USING DYNAMIC ANALYSIS TO ANIMATE ARTICULATED BODIES SUCH AS HUMANS AND ROBOTS

Jane Wilhelms and Brian A. Barsky

Berkeley Computer Graphics Laboratory
Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720
U.S.A.

Abstract

A method of animating articulated (linked) bodies such as humans, animals and robots using *dynamic analysis* is presented. Dynamic analysis predicts motion by analyzing the effect of forces and torques on mass; this is different than the usual *kinematic* method of specifying motion, where positions, velocities, and accelerations are given without considering the forces and torques producing motion. It is difficult to kinematically specify realistic motion, particularly in cases where the body is moving fast, in complex patterns, or with great freedom. In such cases, animation based on dynamic analysis, though more expensive, may be preferable. Animation using dynamic analysis is also useful in the design and control of robots and other mechanical manipulators, and for analyzing the movement of humans and animals in biomechanics and sports.

Résumé

Nous présentons une méthode pour l'animation d'objets articulés, tels que les humains, les animaux et les robots. Cette méthode est basée sur *l'analyse dynamique*, ce qui permet de prédire les mouvements grâce à une analyse des effets des forces et des moments sur les masses. Ceci diffère de l'approche *kinématique* plus habituelle consistant à spécifier les mouvements en précisant uniquement les position, vittesse et accelération des objets sans accorder aucune attention aux forces et moments à l'origine de ces mouvements. Il est ardu de spécifier kinématiquement un mouvement réaliste, surtout lorsque l'objet se déplace rapidement, de façon complexe, ou avec beaucoup de liberté. Dans de tels cas, une animation basée sur l'analyse dynamique peut être préférable malgré son coût plus élevé. Cette approche à l'animation peut également être pratique pour la conception et le contrôle des robots et autres manipulateurs mécaniques, ainsi que pour l'analyse des mouvements humains et animaux aussi bien en biomécanique qu'en sports.

KEYWORDS: animation, human modeling, dynamics, simulation

This work was supported in part by the Defense Advanced Research Projects Agency under contract number N00039-82-C-0235, the National Science Foundation under grant number ECS-8204381, and the State of California under a Microelectronics Innovation and Computer Research Opportunities grant.

1. Overview

Considering the physical properties of the natural world when modeling imaginary worlds in computer graphics has consistently led to more realistic and aesthetically pleasing images. A natural next step in this process is to consider the physical principles governing moving objects in producing animations; these principles form the study of that part of physics known as *dynamics*. The motion of an object is determined by its own nature and its interaction with the environment, and is described dynamically as a relation between forces and torques and the behavior of masses under their influence. A simpler means of describing motion is *kinematics*, which differs from dynamics in that kinematics describes motion only in terms of positions, velocities and accelerations, neglecting the forces and torques responsible.[11] An effective means of motion specification is particularly important in animating articulated (linked) bodies, such as humans, other animals, and robots, since these bodies are capable of extremely complex movement.

Present systems for animating articulated bodies tend to do so kinematically.[2, 5, 6, 16] A major problem with kinematic motion specification is determining a sequence of positions that produces a realistic animation. The two major alternatives for finding such a sequence are: (1) recording the actual movement occurring in the real world (e.g. filming animals) or (2) using trial-and-error, testing motion sequences until an acceptable solution is found. Neither of these methods is completely satisfactory; measurements may not be easily available, and trial-and-error depends on the ability and patience of the user.

The use of dynamic analysis to avoid the limitations of kinematic motion specification has been suggested before,[5] but the cost and complexity involved have kept it from being implemented. An advantage of dynamic analysis is that the motion predicted is accurate for the specified conditions and it would occur under these conditions in the real world. Dynamic analysis coupled with computer animation has applications outside the field of computer graphics proper -- for example, in the design and control of robots and mechanical manipulators, biomechanical exploration of the engineering principles underlying animal motion,[14] and in the evaluation of the safety and efficiency of motion in sports.[9]

Previous use of dynamics to simulate articulated bodies in motion has largely been limited to crash studies,[15] where dynamics is used to predict the uncontrolled motion of bodies under the influence of large external forces, such as those occurring in auto and plane crashes. Graphical output tends to be simple and motion is often limited to two dimensions. The method described here differs from crash simulations in that dynamics is used to *control* the motion, by using internal forces and torques simulating muscles and motors.

