Thesis Proposal

Simulations of Leaping, Tumbling, Landing, and Balancing Humans

Wayne L. Wooten
College of Computing
Georgia Institute of Technology
wlw@cc.gatech.edu

Thesis Proposal Committee:

Christopher Atkeson
Michael Cohen (Microsoft Research)
Norberto Ezquerra
Jessica Hodgins (Chair)
Michiel van de Panne (University of Toronto)

March 28, 1996

Contents

1 Introduction

Virtual environments often require the use of autonomous synthetic actors. For example, games with synthetic opponents, architectural walkthroughs with virtual tour guides, and information spaces with virtual retrieval agents all require synthetic actors that move through the environment reacting to the environment and to the actions of other actors both real and synthetic. Compelling synthetic actors will require the development of perceptual models, behavioral procedures, and motion control systems. This proposal addresses the last problem–generating motion for synthetic actors.

Animators generally use one of three techniques for generating realistic and natural-looking human motion for highly dynamic behaviors: keyframing, motion capture, or dynamic simulation. When keyframing is used, the animator specifies the location of each body segment at key frames in the animation sequence. In traditional cel animation, artists draw the frames in between key frames but with computer-assisted keyframe animation, interpolation methods are used to calculate the body segment positions for the frames in between. Inverse kinematics and other procedural methods simplify the keyframing process by providing constraints on the motion of hierarchical linkages. With motion capture, the computer determines the three-dimensional location of reflective markers placed on the joints of a live actor. The motion of the joints of a synthetic actor are derived from the locations of the markers as they move over time. With simulation, the equations of motion for a physically realistic model of the synthetic actor are combined with control systems to generate motion trajectories. The control systems take input from high level specifications and allow the synthetic actors to move autonomously in an environment by avoiding obstacles, changing speed, and interacting with other actors.

Simulation may prove to have advantages over motion capture and keyframing. Using simulation, the lower level details of the motion are automatically generated by the equations of motion and the control systems, whereas in keyframing the animator must specify many of the details by hand. Simulation generates physically realistic motion while keyframing relies on the skill of the animator to specify realistic or appealing motion. Motion capture data is realistic because it is derived from a live actor moving in the physical world, but the capture process may introduce flaws into the motion because of the limited resolution and accuracy of the sensors. A simulated actor can react autonomously using a control system to provide the basic behaviors such as running, walking, or standing and higher level controls. Each basic behavior would be more general than an equivalent set of motion capture data or keyframed actions because the behaviors could produce a range of different motions. For example, a control system could produce natural-looking running motion for a range of speeds from 1 m/s to 8 m/s but motion capture or keyframing would require many data

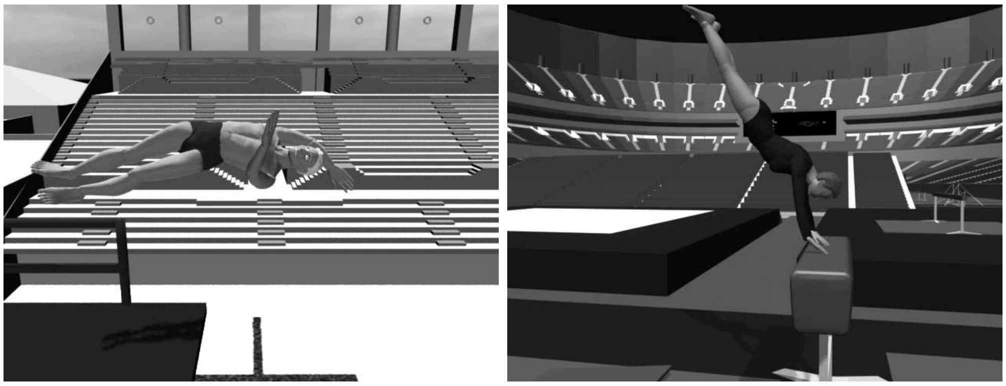


Figure 1: Images from diving and vaulting simulations. The diver on the left is in the flight phase of a backward $1\frac{1}{2}$ somersault pike with $\frac{1}{2}$ twist. The gymnast on the right is about to begin the second flight phase of a handspring vault.

sets and sophisticated interpolation methods to cover the same range of speeds.

