Generating Natural-looking Motion for Computer Animation

Jessica K. Hodgins
Paula K. Sweeney
David G. Lawrence†
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

The automatic generation of motion for animation remains an unsolved problem in computer graphics. One approach to the problem is to combine physically accurate models with control systems. The user specifies high-level goals and the control system computes the forces and torques that the simulated muscles or motors should exert to cause the model to perform the desired task. In this paper we describe control systems for rigid-body models of humans performing four tasks: pumping a swing, riding a seesaw, juggling, and pedaling a unicycle. We designed the control systems with the goal of producing natural-looking motion, and we discuss the techniques that we used to achieve this goal.
Keywords: Animation, Simulation, Control Theory.

Introduction

We would like to be able to automatically generate natural-looking motion for computer animations based on high-level input from the user. We have explored one solution to this problem: combining control systems with physically accurate models. The user specifies a high-level goal ("ride the unicycle from here to there") and the control system computes the forces and torques that will cause the simulated model to perform the desired task. The combination of a carefully designed control system and a realistic physical model can produce natural-looking motion that has much in common with the motion of the animal or human on which it was modeled.

Physical simulation has been used successfully for generating realistic motion of *passive systems* (Barzel and Barr 1988; Hahn, J. 1988; Terzopoulos and Fleischer 1988; Terzopoulos and Witkin 1988; Baraff 1989, 1991; Pentland and Williams 1989; Kass and Miller 1990; Norton et al 1991; Wejchert and Haumann 1991). Passive systems are those that are acted upon by the environment but have no internal source of energy. Bouncing balls, leaves blowing in the wind, and raindrops falling into puddles are all passive systems. To the extent that the computer program accurately models the physical system and the environment, the resulting motion will be natural. In contrast to passive systems, the systems described in this paper, *active systems*, contain simulated muscles or motors that provide an internal source of energy and allow the systems to act on the environment. Humans, animals, robots, and vehicles are all active systems. Simulation of active systems requires not only a physically realistic model of the system being animated but also a control system or computer algorithm that activates the muscles or motors in such a way that the system performs the desired task.

Because the internal workings of biological control systems are much less well understood than the physical systems themselves, the simulation of active systems is, in general, more difficult than that of passive systems. Active systems have been animated by using springs and dampers as the control system (Miller 1988), by programming a control system for a simplified model of the system being animated (Bruderlin and Calvert 1989), and by simulating the actions of the oscillators found in simple biological control systems (McKenna and Zeltzer 1990).

The design of control systems has not yet been automated for systems of the complexity of those that we would like to animate. Others have begun to address the question of automatic generation of motion in the context of optimization problems and optimal control theory (Witkin and Kass 1988; van de Panne, Fiume, and Vranesic 1990). The potential generality of these approaches makes them among the most interesting new methods for animation of dynamic systems. The potential liability is the growth of the search spaces when applied to more complex systems.

Our approach to hand designing these control systems builds on previous work in the control of legged robots (Raibert and Hodgins 1991; Hodgins 1991; Hodgins and Raibert 1990; Raibert 1986). This work provides us with a number of techniques that aid in the design of control systems for dynamic tasks: state machines for structuring the control laws, low-level control through springs and dampers, and symmetry of the motions as a principle for the design of the higher level control algorithms.

In this paper we describe control systems for rigid-body models of humans performing four tasks: pumping a swing, riding a seesaw, juggling, and pedaling a unicycle. In each case, the simulated models are composed of rigid links connected by rotary or sliding joints. The models are derived from measurements of living humans and cadavers (Meredith 1969a,b; Dempster and Gaughran 1965). Each model has enough degrees of freedom to perform the stipulated task but only a small fraction

†Present address: Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305.

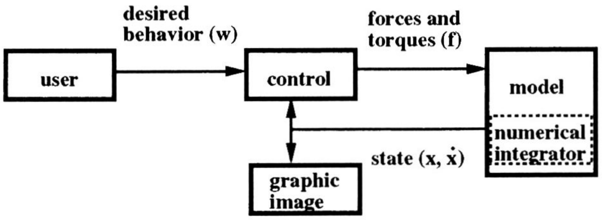

Figure 1: The user specifies the desired behavior at a high level. The control algorithms compute forces and torques that should be applied at each joint based on the state of the system and the desired behavior. The internal forces at the joints and external forces from the environment are applied to the links of the model. The equations of motion are integrated forward in time to produce the new state of the system.

of the number found in humans. The equations of motion for each model were generated with a commercially available package (Rosenthal and Sherman 1986). The package generates efficient subroutines for the equations of motion of the model ($O(n)$ where n is the number of links) using a variant of Kane's method and a symbolic simplification phase.

