MODULAR COMPONENTS FOR DETAILED KINEMATIC MODELLING OF JOINTS

Victor Ng-Thow-Hing[1] and Wei Shao[2]
[1]Honda Research Institute USA, Inc., Mountain View, California, USA, vng@honda-ri.com
[2]New York University, New York, USA, weishao@cs.nyu.edu

INTRODUCTION

Kinematic modeling of joints is an important element for proper motion analysis of tasks, including the estimation of muscle lengths and moment arms [Delp and Loan, 1995]. In general, anatomical axes of rotation and joint centres are obtained from the literature and refined to produce improved fidelity of motion [Buford and Andersen, 2002]. Various applications will require different levels of detail of the kinematic model. This implies that the infrastructure used to model the joints should be scalable in complexity and be modular so that local regions can have their own custom models yet still co-exist within the same skeletal model. In addition, the use of mechanical joint models can result in joint degrees of freedom that may not coincide with a clinician's or physiotherapist's intuitive descriptions of motion. For example, elevation of the scapula (shoulder shrugging) can be specified by rotating the sterno-clavicular joint to create superior motion in the scapula. Unfortunately, the skeleton hierarchy would also propagate the rotation to the humerus, which may not be desired (Figure 1).

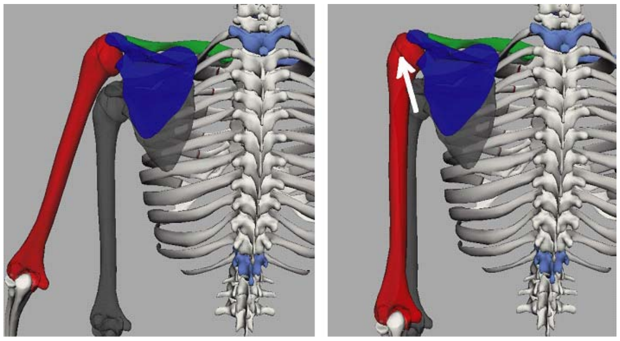

Figure 1: The left image shows unwanted rotation propagation to the humerus while on the right, a more natural elevation of the scapula is achieved when adjusting this degree of freedom using our component model.

PREVIOUS WORK

The origins of articulated models for human joint representations can be found in the study of kinematics of robotic manipulators. Early animation systems, such as PODA [Girard and Maciejewski 1985] made use of the Denavit-Hartenberg link parameter notation from robotics to represent figures with articulated limbs. Although the notation is a convenient way to relate coordinate frames between adjacent segments with four parameters, each parameter set only describes a single degree of freedom between two segments. Multiple sets of parameters must be combined to achieve multiple degree of freedom (DOF) joints. For complex articulations, a higher level of organization to provide a convenient, unified interface for adjusting multiple joint DOFs is desirable.

Physiological joints have been shown to have many complexities that are often neglected in kinematic models. For example, biomechanists routinely specify joints with several non-orthogonal, arbitrary axes of rotation [Delp and Loan, 1995, Buford and Andersen, 2002] that are better aligned to bone articulation. Many joints have translational components and changing centres of rotation, including the knee which is traditionally simplified as a single DOF hinge joint [Bull and Amis, 1998]. In joints like the shoulder, the closed loop consisting of the clavicle, scapula and thoracic surface of the rib cage creates a coupling between the articulations of all these joints. Several groups model this situation by enforcing a constraint on the scapula to stay on the surface of an ellipsoid approximating the rib cage [Garner and Pandy, 1999, Maurel and Thalmann, 2000]. Other structures, like the human spine, exhibit a high degree of coupling behaviour between the vertebrae. For example, kinematic models of the human spine have been built that exhibit coordinated flexion/extension, lateral bending and axial twist rotation of the vertebrae [Monheit and Badler, 1991]. Our framework is designed to accommodate all these desirable biomechanical characteristics.

Judging from the extensive array of previous joint models in the literature, no single representation dominates the capturing of all joint kinematics. In fact, specialized joint models are often needed, as in the shoulder and spine. Similarly, the complexity of these articulations should be made accessible to users by providing meaningful controls. In applications where physiological consistency is desired, a user should not be allowed to configure a skeleton into a non-natural, infeasible posture. These guidelines influenced the design of our joint component model.