Despite the potential advantages of simulation, it does present some difficulties. Robust control systems that generate natural-looking motion are difficult to design by hand and automatic methods are not yet sophisticated enough to generate robust control systems for most human-like behaviors. To produce the desired motion, a user must iteratively adjust the control parameters and control laws until the simulation produces the appropriate trajectory. This tuning process is similar to keyframing except the user adjusts control laws and parameters instead of actual body segment locations.

I have developed simulations for two synthetic athletes, a male diver and a female gymnast. Both simulations use control systems to compute the joint torques required to complete the maneuver in a natural-looking fashion. The control system for diving generates motion for three different dives. One of those dives, the backward $1\frac{1}{2}$ somersault pike with $\frac{1}{2}$ twist, is shown in figure 1. The vaulting control system generates motion for two types of vaults, and a handspring vault is shown on the right of figure 1. Although the motion in the dives and vaults is natural looking, the control systems are brittle and would require further hand tuning to generate natural-looking motion for additional dives and vaults. I plan to address this problem by developing control systems for dives, vaults, and other gymnastic activities that are built on robust lower level behaviors for leaping, tumbling, landing, and balancing.

1.1 Problem Definition

I propose to address the problem of generating natural-looking and physically realistic motion for articulated human figures by building on my existing knowledge of control systems. I will do this by developing control systems for four discrete behaviors and for transitions between those behaviors. Physically realistic motion will be guaranteed within the accuracy of the dynamic model. I will attempt to make the resulting motion look natural by using principles from biological systems for inspiration in the design of the control systems.

I will develop control systems for a set of four tasks: leaping, tumbling, landing, and balancing. The leaping controller will propel the figure into the air with the desired linear and angular velocities. The tumbling controller will perform twists and somersaults using techniques to transform angular velocity from one axis to another by repositioning the body segments. The landing controller will take the high linear and angular velocities at touchdown and reduce them to an acceptable range for the balance controller. The balance controller will prevent the simulated figure from falling in the presence of internal and external disturbances.

The four control systems will share a common set of requirements:

- Each control system should operate under a wide range of initial conditions and disturbances and produce a wide range of final conditions. The control system will be considered robust if disturbances do not prevent the control system from producing the desired final conditions. The disturbances can be internal, introduced by motion of the body segments, or external, introduced by collisions with other objects in the environment.

- The final conditions of the control system should fall within the range of acceptable initial conditions for other control systems. A wide range of overlapping initial and final conditions will make the task of transitioning from one control system to the next trivial. For example, the simulation should be able to perform a backflip by using the leaping, tumbling, landing, and balance controllers in succession.

- The motion produced by the control system should not violate the biological constraints of the human body and should use control strategies similar in appearance to those used by humans.

The next section presents relevant work in the areas of computer graphics, robotics, and biomechanics. Section 3 describes the dynamic models used to simulate vaulting and diving. Section 4 explains the control methods used to generate motion for the diver and gymnast. Section 5 presents the proposed research, outlining the four major goals of this work. The final section provides a timeline for completion and summarizes the proposed research.

2 Background

The proposed research on the simulation of human motion draws on previous results from three different areas. Computer graphics provides methods for animating and rendering synthetic creatures and I will use these methods to develop systems for animating synthetic actors. Robotics provides methods for controlling dynamic simulations and these results will be used to develop robust control systems for articulated figures. Research in biomechanics provides data and insight into how the human body moves. These data will help me develop control systems that generate natural-looking motion.