This paper introduces a method of specifying three-dimensional motion dynamically. Although the principles involved are particularly oriented towarding animating articulated bodies, they are equally applicable to simpler structures. The use of dynamic analysis is explained first, followed by a brief description of the system (*Deva*) that has been implemented to investigate dynamic and kinematic motion specification, and concluding with some discussion of future research directions.

2. Dynamic Analysis Using the Gibbs-Appell Formulation

For dynamic analysis, it is necessary to develop the dynamic equations of motion (one for each degree of freedom) that describe the relationship between masses and the forces and torques affecting them. In the case of articulated bodies, each body segment (e.g., the leg) can be modeled as a rigid body. An unconstrained rigid body is capable of six degrees of freedom of motion; however, in articulated bodies motion is restricted by attachments to neighboring segments. Because the motion of each segment of the articulated body is affected by the motion of others, the dynamics equations are complex and coupled.

A variety of formulations of the dynamics equations are available, all of which produce the same result in slightly different terms. The most familiar formulation is that of Newton, exemplified in simple form in his *Second Law* ($\mathbf{F} = m\mathbf{a}$); that is, the sum of all forces ($\mathbf{F}$) acting on a particle is equal to the particle's mass (m) times its acceleration ($\mathbf{a}$). Another formulation, the Gibbs-Appell, more easily describes the complex dynamics of articulated bodies and was chosen for use here.[1, 8]

The dynamics equations can be solved in either of two directions: the *direct solution* involves providing the accelerations of the bodies involved and solving the equations for the forces and torques that produced these accelerations; the *indirect solution* involves providing the forces and torques, and solving for accelerations. The indirect problem is of relevance here, where the desired output is a prediction of realistic motion. Besides the controlling input (forces and torques), solution of the dynamics equations also requires knowledge of the present state of the body including, for each segment, its mass, the location of its center of mass and the distribution of the mass around the center, and its present position and velocity. Once the equations are formulated, they can be integrated using numerical methods.[7] The cost of dynamic analysis is bounded by the cost of solving the set of equations, and is $O(n^3)$ for n degrees of freedom.

2.1. The Gibbs Formula: The Basis for the Gibbs-Appell Formulation

The Gibbs-Appell dynamics formulation is based on the Gibbs Formula, which describes the *energy of acceleration*.[8] For rigid bodies consisting of n segments, this formula is

$$G = \sum_{k=1}^{n} \left(\frac{1}{2} m_k \mathbf{a}_k^T \mathbf{a}_k + \frac{1}{2} \alpha_k^T \mathbf{I}_k \alpha_k + \alpha_k^T (\omega_k \times I_k \omega_k) + f(\omega_k) \right)$$

where, for segment k,

$m_k =$ *mass*
$\mathbf{a}_k =$ *acceleration vector of center of mass*
$\alpha_k =$ *angular acceleration vector*
$\omega_k =$ *angular velocity vector*
$f(\omega_k) =$ *scalar disappears with differentiation*

$$\mathbf{I}_k = \begin{bmatrix} I_{xx} I_{xy} I_{xz} \\ I_{xy} I_{yy} I_{yz} \\ I_{xz} I_{yz} I_{zz} \end{bmatrix} = \textit{inertial tensor}$$

$I_{xx} = \int (y^2 + z^2)dm$; *etc., moments of inertia*
$I_{xy} = \int xy\ dm$; *etc., products of inertia*

The actual dynamics equations are found by partially differentiating the Gibbs formula with respect to the local acceleration relative to each degree of freedom. For n degrees of freedom, this results in n equations, which relate local accelerations at the degrees of freedom to the *generalized forces* acting on the body. A generalized force can be thought of as the net force (for sliding degrees of freedom) or torque (for revolute degrees of freedom) active at this degree of freedom, and is the result of all forces and torques acting within the system.