A state machine provides the underlying structure for the control system for each animation but allows the control laws to change when the dynamics of the system or the user's commands change. For example, the dynamics of the seesaw change when the legs of a rider touch the ground and the control laws must change to match. Similarly, the control laws change when the user switches from controlling the position to controlling the velocity of the unicycle. Each state has specific control laws and the transitions between the states are determined by changes in the state of the system or by changes in the input to the control system.

Proportional-derivative control is used for low-level position control in most of the simulations. The torque exerted at a joint is a function of the error in position and the relative velocity between the links on either side of the joint:

$$\tau = \mathbf{kp}(\theta - \theta_{\mathbf{d}}) + \mathbf{kv}\dot{\theta}$$

where τ is the control torque, θ is the relative joint angle between the links, θ_d is the rest position of the spring, and $\dot{\theta}$ is the relative velocity between the links. The gains, k_p and k_v depend on the mass of the links and the desired stiffness and damping of the joint. This servo has the same effect on the system as a spring and damper where the rest position of the spring is controlled by the control system.

Animations are produced through high-level interactions with the control system. For example, the user specifies the forward speed of the unicycle, how long the juggler should use one pattern before switching to another, or the maximum height the swing will achieve. At each simulation time step, the control system computes forces or torques for each joint based on the state of the system and the requirements of the task. The equations of motion of the system are integrated forward in time, and the resulting motion is displayed in a simple graphical model and recorded for later use in an animation. The layout of the animation system is shown in figure 1.

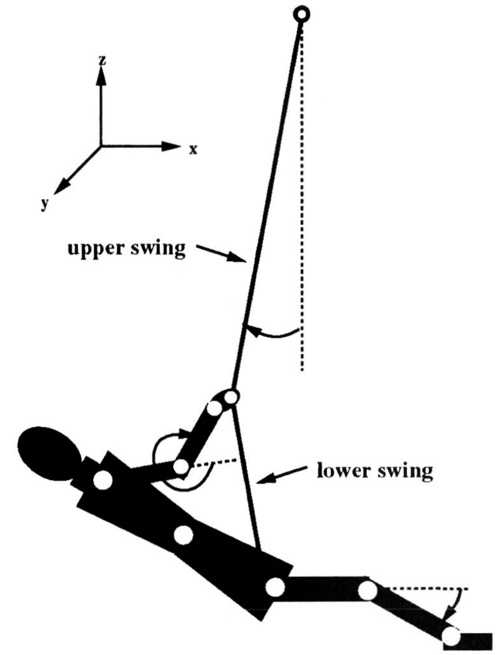

Figure 2: Model used to animate the motion of a human pumping a swing. All joints rotate about the y axis and the motion of the model is constrained to the x-z plane. The angle of the swing, the knee and the elbow are marked to aid in the interpretation of figure 5.

The details of the individual models and control systems are described below.

Pumping a Swing

To pump a swing the control actions of the human must be coordinated with the fore-aft motion of the swing. Figure 2 shows the model used to simulate a human pumping a swing. The links of the model— cylinders, truncated cones, and ellipsoids—are connected by rotary joints at the wrists, elbows, shoulders, waist, hips, knees, and ankles. The swing is modeled as two rods connected by a pivot joint located where the hands attach to the swing. A second pivot joint attaches the bottom of the lower body to the end of the lower swing rod. This joint rotates freely, and the angle between the body and the lower rod of the swing is controlled by the elbow, shoulder, and wrist joints. The grip of the hands on the swing joint is modeled by a pair of orthogonal springs and dampers. The motion of the swing and the human are constrained to the fore-aft plane.

Swinging has been studied for two simple models: a point mass that slides up and down a rigid rod and a system that switches between a double and a compound pendulum (Tea and Falk 1968; Burns 1970; Gore 1970; Gore 1971; McMullen 1972). These models are shown in figure 3. Our model is more complex than either of these, but the simpler models provide some physical intuition about how the control system can move the joints to increase the amplitude of the swinging motion.

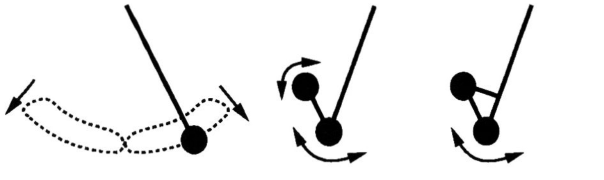

Figure 3: Two simplified models of pumping. The leftmost model, a point mass on a rigid rod, can be pumped by sliding the mass up the rod at the lowest point of the cycle and down at the highest in a pattern like that shown by the dashed line. This pumping action decreases the moment of inertia of the system when all the energy is kinetic and increases it when all the energy is potential. The double/compound pendulum model increases the amplitude of the swing by letting the second pendulum fall backwards at the highest point of the swing. When the fall of the second pendulum is arrested and the model becomes a compound pendulum, the angular velocity of the first pendulum is increased causing the swing to go higher on the next cycle. The motion of a human on a swing is similar to the motion of the double/compound pendulum in that the arms are relaxed at the back of the swing so that the body falls backward and acts like a double pendulum until it is caught by the extended arms. When the arms stop the motion of the body, the system becomes a compound pendulum again but with increased angular velocity.

