Adapting Simulated Behaviors For New Characters

Jessica K. Hodgins and Nancy S. Pollard

College of Computing and Graphics, Visualization and Usability Center*

Abstract

This paper describes an algorithm for automatically adapting existing simulated behaviors to new characters. Animating a new character is difficult because a control system tuned for one character will not, in general, work on a character with different limb lengths, masses, or moments of inertia. The algorithm presented here adapts the control system to a new character in two stages. First, the control system parameters are scaled based on the sizes, masses, and moments of inertia of the new and the original characters. Then a subset of the parameters is fine-tuned using a search process based on simulated annealing. To demonstrate the effectiveness of this approach, we animate the running motion of a woman, child, and imaginary character by modifying the control system for a man. We also animate the bicycling motion of a second imaginary character by modifying the control system for a man. We evaluate the results of this approach by comparing the motion of the simulated human runners with video of an actual child and with data for men, women, and children in the literature. In addition to adapting a control system for a new model, this approach can also be used to adapt the control system in an on-line fashion to produce a physically realistic metamorphosis from the original to the new model while the morphing character is performing the behavior. We demonstrate this on-line adaptation with a morph from a man to a woman over a period of twenty seconds.

CR Categories: I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism: Animation—; G.1.6 [Numerical Analysis]: Optimization—; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically-Based Modeling

Keywords: Human Motion, Motion Control, Dynamic Simulation, Simulated Annealing

1 Introduction

If simulated, humanlike characters are to be useful in animations and virtual environments, we must be able to create new, appealing characters easily. Appealing human motion has several components: the kinematics and dynamics of the figure must be physically correct and the control algorithms must make the figure perform in ways that appear natural and are stylistically appropriate for the setting and character. In this paper, we describe an algorithm for adapting existing control systems to new dynamic models to facilitate the rapid creation of new characters. We demonstrate this approach for running and cycling behaviors. A running behavior designed for a male figure is adapted to control the running motion for a woman, child, and imaginary character, and a cycling behavior designed for a male figure is adapted to control a second imaginary character. The running motion of three different figures with markedly different dynamic properties is shown in figure 1. A simple, or geometric, scaling is not adequate to transform one model or control system into another. For example, the man and the child differ not only in height but also in proportion because the child has a proportionally heavier torso and shorter arms and legs.

Figure 1: Image of running child, woman, and man.

An algorithm such as the one described in this paper should allow an animator to develop a new character by using a commercial modeling package to define the shape of the body parts and then to automatically adapt an existing behavior to animate the new model. The animation process proceeds in two stages. First, the volume and mass distribution of each body part are computed from a polygonal representation of the new model, and an approximation to the new control system is obtained by scaling based on the sizes, masses, and moments of inertia of the new and the old models. Second, a search process is used to tune the new control system. When the old and new models differ substantially, the transformation may require adapting the control system to one or more intermediate models rather than moving directly from the original to the new model.

This algorithm can be used not only to adapt a control system to a new model but also to perform a physically realistic metamorphosis between two models. In the transformation, the graphical model, the dynamic system, and the control system are interpolated on-line while the character performs the behavior. The trajectories of the control system parameters for the transformation are found using a two-stage process of scaling and tuning that is similar to the procedure used for the off-line adaptation. Figure 2 shows the metamorphosis of a man into a woman while running.

*Georgia Institute of Technology, Atlanta, GA 30332-0280, [jkh|nsp]@cc.gatech.edu

Figure 2: Images of the metamorphosis of a man into a woman while running. The metamorphosis occurred over 20 seconds and the images are spaced by 3 seconds.

2 Background

Generating appealing motion is the central problem in animation. Dynamic simulation and other procedural approaches are one potential solution to this problem. Simulation guarantees physical realism (or at least adherence to a set of consistent "pseudo-physical" laws), but the design of control systems for characters with interesting complexity has proved difficult. Control systems or procedural algorithms can be hand-designed[11, 25, 6, 5, 14, 32], but this approach is labor intensive and requires that the animator or programmer have extensive knowledge about the details of the behavior.

A more appealing approach is automatic or semi-automatic design. One such approach treats the problem of generating motion as a trajectory optimization problem. Witkin and Kass[38] used this approach to control a jumping Luxo lamp. Cohen and Liu[8, 19] divided the optimization problem into smaller spacetime windows thereby providing more control for the animator and reducing the time required for the optimization. They used this approach to control a two-link acrobot and a planar diving figure. Liu, Gortler, and Cohen[20] implemented a hierarchical wavelet-based version of spacetime constraints to allow finer detail where necessary without increasing the computation cost uniformly. Zhao and his colleagues at the University of Pennsylvania[40] represented the trajectory of the control variables as a spline and then optimized the locations of the control points for the spline. They used this approach to control a planar human figure performing a vertical jump.

A second approach to automatically generating motion uses techniques from optimal control to find a control algorithm instead of a desired trajectory. Once found, control algorithms have the advantage that different but similar motions can be generated to respond to a disturbance or interaction without further optimization. As optimization techniques for complex systems approach real-time, however, this distinction becomes less significant because trajectories can be computed for new situations as they arise. In the most general case, an optimal controller must contain information on how to get from every state of the system to every other state. This problem is of higher dimension than the problem of finding a trajectory that reaches a particular goal state from a particular start state. As a result, optimal control approaches have focused on simple systems, on problem domains where the space is dense with solutions, or on techniques that allow the space to be represented without a fine discretization.

The first paper to introduce optimal control to the graphics community was Brotman and Netravali[4]. They used a linear quadratic regulator to control the motion of a single body on the plane. Huang and van de Panne[15] used a best-first search to discover a sequence of set points that, when combined with a proportional-derivative servo, allowed a two link acrobot to hop and flip. Closed loop controllers were synthesized by van de Panne and his colleagues[35, 36] for the jumping Luxo lamp and other simple systems using dynamic programming. A generate-and-test strategy was used by van de Panne and Fiume[34] to produce neural network-based control systems for a wide variety of planar creatures with 3-6 links. Ngo and Marks and their colleagues[23, 1] relied on a similar generate and test approach to find stimulus/response systems that animate a variety of behaviors for planar and three-dimensional figures. Sims[30, 29] used genetic algorithms to construct linked creatures and competitive behaviors for the task of capturing a block. Grzeszczuk and Terzopoulos[13] used simulated annealing to learn low-level controllers and higher-level behaviors for locomotion of fish and snakes.

