Biomechanical Modelling and Animating Human Hand Movements

O.A. Kuchar and J.N. Scrimger
School of Computer Science, Technical University of Nova Scotia, Halifax, N.S., Canada, B3J 2X4

Abstract

A different technique for animating human movement is presented and applied to animating the movement of the fingers and thumb. This paper presents the technique of muscle control -- simulating the effects of the tendons and muscles that affect movement of the body. The hand model created for simulating the movement of the fingers and thumb includes "puppet strings" that behave similarly to their original counterparts -- the motor muscles and tendons found in the hand.

In applying the new approach of muscle control movement, the hand was easier to manipulate and animate. In graphics, the simulation of movement is part of a larger project to simulate living creatures, including humans. The technique presented in this paper produced results for the animation of the fingers and thumb. Since the hand, which is an intricate part of the body, can be animated using this technique then in theory, it can be applied to the entire body. Thus, the goal of the research was proven to be a step in the right direction of animating human motion.

Keywords: computer animation, human hand, biomechanical model, human movement visualization, muscles

1. Introduction

Improving motion realism requires many degrees of freedom in the body linkages that increase the difficulty of control. Models can either be simplified or complex. Models can be simplified to ease in control, thereby risking unrealistic movements. Models can also have complicated control with a complex model and hope the resulting motions appear more natural. Even though algorithms have addressed greater animation power with kinematics, dynamics, inverse kinematics, available torque, locomotion, gestural and directional control, the human models themselves tended to be rather simplified versions of real human flexibility. A summary of different animation techniques is discussed in Tost et al. [1988].

Increased realism in the human models would demand more accurate and complicated motion control. Now that the control regimes are improving, it is time to return to the human models and re-evaluate their structures to take advantage of algorithmic improvements. First, human models should be more complex, as depicted in Monheit et al. [1991] which describes a human model with a flexible torso and spine. Secondly, the human model should contain more segments (bones) that can be manipulated. Thirdly, algorithms should be developed such that the user interface is simple in creating motion. The algorithms that would be needed are algorithms that calculate motion and position of limbs, while the user only inputs the beginning position, end position, and environmental factors. Finally, several added features should be implemented in the human model, such as muscle and tissue. Overall, the human models must be re-examined and new, more complicated models need to be formulated. The research presented in this paper attempts to create a more complex human hand model.

2. Movement Creation: The Mathematical Model

Several muscles were modelled according to their original counterparts, as depicted in Table 1. A full description of the tendons and their functions is described in Kuchar [1996].

This section gives a flavour for the underlying mathematical model applied to the muscle effect on the fingers. Two examples of the mathematical model are provided: the Flexor Digitorum Profundus (FDP) tendon; and the second dorsal interosseous.

Abbreviation	Name	Function
FDS	Flexor Digitorum Superficialis	- flexes the fingers
FDP	Flexor Digitorum Profundus	- flexes the fingers
	Flexor Digiti Minimi	- located in the pinky - flexion of the metacarpophalangeal joint
FPL	Flexor Pollicis Longus	- located in the thumb - flexes at the interphalangeal joint
FPB	Flexor Pollicis Brevis	- located in the thumb - flexes, adducts
EDC	Extensor Digitorum	- extends the fingers
	Extensor Indicis	- extends the index
	Extensor Digiti Minimi	- extends the pinky
EPL	Extensor Pollicis Longus	- located in the thumb - extends the distal phalanx
EPB	Extensor Pollicis Brevis	- located in the thumb - extends the proximal phalanx - abducts the hand
AP	Adductor Pollicis	- adducts the thumb
ADM	Abductor Digiti Minimi	- abducts the pinky - flexes the proximal phalanx
APB	Abductor Pollicis Brevis	- adduction of the thumb - flexes the metacarpophalangeal joint
APL	Abductor Pollicis Longus	- abducts the thumb - flexes the metacarpal bone

Table 1. Summary of Modelled Muscles and their Functions.