We have developed a modular joint component framework that allows practitioners to create complex kinematic models by combining a set of components that can implement multi-joint dependencies and limits on circumduction of limbs (Figure 2).

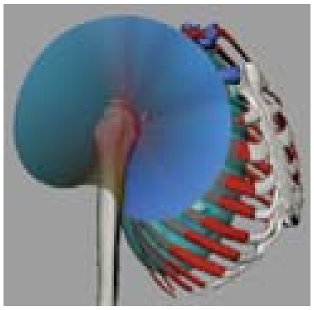

Figure 2: Joint cones delineate circumduction of limbs.

METHODS

Examination of the many existing kinematic models has allowed us to define seven types of components that can be combined to recreate existing and new, extensible kinematic models. An overview of these components are described in this section. More technical details of our implementation are described in [Shao and Ng-Thow-Hing, 2003].

JOINT COMPONENT MODEL

In our model, an articulated figure consists of a set of *segments* that can express only rigid body motion. A hierarchy relates the segments to each other where the motion of segment is expressed relative to its parent in the form of a 4x4 transformation matrix. A segment can have no parent, implying that its motion is relative to the world coordinate frame. Therefore, the segments of a single articulated figure can be partitioned into several hierarchical trees. This is useful if an articulated figure contains free-floating segments.

A *joint set* contains one or more segments whose configuration is described by independent *degrees of freedom* (DOF)} or *generalized coordinates*. For each segment in the joint set, its relative motion with its parent is described as an *articulation* or *joint*. For example, the shoulder joint set consists of four bone segments (clavicle, scapula, thorax and humerus) with four articulations (sterno-clavicular, acromio-clavicular, scapulo-humeral and scapulo-thoracic joints) as shown in Figure 3.

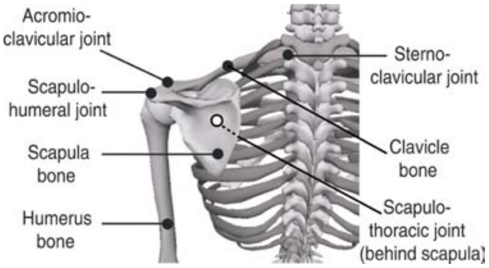

Figure 3: Shoulder joint set

To implement joint set functions, a set of building blocks called *joint components* were designed. Each joint component implements a cohesive function, facilitating its reuse in different contexts. A *joint set function* comprises a network of joint components that is created by connecting the output of one component to the inputs of one or more other components (Figure 4). Generalized coordinates feed into the network with transformation matrices for segments produced as output. A relatively small number of simple joint components can be combined to create a diverse array of behaviours. This framework allows new types of joint components to be added and used with existing components with minimal coupling between modules.

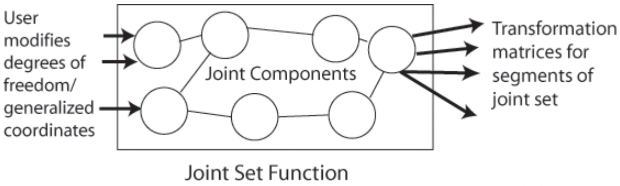

Figure 4: Overview of joint component model

Multiplication components facilitate combination of various segmental transformation matrices and matrix decompositions created from generalized coordinates descriptions. This component take as input a list of several matrices and multiplies them together to produce a single transformation matrix as output. The order of elements in the list determines the multiplication order. The output can either be the final transformation that will be applied to the corresponding joint or an intermediate result that will be used as input to other components.

One-to-Many components allow a single generalized coordinate to influence one or more segmental transformations directly. For example, we have implemented a knee model where a single generalized coordinate is the common parameter of several cubic spline functions that evaluate the Euler angle rotations and translations for the patella and tibia (Figure 5).