Automatic techniques that begin the optimization or search process without significant knowledge of the behavior have not yet been successfully used for complex models such as three-dimensional humanlike figures. For a human figure with a realistic number of degrees of freedom, the search space is substantial and the density of acceptable solutions is low; however, knowledge about the behavior can be used to focus the search. Knowledge can be incorporated in the form of external guiding forces, via an existing but imperfect control system, or through motion capture data. External forces were used by Lamouret and van de Panne[37] to maintain the attitude of the body of a walking human figure, thereby guiding the optimization process towards the desired solution. The external force was eliminated in later stages of the optimization. Laszlo et al.[17] used limit cycle control to stabilize open-loop trajectories for walking of a human model with 19 degrees of freedom. Ringrose[26] adapted the control system of a planar quadruped to carry additional weight or to have longer leg lengths or heavier feet. In this paper, we take a similar approach by automatically adapting an existing control system for a new dynamic model. Because our control system is more complex and has more parameters, we include more extensive knowledge about the behavior in the form of a priori scaling and parameter selection. We believe that incorporating this knowledge improves the resulting control system and allows us to use fewer intermediate models for transitioning between more complex systems.

Other researchers have realized that if generating motion directly proves too difficult, perhaps we can obtain desired motion by adapting or smoothly blending between existing motion sequences derived from procedural approaches, simulation, or motion capture[39, 7, 33, 27, 12]. Techniques for adapting existing trajectories via optimization share with our work the idea that optimization procedures can be used to adapt to new situations. The two approaches differ in the level of the parameters used by the optimization. Motion trajectories contain little explicit knowledge of the task to be performed, but a well-parameterized control system contains extensive knowledge about the task.

We also draw from research in biomechanics for data and inspiration in this work. McMahon's elegant work on scaling between species inspired the idea that the control parameters could be approximately scaled based on knowledge about the dynamics of the system[21]. The biomechanics literature also contains data about

	Woman	Child	Running Character	Cycling Character
Mass (kg)				
head	4.22 (0.90)	5.63 (0.99)	5.98 (1.01)	7.19 (1.07)
torso	17.62 (0.87)	8.47 (0.68)	12.55 (0.78)	28.95 (1.03)
pelvis	11.08 (0.94)	3.71 (0.65)	9.51 (0.89)	7.89 (0.84)
leg	8.32 (0.92)	2.98 (0.65)	6.84 (0.86)	5.03 (0.78)
arm	2.51 (0.87)	1.20 (0.68)	2.38 (0.85)	2.70 (0.89)
wheel				1.93 (1.20)
frame				1.52 (0.76)
Length (m)				
height	1.63 (0.91)	1.08 (0.60)	1.64 (0.92)	1.74 (0.97)
leg length	0.68 (0.91)	0.39 (0.52)	0.72 (0.96)	0.70 (0.93)
arm length	0.50 (1.01)	0.29 (0.59)	0.55 (1.12)	0.54 (1.11)
hip spacing	0.15 (0.77)	0.13 (0.67)	0.17 (0.84)	0.22 (1.10)
wheel radius				0.58 (0.87)
Moment of Inertia ($kg\ m^2$)				
leg/hip	1.17 (0.89)	0.13 (0.58)	1.58 (0.95)	0.48 (0.84)
body/ankle	45.40 (0.90)	10.10 (0.67)	36.55 (0.86)	74.80 (0.99)
body/hip	6.50 (0.90)	2.26 (0.73)	5.16 (0.86)	17.67 (1.10)

Figure 3: Measurements of the mass and size of body and bicycle parts for the woman, child, and two imaginary characters. Moment of inertia parameters express moment of inertia of several body parts about a particular joint. Numbers in parentheses show the geometric scaling factor from the man to the other characters based on that parameter alone. Variation in these numbers is an indication that geometric scaling alone will not adequately describe the transformation from the man to the new character.

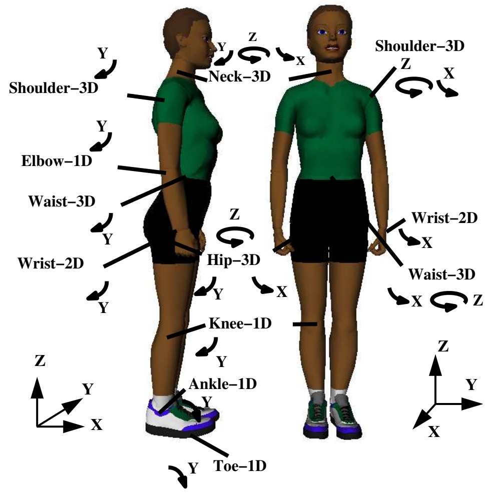

Figure 4: The controlled degrees of freedom for the dynamic model of the runner. The runner models have 17 body segments and 30 controlled degrees of freedom. The bicyclist models are similar, but have only a single degree of freedom at the neck and hips and no degree of freedom at the toes. The direction of the arrows indicates the positive direction of rotation for each degree of freedom.

the running motions of men, women, and children, and we use this information as a point of comparison for the simulated running motion in the last section of the paper. Finally, the biomechanics literature provides data on the anthropomorphic parameters of men, women, and children of various ages that we used in developing our models. In the computer graphics area, these data have been used extensively in the Jack system developed at the University of Pennsylvania[2] to allow the construction of models of various anthropomorphic dimensions for ergonomic analysis and human factors engineering as well as distributed interactive simulation.

In the remainder of the paper, we describe the algorithm for adapting control systems for steady-state running, bicycling, and physically realistic morphing. We briefly describe the dynamic models and the control systems. The algorithm for adapting the control system through scaling and tuning is described next, followed by an analysis of the performance of the algorithm.

3 Dynamic Simulation

The animated motions described in this paper are computed using dynamic simulation. Each simulation contains the equations of motion for a rigid-body model of a human or humanlike character, constraint equations for the interaction with the ground, parameterized control algorithms for running or bicycling, a graphical image for viewing the motion, and a user interface for changing the parameters of the simulation. During each simulation time step, the control algorithm computes desired positions and velocities for each joint based on the state of the system and the requirements of the task as specified by the user. Proportional-derivative servos compute joint torques based on the desired and actual value of each joint. The equations of motion of the system are integrated forward in time, taking into account the internal joint torques and the external forces and torques from interactions with the ground plane or other objects. A description of the graphical and dynamic models and an overview of the control algorithms are given below (for details see [14]).

