Achieving Consistency in a Combined IK/FK Interface for a Seven Degree-of-Freedom Kinematic Chain

John McDonald, Rosalee Wolfe, Karen Alkoby, Jaceck Brzezinski, Roymieco Carter,
Mary Jo Davidson, Jacob Furst, Damien Hinkle, Bret Kroll, Glenn Lancaster,
Lori Smallwood, Jorge Toro, Nedjla Ougouag, Jerry Schnepp

School of CTI, DePaul University
{jmcdonald@cs.depaul.edu | asl@cs.depaul.edu}

ABSTRACT

Many applications in computer animation portray the motion of a human arm and torso. Often such applications can benefit from a combination of Inverse (IK) and Forward (FK) Kinematics controls to manipulate the arms of the model. The human arm is a kinematic chain with seven degrees of freedom. The previous analytic solution to this kinematic chain gives highly detailed IK controls, but problems arise when integrating it with FK controls. These problems impede the artistic process when creating expressive animations.

This work improves on the previous analytic solution to create a hybrid FK/IK control interface for manipulating the chain, and enables the recalculation of all the parameters necessary for the IK solution. Thus IK and FK controls can interact seamlessly to manipulate the arm. The torso is modeled as a separate kinematic chain, and is integrated with the arm linkages. User tests demonstrate the effectiveness and efficiency of the combined FK/IK control interface.

Keywords

Character Animation, Expressive Animation, Inverse Kinematics Forward Kinematics

1. Introduction

Many applications in computer animation portray the actions of a virtual human model. Such models are required to perform an incredible array of tasks, from manipulating objects in a virtual world to conveying emotional context and meaning. Much of the expressiveness of virtual actors is conveyed through the torso and arms. Therefore the control interface for the arms and torso is a key component of any system for animating a human model.

The flowing motions of the human arm and torso require a complex manipulation of a multitude of joints including the wrist, shoulder, elbow, collar, and spinal articulations [Van98]. To model these joints, computer graphics applications often use kinematic chains such as those found in robotic linkages, and rely on a variety of methods to control these joints to achieve the desired positions [Gir85] [Bad93] [Kog94] [Kon94] [Mur94].

Two general categories of controls exist for such kinematic chains, depending on the type of information they use as input:

1. *Forward Kinematics (FK) Controls.* These controls specify input data consisting of a collection of angles for the joints in the kinematic chain. The final orientation of any segment can be computed by multiplying the transformations in the joints.
2. *Inverse Kinematics (IK) Controls.* This class of control identifies an *end effector* in the chain, and a target at which the end effector should be positioned. Analytic or iterative IK solutions require the computation of the joint angles necessary to position the end effector at the target subject to various constraints, which restrict the space of possible solutions [Gir85].

The choice of control depends on the application. For example, in ergonomics simulations, the arms and torso need to be positioned so that the hands of the model can manipulate some object in space. These applications are most conveniently modeled using inverse kinematics [Bad93] [Tol00].

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

However, when positioning the body for expressive postures or narrative pantomime, the precise positioning of an end effector is not as important as the movements of the joints, which must look natural. In these kinds of applications, changes in the orientation of intermediate joints in the chain can dramatically alter the expression or emotion displayed by the figure. Animators must have complete control over all the degrees of freedom available in a model. Traditionally, for these types of applications forward kinematics has been the control of choice [Mae96] [Jon00].

2. Hybrid Interfaces

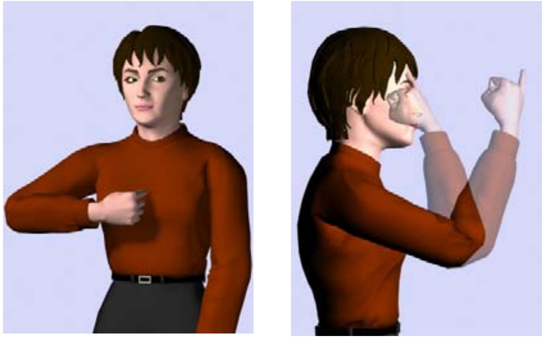
Many applications can benefit from a combination of these two types of controls. Character animation is a case in point. Consider a figure peering through a window with her fingertips touching the glass. If the character is curious or frightened, her posture will draw her elbows in towards her body, whereas if she is angry, her elbows will tend to flare out. Both postures place the character's fingertips on the glass in front of her. Therefore the animator would want IK control over the position of the fingertips, and direct control over the orientations of the intermediate joints.