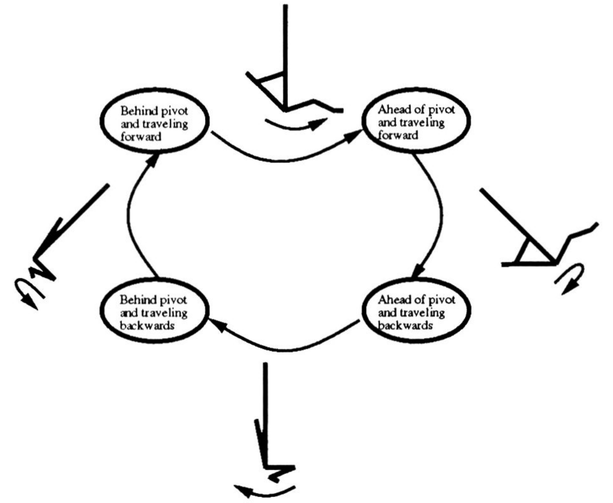

Figure 4: State machine used to control the pumping motion. The state is determined by the angle and velocity of the swing. The state **behind pivot and traveling forward** has the same control laws as the state **ahead of pivot and traveling forward** and serves only to prevent false transitions before the swing has reached its maximum forward position. For the same reason, **ahead of pivot and traveling backwards** has the same control laws as **behind pivot and traveling backwards.**

The state of the control system for pumping depends on the angle and velocity of the swing. The transitions between the states occur when the swing passes through the lowest point of the cycle and when it nears the highest point. The state machine is illustrated in figure 4. The control laws for each state specify the desired angles and the gains for each servo.

The control system uses a proportional-derivative servo or spring/damper system to control each joint. When the swing is moving forward the desired angles cause the body to lean back and the legs to extend. As the swing moves backwards, the desired angles cause the arms to move the body forward towards the lower rod and the legs to bend. The top graph of figure 5 shows the angle of the swing as the control system pumps for twenty seconds, coasts for ten seconds, and pumps again for ten seconds. The middle graph shows the angle of the knee joint as the legs are swung back and forth. The bottom graph shows the angle of the elbow joint as it pulls the body forward and allows the body to fall backwards.

Riding a Seesaw

The seesaw animation has two riders on opposite ends of a plank. The control system varies the rotation of the plank by changing how hard each of the riders pushes off against the ground. The model for the seesaw animation is shown in figure 6. The collision model for the leg and the ground consists of two orthogonal pairs of springs and dampers. The springs are stretched between the touchdown and current positions of the end of the leg. When the lower end of the leg leaves the ground, the springs are disconnected. Like the hands in the swing model, the lower arms are attached to the handles of the seesaw with springs and dampers.

The control for the seesaw uses two independent state machines, one for each rider. The seesaw with two riders is much like a quadruped bounding in place, and the control algorithms are similar to those used for the control of a bounding quadruped (Raibert and Hodgins 1991). The state machine for the control of the seesaw is shown in figure 7. The transitions between the states occur when the legs touch or leave the ground.

The control laws consist of proportional-derivative servos for each joint. The setpoints and gains depend on the state. During **flight**, the leg is moved to an appropriate position for touchdown. During **compression**, a spring at the knee joint stores energy and causes the system to bounce passively. During **extension**, the knee is extended to add energy to the bouncing oscillation of the system. The height of the oscillation can be varied by changing the extension of the knee.

Juggling

To animate juggling, the control system moves the wrist, shoulder, and elbow joints of a model so that the hands catch and throw balls. The user directs the animation by specifying the juggling pattern (cascade, shower, or fountain) and the length of time that the balls are held in the hand (dwell time) or flying through the air (flight time). The control system perturbs the throws so as to maintain the stability of each pattern and produce transitions between the patterns.

Figure 8 illustrates the model used in the juggling simulation. The hands are not anthropomorphic but are of approximately the same size and density as human hands. Collisions are detected between each ball and the surfaces