3.1 Graphical and Dynamic Models

The models we used to animate the running and bicycling motions were constructed from rigid links connected by rotary joints with one, two, or three degrees of freedom. The human graphical models were created by modifying models purchased from Viewpoint Datalabs, and the imaginary characters were modeled in Alias. Intermediate models for the morph scene were created by first shrinking a cube or cylinder onto the polygonal model for each body part. Models created in this fashion have the same number of vertices, and corresponding vertices can be linearly interpolated from one character to the other to perform a morph or create intermediate models. This simple algorithm for three-dimensional morphing is not as elegant as algorithms published in the literature (for example, [16]), but it provides acceptable results for our application because the polygonal models for the body parts are generally convex and because a close correspondence exists between the physical characteristics of the models.

The dynamic models were derived from graphical models by computing the mass and moment of inertia of each body part using algorithms for computing the moment of inertia of a polygonal object of uniform density[18] and density data measured from cadavers[9]. The mass parameters of the four models used to test the scaling and tuning algorithms are given in figure 3.

The controlled degrees of freedom of the models are shown in figure 4. Each internal joint has a torque source that allows the control algorithms to apply a torque between the two links that form the joint. The equations of motion for each system were generated using a commercially available package[28, 31]. The points of contact with the ground are modeled using constraints with Baumgarte stabilization[3].

3.2 Running Control Algorithms

Running is a cyclic behavior in which the legs swing fore and aft and provide support for the body in alternation. Because the legs perform different functions during the phases of the locomotion cycle, the muscles are used for different control actions at various times in the cycle. For example, when the foot of the simulated runner is on the ground, the ankle, knee, and hip provide support and balance. During the flight phase, a leg is swung forward in preparation for the next touchdown. These distinct phases and corresponding changes in control actions make a state machine a natural tool for selecting the control actions that should be active at a particular time. The states correspond to the points of contact with the ground: flight, heel contact, heel and metatarsus contact, and metatarsus contact.

To generate steady-state running, the control system must maintain three parameters: forward speed, flight duration, and balance. Each state includes control laws that compute desired values for each joint with the goal of controlling those three parameters of the running cycle. Because they control the high-level attributes of the running motion, parameters of these control laws will be adjusted in the search step described in Section 4.3.

During flight, one leg is swung forward in anticipation of touchdown. The foot is positioned at touchdown to correct for errors in forward speed and to maintain balance. The disturbances caused by the impact of touchdown can be reduced by decreasing the relative speed between the foot and the ground at touchdown. This technique is commonly called *ground speed matching*. In our control system, ground speed matching is accomplished by swinging the leg further forward in the direction of travel during flight and moving it back just before touchdown.

Flight duration is controlled by extending the ankle and knee joints during stance, causing the heel to lift off the ground and adding energy to the system for the next flight phase. Thrust is initiated when the metatarsus is in contact with the ground and the hip has moved a certain distance in front of the foot.

Throughout stance, proportional-derivative servos are used to compute torques for the hip joint of the stance leg that will cause the attitude of the body (roll, pitch, and yaw) to move toward the desired values. The desired angle for pitch is inclined slightly forward and the desired angles for roll and yaw are zero.

The control laws for forward speed, flight duration, and balance result in desired values for each joint. Proportional-derivative servos are used to control the position of all joints. For each internal joint the control equation is

$$\tau = k(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}) \tag{1}$$

where θ is the angle of the joint, θ_d is the desired angle, $\dot{\theta}$ is the velocity of the joint, $\dot{\theta}_d$ is the desired velocity, and k and k_v are the proportional and derivative gains.

3.3 Bicycling Control Algorithms

The underlying principles of the control system for the bicyclist are similar to those of the runner. High level control laws are used to compute desired values for each joint with the goal of controlling balance, speed, and facing direction. The rider navigates by applying forces to the handlebars and controls speed by applying forces to the pedals. Proportional-derivative servos are used to control the positions of all joints.

4 Scaling and Tuning

For motions as complex as running and bicycling, the parameters of a control system must be carefully tuned to match the physical attributes of each character. A control system that has been tuned for one dynamic model will not, in general, work on a different dynamic model. Figure 5 shows the result of using the controller designed for the male model to control the running motion for a model whose characteristics fall halfway between those of the male and female models. The model fails to run because the control system gains and the desired forward speed that are appropriate for the male model are too high for the smaller, lighter intermediate model. This section describes how the control system of one character can be adapted to produce similar motion in a different character through a two step process involving scaling and search.

4.1 Geometric Scaling

Control systems for geometrically similar characters can be scaled based on size alone[25]. Two characters are geometrically similar if the model for one can be obtained by scaling the model of the other by a constant factor. For example, a model of the female runner can be geometrically scaled to be the same height as the child by scaling the position of each model vertex by the ratio of the characters' heights. A control system can then be generated for the scaled model based on the scaling rules in figure 6. These rules are derived assuming that scaling is uniform in all dimensions and that densities and acceleration due to gravity are the same for the two characters. For example, a geometric scaling factor L and the assumption that acceleration due to gravity (in units of $\frac{L}{T^2}$) is constant means that time must scale as $L^{1/2}$. This relationship in turn implies that desired velocity for the scaled character should scale as $\frac{L}{T} = \frac{L}{L^{1/2}} = L^{1/2}$.

The scaling rules in figure 6 allow us to adapt each parameter of the control system to account for the physical differences between two geometrically similar characters. We apply geometric scaling to the following parameters:

- The state of the system (joint angles, position and orientation of the torso, and their derivatives)
- Gains for all proportional-derivative joint servos
- Values used to control the motion such as desired forward speed and desired time of flight
- Constants referenced by the control system such as the desired clearance of the foot during flight
- The integration time step for the dynamic simulation

Once the designer of the behavior has identified the control parameters and their units, this process is automatic, and requires only that the animator supply the geometric model for the new character.

In general, two characters will not be geometrically similar, as figure 7 demonstrates. Figure 3 further illustrates this point by giving the scaling factors based on various parameters of the dynamic models. Although these scaling factors vary substantially, geometric scaling can serve as a good first approximation, capturing some of the physical differences between two characters. Figure 8 illustrates the effects of geometric scaling on stable running time for characters with physical characteristics ranging from the male model to the female model. For example, the 50% model illustrated in figure 5 runs for less than 4 seconds when the control system has not been scaled, but geometric scaling is sufficient to give it a stable running gait (i.e. the model runs for at least 100 seconds without falling).

Because geometric scaling is approximate for the models that we would like to animate, we must pick an appropriate scaling factor. For the runner, we tried scaling factors based on the height and the leg length of the characters. Leg length was found to be a more reliable scaling factor, perhaps because the control and appearance of the running motion depend much more on the lower body than on the upper body. For the bicyclist, the ratio of wheel radii was found to be a more reliable measure than either height or leg length.