2.1 The Basic Finger Model

The finger model is shown in Figure 1. The finger is reduced to a line structure with four segments and three joints that correspond to the bones and joints of a human finger. A "puppet string" is used to represent a tendon. The puppet string is attached to the bone segment similar to the original tendon insertion. The puppet string is the only part of the model that the user manipulates with the sliders in the interface.

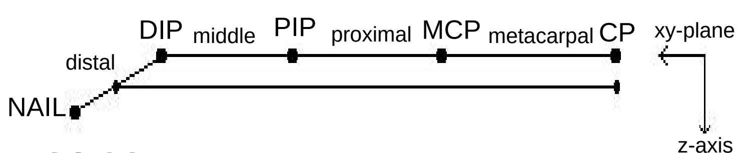

Figure 1. The Finger Model.

2.2 The FDP Mathematical Model

The FDP is primarily a flexor of the distal interphalangeal joint (DIP) joint but flexion of this joint is soon followed by flexion of the proximal interphalangeal joint (PIP) joint, which has no special extensor to antagonize this action. The FDP tendon is modelled by a line that connects the base of the metacarpal bone to the middle of the distal phalanx. For computational reasons, the middle of the distal phalanx was chosen instead of the base of the distal phalanx.

Since a tendon affects all joints that it passes, the FDP affects the DIP, PIP, and metacarpophalangeal (MCP) joints. When modelling the effects of the FDP tendon on the joints, three cases occur:

1. The DIP joint is flexed by the pull of the puppet string. If this joint is flexed past the maximum constraint limit of the DIP angle specified in the input file for the finger, then the DIP joint is placed at the maximum angle and flexion of the PIP joint begins.
2. The PIP joint is flexed by the pull of the puppet string. In this case, the DIP joint has been fully flexed. If the PIP joint is flexed past the maximum limit of the angle specified in the input file for the finger, then this joint is placed at the maximum constraint angle and flexion of the MCP joint begins.
3. The MCP joint is flexed by the pull of the puppet string. In this case, the IP joints have been fully flexed. If the MCP joint is flexed past the maximum limit of the angle specified in the input file for the finger, then this joint is placed at the maximum angle and the finger is fully flexed.

Each case mentioned above is described by the accompanying mathematical model.

CASE 1: Flexion at the DIP Joint

The algorithm that calculates the angle at the DIP joint, indicated as φ in Figure 2, is now presented.

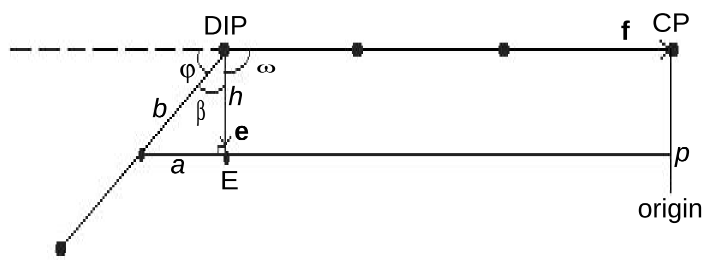

Figure 2. DIP joint is flexed.

The following constraints exist in this algorithm:

1. p is the slider value passed from the interface to the FDP procedure. p is the current length of the puppet string.
2. b is one-half the length of the distal phalanx.

The final algorithm to calculate φ is:

$$\varphi = 180^0 - [\arcsin((p - (CP - DIP)) / b) + (e \cdot f)] = 180^0 - (\beta + \omega)$$

1. **Calculate** β.
   Calculate a as the remainder of the puppet string when the lengths of the middle, proximal, and metacarpal bones are subtracted from p. Thus, β is calculated as the arcsine of (a / b).
2. **Calculate** ω.
   First, point E needs to be determined. The (x, y) values of E are obtained using the (x, y) coordinates of the DIP joint; however, the z

coordinate needs to be calculated. Thus, using the Pythagorean theory, the variables *a* and *b* can be used to calculate *h*. The z coordinate of E is (-*h*). Secondly, a vector **e** is constructed from the points E and the DIP joint. Thirdly, a vector **f** is constructed to the base of the metacarpal bone from the current flexing joint. Finally, ω can be calculated using the dot product of the two vectors **e** and **f**.