2.2. Dynamic Analysis Using the Gibbs-Appell Formulation

The dynamics equations can be stated in a succinct form as

$$\mathbf{M}\,\ddot{\mathbf{c}} + \mathbf{V} = \mathbf{q} \qquad or \qquad \mathbf{M}^{-1}(\mathbf{q} - \mathbf{V}) = \ddot{\mathbf{c}}$$

For a body with n degrees of freedom, $\ddot{\mathbf{c}}$ is an n-length vector of the local acceleration occurring at each degree of freedom (e.g. a rotation about the X-axis or a translation along the Z-axis) and $\mathbf{q}$ is an n-length vector specifying the generalized force active at each degree of freedom. $\mathbf{M}$ is an $n \times n$ inertial matrix dependent upon the configuration of the system masses. $\mathbf{V}$ is an n-length vector dependent upon the position of the masses and their motion relative to each other. Thus, if the generalized force vector $\mathbf{q}$ is available, the equations can be solved for accelerations $\ddot{\mathbf{c}}$ which can then be used to update the positions of the body segments.[8]

The elements of both the matrix $\mathbf{M}$ and the vectors $\ddot{\mathbf{c}}$, $\mathbf{V}$, and $\mathbf{q}$ are arranged so that those referring to revolute degrees of freedom precede those referring to sliding degrees of freedom. A coordinate frame is associated with each degree of freedom such that its Z-axis is the axis of sliding or revolution. Indexing is such that degree of freedom k lies between segments k–1 and k.

For dynamic analysis, multiple-degree-of-freedom joints can be depicted as a sequence of single-degree-of-freedom joints connected by massless and dimensionless segments. Thus each degree of freedom has its own *joint* and *segment*, and in the following dynamics discussion "joint" and "segment" refer to these generalized forms.

The formulation of the dynamics equations described below does not exploit the recursive nature of the terms, and the cost of setting up the equations is $O(n^4)$. A recursive formulation has been suggested by Athan and Horowitz[1] which is $O(n^3)$. It is unlikely that any further speedup can be gained, however, because of the cost of solving the set of equations.

2.2.1. Explanation of Terms

The partial differentiation of the Gibbs formula leaves motion described in inertial world space terms; however, these equations can be restated in terms of local joint configuration because of the known relation between local and world frames. The following kinematic configuration information is necessary for dynamic analysis ($k = 1, \ldots, n$ for n degrees of freedom).

$\mathbf{r}_k$ = *3-D position vector of joint connecting segments k–1 and k*
$\mathbf{g}_k$ = *3-D position vector of center of mass of segment k*
$\mathbf{u}_k$ = *3-D unit vector of rotation or sliding axis of joint k*
$s_k, \dot{s}_k, \ddot{s}_k$ = *sliding along axis $\mathbf{u}_k$, its velocity, and acceleration*
$\theta_k, \dot{\theta}_k, \ddot{\theta}_k$ = *rotation angle about axis $\mathbf{u}_k$, its angular velocity, and acceleration*
$\mathbf{R}_k$ = *3×3 matrix describing rotation about axis $\mathbf{u}_k$*

2.2.2. Finding the Inertial Matrix (M-Matrix)

$\mathbf{M}$ is an $n \times n$ matrix which takes into account the present distribution of body mass. $\mathbf{M}$ consists of four submatrices.

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}^{\theta} & \mathbf{M}^{\theta s} \\ [\mathbf{M}^{\theta s}]^T & \mathbf{M}^{s} \end{bmatrix}$$

The upper left submatrix $\mathbf{M}^{\theta}$ is an $r \times r$ matrix describing the relation between revolute degrees of freedom. Its elements are defined by the following equation

$$m^{\theta}_{ij} = m^{\theta}_{ji} = \sum_{k=distal(i,j)}^{Todistal} (m_k [\mathbf{u}_i \times (\mathbf{g}_k - \mathbf{r}_i)]^T [\mathbf{u}_j \times (\mathbf{g}_k - \mathbf{r}_j)] + \mathbf{u}_i^T \mathbf{I}_k \mathbf{u}_j)$$

(for $i = 1, \ldots, r$ and $j = 1, \ldots, r$).

(Note that *distal*(i,j) refers to whichever of i or j lies further from the initial world segment, and if i and j lie on separate branches the calculation does not take place. *Todistal* refers to the furthest segments continuing out this branch.)

The upper right submatrix $\mathbf{M}^{\theta s}$ (whose transpose is the lower left submatrix) is an $r \times t$ matrix describing the relation between revolute and sliding degrees of freedom. Its elements are

$$m^{\theta s}_{ij} = \sum_{k=distal(i,j)}^{Todistal} m_k \mathbf{u}_j^T [\mathbf{u}_i \times (\mathbf{g}_k - \mathbf{r}_i)]$$

(for $i = 1,..,r$ and $j = 1,..,t$).