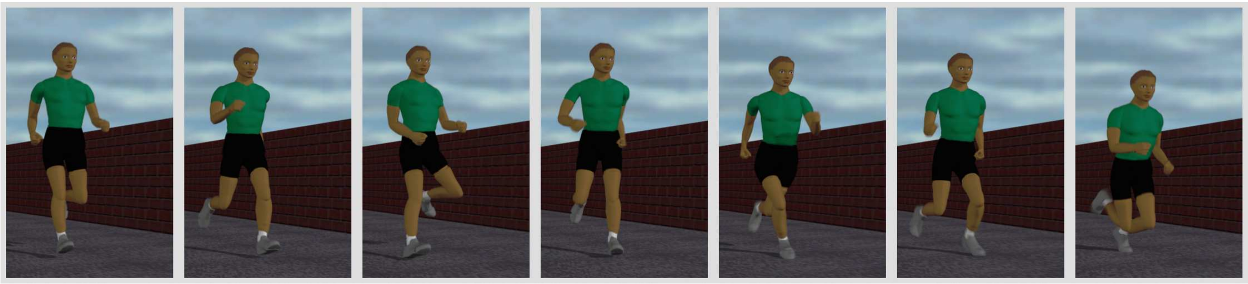

Figure 5: Images showing the result of using the control system designed for the man to control the running motion for a model that is halfway between the man and the woman. The images show successive touchdowns (1.8, 2.17, 2.53, 2.8, 3, 3.53 seconds).

Quantity	Units	Geom. Scaling	Mass Scaling
Basic variables			
length	L	L	–
time	T	$L^{1/2}$	–
force	F	L^3	M
torque	FL	L^4	IL^{-1}
Motion variables			
displacement	L	L	–
velocity	LT^{-1}	$L^{1/2}$	–
acceleration	LT^{-2}	1	–
angular displacement	–	1	–
angular velocity	T^{-1}	$L^{-1/2}$	–
angular acceleration	T^{-2}	L^{-1}	–
Mechanical parameters			
mass	$FL^{-1}T^2$	L^3	M
stiffness	FL^{-1}	L^2	ML^{-1}
damping	$FL^{-1}T$	$L^{5/2}$	$ML^{-1/2}$
moment of inertia	FLT^2	L^5	I
torsional stiffness	FL	L^4	IL^{-1}
torsional damping	FLT	$L^{9/2}$	$IL^{-1/2}$

Figure 6: Scaling rules that capture differences in size, mass, and moment of inertia. The geometric scaling factor is derived assuming uniform scaling by factor L in all dimensions (geometric similarity), and assuming that the acceleration of gravity and the density of the material are invariant to scale. The mass scaling factor assumes also that mass scales by factor M and moment of inertia scales by factor I. A "–" in the mass scaling column indicates that there is no change in the scaling rule.

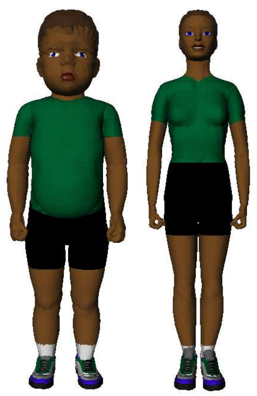

Figure 7: Woman scaled to the height of a 3-year-old child.

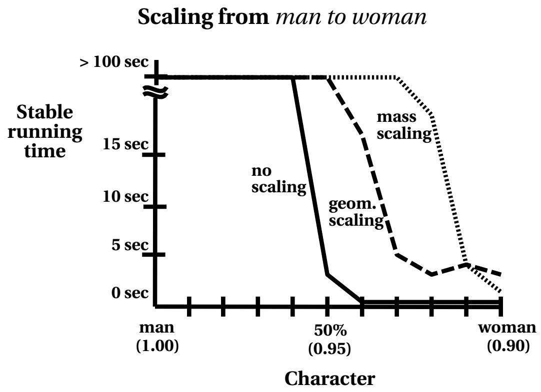

Figure 8: Effects of scaling on stable running time for characters with physical characteristics ranging from the man to the woman. The control system for each character was derived in three ways: (1) directly from that of the man, (2) by geometric scaling, and (3) by mass scaling. The scaled control systems were not tuned. Scaling factors for the characters are shown in parentheses.

4.2 Mass Scaling

An additional scaling step is needed to adapt the control system to physical differences not captured through geometric scaling. Differences in proportion between the man and woman, for example, are sufficient to prevent a geometrically scaled control system for the man from producing steady-state running in the woman (figure 8). Many of these differences can be accounted for through mass scaling rules, which correct for differences in masses and relative moments of inertia.

As an example, we first derive the mass scaling rule for torsional stiffness. For a system with one rigid body and one angular joint, torque τ at the joint produces angular acceleration $\ddot{\theta}$:

$$\tau = I\ddot{\theta} \tag{2}$$

where I is the moment of inertia of the body about the axis of rotation. This joint is controlled with gains k and k_v as in equation 1.

Based on equations 1 and 2, a second system with moment of inertia I' and the same link lengths as the first system could be controlled to have the same angular positions, velocities, and accelerations over time by adjusting the gains as follows:

$$k' = k\left(\frac{I'}{I}\right) \quad \text{and} \quad k'_v = k_v\left(\frac{I'}{I}\right) \tag{3}$$

Given appropriate ratios for the moment of inertia terms, equation 3 expresses the scaling relationships for torsional stiffness and damping that are required to account for differences in moments of inertia between two characters that have the same link lengths but very different inertia properties.

If link lengths also differ, equation 3 can be combined with the rules for geometric scaling. For torsional stiffness:

$$k' = kL^4 \left(\frac{I'}{IL^5}\right) = k\left(\frac{I'}{I}\right) L^{-1} \qquad (4)$$

Torsional stiffness k has been scaled by geometric scaling factor L^4, and moment of inertia I has been scaled by geometric scaling factor L^5.

The complete set of rules for scaling based on mass and moment of inertia properties is given in the rightmost column of figure 6. All of the mass scaling rules in figure 6 can be derived from the single design decision to eliminate the observable effects of differing mass distributions. Gains are scaled to keep angular and positional terms the same over time for two systems of the same scale. The derivation of all mass scaling rules is similar to that for torsional stiffness, although terms related to linear motion rather than angular motion will scale based on mass ratios rather than moment of inertia ratios. For example, the hands of the cyclist are attached to the handlebars with linear springs, and the stiffness of those springs will scale based on the ratio of relevant masses of two characters.