2.1 Computer Graphics

Generating physically realistic motion with simulation has been a topic of interest in computer graphics for a number of years. One of the first systems for simulating articulated figures was developed by Armstrong, Green, and Lake [1]. Their system addressed the problem of computational complexity by using a formulation for the equations of motion that grew linearly with the number of links in the articulated figure. Wilhelms developed a similar system to reduce the complexity of the computation by using the Gibbs-Appell formulation [56]. Both systems allowed users to control the motion of the articulated figure by applying forces to the joints. Developing ad hoc equations and allowing direct user specification of joint torques did not allow easy generation of high level, coordinated motion and was not intuitive for the user. Isaacs and Cohen presented a control method that used kinematic constraints and inverse dynamics to determine the forces required to control articulated figures [24]. However, this method requires that body segment trajectories or close approximations be known in advance.

During this time, computing resources were inadequate for simulations of complex articulated figures. Girard developed a system called PODA, that required less computational power [16]. Figure 2A shows a graphical representation of an articulated figure animated with PODA. Girard's system used inverse kinematics to position the body segments based on the desired location of the end effector. The animator would keyframe the motion of the end effector in a low level fashion, but would not have to specify the motion of each body segment. In addition, Girard used dynamics to compute the motion of the main body of the figure, allowing it to accelerate under gravity while in flight.

The original version of the Jack system, developed at the University of Pennsylvania by Badler and his colleagues [2], did not incorporate a dynamics model. Jack included an interactive system that formulated inverse kinematics as a nonlinear programming problem to position the body seg-

