3d 6 degree of freedom model and the inverse kinematics (ik) model is that constraints can be added between segments that restrict the relative motion between the segments. this is accomplished by creating one or more ik chains.

inverse kinematics is the process of determining the parameters of a jointed flexible object (a kinematic chain) in order to achieve a desired pose.

an inverse kinematics solution is dependent on the choice of hierarchical model 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

global optimization

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 the 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.

theory

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 noncolinear points be fixed to each segment. these non-colinear 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.



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. lu and o'connor termed this process global optimization while other inside the biomechanics community prefer the term inverse kinematics. (inverse kinematics is the term used by visual3d but visual3d's approach is based on the lu and o'connor technique.)

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:



and the expression that is minimized becomes:



where now mt is the total number of targets on all the segments in the inverse kinematics chain.

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:



where n is the number of segments in the ik chain. k_n is the weight factor for the mobilizers associated with the segment and m is the number of targets used to track that segment.

boundary conditions

this functionality is new and has limited documentation.

here is some preliminary information.

to use the lbfgsb optimizer you first have to setup a traditional [inverse kinematics model](#):

next:

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 inverse kinematic problem within the limits of any boundary conditions you specify.

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.

From:
<https://has-motion.com/wiki/> - **Wiki Documentation Site**

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:kinematics_and_kinetics:inverse_kinematics&rev=1718804151

Last update: **2024/06/19 13:35**