Applying the mass scaling rules requires selecting the relevant body segments for each gain so that mass ratio M or moment of inertia ratio I can be computed. The relevant body segments are determined for each state based on knowledge of the behavior. For example, during the flight phase of the running motion, torque applied at the hip is primarily responsible for swinging the entire leg forward, and so the flight gains for the hip should depend on the moment of inertia of the entire leg about the hip.

For each gain in the control systems for the running and cycling behaviors, the appropriate mass or moment of inertia was determined by identifying the body parts whose motion is significantly affected by that gain and summing their masses or their moments of inertia about the joint. Identification of the body segments most closely associated with each gain is something that must be done once by the designer of the behavior. Mass scaling rules can then be automatically applied to create a new control system based only on the geometric model of a new character.

Figure 8 illustrates the effect of mass scaling on characters ranging from the male model to the female model. Application of mass scaling rules can result in a substantial improvement over geometric scaling alone. For example, a character generated by transforming the physical attributes of the male model 70% of the distance toward the female model does not run a single step when the control system of the man is used without scaling or tuning, runs for 5 seconds with geometric scaling based on the ratio of leg lengths of the man and the new model, and runs with a stable gait for at least 100 seconds when mass scaling rules are applied. This plot is typical of the results we observed when scaling between any two characters, although when the physical characteristics were very different (e.g. when scaling the control system from the woman to the child) the curves would be shifted to the left on the x-axis of the plot.

Although the mass scaling rules result in a far better control system than geometric scaling alone, these rules are still approximate because they depend on the choice of a single ratio for each gain. One source of error is introduced by scaling each gain based only on a subset of the body parts, although torque applied at a joint actually affects all parts of the body. A second source of error is that the moment of inertia of a set of body parts about a joint changes as the joint angles change. For example, the moment of inertia of the leg about the hip is smaller when the leg is bent than when it is straight. To obtain a single scaling factor for each gain, we calculated the moments of inertia with the body standing upright in the position shown in figure 4. This approximation is similar to the assumption underlying the use of a proportional-derivative servo with constant gains despite changes in joint angles.

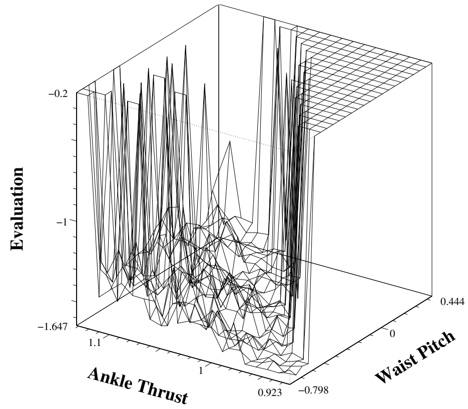

Figure 9: The evaluation function used to tune the runner control system has many local minima. This plot shows a sampling of the evaluation function as two of the tuning parameters responsible for body attitude and thrust are varied. The front corner of the figure shows a near optimal solution for this part of the search space. Values of -0.2 for the evaluation function indicate that the runner did not run for the desired running time.

4.3 Tuning the Motion

Development of a control system for steady-state performance of a behavior requires a final search step to compensate for the approximations inherent in the mass scaling rules. Because the mass scaling rules provide us with a good initial guess at a control system, a search over a small number of high-level parameters is sufficient to produce a control system for stable, repeatable motion.

4.3.1 Running

To constrain the search process for the running behavior, we restricted the search to five high-level control parameters covering forward speed, body attitude, and flight duration. One parameter, ground speed matching, controlled foot velocity relative to the ground at landing and affected the running speed. The second parameter controlled the desired pitch angle of the body during stance. Three parameters affected the duration of the flight phase: the timing of thrust and the extension of the ankle and knee during stance.

The direction of the search was determined by an evaluation function designed to capture the quality and appearance of the running motion. Values assigned to the search parameters defined a control system. The runner was commanded to run for a fixed duration (15 seconds in our experiments), and the resulting motion was evaluated. The evaluation function contained penalties for falling, for errors in velocity, for head acceleration, and for deviations in roll, pitch, and yaw from one stride to the next. Penalty coefficients could be adjusted to alter the style of the motion. For example, the evaluation function was adjusted to convert a stride where the feet of the runner were barely skimming over the ground to a stride that was more bouncy, to pull a runner out of a crouched running stride, and to eliminate a limp from a running character.

The search space for this problem contains a large number of local minima. For example, figure 9 shows the evaluation function as two of the search parameters controlling body attitude and thrust are varied. The local minima in this plot result from the fact that the optimization function combines a variety of criteria that together provide a high-level evaluation of a complex motion. In an informal evaluation of search techniques, simulated annealing[24] appeared to provide better results on this search space than Powell's method, a line-search technique. Simulated annealing was used to tune the control system for the characters described in this paper. A typical search required approximately 1500 evaluations and 15 hours of processing time on an SGI R8000 computer.

The tuning process required several stages, because scaling the control system of one character directly to reflect the parameters of a second character did not, in general, result in a stable running motion for the second character. To tune the control systems, we used intermediate characters that created smaller jumps in the physical parameters. The intermediate characters were then tuned in sequence. For example, in creating a control system for the woman, we scaled and tuned control systems for intermediate characters at 50%, 70%, and then 90% of the distance from the man to the woman. Similarly, two intermediate characters were used in tuning the control system of the child from that of the woman, and three intermediate characters were used in tuning the control system of the imaginary character from that of the man. We obtained good performance from the tuning process when a scaled, but untuned, control system would allow a character to run for approximately 10 seconds without falling. Such a heuristic could be used to determine automatically which intermediate characters should be created.

The results of scaling and tuning a control system for the man to control a woman, child, and imaginary character are shown in figure 10 along with a sequence of video frames showing a four-year-old child. Each of the animated characters had a stable running motion, measured by verifying that they would run with a stable gait for at least 100 seconds.

4.3.2 Bicycling

The search process for the bicycling behavior was similar to that for running. To constrain the search process, we grouped search parameters into high level categories for control of roll and yaw of the bicycle, control of the handlebars and pedals, and stiffness of the arms and shoulders. The bicyclist was commanded to ride a challenging course involving straight riding, turning, and speed changes, and the evaluation function included penalties for falling, excessive yaw velocity, head acceleration, and poor tracking of the desired course.

We used scaling and search to obtain a bicycling control system for an imaginary character from a working control system for a man (figure 11). The scaling and tuning process worked very well for this problem, requiring no intermediate characters, although we found that the bicycle design had to be well adapted to the character. If the bicycle was too heavy or too light, or if the handlebars could not be reached in a “comfortable” position, the resulting motion was poor. For an integrated system such as this, the constraints of physical realism may require the animator to experiment with different bicycle designs.