To further illustrate the different kinds of controls required in expressive character animation, consider the following two examples:

1. Moving the arm so that a finger on one hand is placed in contact with the body, the face or the other hand, as in Figure 1A. An IK control easily achieves this contact.
2. Moving the hand away from the body in a natural arcing motion, as in Figure 1B. An FK control easily achieves this effect.

Forward kinematics has the advantage of being a straightforward process of building an object hierarchy. Rotating a single joint, as in Figure 1B, will move the joint and all of its children in a natural arcing motion. However, altering multiple joint angles causes the end effector to follow complicated arcs. While generally useful for positioning the body expressively, forward kinematics is cumbersome when used for placing an end effector at a specific location.

At the same time, these natural arcs may be precisely what the animator wants, but these arcs can be difficult to describe to an IK engine. If an IK rig animated the motion between the endpoints in Figure 1B, it would move the tip of the pinky along a straight line, and would introduce unwanted movement at the elbow. While one could describe this arc by a spline, the result is more cumbersome, and can involve more extraneous joint movement than simply letting FK have its way. On the other hand, when animating linear motions, as when the hands trace the rectangular shape of an object, an IK solution is far more effective.



A: IK to Target **B: FK to Move Joint**

Figure 1: IK and FK Applications

For these reasons, a system for expressive animation must provide an integrated set of FK and IK controls to support both kinds of movements. Therefore, it must conform to the following criteria:

1. Provides IK controls for the arms, which allow the specification of a target for any point on the end effector.
2. Provides the animator direct control over all redundant degrees of freedom, while preserving end effector/ target contact.
3. Provides FK controls for direct movement of each joint.
4. Provides the seamless integration of these FK and IK controls so that they may be used in any combination.

3. Available Techniques

Most comprehensive IK systems allow an animator to break out of IK control to manipulate the model with FK [Mae96]. However, the kind of control provided does not always conform to the criteria outlined in the last section.

Building a hybrid interface suitable for such applications requires the consideration of the available IK techniques in relationship to these four criteria. This includes investigating how well the IK solution provides for control over all the available redundant degrees of freedom and how well it integrates with FK controls.

Inverse Kinematics

IK techniques fall into two categories: analytic and iterative solutions. A great deal of work has been done in developing both types of solutions for the kinematic chains in the arm and torso [Kon94] [McD00] [McD02] [Tol00].

Iterative IK solutions work from a given configuration of the model and incrementally move the end effector towards a given target. Such systems will generally try to calculate a set of “best” angles for the redundant degrees of freedom in the linkage. However, the criteria from Section 2 require that these angles be under direct control of the animator, allowing interactive exploration of the entire collection of postures that fix the end effector [McD02].

Another problem arises because certain targets on the body or in space will correspond to singular configurations for the iterative solution. When the joints approach such configurations the model will behave in a chaotic manner. Artists are quite familiar with the unfortunate choices that an iterative IK solution can make, as the final configuration of the model is not uniquely determined by the position of the end effector [Tol00]. While implementations reduce these effects using range limiting and optimization techniques [Gir85] [Mur94] [Zha94], analytic solutions, when available, are preferable [McD02] [Tol00].

Several analytic solutions have been investigated for different types of kinematic chains. The first analytic solutions were developed for robotic linkages [Mur94] and do not model the complete expressiveness of the human arm and torso.

Most of the analytic solutions for the human arm in modern software work with linkages of just two bones, leaving the third segment, the hand, to be manipulated by FK or to be specified as a global orientation in space [Kon94] [Tol00]. This type of linkage has one redundant degree of freedom in the two joints of the linkage itself. This degree of freedom allows the user to raise and lower the model's elbow about the axis through the shoulder and wrist.