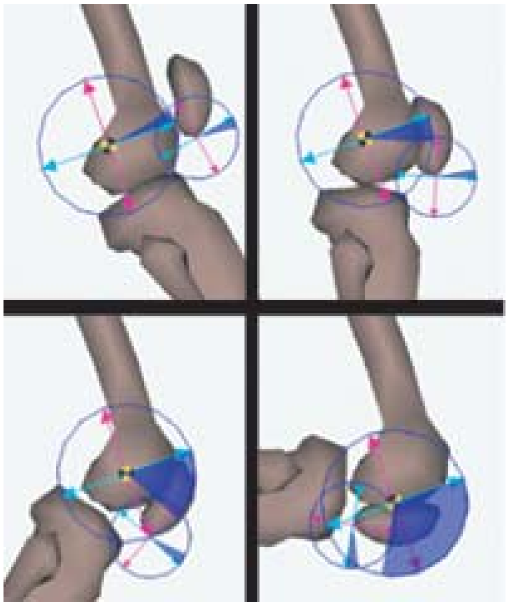

Figure 5: Knee model showing translation and rotation of patella and tibia driven by a single degree of freedom.

Compensation components allow a segment's parent transformation to be cancelled out to produce segment motions without propagated parent transformations. In a standard hierarchical skeleton tree, the child segment inherits the transformations of a parent segment. However, this may produce undesirable behaviour in some situations. For instance, if we wanted to shrug the shoulders of a human model, the rotation in the clavicle would propagate to the humerus, causing the humerus to rotate away from the body (Figure 1). In addition to canceling out any rotation transformations inherited from any ancestor segment, connectivity at the joint is maintained by adjusting the translation of the segment.

In the shoulder joint set we created, two compensation components are used to create an independent humerus orientation from the scapula and clavicle rotation. The first component cancels the effects of the acromio-clavicular joint (connecting the scapula to the clavicle), and the second component nullifies the sterno-clavicular joint (connecting the clavicle to the sternum of the thoracic cage).

Rotation components handle rotations with changing joint centres of rotation, and nonintersecting, non-orthogonal axes of rotation. Joints with multiple DOF rotations are created by combining rotation components, each of which produces a rotation matrix for a single axis rotation. An important simplifying assumption being made is that the joint centres for each axis rotation are independent of the rotations about the other axes. While this may not be the case in reality, reasonable articulations were observed in the joints we created. A rotation component can have several angle intervals with each interval having a different joint rotation centre. The ability to model a changing joint rotation centre is important to accurately describe rotations in the knee and humerus of the shoulder [Kapandji, 1982].

Dependency components allow dependencies to be established between segment joints, including joint limits as a function of another joint's degree of freedom. In each dependency component, a pair of joints are specified, one as the active joint *a* that drives the other passive joint *p*. The movement of a DOF of *p* is set to be dependent on a DOF of *a* through a mapping function. The actual nature of the mapping function used in the dependency component can be any linear or nonlinear function. Interpolating splines are often convenient to match dependency relations to experimental data points. Typically, the DOF corresponds to Euler angles that define the rotation matrix of each joint. The dependency component takes two input Euler angles, one from each joint, and contains a mapping function to output a modified angle for the passive joint. We implemented several types of mapping relationships:

> **One-to-one mapping:** For any given DOF value of *a*, a DOF value for *p* is defined. For instance, the rotation of the scapula around an axis perpendicular to its outward surface tangent plane is almost linearly dependent on the abduction of upper arm. A linear one-to-one mapping can capture this relationship.
>
> **One-sided bound mapping:** This is a one-to-one map where values of *p* are bounded on one side by a lower or upper limit that is a function of a DOF of *a*. An example of this is the dependency between the abduction of the humerus bone and the elevation of the clavicle bone. The higher the upper arm is raised, the more restricted is the vertical movement of the shoulder's clavicle. The restriction is due to a lower limit placed on clavicle elevation, which can be implemented as a one-sided bound that is dependent on the amount of abduction of the humerus.
>
> **Two-sided bound mapping:** The value of a DOF of *p* is bounded on both sides by limits dependent on a DOF of *a*. Again using the shoulder as an example, when the left upper arm is rotating in the horizontal plane from the left side to the front right of the body, the horizontal movement of the shoulder (at the clavicle bone) becomes more restricted. A similar phenomenon occurs when the left upper arm is rotating to the back of the body. Since there are both upper and lower limits, a two-sided mapping is appropriate.

Joint cone components model more accurate physiological limits on limb circumduction than having separate limits along each rotation axis (Figure 2). Joint sinus cones [Maurel and Thalmann, 2000, Wilhelms and van Gelder, 2001] have been used to provide a better mechanism for joint limits for ball-and-socket joints than pairs of Euler angle bounds for each joint DOF.