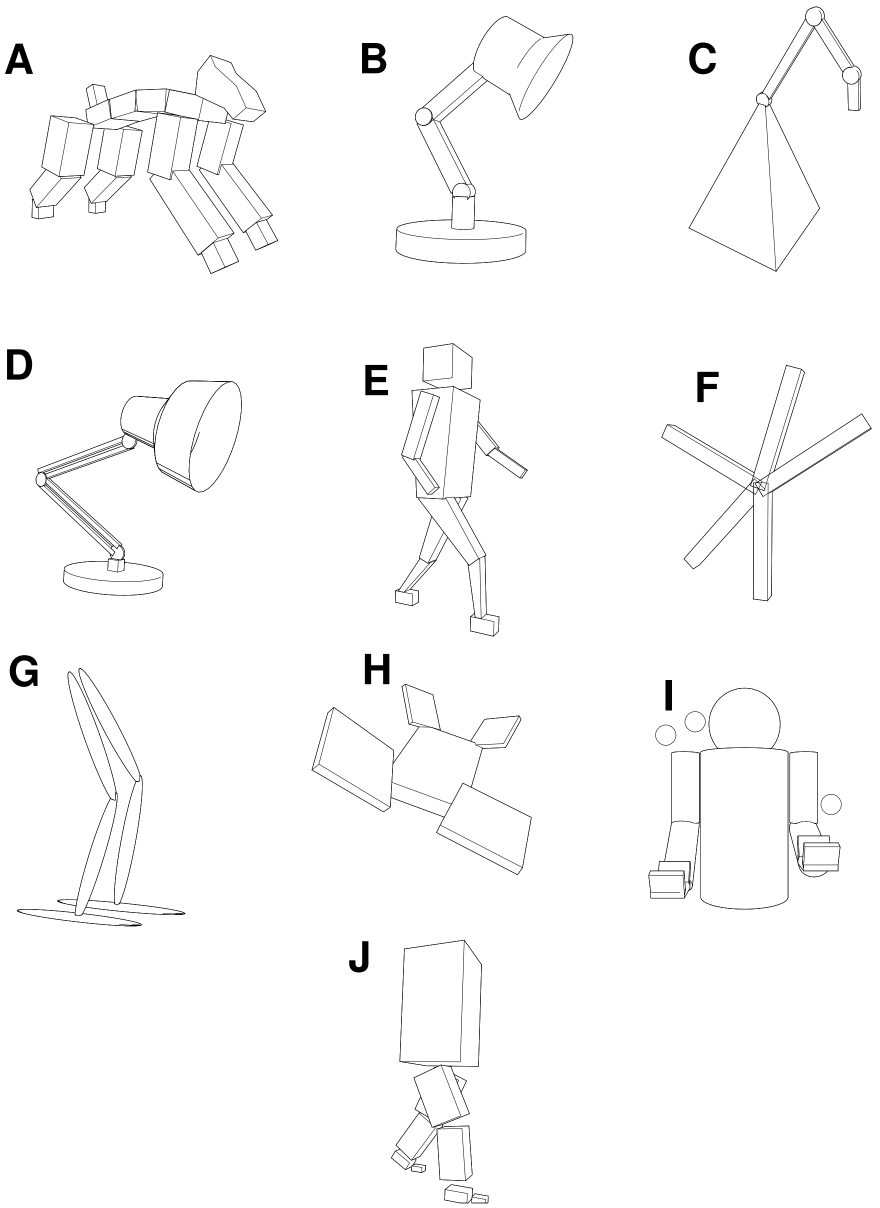


Figure 2: A diagram of a number of creatures presented in computer graphics research. A: Girard [16]; B: Witkin and Kass [58]; C: Cohen [7]; D: van de Panne et al. [53]; E: van de Panne et al. [54]; F: Ngo and Marks [34]; G: van de Panne and Fiume [52]; H: Sims [48]; I: Hodgins et al. [22]; J: Stewart and Cremer [49]

ments based on the location of end effectors. Control programs ensured balance by moving the foot when the projection of the center of mass on the ground plane moved outside the polygon defined by the points of contact of both feet. Many enhancements have been added to Jack over the years, including a strength model to constrain the motion based on muscle strength and improve the natural look of the motion [26]. Ko and Badler developed a generalized model of walking using digitized walking data and inverse dynamics [25]. With their generalization, motion could be generated for models with arbitrary anthropometry taking different step lengths from those obtained using motion capture.

After dynamic simulation algorithms were understood in the computer graphics community, researchers began to turn their attention towards control. Witkin and Kass presented one solution to the problem of automatically generating natural-looking and physically realistic motion for articulated figures [58]. Their method, which they called spacetime constraints, determined a trajectory for each degree of freedom in the system using a two point boundary problem. Constraints were used to specify a desired behavior for the system and to enforce the Newtonian laws of physics (thereby ensuring physical realism). The constrained optimization problem was solved numerically and the result was an optimal, natural, and physically realistic trajectory curve for each degree of freedom. Witkin and Kass used this approach to animate an articulated, lamp with six degrees of freedom (figure 2B). The lamp's optimization criteria minimized the power consumed by each of the joint actuators, resulting in natural-looking motion that exhibited anticipation.

In 1992, Cohen refined the spacetime constraints technique in several ways [7]. His system allowed interactive specification of the constraints and objective functions. The system allowed the subdivision of spacetime into windows and allowed optimization to take place over a subset of spacetime selected by the user. Cubic B-splines were used instead of finite differences to define a C^2 continuous function describing the trajectory of each degree of freedom. In 1994, Liu, Gortler, and Cohen improved the speed of the spacetime algorithm using hierarchical wavelets [29]. The nature of the wavelet formulation allowed the functions specifying the trajectories of the degrees of freedom to have detail only where needed. The wavelet formulation led to better conditioned systems, that in turn allowed fast convergence. They used common subexpression elimination to reduce the number of expressions used to describe the system, constraints, and objective functions. The creature used to demonstrate the wavelet formulation had six degrees of freedom and is shown in figure 2C.

Spacetime constraints have not yet been used to generate motion for systems as complex as the human body. The computational complexity of spacetime constraints is high, even with the enhancements developed by Cohen and his colleagues. The inverse of the Hessian matrix must be computed to solve the constrained nonlinear optimization problem. This operation takes $O(N^3)$

time, where N is the dimension of the matrix. Even relatively simple simulations with four degrees of freedom could have thousands of nonzero entries in the matrix. Real-time response for systems with many degrees of freedom is not feasible with today's computer hardware because the algorithm is not able to produce new trajectories rapidly enough.