Such analytic solutions do not, however, take the full linkage of the human arm including the wrist into account in the solution itself. The full linkage has four redundant degrees of freedom. This case was considered in [McD02] where an analytic IK solution was presented for the entire seven degrees of freedom in the human arm.

[McD02] gives the animator direct control over the four redundant degrees of freedom in the linkage. The wrist orientation and the elbow elevation can both be altered while preserving end effector/target contact. Thus the solution satisfies three of the four criteria for an IK control outlined in Section 2.

Unfortunately problems arise when integrating this method with forward kinematic controls for the arm. Such integration was not considered in [McD02]. Forward kinematic manipulation of the shoulder, elbow, or wrist will cause the IK parameters to change. The system must recompute these parameters if the interface for manipulating the model is to work seamlessly. If it does not, and the user subsequently manipulates the IK controls, the model's arm will jump discontinuously as the recorded IK parameters reassert themselves.

For example, consider choosing the tip of the index finger with a bent wrist. Choose a target for the fingertip out in front of the body. Then straighten the elbow with an FK control. This moves the fingertip, and so the system records its new position. Then reapply the method from [McD02] using this new position as a target. Since the solution does not need to move the fingertip, it should not move the model at all. Instead, the position of the elbow jumps. Figure 2 shows the disparity between the resulting position and the correct one.

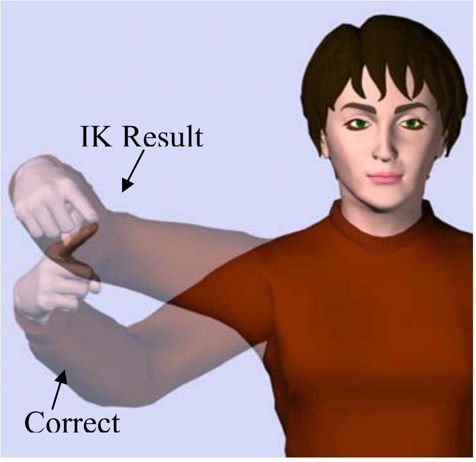


Figure 2: Discontinuity Caused by [McD02]

Such recalculations must always be considered when integrating an IK solution with FK. An advantage of iterative solutions is that their only input is the end effector's position, which can be easily recalculated from the model's joints. But, as described above, the unpredictable nature of iterative solutions makes them unsuitable for expressive animation.

The analytic solutions such as those outlined in [Tol00] and those found in many commercial packages must also recalculate all of the IK parameters if they are to be integrated with FK, and many do. However, they only work with the shorter two-segment IK chains, which do not extend to include the wrist's rotation in the IK solution itself.

4. Extending the IK Solution

To see why problems arise in the current techniques, we must carefully investigate the method presented in [McD02].

The Previous Analytic IK Solution for the Arm

The human arm has three main joints: the wrist, elbow and shoulder. Together, these joints have a total of seven degrees of freedom. A discussion of these joints can be found in [McD02], where the following solution to the kinematic chain was developed. The algorithm requires the following input:

1. The position A, on the model's hand, called an *articulator*, to be used for targeting

2. The target point T in space or on the body
3. The local orientation of the model's wrist including the radial twist of the forearm
4. The elevation angle δ for the elbow, which raises and lowers the elbow while keeping the chosen articulator fixed in space.

The solution then computes the remaining angles necessary to position the arm so as to place the articulator A at the chosen target T. Note that if the articulator is chosen at the wrist, then the system degenerates to the previous two-segment cases such as in [Tol00] and those implemented in several commercial packages.

The method assumes that a coordinate system has been chosen with its origin at the model's shoulder, and with the z-coordinate axis corresponding to the primary vertical axis of the body but pointed downwards, i.e. running parallel to the line from the neck through the hips. The y-axis will protrude straight out of the body perpendicular to the plane formed by the two shoulders and the hips. See Figure 3.