In a joint cone component, the joint sinus cone is defined using a reference point $\boldsymbol{p}$ and a space curve $\boldsymbol{c}$. The reference point $\boldsymbol{p}$ is the apex of the cone and is located at the joint centre. The curve $\boldsymbol{c}$ creates an irregular boundary at the base of the cone and is defined by an initial list of user-selected control points. An additional vector $\boldsymbol{v}_{\mathbf{rest}}$ is defined with origin at $\boldsymbol{p}$ with direction along the bone's longitudinal axis in an initial rest configuration. This cone provides a way of bounding the movements of two DOFs of a joint, such as abduction/adduction and flexion/extension in the humerus at the shoulder. To limit the third twist DOF, an additional pair of angle bounds is associated with each control point on curve $\boldsymbol{c}$ and the tip of $\boldsymbol{v}_{\mathbf{rest}}$. Interpolation between these values determines the twist angle bounds for the interior of the cone. Whenever a bone's current longitudinal axis lies outside of a joint cone's boundary curve, new rotation angles are computed to restrict the bone's orientation back to the bounds of the cone.

Custom components handle specialized joint kinematics such as the scapulo-thoracic motion in the shoulder that creates a closed-loop in the joint hierarchy. For example, a scapula constraint component was created to handle the specific situation of the scapulo-thoracic constraint in the shoulder. This example illustrates how the component framework can be extended for special handling of an individual joint. The scapula is always gliding on a curved surface defined by ribs, muscles and fatty structures. To represent this in our model, we use an ellipsoidal constraint as others have done in the past [Garner and Pandy 1999, Maurel and Thalmann 2000]. However, instead of using only one ellipsoid, we have chosen to use two, with one for each side of the rib cage (Figure 6).

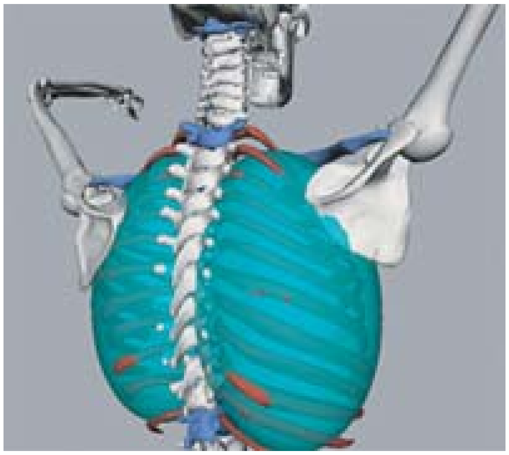

Figure 6: Two ellipsoids are used to constrain the scapulas to slide over the thoracic cage.

This allows the sliding constraints on both sides of the rib cage to be properly maintained as the spine is twisted or laterally bent. In order to constrain the scapula bone to be gliding on the surface of an ellipsoid, we define pairs of reference points on the scapula, and make sure that at least one active pair stays on the ellipsoid at all times.

We define our pairs of reference points to lie near the outer perimeter of the scapula as shown in Figure 7.

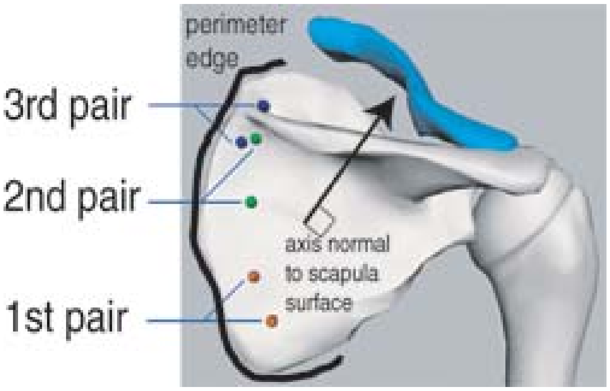

Figure 7: Reference contact pairs on the scapula

Having several pairs of reference points allows the contact area between the scapula and rib cage to change depending on other joints in the shoulder. Referring to Figure 7, the area close to the 1st pair is more likely to be in contact with the thoracic cage when the shoulder is lifted. The 2nd pair is more likely to be in contact when the shoulder is lowered. The 3rd pair is active when the scapula is fully rotated clockwise around the axis normal to its surface. Therefore, these three pairs of reference points are used to find an interpolated pair over two DOFs corresponding to the amount of shoulder lift and rotation about the scapula.