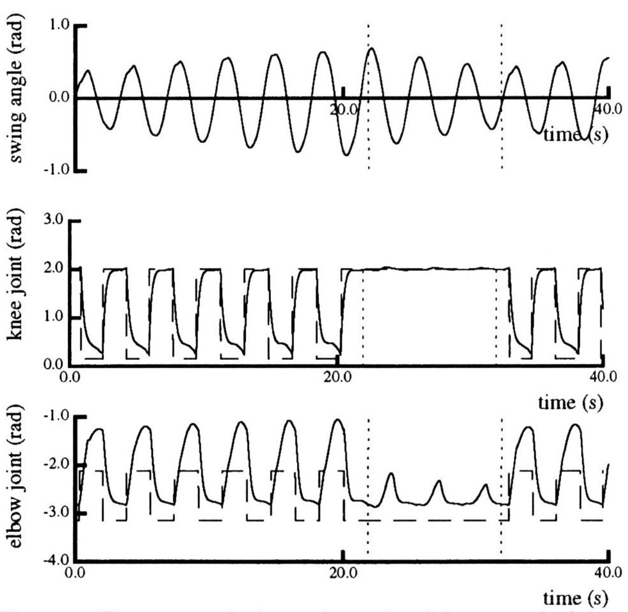

Figure 5: The top graph shows the angle of the top rod of the swing with respect to vertical. The control system pumped the swing for twenty seconds, coasted for ten seconds, and pumped again for ten seconds. The vertical dashed lines indicate when the task changed from **pump** to **coast**. The middle graph shows the movement of the knee joint as the legs are swung back and forth. The bottom graph shows the elbow joint as it pulls the body forward and allows it to fall backwards. The dashed lines in the lower two graphs indicate the desired angle for the knee and elbow. The elbow is pulled away from the desired angle by the weight of the body when the body is leaning back.

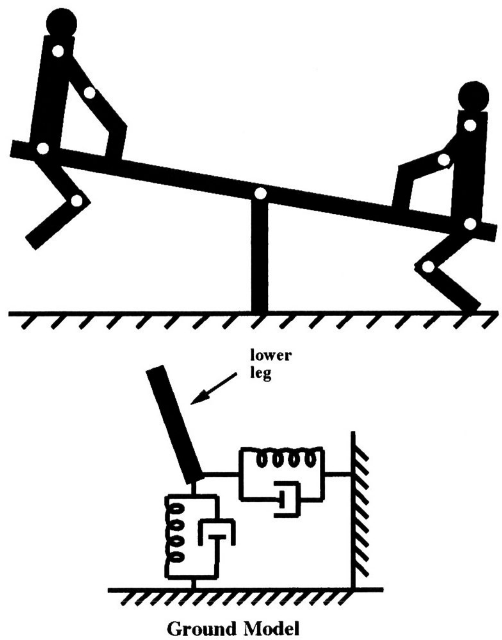

Figure 6: The model for the seesaw animation. The ground model is two orthogonal pairs of springs and dampers. The ends of the arms are also attached to the handles of the seesaw with springs and dampers.

of the fingers, thumbs, and palms and are modeled by a spring normal to the plane of the surface and by two dampers aligned with the surface and perpendicular to each other. Collision forces are applied to both the hand and the ball when the ball is in contact with the hand.

Control System

The basis of all juggling patterns are accurate throws and robust catches. The control system causes the ball to be thrown by generating a desired trajectory for the hand in cartesian coordinates that will accelerate the ball and the hand to the desired velocity by the time they reach the release point. The control system computes torques that will cause the hand to match the speed of the ball as it falls and then close around the ball after contact. This method of catching was implemented for a one degree of freedom juggling robot by Bühler, Koditschek, and Kindlmann (1989).

The control system has four states: **meet**, **decelerate**, **accelerate**, and **follow**. In each state the control system generates a desired trajectory for the hand that will cause it to move to the desired position and arrive with the desired velocity and acceleration. Using inverse kinematics, the desired hand position is transformed to desired positions for the joints of the upper and lower arm and wrist. Proportional-derivative control coupled with a simplified version of the forward dynamics cause each joint to track the trajectory.

During the **meet** state, the hand moves up to meet the falling ball. The control system generates a polynomial trajectory that will cause the hand to meet the ball at the desired catching position with a velocity that matches the velocity of the ball and an acceleration equal to gravity. Matching the descent velocity of the ball reduces the chance that the ball will bounce out of the hand before the finger and thumb close to constrain it.

When the ball is in the hand, the control system is in either the **decelerate** or **accelerate** state. During these states the control system chooses a trajectory that reverses the motion of the hand and ball in the z direction and moves it towards the desired release position. When the hand nears the release position, the finger and thumb open and the hand follows the ball briefly (0.05 sec) in x and y while decelerating in z to prevent collisions which would disturb the trajectory of the ball. After the ball leaves the hand, the control system generates a trajectory that will cause the hand to meet the next ball.