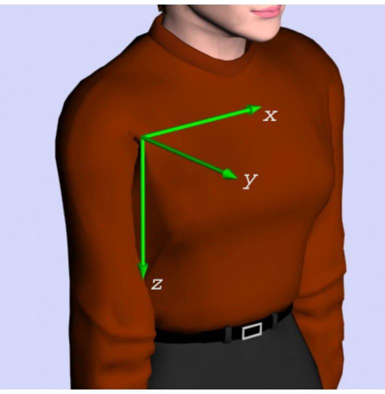


Figure 3: The Shoulder Coordinate System

With this setup, the method calculates the transforms for both the elbow and the shoulder. Let S be the position of the shoulder in space and let $d = \text{dist}(S, T)$ be the distance from the shoulder to the desired target. The solution is calculated in four steps:

1. Calculate the bend angle γ of the elbow, as shown in Figure 4. With this, the articulator will lie at a distance of d from the shoulder.
2. Position the articulator on the y-axis in front of the body by calculating two angles in a spherical coordinate system ϕ_s and θ_s for the shoulder, as shown in Figure 4.
3. Rotate the shoulder by the elevation angle δ about the y-axis, which is currently the axis through the shoulder and the articulator.
4. Use the spherical coordinates of the articulator's target to calculate two more spherical angles ϕ_a and θ_a for the shoulder, which will rotate the articulator from the y-axis to the chosen target.

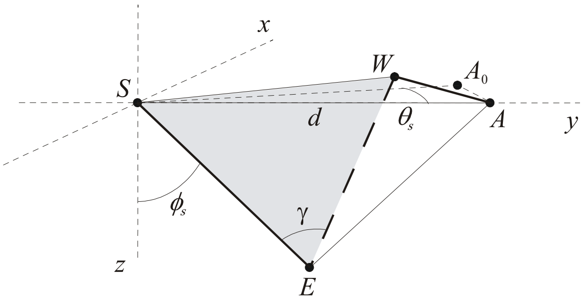


Figure 4: The Spherical Angles of the Upper Arm in the Default Position

Thus, the method decomposes the shoulder rotation into five angles, ϕ_s, θ_s, δ, ϕ_a, and θ_a about the axes x, z, y, x and z respectively. The final transformation of the shoulder is computed as the product

$$M_s = R_D R_{\phi_s} R_{\theta_s} R_\delta R_{\phi_a} R_{\theta_a}$$

where R_D is the default rotation of the shoulder. The benefits of this parameterization were detailed in [McD02].

Effects of FK Controls

The method presented in [McD02] alone is sufficient if the only interface given to the user is:

1. The IK controls for the position of the articulator
2. The elbow elevation angle
3. The local orientation of the wrist.

It is important to note that wrist orientation, while under direct control, results in a call to the IK system so that the articulator position remains fixed as demonstrated in the left image in Figure 6 of Appendix A. This is distinctly different from the desired FK control for the wrist, which would move the articulator as in the left image of Figure 7.

In addition to these IK controls, we wish to allow the user to rotate the elbow, wrist and shoulder through FK. We will work with the underlying IK parameters as our basis since the IK solution will still be the primary means for controlling the model. Any forward kinematic moves made later will be recast in terms of the IK parameters so that if the user then moves back to the IK controls, the model will not jump as the IK system reasserts itself. This is the step that is missing from the previous method and which will satisfy the final criterion from section two.

Consider the effect of moving the model's arm with the above analytic IK method, followed by an adjustment of one of the joints with an FK control. *All* of the IK parameters will change:

1. The position of the articulator will change. This is not a problem since we can simply recompute

the articulator's position from the model and set the IK target to the new position.

2. The elbow bend angle and wrist orientation might change as the result of an FK move. Again, we can read their new values straight out of the FK data.
3. The elevation angle δ will change. This can *not* be directly read from the FK data.

Certainly, the elevation angle will change if the shoulder is rotated, but even if the shoulder does not move, as when the wrist is flexed, δ will change. Recall that this analytic solution decomposes the shoulder into five angles: two pairs of spherical angles and the elevation angle. There are many subtle interactions that can occur in the shoulder to change these angles, including δ.