The lower right submatrix $\mathbf{M}^{s}$ is a $t \times t$ matrix describing the relation between sliding degrees of freedom. Its elements are

$$m^{s}_{ij} = m^{s}_{ji} = \sum_{k=(i,j)}^{Todistal} m_k \mathbf{u}_i^T \mathbf{u}_j$$

(for $i = 1, \ldots, t$ and $j = 1, \ldots, t$).

2.2.3. Velocity-Dependent V-Vector

The **V**-vector takes into account such velocity-dependent contributions as the Coriolis and centrifugal forces. The **V**-vector contains two sub-vectors: $\mathbf{V}_\theta$ is an r-length vector representing revolute degrees of freedom, and $\mathbf{V}_s$ is a t-length vector representing sliding degrees of freedom.

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_\theta \\ \mathbf{V}_s \end{bmatrix} \quad \mathbf{V}_\theta = \begin{bmatrix} V_{\theta 1} \\ \cdots \\ V_{\theta r} \end{bmatrix} \quad \mathbf{V}_s = \begin{bmatrix} V_{s1} \\ \cdots \\ V_{st} \end{bmatrix}$$

The elements of the $\mathbf{V}_\theta$ vector for revolute degrees of freedom ($V_{\theta k}$ for $k = 1, \ldots, r$) are found using the following equation.

$$V_{\theta k} = [\dot\theta_1,..,\dot\theta_r,\dot s_1,..,\dot s_t] \begin{bmatrix} \mathbf{N}_\theta^{k\theta} & \mathbf{N}_\theta^{ks} \\ [\mathbf{N}_\theta^{ks}]^T & 0 \end{bmatrix} [\dot\theta_1,..,\dot\theta_r,\dot s_1,..,\dot s_t]^{\mathbf{T}}$$

where the components of the $\mathbf{N}_\theta$ matrix are found by

$$n_{\theta ij}^{k\theta} = n_{\theta ji}^{k\theta} = \sum_{l=distal(k,j)}^{distal} (m_l[\mathbf{u}_k \times (\mathbf{g}_l - \mathbf{r}_k)]^T [\mathbf{u}_i \times (\mathbf{u}_j \times (\mathbf{g}_l - \mathbf{r}_j))]$$
$$+\mathbf{u}_k^T[\frac{1}{2} trace(\mathbf{I}_l)\mathbf{u}_i \times \mathbf{u}_j + \mathbf{u}_j \times \mathbf{I}_l \mathbf{u}_i])$$

where $trace(\mathbf{I}_l) = I_{l11} + I_{l22} + I_{l33}$
(for $i = 1, \ldots, r$ and $j = 1, \ldots, r$)

and

$$n_{\theta ij}^{ks} = \begin{cases} \displaystyle\sum_{l=distal(k,j)}^{distal} m_l[\mathbf{u}_k \times (\mathbf{g}_l - \mathbf{r}_k)]^T [\mathbf{u}_i \times \mathbf{u}_j] & \textit{for } i \textit{ proximal to } j \\ 0 & \textit{for } i \textit{ distal to } j \end{cases}$$

(for $i = 1, \ldots, r$ and $j = 1, \ldots, t$)

The elements of $\mathbf{V}_s$ vector for sliding degrees of freedom (V_{sk} for $k = 1, \ldots, t$) are found using the following equation.

$$V_{sk} = [\dot\theta_1,..,\dot\theta_r,\dot s_1,..,\dot s_t] \begin{bmatrix} \mathbf{N}_s^{k\theta} & \mathbf{N}_s^{ks} \\ [\mathbf{N}_s^{ks}]^T & 0 \end{bmatrix} [\dot\theta_1,..,\dot\theta_r,\dot s_1,..,\dot s_t]^{\mathbf{T}}$$

where the components of the $\mathbf{N}_s$ matrix are found by

$$n_{sij}^{k\theta} = n_{sji}^{k\theta} = \sum_{l=distal(k,j)}^{distal} m_l \mathbf{u}_k^T [\mathbf{u}_i \times (\mathbf{u}_j \times (\mathbf{g}_l - \mathbf{r}_j))]$$

(for $i = 1, \ldots, r$ and $j = i, \ldots, r$)

and

$$n_{sij}^{ks} = \begin{cases} \displaystyle\sum_{l=distal(k,j)}^{distal} m_l \mathbf{u}_k^T [\mathbf{u}_i \times \mathbf{u}_j] & i \textit{ proximal to } j \\ 0 & i \textit{ distal to } j \end{cases}$$