Brotman and Netravali presented an alternative to spacetime constraints [4]. They used an LQR optimal control system to apply the minimal amount of control energy needed to ensure that rigid bodies passed through user-specified keypoints. This technique is different from the spacetime approach because it calculates a control value that affects motion, instead of computing the trajectories themselves. They presented examples of single body systems–a truck moving along a straight line and an aircraft moving along its equilibrium flight direction. For the truck, the control value corresponded to application of acceleration or deceleration. For the aircraft, the control value altered the orientation of the plane along the flight path. Brotman and Netravali did not present examples of articulated figures or nonlinear systems, but do mention in a footnote that their methods could be used for these systems by linearizing the problem first.

van de Panne, Fiume, and Vranesic developed a method similar to Brotman and Netravali's [53]. Instead of using an LQR optimal control system to produce the motion of the simulation, they derived a controller using dynamic programming and an optimization criteria of minimum energy consumption. The derived controller could be reused to guide the simulation from any initial state included in the design of the controller to the final desired goal state. They used this method to generate the motion for a lamp with five degrees of freedom (figure 2D). Unfortunately dynamic programming solutions are exponential in the number of degrees of freedom and current methods become prohibitively expensive in both time and space as the size of the state space becomes large.

Pai developed a model of a human-like walker based on the notion of least constraint [37]. Least constraint generates motion using inequality constraints, which are time and state based assertions on the allowed states of the simulation. The set of constraints are satisfied using a nonlinear extension of the relaxation method. Pai used a dynamic simulation system called *Newton* [10] and a set of these assertions to generate motion for a walker with 28 degrees of freedom. The control program included constraints to ensure that the foot clears the ground during the swing phase, the pelvis remains above a certain height, and the placement of the swing foot is suitable for dynamic balance.

van de Panne, Fuime, and Vranesic developed a simulation of a two-dimensional, human-like biped that walked up and down gently sloping terrain and stairs [54]. The controller for the stance leg was generated with dynamic programming [53]. All of the other body segments were controlled with proportional-derivative controllers. The control tables generated by the state-space controller allowed the simulation to ascend and descend both stairs and slopes as well as walk over level terrain.

Figure 2E shows a diagram of the walking creature.

Ngo and Marks used forward dynamics to simulate planar creatures with three to five links that were controlled by stimulus-response systems [34]. An example of a five link creature is shown in figure 2F. The control system for each creature was composed of response functions that generated motion based on the input of the stimulus functions. The stimulus functions were used to sense conditions in the environment, like contact with the ground, and the response functions changed the configuration of the creature over time. The parameters for the stimulus-response system were generated with a trial and error genetic search algorithm. Optimization criteria, such as longest distance traveled or highest jump achieved, were used to select the best stimulus-response controllers.

van de Panne and Fiume developed a similar method that used sensor-actuator networks [52]. They modeled ten planar creatures composed of two to six links. Their walking creature with eight degrees of freedom is shown in figure 2G. Sensors determined ground contact, joint angles, joint lengths, and light locations. The actuators used proportional-derivative servos to move the joints and were connected to the sensors through a network of hidden nodes. The networks were similar to artificial neural networks with weights existing between connections of the sensors, hidden nodes, and actuators. Both simulated annealing and stochastic gradient ascent methods were used to search for optimal weights for use in the sensor-actuator network. The evaluation metric for the search methods attempted to generate creatures that could travel the farthest or follow a moving point in space.

Sims presented a system that generated not only a control system using genetic algorithms, but the creature's morphology as well [48, 47]. The creatures were three-dimensional and had many degrees of freedom. Sensors measured joint angles, contact between linkages, and determined the location of light sources. The control programs were based on a dataflow computer paradigm instead of an artificial neural network. Each neuron could take up to three inputs and perform arbitrary computations on them to produce an output. The number of neurons and the connections between them were derived using a genetic algorithm. A number of objective functions were used in the genetic search algorithm to select different behaviors. Sims produced creatures that could swim, crawl, jump, and follow light sources. An example of a swimming creature is illustrated in figure 2H.