Suppose that the user has chosen an articulator on the tip of the index finger, and uses the IK solution to place the tip of the finger at a point out in space in front of the body. What happens if the user subsequently flexes the model's wrist via FK? The articulator target *A* will change position. Call the new target *A'*. This causes the spherical angles of the target, ϕ_a and θ_a, to change to match the new location.

Note that flexing the wrist does not actually change the orientation of the shoulder, and so the total shoulder transformation M_s remains constant. Since the default rotation of the shoulder has not changed, one or more of the other three angles ϕ_s, θ_s and δ, have changed to compensate. In fact, most often the change in ϕ_a and θ_a will be compensated by a change in all three of the other angles.

In order to integrate FK and IK controls seamlessly for this method, we need to recompute all five of the angles in the shoulder transformation's decomposition, ϕ_s, θ_s, δ, ϕ_a, and θ_a. All five must be recalculated each time the user moves the model using the FK control. This will complete the recalculation of the IK parameters in this analytic method and will correct the jumping problem seen in Figure 2.

Extending the Method: Recomputing the Angles

To recompute the five angles in the IK solution, we need to take a close look at the decomposition of the shoulder transformation. Suppose that the user has made a sequence of FK changes to the arm's joint rotations. These FK moves on the model result in a new local transformation M' at the shoulder.

We know from the development of this analytic IK solution that whatever this matrix is, it can be decomposed into a product of six rotations:

$$M' = R_D R_{\phi_s'} R_{\theta_s'} R_{\delta'} R_{\phi_a'} R_{\theta_a'}$$

The first transformation in this product is the default rotation for the shoulder, which is a constant and is therefore known. It remains to recompute the other five angles in the decomposition.

On the right side of the chain, we have the rotations for the two spherical angles corresponding to the new articulator position *A'* = (*x'*, *y'*, *z'*). Therefore, these angles may be calculated from the coordinates of *A'*.

$$\theta_a' = \arctan\left(\frac{-x'}{y'}\right)$$

$$\phi_a' = \frac{\pi}{2} - \arccos\left(\frac{z'}{\sqrt{(x')^2 + (y')^2 + (z')^2}}\right)$$

The differences in these formulas from their normal spherical coordinate presentation are due to the fact that the spherical coordinates of the articulator's target are measured from the *y*-axis.

Next, the first stage of the original analytic solution can be run to find the values of ϕ_s' and θ_s'. See Figure 4. Determining these angles is accomplished by first calculating the projection of *A* to the plane *SEW*. Call this projection A_0. Then ϕ_s' and θ_s' can be calculated as

$$\phi_s' = \pi/2 - \arccos\left(\frac{SE \bullet SA_0}{|SE||SA_0|}\right)$$

$$\theta_s' = \arccos\left(\frac{SA \bullet SA_0}{|SA||SA_0|}\right)$$

These calculations are most conveniently accomplished in the elbow's coordinate system, in which *A'* and *W* have a known representation in terms of the initial input data.

Finally, we come to the elevation angle δ'. While we could try to use elementary trigonometry to calculate the new elevation angle, the decomposition of *M'* admits a more elegant solution with fewer special cases. Since each of the transformations in the decomposition is a rotation and is therefore invertible, the rotation matrix for the elevation angle can be calculated from the decomposition

$$M' = R_D R_{\phi_s'} R_{\theta_s'} R_{\delta'} R_{\phi_a'} R_{\theta_a'}$$

by multiplying both sides of the equation by the inverses of the rotations for ϕ_s, θ_s, ϕ_a, θ_a and the default rotation. Thus,

$$R_{\delta'} = \left(R_{\theta_s'}\right)^{-1}\left(R_{\phi_s'}\right)^{-1}\left(R_D\right)^{-1} M' \left(R_{\theta_a'}\right)^{-1}\left(R_{\phi_a'}\right)^{-1}$$