(for $i = 1, \ldots, r$ and $j = 1, \ldots, t$)

2.3. Forces and Torques Contributing to Motion

A variety of forces and torques contribute to motion, some of which can be calculated automatically, others of which can be input by the user to control the movement. The forces and torques that can be computed automatically include those due to gravity, to collisions between parts of the body, to joint limits, and to contact with external objects such as the ground or objects in the environment whose position and behavior is known to the system. Those forces and torques that should be input include internal forces applied across joints and external applied forces unknown to the system; these forces and torques will be called *controlling forces* and *torques*. All forces and torques are calculated for each degree of freedom; by summing their contributions to each degree of freedom, the generalized force for that degree of freedom is found. A more detailed description of the calculation of these forces is forthcoming.[12]

2.3.1. Gravitational Forces and Torques

The gravitational component of each generalized force can be determined simply by considering the effect of the mass of distal segments at a particular degree of freedom. In the case of each revolute degree of freedom, this involves finding the torque around its axis of rotation, which depends upon the gravitational force acting on each distal segment and the perpendicular distance of the point of force's application from the axis of rotation. The equation for torque due to gravity at degree of freedom k is

$$\tau_{gk} = -g_c \sum_{i=k}^{distal} m_i \mathbf{z}_{0T}[\mathbf{u}_k \times (\mathbf{g}_i - \mathbf{r}_k)]$$

where $\mathbf{z}_0 = (0,0,1)$ is a vertical direction vector, $g_c = 9.81$ m /sec^2 is the acceleration due to gravity, and the other terms are as explained previously.

In the case of each sliding degree of freedom, the gravitational component (f_{gk}) contributing to the generalized force there is dependent upon the component of the gravitational force acting on each distal segment that lies along the axis of sliding; that is,

$$f_{gk} = -g_c \sum_{i=k}^{distal} m_i \mathbf{z}_0^T \mathbf{u}_k$$

By altering the gravitational constant, motion on other planets or in space can be simulated.

2.3.2. External Applied Forces

Where the magnitude and direction of an external applied force (such as a pull from a rope or a shove) is known, the contribution of this force to the generalized force at each degree of freedom can be calculated using a method very similar to that used for gravity. Assume that a force represented by vector $\mathbf{F}_a$ is applied to segment a at a location designated by the world-space vector $\mathbf{a}$. Each revolute degree of freedom i which lies proximal to segment a feels the effect of this applied force as the torque τ_{app_i} defined by the equation

$$\tau_{app\_i} = \mathbf{F}_a[\mathbf{u}_i \times (\mathbf{a} - \mathbf{r}_i)]$$

Each sliding degree of freedom j which lies proximal to segment a feels the effect of the applied force as the force F_{app_j} defined by the equation

$$F_{app\_j} = \mathbf{F}_a \mathbf{u}_j$$

At present, arbitrary external applied forces cannot be specified by the system *Deva*.

2.3.3. Joint Limit Forces and Torques and Damping

Although the specification of the dynamics equations automatically restricts motion to the designated degrees of freedom, it does not restrict motion within a particular degree of freedom; thus, segments are quite capable of moving through each other. To restrict motion to more realistic patterns, limiting positions for joints must be established and appropriate counteracting forces and torques applied when the motion threatens to pass beyond these limits. Simulating exact joint limit forces and torques, particularly for multiple-degree-of-freedom joints, would be extremely difficult. However, they can be adequately represented by spring and damper pairs that become active as joint limits are approached. The natural damping of friction can also be simulated by applying a damping force or torque during all joint movements.[10] Linear springs are simply specified as the product of a *spring constant* times the distance that the spring is compressed. Damping force or torque is the product of a *damping constant* times the local speed at that degree of freedom. However, the use of such a simple spring and damper is undesirable because the user must determine appropriate constants for each degree of freedom (the opposing force needed being dependent upon the nature of the joint). The method implemented is somewhat more complex, but automatically calculates joint limit forces and torques. The spring and damper used are linear; however, the stiff spring necessary to quickly stop motion might be better simulated with an exponential function.