3. **Calculate** φ.
   To calculate φ, the angle at the DIP joint, subtract the sum of β and ω from 180^0. This is the flexion of the finger at the DIP joint. If φ is flexed past the maximum constraint limit of the angle specified in the input file for the finger, then φ is placed at the maximum angle and flexion of the PIP joint begins.

CASE 2: Flexion at the PIP Joint

The algorithm that calculates the angle at the PIP joint, indicated as γ in Figure 3, is now presented.

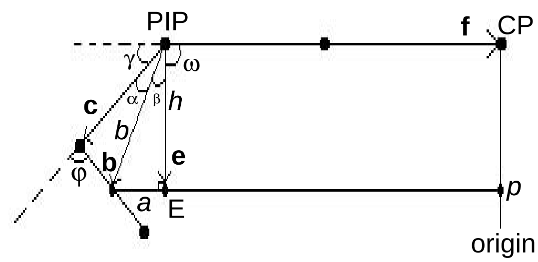

Figure 3. PIP joint is flexed.

The following constraints exist in this algorithm:

1. *p* is the slider value passed from the interface to the FDP procedure. *p* is the current length of the puppet string.
2. *b* is the constant length between the midpoint of the distal phalanx to the PIP joint.
3. φ is the maximum angle of the DIP joint from case 1.

The algorithm to calculate γ is:

$$\gamma = 180^0 - [\arcsin((p - (CP - PIP)) / b) + (e \cdot f) + (b \cdot c)] = 180^0 - (\beta + \omega + \alpha)$$

1. **Calculate** β.
   Calculate *a* as the remainder of the puppet string when the lengths of the proximal and metacarpal bones are subtracted from *p*. Thus, β is calculated as the arcsine of (*a* / *b*).
2. **Calculate** ω.
   First, point E needs to be determined. The (x, y) values of E are obtained using the (x, y) coordinates of the PIP joint; however, the z coordinate needs to be calculated. Thus, using the Pythagorean theory, the variables *a* and *b* can be used to calculate *h*. The z coordinate of E is (-*h*). Secondly, a vector **e** is constructed from the points E and the PIP joint. Thirdly, a vector **f** is constructed to the base of the metacarpal bone from the current flexing joint. Finally, ω can be calculated using the dot product of the two vectors **e** and **f**.

3. **Calculate** α.

   First determine two vectors **b** and **c**. Vector **b** is constructed from the PIP joint to the midpoint of the distal phalanx. Vector **c** is constructed to the DIP joint from the PIP joint. Thus α is calculated using the dot product of **b** and **c**.

4. **Calculate** γ.

   To calculate γ, the angle at the PIP joint, subtract the sum of α, β, and ω from 180^0. This is the flexion of the finger at the PIP joint. If γ is flexed past the maximum constraint limit of the angle specified in the input file for the finger, then γ is placed at the maximum angle and flexion of the MCP joint begins.

CASE 3: Flexion at the MCP Joint

The algorithm that calculates the angle at the MCP joint, indicated as θ in Figure 4, is now presented.

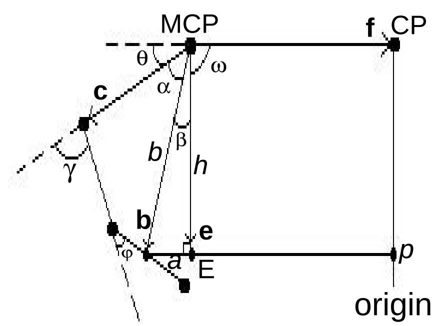

Figure 4. MCP joint is flexed.

The following constraints exist in this algorithm:

1. *p* is the slider value passed from the interface to the FDP procedure. *p* is the current length of the puppet string.
2. *b* is the constant length between the midpoint of the distal phalanx to the MCP joint.
3. φ is the maximum angle of the DIP joint from case 1.
4. γ is the maximum angle of the PIP joint from case 2.