This gives us the transformation as a matrix. To calculate the elevation angle for the elbow, we use a surprising fact arising from the construction of the analytic solution. $R_{\delta'}$ is a rotation about the *y*-axis, and will therefore be of the form

$$R_{\delta'} = \begin{pmatrix} \cos\delta' & 0 & \sin\delta' \\ 0 & 1 & 0 \\ -\sin\delta' & 0 & \cos\delta' \end{pmatrix}$$

Therefore, we can find δ' by taking

$$\delta' = \operatorname{sgn}\left(\left(R_{\delta'}\right)_{1,3}\right)\arccos\left(\left(R_{\delta'}\right)_{1,1}\right)$$

This completes the recalculation of the IK parameters after a forward kinematic move. Applying this recalculation finishes the computation of the five IK parameters.

5. The Integration with the Torso

The completion of the control interface requires the integration of the arm's IK chain with the linkages for the torso and back. The FK and IK controls should continue to work seamlessly as the animator manipulates the model's torso.

The interface uses a simplified model of the torso consisting of a sequence of three evenly spaced joints starting at the hips, and having three degrees of freedom, corresponding to flexion, abduction and radial twist about the local tangent to the spinal column.

The sterno-clavicular joint in the collar is modeled as a rotational joint at the top of the spine and articulating with the shoulder joint. It has three degrees of freedom, a rotation that moves the shoulder forwards and backwards, a rotation that moves the shoulder up and down, and a small amount of radial twist. Though simplified, this model is capable of representing many movements of the shoulder and torso. This method is also compatible with more realistic, physically based models, as described in [Mau00] and [Sha03].

The first step of the new IK solution calculates the coordinates of the target relative to this hierarcy, i.e. relative to the arm's rest orientation, which has been rotated by the torso and collar joints. Figure 5 displays two examples of the added expressivity enabled by integrating the FK/IK system with the torso.

6. The Interface

We implemented the FK and IK controls enabled by these techniques in a custom software package allowing animators to control a human model interactively. The integrated FK/IK interface adds FK controls for the wrist, elbow, shoulder and torso to the IK controls. Figures 6 and 7 in Appendix A show the effects of the wrist and elbow controls in the combined FK/IK interface, emphasizing the differences in effect from the FK and IK controls for these joints. The remaining shoulder and torso controls are slider controls accounting for all the degrees of freedom.

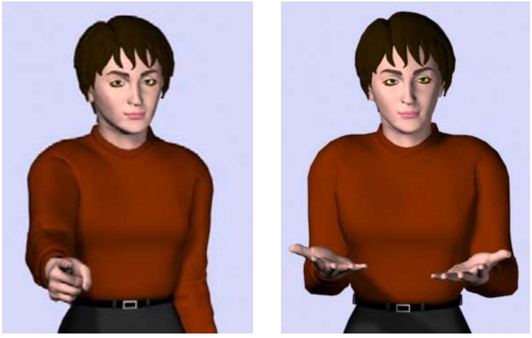


Figure 5: Torso Movement

7. Testing the Interface

A usability test evaluatd the effectiveness of the integrated FK/IK interface for positioning a model's arms. The test compared the integrated interface with the controls found in the most recent version of a widely used, commercially-available animation package, which allowed the subjects to use either FK, a traditional IK solution, or a combination of both to move the model.

The test participants had previous experience with the commercial package ranging from six months to three years. While three of the participants also had substantial experience in using the integrated FK/IK interface, three of them had minimal prior exposure. See Table 1 in Appendix B.

The participants created two versions of three American Sign Language (ASL) signs: FOOD, IDEA and CLOTHES. Each sign contained successively more complicated motion. The first, FOOD, required a small arcing of the forearm and a localized oscillation of the wrist along a single axis. The second, IDEA, required a twisting and arcing motion of the forearm. The last sign, CLOTHES, required a circling motion of the wrists as well as a twisting and arcing motion of the forearms. For a background on ASL animation see [McD00] and [Wol99]. Each participant created one version of the sign using the commercial package, and another using the integrated FK/IK interface, yielding a 3x2 experimental design.