The end limit forces (for sliding joints) or torques (for revolute joints) each have three components. The first component counteracts other forces and torques pushing the joint beyond its limit and is simply equal in magnitude and opposite in direction to all other forces (for sliding joints) or torques (for revolute joints) contributing to the generalized force at that degree of freedom (such as gravity or actuator forces). The second component is a spring whose strength is a function of the amount the joint limit has been exceeded, the local velocity, and the mass (for sliding joints) or moment of inertia (for revolute ones) distal to this degree of freedom. The third component is a damper which is a function of the local velocity and the mass or moment of inertia due to distal segments.

A damping action is normally active throughout the range of joint motion. This damper is calculated in the same manner as the end limit damper but is weaker (usually 60% of end damping).

2.3.4. Ground Reaction Forces

Bodies must be prevented from moving through the ground by counteracting reaction forces. Ideally, reaction forces can be calculated by considering the present state of the entire body. However, they are not automatically calculated in the course of dynamic analysis as done here, because the Gibbs-Appell formulation avoids the calculation of all reaction forces in the interest of speed. Reaction forces can be included, but this would involve considerable increase in the number of degrees of freedom that must be analyzed. An alternative is to set up and solve another set of dynamics equations that will predict reaction forces when a portion of the body is in contact with the ground. In order to avoid this considerable cost, a method of simulating ground reaction forces with an approximated force plus springs and dampers has been found satisfactory.

Simulating ground reaction forces proved an interesting problem, because the necessary normal force can vary from one to several times the body weight, because it can be distributed over several support points, and because a suitable tangential frictional force must be found to provide the desired amount of horizontal slipping on the ground. The normal force must be sufficient to stop vertical motion before the body descends noticeably into the ground, and yet not be so much as to cause unrealistic bouncing. A further problem involves the sampling rate at which dynamic analysis is done. If the body is moving rapidly it may descend significantly below the floor between sample times. A combination of an estimated counteracting force plus a spring and damper has proved an acceptable solution. Reaction forces consist of a normal force perpendicular to the ground and two orthogonal tangential forces.

Calculating Normal Forces. Normal forces are calculated as a combination of an estimated reaction force plus, possibly, a contribution from a spring and damper. The estimated force is based upon the body's total mass and momentum. This force is distributed among all contact points in proportion to their depth below the ground surface. Floor contact points are recognized by checking the height of each of the eight corners of the max/min boxes of each of the body segments against the height of a planar horizontal ground.

Because this force is only approximate, additional force is added as the contact point continues to descend. This force is due to a spring whose strength is a function of estimated normal force described above and the amount of descent beyond the original contact depth. A damper is also introduced to slow motion.

Calculating Tangential Frictional Forces. Frictional forces act to oppose tangential motion along the surface and are dependent upon both the normal force pressing into the ground and the characteristics of the surfaces. To simulate frictional forces, a combination of an estimated force plus a spring and damper are again used.

The estimated tangential force for a particular contact point is merely the product of a coefficient of friction and the normal force calculated as indicated above and applied in a direction to oppose sliding. If the contact continues to be displaced tangentially, a tangential spring is applied to contribute further opposition. (A maximum tangential force can be indicated so that sliding will occur where desired.) A damping force is also added to subdue motion.

The reaction force is then treated as an applied force acting on the contact point, as described in Section 2.3.2. The effect of reaction forces can be altered by varying the constants described above to simulate more or less springiness, damping, and friction; e.g. bodies sliding on ice or bouncing on trampolines.

2.3.5. Actuator and Muscular Forces

Actuators in mechanisms such as robots and muscles in animals serve the same function: applying a force or torque across the joint that joins segments for the purpose of movement. Actuators are far simpler than muscles, since they can (ideally) be designed to apply the force or torque relative to only one degree of freedom, and as such can map directly into a component of a single generalized force. Muscles are more complex for several reasons. In particular, they apply a force between two or more fixed points so the resultant torque varies with the joint angle. The joint is usually capable of more than one degree of freedom. Typically several muscles cross one joint and some muscles cross more than one joint. The effect of all the muscles controlling a joint can, however, be simplified to a single net force or torque, and components of these net forces and torques can be found relative to the axis of motion of each of the joint's degrees of freedom. These components are then equivalent to ideal actuator forces or torques and can be mapped directly onto the generalized forces for each degree of freedom. Because of the great complexity of modeling individual muscles, the present implementation of *Deva* starts with a controlling input of actuator forces and torques, rather than muscles.