Grzeszczuk and Terzopoulos automatically generated controllers for hand-designed, flexible creatures such as fish, snakes, and sting rays [17]. The creatures were not modeled using rigid body dynamics, instead they were modeled with point masses connected by springs that acted as actuators. Time and frequency domain discrete controllers were found by optimizing an objective function composed of two terms, one for the amount of control energy used and the other for the smoothness of the trajectory. Both simulated annealing and the simplex method were used to search for the optimal controllers. Higher level control was achieved by combining and smoothly blending the automatically derived lower level controllers.

Many problems must be addressed before automatically generated control systems can be used for simulations with many degrees of freedom. Most of the methods mentioned above either produce natural-looking motion for creatures with few degrees of freedom or produce unnatural-looking motion for creatures with more degrees of freedom. Often natural-looking motion is produced when a simple energy minimization function is used as the optimization criteria but this will not produce many of the movements seen in more complex systems. For example, when people gesture as they speak they are not using minimum energy. Using search techniques, users have little direct control over the desired motion and must be content with motion trajectories derived through a combination of optimization criteria and search algorithms.

An alternative to automatically generated control systems are hand-designed control systems. Bruderlin and Calvert developed a simulation of a walking human using a hand-designed control system for a single leg and biological data for parameters such as pelvis rotation [5]. The upper body motion was added kinematically after the motion of the legs had been computed. The system allowed the animator to adjust parameters of the walking simulation such as forward velocity, step length, step frequency, and pelvic tilt.

McKenna and Zeltzer developed a simulation of a six-legged, articulated insect with a gait controller that coordinated the movement of the six legs [30]. The gait controller, derived from observations of gait patterns of walking insects, issued commands to lower level motor controllers. The motor controllers generated forces using exponential springs and were responsible for maintaining support and generating forward motion.

Raibert and Hodgins developed dynamic simulations to model a biped robot, a quadruped robot, and a planar kangaroo [45]. Control systems composed of state machines and hand-tuned, proportional-derivative servos were used to generate motion for running, trotting, bounding, galloping, and jumping. The user could direct the speed, gait, and path of the simulated creatures, and the motion was used to produce the animation, "On the Run," shown in the 1991 SIGGRAPH Electronic Theater. Hodgins, Sweeney, and Lawrence developed feedback control techniques for several behaviors of human-like creatures: swinging, juggling balls and clubs, and bouncing on a see-saw [22]. A diagram of the juggler is shown in figure 2I.

Stewart and Cremer developed a simulation system based on *Newton* and they used control systems similar in basic design to those of Raibert and Hodgins [49]. The animator used state machines and proportional-derivative servos to control the motion of the simulation. Stewart and Cremer used the system to simulate a biped with fourteen degrees of freedom that could walk in a straight line and up and down stairs. A diagram of the biped model can be seen in figure 2J.

More recently, Hodgins and colleagues developed simulations of human athletes using a three-dimensional dynamic model of the human body [19, 59, 23]. Diving, vaulting, cycling, and running were simulated using a collection of control techniques that included state machines to activate control laws required for a specific phase of each behavior, synergies to control several degrees of freedom simultaneously, inverse kinematics to derive desired joint angles given desired end effector positions, and proportional-derivative servos to determine the torques required to move joints to the desired positions.

Hand-designed control systems have been used to produce motion for creatures with many degrees of freedom, but natural-looking motion requires careful design of the control system. All of the control systems discussed above performed single tasks. Much time and effort will have to be spent developing control systems that perform a wide variety of tasks in a robust fashion.