In general, it is desirable to create joint components that can be reused. Nevertheless, the ability to create very specialized constraints can be useful to create tailored, intuitive parameters to simplify the description of complex articulations unique to a particular joint. More biomechanical detail can be added to a joint component as deemed necessary for the application.

RESULTS

Having described all the joint components, we can connect them in a network to construct joint set functions for the segments of our skeleton. We will describe two cases of fairly complex joints we created using our framework: the spine and the shoulder. Initial estimates of parameter data for the joints were determined from literature on joint physiology [Kapandji 1982]. Custom software plug-ins were developed for the Maya 3-D modelling software [Alias|wavefront 1999] to allow interactive placement of the bones and adjustment of joint parameters. Maya's advanced modelling environment allowed articulation of joint sets to be evaluated interactively. Once we were satisfied with the joint model, we exported the parameters for all the joint components in an XML-based file format, which is loaded into our own OpenGL-based custom application software.

We implemented our joint component framework in these two different software environments to test our ability to interchange joint models between them. We can achieve interactive rates on a Pentium III 933 MHz Windows 2000 computer, with a Nvidia GeForce4 graphics card.

Spine Model: There are twenty-four movable vertebrae in the spine of a human. According to their position and functionality, they are divided into three joint sets: the cervical region (seven vertebrae in the neck), the thoracic region (twelve vertebrae in the thorax), and the lumbar region (five vertebrae in the abdomen). For the thoracic joint group, we also included all the ribs and the sternum, creating the thoracic cage. For all three spine joint sets, the same type of joint function is used. The difference between them is just the joint parameters given for each joint group, where the amount of rotation in the thoracic vertebra is considerably less than the cervical and lumbar regions. For example, the cervical joint set has seven joints (c1-c7) as well as seven bones (including both the vertebrae and the discs between any two vertebrae). Each joint alone has three DOF of rotation and thus has three rotation components. Rotation axes and rotation centres are estimated from [Kapandji, 1982] for each rotation component. The three rotation components for a single joint may have quite different rotation centres and non-orthogonal rotation axes. A pair of joint limit angles defined by a one-to-many mapping component is provided to bound each of the rotations. Since rotation behaviour of the vertebrae in the spine are coupled together, we simplify movement control to have only three DOF: flexion/extension, lateral-bending, and twisting along the vertebra axis. In each joint set of the spine, a one-to-many mapping component first converts the input DOF to a rotation angle for each vertebra in the joint set.

Because the thoracic cage creates a closed chain with the spine and sternum, it tends to resist thoracic spine movement that would otherwise cause the individual ribs to rotate away or into each other during lateral bending. By carefully choosing the joint parameters for the ribs, the rotation of a rib can be set to be dependent on the amount of motion of its attached thoracic vertebra to maintain the overall shape of the rib cage. Intuitively, we should rotate the ribs in a direction opposite to that of the spine's rotation with the ribs always rotating less than spine. Therefore, we choose to define the axes of ribs to be opposite to those for vertebrae and define their rotation limits to be smaller. As shown in Figure 8, we can successfully approximate the non-rigidity of the whole rib cage (but not that of single ribs). For more accurate deformations of the rib cage, a custom joint component can be designed.

Figure 8: Flexible spine model

To summarize, our spine model is composed of three joint sets. Each joint set has three DOF of rotation for flexion/extension, lateral bending, and twist, making a total of nine DOF to control the entire spine and the rib cage. This is considerably less than the total number of articulations achievable in our model because we have implicitly built in the various dependencies. Although our model is probably still not as accurate as a real human spine, we can achieve fairly realistic spine configurations with a lightweight, intuitive manipulation interface. We believe that our spine model is an acceptable compromise between the need for accuracy and simplicity of control.

Shoulder Model: The shoulder is one of the most complex joints in the human body, making it a good test of the versatility of our joint component framework model. The shoulder comprises four articulations (the scapulo-humeral joint, the acromio-clavicular joint, the sterno-clavicular joint, and the sliding scapulo-thoracic joint) (Figure 3). In addition to the bone articulations, ligaments, cartilage and muscles also play important roles in the shoulder to create coupling behaviour and dependencies among the shoulder's joints.