Controlling forces are input by the user, usually as control functions which describe the force or torque versus time for each degree of freedom. Although this method is suitable for experimentation, the number of iterations necessary to develop a desired movement makes it unsuitable for a practical animation system.

Various low-level control schemes are being investigated to give the user more general control over the motion. A simple example of this is the *freeze* function which allows clamping of specified degrees of freedom to their present position by applying a spring and damper combination on either side of this position. A similar method is being explored to automatically adjust for balance, a major problem in unstable two-legged animals.

3. Deva: A System for Animating Articulated Bodies

Deva is an experimental graphical system for simulating the motion of articulated bodies. More detailed descriptions of the system are available elsewhere.[13] *Deva* consists of a central ***body model*** describing the present characteristics (dimensions, mass properties, display properties, connectivity, etc.) and configuration (positions of segments) of the body, and a set of subroutines for modifying, displaying, and animating the body. The body can be displayed either as a vector figure on a calligraphic display (see Figure 1) or as a solid figure on a raster display.

The body model consists of a number of segments (e.g., a human body is typically simplified to 15 segments) connected by joints capable of from one to six degrees of freedom of motion (three translational, three rotational). Each segment has its own local coordinate frame whose position and orientation are defined relative to its more proximal segment. It is possible to associate limits with joints to constrain their range of motion.

The body can be moved either kinematically or dynamically. For kinematically-specified motion, the user describes a sequence of local positions taken by each degree of freedom. For dynamically-specified motion, the user describes the controlling forces and torques acting on and in the body. Dynamic analysis routines evaluate the effect of these forces and torques and output an updated configuration of the body. As the body configuration is altered, the calligraphically-displayed image of the body is repositioned.

& Baecker Reeves

The use of ***control functions*** provides a convenient means of specifying motion control information for either kinematic or dynamic animation. A control function is a curve of either position versus time (for kinematic control) or force/torque versus time (for dynamic control). A unique control function is associated with each degree of freedom, and evaluating the control functions provides positions or forces (for sliding joints) and torques (for revolute joints) that specify the motion at that degree of freedom. An interactive graphical editor, *Virya*,[13] has been developed to define control functions for each degree of freedom (see Figure 2). Control functions are ***cubic interpolatory spline curves***[4, 3] which are defined by user-specified ***defining points***. This method allows the user to specify quite complex and smooth curves by designating relatively few defining points.

4. Conclusions

The use of dynamic analysis for the animation of articulated bodies has been briefly presented, and a system (*Deva*) using this method described. In computer graphics, dynamic analysis provides an alternative means of motion specification that avoids some of the problems involved in pure kinematic animation. Graphical simulation of dynamically-predicted motion is also a tool useful in such fields as robotics and bioengineering. Kinematic motion specification is much less expensive and far easier to implement than dynamic specification and is preferable in cases where an acceptable kinematic description is available. When this is not the case, particularly when motion is complex, fast, or involves contact with powerful external forces and torques, dynamic analysis may be preferable. A combination of kinematic and dynamic motion specification which provides the best of both techniques could provide the best of both worlds.