4.4 Metamorphosis

The algorithm described here can be used not only for adapting control systems to achieve steady-state running but also to perform an on-line metamorphosis from one model to another. To perform the metamorphosis, the graphical models, the physical models, and the control system are interpolated to create a dynamic simulation where one model is transformed to another over a period of time. Figure 2 shows the transition from a running man to a running woman over a period of 20 seconds.

The physical metamorphosis from the man to the woman was performed by creating nine equally spaced dynamic models between the man and the woman and linearly interpolating the limb lengths, masses, and moments of inertia to create a new dynamic model for each simulation step. The initial control system was created in a similar fashion by interpolating between the scaled parameters for each of the nine intermediate models. This initial control system was not stable, and the morphing runner fell over before it had completed the transition to the woman model. As in the adaptation of the steady-state control system, five parameters of the running motion were tuned to achieve stable running for the 20-second morph and steady-state running using the female model for a subsequent 7 seconds. The initial values for the five parameters were those that were found by tuning the running motion of the male model. The tuning process then determined a rate of change for each parameter, creating a linear ramp for each parameter as the dynamic system made the transition from the man to the woman.

The metamorphosis shown in figure 2 was performed over a 20-second period. A faster morph would be preferable, but physically realistic metamorphosis is difficult for several reasons. The primary difficulty is that the dynamic system, while physically realistic at each moment in time, is changing in a way that violates physical laws. For example, the body is changing in mass during the flight phase so angular momentum is not conserved. Similarly, while the foot is on the ground, the ground contact forces are applied to a constantly changing model. Physically realistic morphing of the control system is also difficult because the running is not steady-state. The control system has step-to-step goals of maintaining forward speed, flight duration, and balance and immediate goals of moving the joints to the right angles. When the dynamic model is continuously changing, these two sets of goals and the choice of corresponding gains are no longer synchronized. For example, forward speed is controlled by the position of the foot at touchdown with the new desired speed achieved by liftoff. When the dynamic model is morphing, the system in effect at the time the desired foot position angle was selected is no longer active when the new speed is achieved, resulting in errors in the control of forward speed.

5 Discussion

One goal of this research is to demonstrate that simulations of human motion can be automatically adapted to new dynamic models while maintaining the important properties of the running motion. Figure 10 compares video footage of a human child with images of a simulated child, and figure 12 compares biomechanical data from the literature[22, 10] with measurements of the simulated runners. We found that our simulation results were very similar to the quantitative data from the human runners. These results also captured some of the male/female and adult/child differences found in the literature. A comparison of the video footage for the child, however, showed that the human child had more variability in his motion.

We have presented algorithms that allow an animator to generate running or bicycling motion for several different dynamic systems in an automatic fashion. By dividing the algorithm into two stages, scaling and tuning, we chose a hybrid approach based on explicit knowledge about the system and on automatic search. We made this decision because of our intuition that a fully automatic approach that attempted to tune approximately 100 control gains would not be successful. We also felt that an exclusively knowledge-based approach would fail because our understanding of human running and bicycling, as represented by the control laws, is far from complete. The tuning process allows for some imprecision in the exact form of the control laws by adjusting several important parameters for an individual model.

To be widely applicable, this approach to adapting control systems must be independent of the particular behavior that we chose for our experiments. Would the same approach work for adapting control systems for diving and vaulting to new models? Both the scaling laws and the selection of parameters for tuning are based on

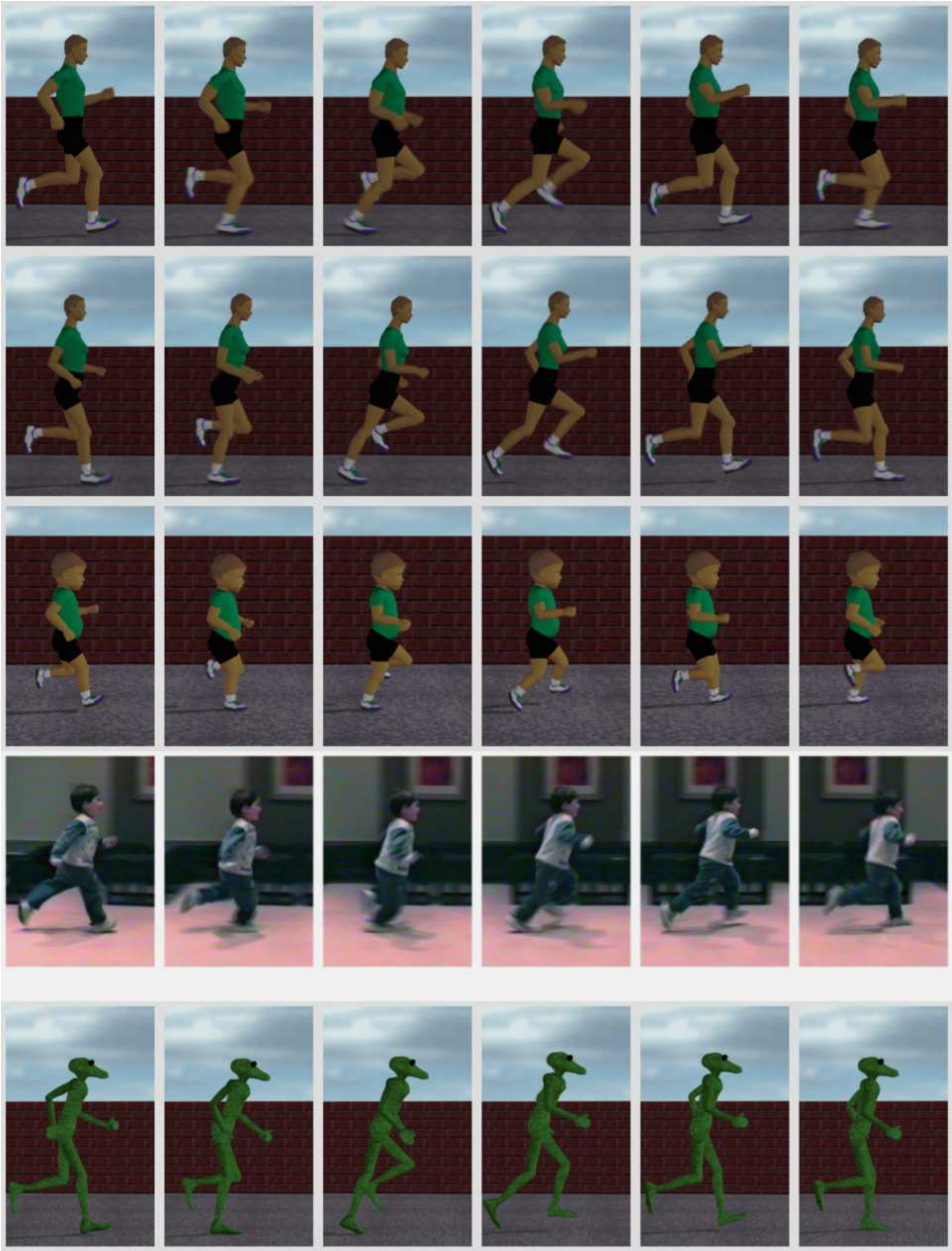

Figure 10: Images of a running man, woman, child, and imaginary character, and a video of a four-year-old human child. The human child weighs 20 kilograms and is 1.07 meters tall. In each case, the spacing of the images is 0.066 seconds.