We put the shoulder complex (four joints and three bones) into a shoulder joint set. Its joint set function has five DOF, of which three control the scapulo-humeral joint (flexion/extension, abduction/adduction and twist of the humerus bone) and two control the sterno-clavicular joint (vertical and horizontal rotation of the clavicle bone). The acromio-clavicular joint has zero controllable DOF because its movement can be fully determined by movements of the other two joints and the ellipsoidal surface constraint with the thoracic cage. The joint set function outputs three transformation matrices for the three joints respectively. Inside, the function has three parts for each of the scapulo-humeral, sterno-clavicular and acromio-clavicular joint articulations. In summary, the shoulder in our model is deterministically controlled by a complex component network with a very simple interface (only five DOFs). We can model the independent movements of the shoulder and upper arm as well as their coupling behaviours (Figure 9).

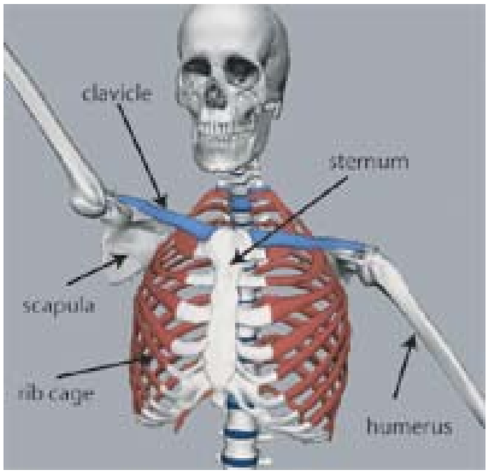

Figure 9: Shoulder kinematic model

CONCLUSION AND DISCUSSION

We have successfully used the joint component framework to create kinematic models of the human shoulder and spine (Figures 8 and 9). Despite the large number of articulations in these joint complexes, they can be manipulated with relatively few degrees of freedom (5 for the shoulder and 9 for the spine). This is achieved by building in many joint interdependencies in the component model. Using this framework, joint models can be iteratively refined by replacing or adding components to the kinematic model.

In future work , we would like to compare the kinematic data from our component model with previously defined models and experimental data in the literature. One current limitation of our model is that we do not account for any dynamic effects or dependencies on joint loading. It is possible to design new components to take into account load forces and torques, but a source for the loads would be required. These loads could be obtained through the use of dynamics simulation software or measured data.

We are currently using this framework to aid in determining subject-specific parameters that will allow the same joint set functions to be customized for different individuals. The seven joint components we defined do not represent a complete set for modelling all joints at every level of accuracy. We would like to convert the various joint models developed in the biomechanics community into our framework to make them accessible within a unifying toolbox. As the joint component model provides an XML-based format for data interchange, it can be used as a standard description specification for sharing kinematic joint models.

ACKNOWLEDGEMENT

This project is supported by the Honda Research Institute.

REFERENCES

Alias|wavefront (1999) *Maya 3-D Modelling Software*.
Buford, W.L. and Andersen, C. (2002) *Proceedings of IV World Congress Biomechanics 2002*.
Bull, A. and Amis, A. (1998) *Proc. Instn. Mech. Engrs.,* **212,** part H, 257-372.
Delp, S.L. and Loan, P. (1995) *Comput. Biol. Med.* **25**(1): 21-34.
Garner, B. and Pandy, M. (1999) *Computer Methods in Biomechanics and Biomedical Engineering* **2**, 107-124.
Girard, M. and Maciejewski, A. (1985) *Computer Graphics (Siggraph '85 Proceedings)* **19** 263-270.
Kapandji, I. (1982) *The Physiology of the Joints: Upper Limb,* 2nd Ed., **1**. Churchill Livingstone.
Kapandji, I. (1982) *The Physiology of the Joints: The Trunk and the Vertebral Column,* 2nd Ed, **3**. Churchill Livingstone
Maurel, W. and Thalmann, D. (2000) *Computers and Graphics* **24**(2), 203-218.
Shao, W. and Ng-Thow-Hing, V. (2003) *Proc. of ACM Siggraph Symposium on Interactive 3-D Graphics.*