Juggling Patterns

Jugglers commonly use three different three-ball patterns: the cascade, shower, and fountain (Buhler and Graham 1984; Beek 1989). The three patterns are shown in figure 9. In the cascade, each hand throws the balls across to the other hand and each throw passes under the arriving balls. In the shower, the balls move roughly in a circle with the throw from one hand passing above the throw from the other hand. In the fountain, each

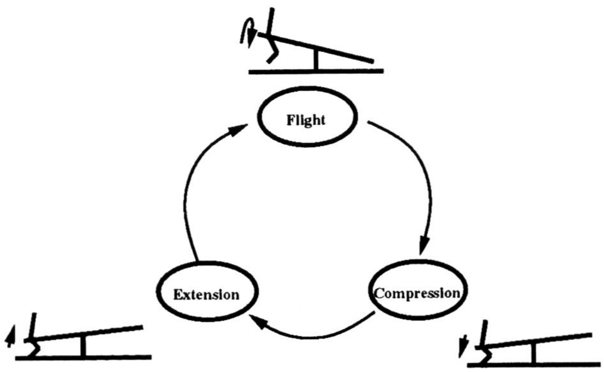

Figure 7: The state machine for one seesaw rider. The transition from **flight** to **compression** occurs when the leg touches the ground. The transition from **compression** to **extension** occurs when the knee joint begins to extend. The transition from **extension** to **flight** occurs when the leg leaves the ground.

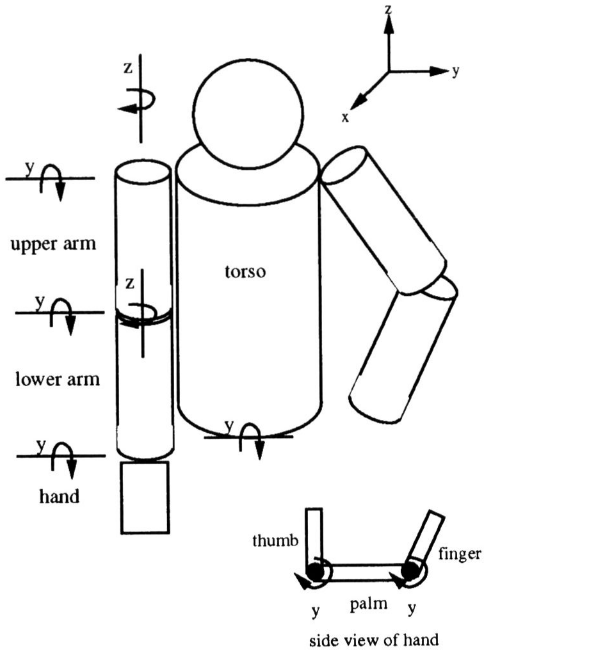

Figure 8: The rigid-body model used in the juggling animation. The model has eleven links with a total of fifteen degrees of freedom. The parameters for the mass, moment of inertia, and dimensions of all the links except the hands were obtained from Dempster and Gaughran (1965). The hands are of approximately the same size and density as the human hand measured in Dempster and Gaughran (1965).

hand juggles separately, throwing and catching the balls without passing them to the other hand.

To produce these three juggling patterns we chose release positions, catch positions, and flight times for each pattern. The catching and throwing routines were the same for all patterns.

Transitions between Patterns

The control system performs transitions between patterns by waiting until the first ball in a pattern is caught and then setting the throw and catch positions and the flight times to those used in the new pattern. Two feedback laws make the transitions robust: phase correction and collision avoidance.

The phase of each ball is the time in the cycle at which it is caught. In a three-ball cascade or shower, catching the first ball signals the beginning of a cycle, the second ball is caught 1/3 of the way through the cycle, and the third 2/3 of the way through. Phase correction is performed by adjusting the dwell time so that the next time the ball is caught it will be closer to the correct phase. In the fountain pattern, the two hands operate independently and the phase correction algorithm is used to keep both the balls and the hands operating in phase.

The feedback law for collision avoidance is used when errors in phasing or the changing patterns of the throws would cause two balls to collide. On each throw, the control system uses the ballistic equations for the balls to predict if the ball in the hand will collide with either of the balls in the air. If a collision is expected, the throw position is moved by twice the ball radius in x to prevent the collision.

Balancing on a Unicycle

To maintain balance a unicycle rider must push on the pedals in such a way as to correct errors in balance and forward speed. The user directs the animation by specifying a desired velocity or a desired position.

The model contains a wheel, a seat, and a body with two legs (figure 11). The legs push on the pedals through a pair of orthogonal springs and dampers. The springs allow the legs to push down and sideways on the pedals but not to pull up. The contact model between the wheel and the ground is also a pair of springs and dampers. This model allows the wheel to roll but does not allow it to slip.

Control

Unlike the other animations described in this paper, the unicycle control problem is continuous and there are no impacts that cause the dynamics of the system to change. The unicycle wheel is always touching the ground, and the control laws do not depend on the state of the system. As a result, the state machine is used only to handle changes in the control laws when the user switches from position control to velocity control.