As a guide for creating the animations, participants received videotaped demonstrations of each sign recorded from side and front views. For each sign, participants received a model whose fingers were already in the correct position. They began the task of creating the animation from that point. Only the time required to position the arms was recorded.

Each participant completed the two versions of a sign, and then proceeded to complete two versions of the next sign. To control for transfer of learning, the order in which they used the commercial software and the integrated FK/IK interface was randomized.

After the participants completed the animations, a team of two animators familiar with ASL critiqued the results jointly. This team had access to both the completed animations and the reference video footage. They examined the accuracy of the arm placement. The reviewers were unaware of which software was used to create each animation.

Table 2 and Table 3 of Appendix B display the results. For the simplest sign, FOOD, the average completion times were the same. However, for the more complex signs, participants required less time when using the integrated FK/IK interface. For the sign IDEA, participants using the integrated interface required only 57 percent of the time that they needed to complete the same sign using the commercial software. This percentage dropped even further with the sign CLOTHES. When using the integrated FK/IK interface, participants required less than 50 percent of the time they needed when using the commercial software.

While the completion times examined the efficiency of the new interface, the reviewer's critique evaluated its effectiveness. Table 3 in Appendix B shows the results. The preferred versions of each animation are listed for each test participant. When both versions were judged to be of equal quality, the word "tie" appears.

For all three signs, the new integrated interface produced better results, but it is particularly striking for the most complex sign, CLOTHES. For all six participants, the preferred animation was the one created with the new FK/IK interface. Since five of the six participants were more familiar with the commercial package, this clearly demonstrates the advantages of the new approach.

8. Conclusion and Next Steps

The techniques in this paper extend the results from [McD02] to build an integrated FK/IK control interface for a three segment kinematic chain with seven degrees of freedom, such as the human arm. These controls allow animators to create expressive motions in less time than with previous FK/IK techniques.

Next steps include the incorporation of a system of joint correlations similar to those described in [McD00] and [Sha03] to help animators coordinate the movements of various segments of the arms and torso. This would increase the efficiency of the interface. Also, while the controls for the arm are integrated into a simple model for the spine and shoulders, a more realistic model of the torso, collarbone and neck as in [Mau00] would further expand the expressiveness of the model.

9. Acknowledgements

The authors wish to express their thanks to Tahseen Basheeruddin, Ryan Burnett, Cynthia Dwyer, Bret Kroll, Jerry Schnepp and Prabhakar Srinivasan for their participation in the user test.

10. References

[Bad93] Badler N, Phillips C, Webber B. Simulating Humans: Computer Graphics Animation and Control. Oxford University Press. New York, NY. 1993.

[Gir85] Girard M, Maciejewski A. Computational Modeling for the Computer Animation of Legged Figures. Computer Graphics. 19(3), July 1985, Pages 253-270.

[Jon00] Jones A, Bonney S. 3Dstudio Max 3 Professional Animation. New Riders, Indianapolis, 2000.

[Kog94] Koga Y, Kondo K, Kuffner J, and Latombe J. "Planning Motions with Intentions," Proc., SIGGRAPH'94, Orlando, FL, July 24-29, 1994. Pages 395-407.

[Kon94] Kondo K. Inverse Kinematics of a Human Arm. Technical Report STAN-CSTR-94-1508, Dept. of Computer Science, Stanford University.

[Mae96] Maestri, G. Digital Character Animation. New Riders, Indianapolis, 1996.

[Mau00] Maurel, W, Thalmann, D. Human Shoulder Modeling Including Scapulo-Thoracic Constraint and Joint Sinus Cones. Computer & Graphics, New York, Vol 24, No 2, p. 203-218, 2000.

[McD00] McDonald J, Toro J, et. al. An Improved Articulated Model of the Human Hand. Proceedings of the 8th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media. 2000. Pages 306 – 313.

[McD02] McDonald J, et al, A Direct Method for Positioning the Arms of a Human Model. Graphics Interface 2002, Proceedings, 99-106, 2002.