The algorithm to calculate θ is:

$$\theta = 180^0 - [\arcsin((p - (CP - MCP)) / b) + (e \cdot f) + (b \cdot c)] = 180^0 - (\beta + \omega + \alpha)$$

1. **Calculate** β.

   Calculate *a* as the remainder of the puppet string when the length of the metacarpal bone is subtracted from *p*. Thus, β is calculated as the arcsine of (a / b).

2. **Calculate** ω.

   First, point E needs to be determined. The (x, y) values of E are obtained using the (x, y) coordinates of the MCP joint; however, the z coordinate needs to be calculated. Thus, using the Pythagorean theory, the variables *a* and *b* can be used to calculate *h*. The z coordinate of E is (-*h*). Secondly, a vector **e** is constructed from the points E and the MCP joint. Thirdly, a vector **f** is constructed to the base of the metacarpal bone from the current flexing joint. Finally, ω can be calculated using the dot product of the two vectors **e** and **f**.

3. **Calculate** α.
   First determine two vectors **b** and **c**. Vector **b** is constructed from the MCP joint to the midpoint of the distal phalanx. Vector **c** is constructed to the PIP joint from the MCP joint. Thus α is calculated using the dot product of **b** and **c**.
4. **Calculate** θ.
   To calculate θ, the angle at the MCP joint, subtract the sum of α, β, and ω from 180^0. This is the flexion of the finger at the MCP joint. If θ is flexed past the maximum constraint limit of the angle specified in the input file for the finger, then θ is placed at the maximum angle and the finger is fully flexed.

2.2 The Dorsal Interosseous Mathematical Model

The dorsal interossei occupy the spaces between the metacarpal bones and the main effect of the dorsal interossei is the abduction of the fingers. The mathematical model involves two fingers, as depicted in Figure 5. The mathematical model is explained below for the second dorsal interosseous.

The following constraints exist in this algorithm:

1. p is the slider value passed from the interface to the dorsal procedure. p is the current length of the puppet string.
2. C is the midpoint of the metacarpal bone on finger 2.
3. A is the midpoint of the proximal phalanx on finger 1.
4. a is one-half the length of the proximal bone on finger 1.
5. b is the length from the MCP joint on finger 1 to point C.

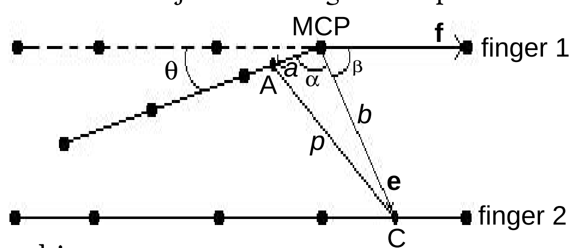

Figure 5. The second dorsal interosseous.

The algorithm to calculate θ is:

$$\theta = 180^0 - [(180^0 / \pi)\arccos((p^2 - a^2 - b^2) / (-2ab)) + (e \cdot f)] = 180^0 - (\alpha + \beta)$$

1. **Calculate** α.
   Applying the cosine rule of a triangle, α can be calculated using p, a, and b.
2. **Calculate** β.
   A vector **e** is constructed from the MCP joint to point C. A vector **f** is constructed to the base of the metacarpal bone from the MCP joint. β is then calculated using the dot product of **e** and **f**.
3. **Calculate** θ.
   To calculate θ, subtract α and β from 180^o. If the MCP joint is abducted past the maximum limit of the angle specified in the input

file for the finger, then this joint is placed at the maximum angle and the finger is fully abducted.

3. Assumptions

In modelling the human hand, analysis of constraints is essential to achieve the following:

1. avoid unrealistic motions during hand animation;
2. reduce the search space in model-based analysis of hand images.

Inevitably, a trade-off arises between the degree on constraints contained in a model and its resultant performance; that is, a lack of constraints leads to a useless model, whereas too many of them require complex procedures necessitating expensive computation time. The hand model described in this paper attempts to effectively balance these considerations.