The desired torque at the wheel is a function of the error in forward speed and the angle of the stem:

$$\tau = k_\phi \phi + k_{\dot{\phi}} \dot{\phi} + k_{\dot{x}}(\dot{x}_d - \dot{x})$$

where τ is the desired torque at the wheel, ϕ is the angle of the stem relative to vertical, $\dot{\phi}$ is the velocity of the angle of the stem, $\dot{x}_d$ is the desired speed of the unicycle, $\dot{x}$ is the actual speed, and k_ϕ, $k_{\dot{\phi}}$, and $k_{\dot{x}}$ are gains.

There is no motor at the hub, and the desired torque at the wheel is produced by moving the hip and knee

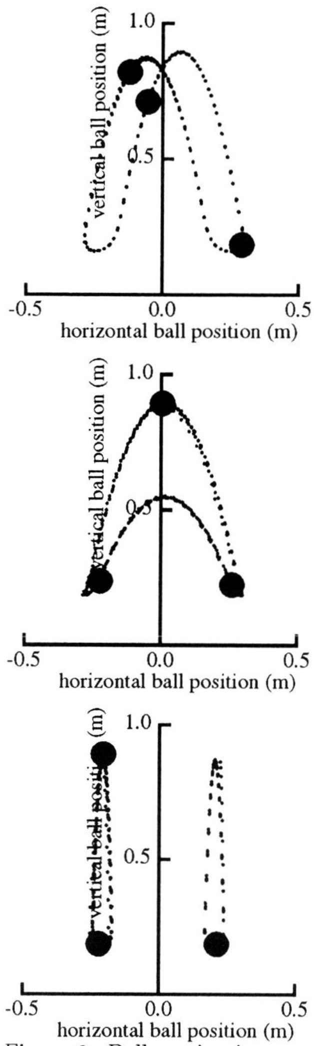

Figure 9: Ball motion in cascade, shower, and fountain patterns.

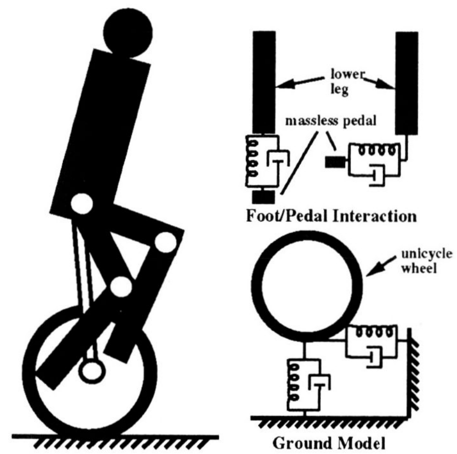

Figure 11: Schematic drawing of the model used in the unicycle animation. The model consists of a wheel, a seat, and a body with two legs divided into upper and lower segments. The body is attached to the seat by a pivot joint. The interaction between the lower legs and the massless pedals is modeled by a pair of orthogonal spring/dampers. The springs allow the legs to push down and sideways on the pedals but not to pull up. The ground model for the unicycle wheel is a pair of orthogonal spring/dampers. The x spring is stretched between the point marking the distance that the wheel has rolled and the point on the ground where the wheel is touching. The point to which the wheel has rolled is $x_0 + 2\pi r\theta$ where x_0 is the starting location, r is the radius of the wheel, and θ is the number of revolutions of the wheel. The point at which the wheel touches the ground is assumed to be the point directly beneath the hub. This model does not allow the wheel to slip.

joints so that the legs press down on the pedals with the appropriate force. Each leg is assigned a percentage of the desired torque based on the sign of the desired torque and the current pedal position:

$$\text{weight} = \frac{1}{2} + \frac{1}{2}\sin\theta$$

The other leg has a weighting of 1 − weight. When the torque is positive and the pedals are horizontal ($\theta = \pm\pi/2$) the front leg has a weighting of one and the rear leg has a weighting of zero. When the pedals are vertical, both legs have a weighting of one-half and the force is divided evenly between the two legs. The desired force along the axis of the leg is increased by a preload to ensure that the legs always push down on the pedals and remain on the pedals.

The desired force at the pedal is converted to desired torques at the knee and hip by taking the kinematics of the leg into account and assuming massless legs:

$$\tau_{knee} = -\text{ll} f_x$$

$$\tau_{hip} = -\text{ul}(\sin(\alpha) f_z + \cos(\alpha) f_x) - \text{ll} f_x$$

where τ_{knee} is the desired torque at the knee, τ_{hip} is the desired torque at the hip, ll is the length of the lower leg, ul is the length of the upper leg, α is the angle of the knee, and f_x and f_z are the desired forces between the pedal and the lower leg in the coordinate system of the lower leg.

The user interacts with the animation of a unicycle rider by specifying a desired speed or a desired position. The desired position is converted to a desired speed:

$$\dot{x}_d = k_x(x - x_d)$$

where $\dot{x}_d$ is the desired speed, x is the current position, x_d is the desired position, and k_x is a gain. The desired speed is limited by a maximum desired speed. Figure 12 shows the behavior of the system as it starts from rest and cycles to a specified location.