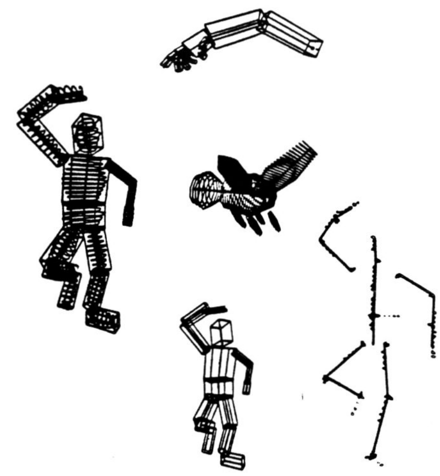

Figure 1. Calligraphic Figures

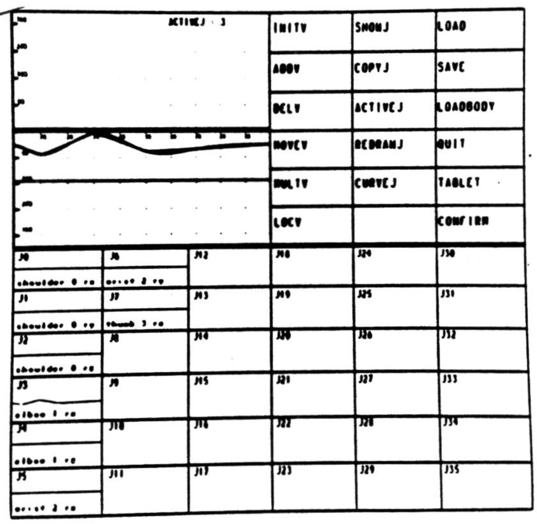

Figure 2. Virya Graphical Editor Display

Research is now concentrating upon the practicality and usefulness of dynamic specification for producing realistic animation. This involves a number of problems and complexities. For example, how does one choose the appropriate applied forces and torques that produce realistic motion? How can one best model motion such as walking, where the body is sometimes restricted by the ground, and at other times not? How can one best provide a balancing mechanism, a particularly complex problem for two-legged bodies? What kind of high-level control will make the system practical to use? These issues provide fertile ground for future research and are currently being investigated.

Acknowledgements

The authors wish to thank Professors Roberto Horowitz, Al Pisano, and Lawrence Stark and former graduate student Tim Athan for their aid in this research.

References

1. Tim Athan, *An Analysis of Complex Multi-Link Chains,* Master's Thesis, Dept. of Mechanical Engineering, University of California, Berkeley (June, 1984).
2. Norman I. Badler, Joseph O'Rourke, Stephen Platt, and M. A. Morris, "Human Movement Understanding: A Variety of Perspectives," pp. 53-55 in *Proceedings 1st Annual National Conference of Artificial Intelligence,* (1980).
3. Brian A. Barsky and Spencer W. Thomas, "TRANSPLINE -- A System for Representing Curves Using Transformations among Four Spline Formulations," *The Computer Journal,* Vol. 24, No. 3, August, 1981, pp. 271-277.
4. Richard H. Bartels, John C. Beatty, and Brian A. Barsky, *An Introduction to the Use of Splines in Computer Graphics,* Technical Report No. UCB/CSD 83/136, Computer Science Division, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, California, USA (August, 1983). Also Tech. Report No. CS-83-9, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.
5. T. W. Calvert, J. Chapman, and A. Patla, "The Integration of Subjective and Objective Data in the Animation of Human Movement," pp. 198-203 in *SIGGRAPH '80 Conference Proceedings,* ACM,(July, 1980).
6. Don Herbison-Evans, "NUDES 2: A Numeric Utility Displaying Ellipsoid Solids, Version 2," pp. 354-356 in *SIGGRAPH '78 Conference Proceedings,* ACM,(August, 1978).
7. Robert W. Hornbeck, *Numerical Methods,* Quantum Publishers, Inc., New York, New York (1975).
8. Roberto Horowitz, *Model Reference Adaptive Control of Mechanical Manipulators,* Ph.D. Thesis, Mechanical Engineering, University of California, Berkeley, CA (May, 1983).
9. D. I. Miller, "Computer Simulation of Human Motion," in *Techniques for the Analysis of Human Motion,* ed. D. W. Grieve, et. al.,Lepus Books,London(1975).
10. Al Pisano, Department of Mechanical Engineering, University of California, Berkeley, CA (1984). Personal communication.
11. Robert Resnick and David Halliday, *Physics Part I,* John Wiley and Sons, Inc., New York (1966).
12. Jane Wilhelms, *Computer Graphics Simulation of the Motion of Articulated Bodies such as Humans and Robots, with Emphasis on the Use of Dynamic Analysis (Tentative Title),* Ph.D. Thesis, Computer Science Division, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. Forthcoming.
13. Jane Wilhelms and Brian A. Barsky, "Kinematic and Dynamic Animation of Articulated Bodies Using Control Functions." Submitted for publication.
14. R. J. Williams and Ali A. Seireg, "Interactive Computer Modeling of the Musculoskeletal System," *IEEE Transactions on Biomedical Engineering,* Vol. BME-24, No. 3, May, 1977, pp. 213-219.
15. K. D. Willmert, "Visualizing Human Body Motion Simulations," *IEEE Computer Graphics and Applications,* Vol. 2, No. 9, November, 1982, pp. 35-43.
16. David Zeltzer, "Motor Control Techniques for Figure Animation," *IEEE Computer Graphics and Applications,* Vol. 2, No. 9, November, 1982, pp. 53-60.