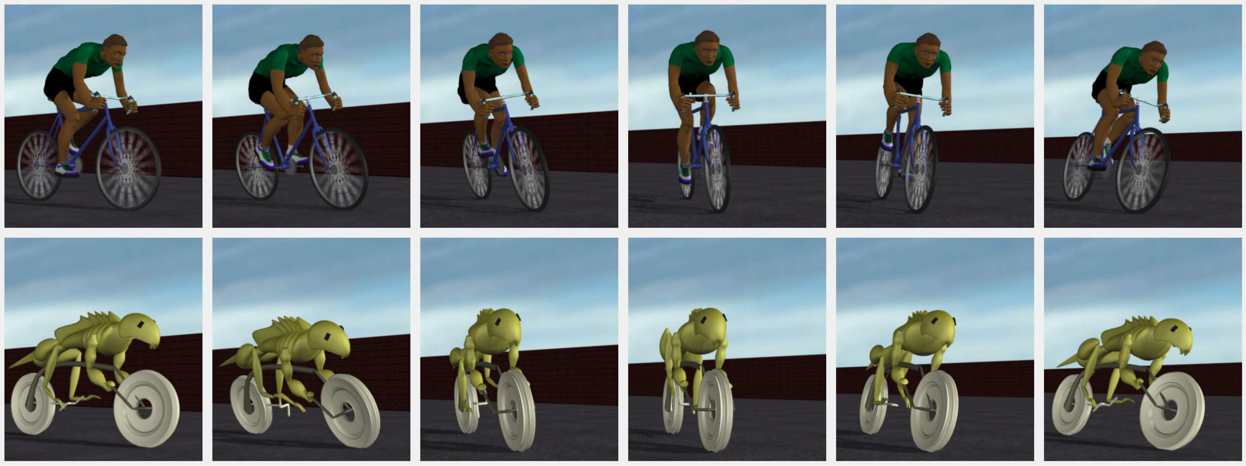

Figure 11: Images of a bicycling man and imaginary character following a slalom course. The spacing of the images is 1 second.

Quantity	Man	Woman	Child
Speed (m/s)			
human	4.83	4.83	2.13
simulated	4.41	4.45	2.52
Step Frequency ($steps/s$)			
human	2.92	3.05	3.8
simulated	3.18	3.33	4.3
Step Length (m)			
human	1.66	1.60	0.56
simulated	1.39	1.33	0.58
Stance Time / Step Time ($\%$)			
human	57.9	55.2	68.5
simulated	59.6	54.3	63.3

Figure 12: A comparison of data from the biomechanical literature with data recorded from our simulated runners. The match between the human child and the simulated child is within the variability between subjects. Simulated data from the male and female runners captures the male/female differences found in the literature.

information about the control system and behavior. This information, however, is readily available to the designer of the behavior. For example, geometric scaling requires that we know the units of each gain in the control system, but that information can be easily determined by examining the units of each control equation. Scaling based on mass and moment of inertia requires that we identify the bodies that are most affected by a particular joint gain. We assumed that the lighter of two bodies (or chains of bodies) would be most affected by joint motion except when that body was in contact with the ground. Only the selection of parameters for tuning is specific to the behavior. We chose higher-level parameters that we thought best represented the important properties of each motion: forward speed, flight duration, and balance for the runner and balance, support, and control of handlebars and pedals for the bicyclist. In principle, it should be possible to pick a similar set of parameters for other behaviors.

To be widely applicable, the scaling and tuning process must also be very easy for animators and researchers to use. Once the designer of the behavior has incorporated the scaling laws into the behavior and selected a set of search parameters, the scaling and tuning process can be completely automatic, requiring only that a geometric model for the new character be supplied. In practice, we found that creating appealing motion may require experimenting with the evaluation function, but this could easily be done without knowledge of the control system for that behavior. Although the search process is time consuming, it makes scaling accessible to users with no knowledge of the control system.

To be useful as part of a modeling and animation package, an algorithm for automatically adapting control systems must be robust for a wide variation in models. The child is markedly different from the woman or man, but there are certainly parameter changes that are too extreme to be accounted for by the combination of scaling laws and tuning presented here. For example, creatures whose physique demands a fundamentally different style of bipedal running could not be controlled using this approach. Birds and bipedal dinosaurs, for example, are “toe-strike” runners rather than “heel-strike” runners. The control laws for toe running probably differ not only in parameter values but also in structure from the control laws for heel-strike running.

The scaling rules presented in this paper were applied to control systems developed for simulation, but they apply equally well to data obtained through motion capture. Motion capture data describing, for example, the position of the torso and the joint angles of a human actor over time can be scaled according to the rules in figure 6. Only torso positions and the length of time between data points are affected. Because joint angle positions are dimensionless, they do not scale.

Scaling is not sufficient for adapting motion capture data to fit new characters, however. Kinematic constraints for characters with different relative link lengths still need to be resolved. For example, the feet of the runners must make contact with the ground, and the feet of the bicyclist must touch the pedals. A more subtle issue is the expected variation of motion with changes in physical characteristics such as mass. This variation is not captured by the scaling rules, which attempt to generate identical motions over time for characters with the same geometric scaling parameters but different masses or mass distributions. Perhaps simulation results demonstrating physical responses to changes in physical properties will provide some insight into more sophisticated scaling rules to capture effects such as these.

The animations described in this paper can be seen on the WWW at: `http://www.cc.gatech.edu/gvu/animation/`.

Acknowledgments

The authors would like to thank Elizabeth de Goursac for her help in developing the early versions of the tools used in this paper, James O'Brien for the use of his rendering and modeling software, Joe Marks for the loan of his child, Christopher, and Sherry Strickland and Victor Zordan for use of their Alias models. We would also like to thank the anonymous reviewers for their detailed comments. This project was supported in part by NSF NYI Grant No. IRI-9457621, by Mitsubishi Electric Research Laboratory, and by a Packard Fellowship.