The control for the unicycle simulation has constant gains. Vos (Vos 1989; Vos and von Flotow 1990) states that the gains must be adjusted based on the state of the system for good performance of a three-dimensional robot unicycle with a motor at the hub and a horizontal turntable for yaw control. When the simulation is extended to three dimensions and the operating range of

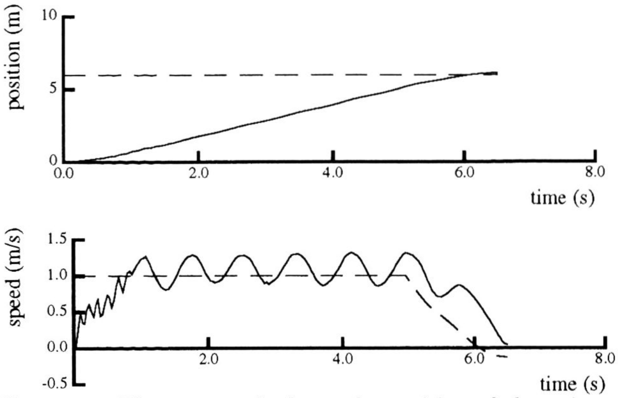

Figure 12: The top graph shows the position of the unicycle as it moves towards the goal. The bottom graph shows the forward speed. The dashed lines indicate the desired position and velocity.

the simulation is extended, we may also find this to be true but constant gains provide adequate performance for the two-dimensional model.

Discussion

We have presented descriptions of the models and control systems for four animations: a swing, a seesaw, a juggler, and a unicycle. In each case, the control system is built upon a state machine and the transitions between the states are determined by changes in the state of the system or changes in the input commands. The current state determines which control laws should be used to control the system. The interactions with the environment and many of the low-level control laws are implemented with springs and dampers. The state machines and some of the control laws are hand designed for each animation.

If properly designed, a control system will produce natural-looking motion when it is used to activate a physically realistic model. The combination of a physically realistic model and a control system can easily produce bad motion too, by violating joint or torque limits or by performing the task in an unexpected way or with extraneous motions. This problem is particularly apparent in underconstrained problems like the swing. There are many ways to pump a swing but only a small fraction produce motion that resembles the movements made by a human pumping a swing.

Our experiments during the design of these animations suggest that several features are required for the generation of natural-looking motion:

- The model must contain the key features of the system being modeled. If the model is too simple, the motion may not appear natural. We originally modeled the swing as a single rigid rod (no joint where the hand is attached) and discovered that the setpoints had to be adjusted very carefully for the swinging motion to increase in amplitude. With the extra degree of freedom, the control is much more robust and the motion appears more natural. We also compared a unicycle animation that was powered by the motion of the legs with one in which the control was provided by a torque source at the wheel and the legs were positioned kinematically. The resulting motions differed because the motor could exert a uniform torque in all pedal positions while the legs could not.
- The model should include the control parameters. In designing and tuning a control system, we found it easy to set the gains too high so that the simulation produced jerky motion. A strength model should be part of the physical model and control commands that require the physical system to go beyond those limits should be filtered. Lee et al (1990) implemented such a system for the task of lifting a load.
- Biological data can provide information about setpoints and control actions. In an underconstrained problem like the swing, choosing the control actions is not easy. A wide range of actions cause the swing to gain amplitude but only a much smaller range are commonly used by humans on a swing. We studied measurements of people pumping a swing to learn about the joint angles that were used and the transition points of the control system. More rigorous comparison of simulation data with biological data can also be used to show to what extent the simulated motion resembles the biological motion.
- Perfectly smooth motion is not natural. Steady-state simulations often produce motion that is too repetitive. Each of the phase plots of juggling shown in figure 9 contains data from several repetitions of the pattern, but there is little variation from one toss to the next. This too-perfect motion appears "robot-like" when played back through a graphical model. We need to take advantage of all opportunities for adding interest to the motion: changes in high-level commands from the user, disturbances from the environment, and the addition of noise within the system.

We have not addressed the question of automatic generation of control systems. Large portions of each of the control systems described here were designed by hand with the aid of principles gleaned from our experience in controlling robots. We are interested in exploring other approaches to the problem of generating control systems, perhaps by formalizing the principles that were outlined here to allow the automatic generation of control systems for limited domains or by applying techniques from optimal control theory to design the control systems with less input from the human designer.

By approaching the problem of motion generation from the perspective of physical realism, we have gained rules that make it possible to automatically generate motion through simulation. Within this framework we can generate many different motions that accomplish a single task by varying the physical system and the control strategies. For example, an adult pumps a swing differently than the eight-year-old child we modeled and a skilled unicyclist would use different gains and perhaps even different control laws than a beginner. Even greater variety could be produced by violating physical laws. In hand-drawn animations emotion and humor are often communicated through exaggerated motions that violate physical laws. Eventually, we would like to take advantage of this part of the design space; however, we feel that a better understanding of techniques for the generation of physically correct motion is needed first.