2.2 Robotics and Control

Many of the ideas used for the control of articulated figures have been borrowed from control techniques developed for robots. Raibert and his colleagues built a series of running machines that performed a number of dynamic tasks [44]. The control algorithms allowed two-legged and four-legged robots to run in circles, run up and down stairs, hop in place and change gaits [20]. Many of the control algorithms developed for the biped robots are relevant to dynamic simulations of human figures. For instance, the notion of adjusting the touchdown position of the foot to control forward velocity could be refined for use in controlling the running speed of a human model.

Hodgins and Raibert developed control algorithms that allowed a planar biped robot to perform a forward somersault [21]. The robot's control system performed this maneuver with a series of control states: approach, hurdle, flip, landing, and recovery. Hodgins and Raibert related the control strategies for the flip to possible biological control strategies such as open-loop control and the motor-tape model. Control of the biped's pitch rate was not performed during flight, and the success of the maneuver was dependent on control actions taken during the ground contact preceding the flight phase. Playter and Raibert developed control algorithms that allowed a three-dimensional biped robot to perform a forward somersault [42, 43]. They implemented a tuck servo to alter the rate of rotation in flight to produce better landing conditions. The tuck servo was not compared to open-loop control to determine whether the tuck servo reliably produced better landings.

Several researchers have explored optimal control systems for simulations of planar platform diving. Murthy and Keerthi presented an optimal control system for a two-dimensional, diver with four degrees of freedom [31]. They formulated a time-optimal control problem using state space and control constraints with a numerical approach to compute the solution. The simulated diver performed both forward and backward somersaults. Solutions required about 10 minutes of computation on an Intel 486-based machine. Liu and Cohen presented a different approach to the problem posed by Murthy and Keerthi [28]. They decomposed the optimal control problem into two subproblems, one involved finding the trajectory of each joint by minimizing the second derivatives and the other involved finding the initial velocity of the torso to ensure it entered the water vertically. They simplified the conservation of momentum equation by assuming untucking is the reverse of tucking, thereby ignoring the effect of the rotation rate of the limbs during flight. Their solutions took about 0.7 s on an R4000 SGI workstation. Crawford and Sastry developed an adaptive controller to search a restricted control space for a planar diver with five degrees of freedom [9]. The adaptive control system searched the state space for control parameters that resulted in a $1\frac{1}{2}$ somersault pike with a vertical entry. Gradient descent was used to perform the search and random variations in the control parameters were used to prevent local minima from interfering with the search. The results were qualitatively comparable to humans performing a $1\frac{1}{2}$ somersault pike.

2.3 Biomechanics

Maximum-height jumping is an important measure of athletic fitness and has led to the development of many theories and models. Data collected from experiments and mathematical models could provide useful insight into the development of leaping controllers that generate natural-looking motion.

Pandy and his colleagues developed a model and control system to study maximum-height human jumping [39]. The model was two-dimensional and consisted of four rigid segments connected with revolute joints. Eight musculotendon actuators, modeled using a variation of the Hill muscle model [14], were used to apply forces to the skeletal elements. They solved the optimal control problem by assuming the neural control signals provided either no excitation or full excitation. The height and joint trajectories produced by the simulation were similar to those of human athletes performing a maximum-height squat jump.

Soest and his colleagues developed a similar model consisting of four links with six musculotendon actuators [55]. Soest was concerned with the role of the gastrocnemius muscle in generating a maximum-height vertical leap. His hypothesis was that the gastrocnemius muscle provided more power as a biarticular muscle that crossed two joints, than as a monoarticular muscle that crossed only one. He developed a simulation with the gastrocnemius attached in biarticular fashion and in a monoarticular fashion. The model jumped 10 mm higher in biarticular mode than in monoarticular mode with a variation of 0.1 mm over many trials, lending credibility to the hypothesis.

Researchers in biomechanics have also analyzed techniques used by humans in performing aerial maneuvers. Yeadon and his colleagues recorded three-dimensional motion of aerial maneuvers with high-speed film and digitized the resulting footage [60, 61]. Yeadon developed a dynamic model of the human body with eleven segments and eighteen degrees of freedom. His simulation was designed for the analysis of twisting somersaults and used inverse dynamics to compute the forces at the joints. Yeadon's data from film capture for three test subjects compared favorably with his simulation.