Normally, movements of the finger joints are coordinated by constraints that make some configurations impossible. After analyzing finger movements, some prominent constraints were incorporated into the hand model, broadly classified as follows (based on movement types and the involved joints):

1. joint angle limits and movement types;
2. flexion of the IP joints;
3. flexion of the MCP joints;
4. adduction and abduction of the MCP joints.

3.1 Constraints on Joint Angle Limits and Movement Types

Possible movements of the MCP joint of the fingers are only flexion, extension or side movements, and that of the PIP and DIP joints is only flexion and extension in the same direction. Although the allowable ranges of joint angles vary slightly from person to person, they do fall into general ranges.

The range of joint angles varies based on passive and active movements. The former movement is externally forced, whereas the latter is activated by tensors and muscles of the hand without external interaction. Joints generally have a greater range for passive movement. Only active hand motions are included in this model, since inclusion of external forces exceeds the scope of this study.

3.2 Constraints on Adduction and Abduction of the MCP Joint

The adduction and abduction movements of the fingers are referenced by the axis of the hand running through the third metacarpal bone and finger. In such movement, the middle finger does not move appreciably without intentional forcing. In their naturally open position, the fingers can freely carry out adduction or abduction. However, when clenched into a fist with the DIP joints extended, the axes of the two distal phalanges of the fingers and the axis of the thumb converge to a point. That is, the abduction and adduction angles continuously decrease as the flexion angles of the MCP joints increase. Such behavior might initially seem to prevent collisions between the fingers, but closer observation reveals that this convergence takes place independently of collisions with other fingers.

3.3 Multitasking

As mentioned previously, finger movement is complicated since more than one muscle can be participating in a movement to obtain a final goal. To incorporate the idea of muscles working together simultaneously to achieve a certain motion, the program would require the ability to multitask the required muscles. This ability was not included in this project. Only one slider in the muscle interface can be manipulated by the user at a given time.

3.4 Finger Dependence

As noted previously, some muscles that move the fingers and thumb are located in the forearm. Since the hand model does not include the forearm, finger dependence was not implemented. This is mainly seen in the manipulation of the ring finger.

3.5 Other Constraints

Obviously other miscellaneous constraints exist which were not included in the model for they increase model complexity without providing meaningful gain. For example, the fibrous sheaths were not included in the hand model for in computer graphics, the tendon can be restricted to the bone.

4. Computer Implementation

Two programs were developed for the basis of this research. The first program is a hand muscle interface that allows the user to manipulate the muscles and tendons of the fingers and thumb. The second program is a keyboarding program that allows the user to view an animated typing hand. Both programs are discussed in this section.

4.1 Hand Muscle Program

The ability to manipulate the muscles and tendons of the human hand is a very important step in the development of creating any finger movement. The interface to the hand muscle program allows a user to manipulate a computer-generated hand by activating tendons, represented as sliders, to create movement. The sliders that are present in the interface relate to the muscles and tendons described in Table 1, and the value of the slider is interpreted as p in the algorithms described in section 2. A detailed discussion of the interface is given in Kuchar [1996].

4.2 Keyboarding Program

The ability to manipulate the muscles and tendons of the human hand is a very important step in the development of creating any finger movement. Once the animator has constructed movement in the interface described in section 3.1, the programmer can then insert the muscle movement into a program to present realistic movement in an application. As a choice, the author has decided to present an animation of typing fingers to demonstrate the intricate movements of the fingers. The program displays the left hand and keyboard in the initial typing stance. The user can type letters in the space provided at the bottom of the interface. Once the user presses the return key, the fingers will reproduce the letters that were typed by the user.

During the implementation of the keyboarding program, it was noted that the pinky would not be able to reach the top row of keys on the keyboard without wrist movement. No matter which orientation is used for the original position of the hand, there will always be a key that is unreachable without wrist movement.

5. Conclusions

In applying the new approach of muscle control movement to the human hand, the fingers and thumb were easier to manipulate. The muscle control algorithms proposed in this research can be implemented, as shown in section 3. Overall, the goal of this research was to create a more realistic human movement animation based on a complex human model. A complex model was developed to include the bones, motor muscles, and tendons for the fingers and thumb of an average human hand.