Acknowledgments

We thank Marc Raibert and the MIT Leglab for loaning us part of the simulation environment and Dave Haumann and Alan Norton for their enthusiastic support of this project.

References

Baraff, D. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics*, 23(3): 223–232.

Baraff, D. 1991. Coping with Friction for Non-penetrating Rigid Body Simulation. *Computer Graphics* 25(4):31–40.

Barzel, R., Barr, A. H. 1988. A Modeling System Based on Dynamic Constraints. *Computer Graphics* 22(4):179–188.

Beek, P. J., 1989. *Juggling Dynamics.* (Free University Press: Amsterdam).

Bruderlin, A., Calvert, T. W. 1989. Goal-Directed, Dynamic Animation of Human Walking. *Computer Graphics* 23(3):233–242.

Bühler, M., Koditschek, D. E., Kindlmann, P. J. 1989. Planning and Control of Robotic Juggling Tasks. In *Robotics Research: The Fifth International Symposium.* (MIT Press: Cambridge).

Buhler, J., Graham, R. 1984 Fountains, Showers, and Cascades. *The Sciences*, 24:44–51.

Burns, J. A. 1970. More on Pumping a Swing. *American Journal of Physics* 38:920–922.

Dempster, W. T., Gaughran, G. R. L. 1965. Properties of Body Segments based on Size and Weight. *American Journal of Anatomy* 120: 33–54.

Gore, B. F. 1970. The Child's Swing. *American Journal of Physics* 38:378–379.

Gore, B. F. 1971. Starting a Swing from Rest. *American Journal of Physics* 39:347.

Hahn, J. 1988. Realistic Animation of Rigid Bodies. *Computer Graphics* 22(4):299–308.

Hodgins, J. K. 1991. Biped Gait Transitions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA.

Hodgins, J., Raibert, M. H. 1990. Biped Gymnastics. *International Journal of Robotics Research*, 9(2):115-132

Kass, M., Miller, G. 1990. Rapid, Stable Fluid Dynamics for Computer Graphics. *Computer Graphics* 24(4):49–57.

Lee, P., Wei, S., Zhao, J., Badler, N.I. 1990. Strength Guided Motion. *Computer Graphics* 24(4):253–262.

McKenna, M., Zeltzer, D. 1990. Dynamic Simulation of Autonomous Legged Locomotion. *Computer Graphics* 24(4):29–38.

McMullen, J. T. 1972. On Initiating Motion in a Swing. *American Journal of Physics* 40;764–766.

Meredith, H. V. 1969a. Body Size of Contemporary Groups of Eight-year-old Children Studied in Different Parts of the World. Monographs of the Society for Research in Child Development 34(1).

Meredith, H. V. 1969b. Body Size of Contemporary Youth in Different Parts of the World. Monographs of the Society for Research in Child Development 34(7).

Miller, G. S. P., 1988. The Motion Dynamics of Snakes and Worms. *Computer Graphics* 22(4):169–178.

Norton, A., Turk, G., Bacon, R., Gerth, J., Sweeney, P. 1991. Animation and Fracture by Physical Modeling. *The Visual Computer*, 7(4):210–219.

Pentland, A., Williams, J. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. *Computer Graphics* 23(3):215–222.

Raibert, M. H. 1986. Legged Robots that Balance. (MIT Press: Cambridge).

Raibert, M. H., Hodgins, J. K, 1991. Animation of Dynamic Legged Locomotion. *Computer Graphics* 25(4):349–358.

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

Tea, P. L., Jr., Falk, H. 1968. Pumping on a Swing. *American Journal of Physics* 36:1165–1166.

Terzopoulos, D., Witkin, A. 1988. Physically Based Models with Rigid and Deformable Components. *IEEE Computer Graphics and Applications* 8(6).

Terzopoulos, D., Fleischer, K. 1988. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *Computer Graphics* 22(4), 269–278.

van de Panne M., Fiume, E., Vranesic, Z. 1990. Reusable Motion Synthesis Using State-Space Controllers *Computer Graphics* 24(4):225–234.

Vos, D. W., von Flotow, A. H. 1990. Dynamics and Nonlinear Adaptive Control of an Autonomous Unicycle: Theory and Experiment. Proceedings of the 29th Conference on Decision and Control. Honolulu, Hawaii, 182–187.

Vos, D. W. 1989. Dynamics and Nonlinear, Adaptive Control of an Autonomous Unicycle. M.S. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Wejchert, J., Haumann, D. 1991. Animation Aerodynamics. *Computer Graphics* 25(4):19–22.

Witkin, A., Kass, M., 1988. Spacetime Constraints. *Computer Graphics* 22(4):159–168.