[Mur94] Murray R, Li Z, and Sastry S. A Mathematical Introduction to Robotic Manipulation. CRC Press, Inc., 1994.

[Sco03] Scott, R. Sparking Life, Notes on the Performance Capture Sessions for The Lord of The Rings: The Two Towers. Computer Graphics, Vol 37, No 4, 2003, pp. 17-21.

[Sha03] Shao, W, Ng-Thow-Hing, V. A General Joint Component Framework for Realistic Articulation in Human Characters. Proceedings of the 2003 symposium on Interactive 3D graphics, 2003, pp 11-18.

[Tol00] Tolani D, Goswami A, and Badler N. Realtime inverse kinematics techniques for anthropomorphic limbs. Graphical Models 62 (5), September 2000, Pages 353-388.

[Van98] Van De Graaff, K, Human Anatomy 5th Edition. WEB McGraw-Hill. Boston, MA. 1998.

[Wol99] Wolfe R, et al. An Interface for Transcribing American Sign Language. SIGGRAPH 99 Sketches, August 11, 1999, Page 229.

[Zha94] Zhao J, Badler N, Inverse kinematics positioning using nonlinear programming for highly articulated figures, *ACM Transactions on Graphics*, Vol 13, No 4, October 1994, pp. 313-336.

Appendix A: Interface Examples

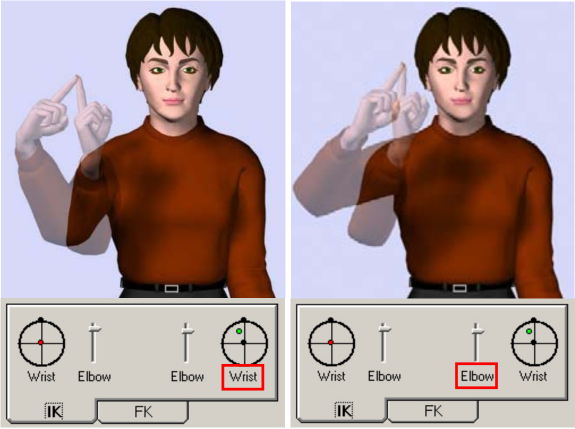


Figure 6: The IK Interface

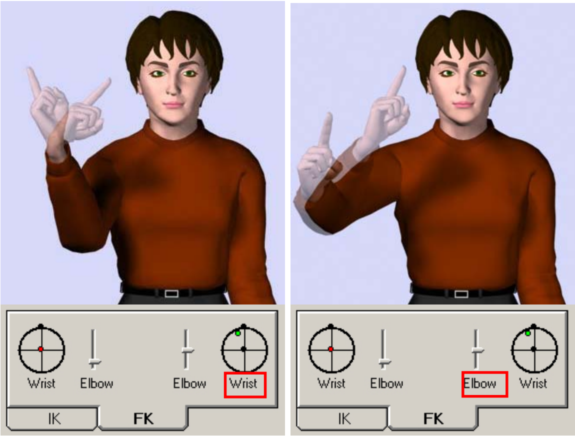


Figure 7: The FK Interface

Appendix B: Data from User Test

Participant	Commercial Package	Integrated FK/IK control
1	18 Months	1 Hour
2	36 Months	18 Months
3	36 Months	1 Hour
4	12 Months	12 Months
5	6 Months	2.5 Hours
6	36 Months	20 Hours

Table 1: Previous Experience of Test Participants

Sign	Commercial package	Integrated FK/IK control
FOOD	20.2	20.7
IDEA	21.8	12.5
CLOTHES	42.2	21.0

Table 2: Average Completion Times.

Participant	FOOD	IDEA	CLOTHES
1	Commercial	Commercial	FK/IK
2	FK/IK	Commercial	FK/IK
3	FK/IK	Tie	FK/IK
4	FK/IK	FK/IK	FK/IK
5	Tie	FK/IK	FK/IK
6	FK/IK	FK/IK	FK/IK

Table 3: Results of Animation Critique, Software Resulting in Preferred Animation