The underlying structure of the hand model is the incorporation of the motor muscles and tendons of the fingers and thumb. Two examples were provided to demonstrate the mathematical calculations required to appropriately affect the joints of the finger.

Overall, the approach of using motor muscles to manipulate human movement solves the problems of:

1. realistic animation;
2. simple manipulation and control of human finger and thumb movement.

The idea of motor muscles is not new to the area of biomechanics, but is new to computer animation. Computer animation of human figure motion is a current research issue. There are many problems in animating humans because of the number of degrees of freedom that are associated with the human model. As seen from Monheit et al. [1991], researchers have started to return to the human body model to devise more accurate motion. This research is a part of that process. Some of the future work in Lee et al. [1995] targets: "*incorporation of a more accurate model for simulating palm movement*" and "*development of special indicators to identify the seven characteristic points on the real hand in an ordinary indoor environment*". This research would complement Lee et al. [1995] since the developed computer program mentioned in section 3 would lead to a better understanding of finger movements and limits the number of possible outcomes. Thus, Lee et al. [1995] would not need to develop special indicators since the computer program would have some knowledge of where the fingers may be positioned.

In conclusion, researchers need to attempt not to oversimplify the human movement model, but to place emphasis towards more complex human models. The hypothesis of this research was to create a more realistic human movement animation based on a complex human model. The hand model created was based on a complex human hand and the movement control of the fingers and thumb was based on muscle control. Using this model, the user is able to create, by simple manipulation of the muscles, a more realistic animation of the movement of the fingers and thumb. The research presented in this thesis allows an animator to create and manipulate the movement of the fingers and thumb in a simplistic fashion, then in theory, the research can be applied to the motion of the entire human body. Thus, a complex human model will add to the realism of animating human motion. This research is a step in that direction.

5.1 Future Work

Human movement, performed with its usual gracefulness, demands the coordination of many muscles. One aspect to consider for a future version of this research would be to include the multitasking ability for the muscles to act simultaneously. One of the problems stated by Zajac [1993] is to determine how muscles coordinate any one movement, much less a repertoire of movements. Many movements need to be understood before researchers can hope to postulate broadly applicable muscle coordination principles. The point of the research is for human movement animation to look real to the user. Thus the user may be able to dictate which muscles should work together to achieve the end result.

Another direction for the presented research would be to increase the scope of the model to include the wrist and forearm. By including the wrist and forearm, two problems may be solved:

1. finger dependence;
2. in the keyboarding program, the pinky would be able to reach the top row of keys.

Understanding how and why the body coordinates muscles intrigues professionals spanning diverse disciplines. This research has a vast potential in many fields: medicine, performing arts, rehabilitation, engineering, and computer animation. There are many applications that can be made from the research presented in this paper. First, a virtual reality hand can be created to work more accurately than what currently is available. Secondly, medical students can use the muscles program to understand what each muscle does and how the lack of a muscle can have consequences to certain movements. Finally, rehabilitation professionals can use the muscle program to explain the affects of muscle loss to a patient in a clearer, more concise fashion.

5. References

Kuchar, O.A., (1996) Finger Movements on a Keyboard: An Animated Simulation of the Biomechanics of the Hand, School of Computer Science, Technical University of Nova Scotia, thesis

Lee, J., and Tosiyasu, L.K., (1995) Model-Based Analysis of Hand Posture, IEEE Computer Graphics and Applications, Volume 15, Issue 5, pp. 77-86

Monheit, G., and Badler, N.I., (1991) A Kinematic Model of the Human Spine and Torso, IEEE Computer Graphics and Applications, March, pp. 29-38

Tost, D., and Pueyo, X., (1988) Human body animation: a survey, The Visual Computer, Issue 3, pp. 254-264

Zajac, F.E., (1993) Muscle Coordination of Movement: A Perspective, Journal of Biomechanics, Volume 26, pp. 109-124