References

[1] J. Auslander, A. Fukunaga, H. Partovi, J. Christensen, L. Hsu, P. Reiss, A. Shuman, J. Marks, and J.T. Ngo. Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics*, 14(4):311–336, October 1995.

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

[3] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[4] L. S. Brotman and A. N. Netravali. Motion interpolation by optimal control. In J. Dill, editor, *Computer Graphics (SIGGRAPH 88 Proceedings)*, volume 22, pages 309–315, August 1988.

[5] A. Bruderlin and T. Calvert. Interactive animation of personalized human locomotion. In *Proceedings of Graphics Interface '93*, pages 17–23, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.

[6] A. Bruderlin and T. W. Calvert. Goal-directed, dynamic animation of human walking. In *Computer Graphics (SIGGRAPH 89 Proceedings)*, volume 23, pages 233–242, July 1989.

[7] A. Bruderlin and L. Williams. Motion signal processing. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 97–104. ACM SIGGRAPH, Addison Wesley, August 1995.

[8] M. F. Cohen. Interactive spacetime control for animation. In E. E. Catmull, editor, *Computer Graphics (SIGGRAPH 92 Proceedings)*, volume 26, pages 293–302, July 1992.

[9] W. T. Dempster and G. R. L. Gaughran. Properties of body segments based on size and weight. *American Journal of Anatomy*, 120:33–54, 1965.

[10] V. L. Fortney. The kinematics and kinetics of the running pattern of two-, four-, and six-year-old children. *Research Quarterly for Exercise and Sport*, 54(2):126–135, 1983.

[11] M. Girard and A. A. Maciejewski. Computational modeling for the computer animation of legged figures. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH 85 Proceedings)*, volume 19, pages 263–270, July 1985.

[12] M. Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 139–148, Providence, RI, April 1997.

[13] R. Grzeszczuk and D. Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 63–70. ACM SIGGRAPH, Addison Wesley, August 1995.

[14] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 71–78. ACM SIGGRAPH, Addison Wesley, August 1995.

[15] P. S. Huang and M. van de Panne. A planning algorithm for dynamic motions. In *7th Eurographics Workshop on Animation and Simulation*, pages 169–182, 1996.

[16] J. R. Kent, W. E. Carlson, and R. E. Parent. Shape transformation for polyhedral objects. In E. E. Catmull, editor, *Computer Graphics (SIGGRAPH 92 Proceedings)*, volume 26, pages 47–54, July 1992.

[17] J. Laszlo, M. van de Panne, and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH 96 Proceedings*, Annual Conference Series, pages 155–162. ACM SIGGRAPH, ACM Press, August 1996.

[18] S. Lien and J. T. Kajiya. A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. *IEEE Computer Graphics and Applications*, 4(10):35–41, 1984.

[19] Z. Liu and M. Cohen. Decomposition of linked figure motion: Diving. In *5th Eurographics Workshop on Animation and Simulation*, 1994.

[20] Z. Liu, S. J. Gortler, and M. F. Cohen. Hierarchical spacetime control. In *SIGGRAPH 94 Proceedings*, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994.

[21] T. A. McMahon. *Muscles, Reflexes, and Locomotion*. Princeton University Press, Princeton, 1984.

[22] R. Nelson, C. Brooks, and N. Pike. Biomechanical comparison of male and female distance runners. *Annals of the NY Academy of Sciences*, 301:793–807, 1977.

[23] J. T. Ngo and J. Marks. Spacetime constraints revisited. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH 93 Proceedings)*, volume 27, pages 343–350, August 1993.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, New York, 1992.

[25] M. H. Raibert and J. K. Hodgins. Animation of dynamic legged locomotion. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH 91 Proceedings)*, volume 25, pages 349–358, July 1991.

[26] R. Ringrose. Simulated creatures: Adapting control for variations in model or desired behavior. M.S. Thesis, Massachusetts Institute of Technology, 1992.

[27] C. F. Rose, B. Guenter, B. Bodenheimer, and M. F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *SIGGRAPH 96 Proceedings*, Annual Conference Series, pages 155–162. ACM SIGGRAPH, Addison Wesley, August 1996.

[28] D. E. Rosenthal and M. A. Sherman. High performance multibody simulations via symbolic equation manipulation and Kane's method. *Journal of Astronautical Sciences*, 34(3):223–239, 1986.

[29] K. Sims. Evolving 3d morphology and behavior by competition. In *Artificial Life IV*, pages 28–39, 1994.

[30] K. Sims. Evolving virtual creatures. In *SIGGRAPH 94 Proceedings*, Annual Conference Series, pages 15–22. ACM SIGGRAPH, ACM Press, July 1994.

[31] Symbolic Dynamics Inc. *SD/Fast User's Manual*. 1990.

[32] X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *SIGGRAPH 94 Proceedings*, Annual Conference Series, pages 43–50. ACM SIGGRAPH, ACM Press, July 1994.

[33] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 91–96. ACM SIGGRAPH, Addison Wesley, August 1995.

[34] M. van de Panne and E. Fiume. Sensor-actuator networks. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH 93 Proceedings)*, volume 27, pages 335–342, August 1993.

[35] M. van de Panne, E. Fiume, and Z. Vranesic. Reusable motion synthesis using state-space controllers. In F. Baskett, editor, *Computer Graphics (SIGGRAPH 90 Proceedings)*, volume 24, pages 225–234, August 1990.

[36] M. van de Panne, E. Fiume, and Z. G. Vranesic. Optimal controller synthesis using approximating-graph dynamic programming. In *Proceedings of the American Control Conference*, pages 2668–2673, 1993.

[37] M. van de Panne and A. Lamouret. Guided optimization for balanced locomotion. In *Eurographics Workshop on Computer Animation and Simulation '95*, pages 165–177, 1995.

[38] A. Witkin and M. Kass. Spacetime constraints. In J. Dill, editor, *Computer Graphics (SIGGRAPH 88 Proceedings)*, volume 22, pages 159–168, August 1988.

[39] A. Witkin and Z. Popović. Motion warping. In *SIGGRAPH 95 Proceedings*, Annual Conference Series, pages 105–108. ACM SIGGRAPH, Addison Wesley, August 1995.

[40] X. Zhao, D. Tolani, B. Ting, and N. I. Badler. Simulating human movements using optimal control. In *Eurographics Workshop on Computer Animation and Simulation '96*, pages 109–120, 1996.