For many years, researchers have debated how cats land on their feet and how humans perform free fall aerial maneuvers. Frohlich presents an enlightening discussion of the techniques used by humans for initiating somersaults and twists [13]. Through simulation and informal human experiments, he demonstrated that humans could perform somersaults and twists using torque generated by pushing on a platform. He also showed how they could perform similar maneuvers with no angular momentum from the platform. I will use these strategies when developing the tumbling control system.

Studies of strategies used in landing and the impact forces produced during landing could provide useful theories for the development of landing control systems. Caster and Bates assessed the strategies used for landing by four male subjects dropped from a vertical height of 0.6m [6]. Their study tried to determine the relative contributions of passive control (mechanical) or active control (neuromuscular) used during landing. They hypothesized that if weights were added around the subject's ankles and the ground reaction force increased in proportion to the weight added, then only mechanical strategies were used. However, if the ground reaction force was attenuated, then the subjects used neuromuscular response as a protective measure to avoid large, damaging forces. EMG data was collected to help determine the strategy employed by the subjects. The results showed both mechanical and neuromuscular strategies were used by all subjects, based on the amount of added mass and prior experiences.

Özgüven and Berme presented a model of impact forces experienced by humans landing from heights of 0.45m and dismounts from a horizontal bar onto regulation gymnastic mats [36]. A simple model composed of two masses connected by springs and dampers was developed to predict the maximum ground reaction force experienced under landing conditions. Parameters for the model were obtained from 12 landing trials. Experimental data showed that the maximum reaction force ranged from 8.2 to 11.6 times body weight for the dismount and from 5.0 to 7.0 times body weight for the vertical landing. The predicted reaction forces were very close to the measured reaction forces.

Research into how humans respond to disturbances in balance provides useful data for developing balance control systems. Nashner and his colleagues studied how humans respond to disturbances in balance while standing and walking [33]. They observed that muscles react in response to a disturbance in a distal to proximal order and the shortest latencies in response to a disturbance are 70-100ms. In summarizing Nashner's research, Ghez [15] emphasized that postural stability in humans is maintained by a combination of preprogrammed feedforward responses and compensatory feedback responses. Nashner observed that after only a few repeated trials of similar disturbances, the preprogrammed response was attenuated [32].

Winter, Patla, and Frank provided a comprehensive review of the balance control literature in addition to presenting findings of their own [57]. One of their interesting observations was that the nervous system anticipates disturbances to balance caused by voluntary movements and applies corrective action to compensate for the disturbance. When subjects disturbed their balance by pulling on a fixed handle while standing, EMG data from the arm and leg muscles showed the gastrocnemius and hamstrings reacted in a distal to proximal order before the bicep muscle began to pull on the handle.

3 Dynamic Human Model

Simulation produces motion using a dynamic model of the synthetic actor's body. The model specifies the mass and moment of inertia of each limb and the locations of joints connecting each pair of limbs. The simulation uses this information to compute physically correct motion within the accuracy of the model. This section will describe the models for a male and female character used in diving and vaulting simulations.

The system used to generate the motion for human characters is composed of a dynamic model, control system, user interface, and graphical renderer (figure 3). The polygonal models used in calculating the parameters of the dynamic models are shown in figure 4. The segments of the human body are modeled as rigid links. The connections between links are modeled as rotary joints with one, two, or three degrees of freedom. Some joints, like the knee, are modeled as a single pin joint whereas others, like the shoulder, are modeled as a three-axis gimbal joint. Figure 4 shows the location and number of degrees of freedom at each joint. Tables 1 and 2 give the coordinates of each joint relative to the center of mass of that segment.

The volume and center of mass of each body segment were calculated from the polygonal model by integrating over the sets of tetrahedra formed by the triangular faces of the model and the origin [27]. Density data for each body segment was combined with the volume and center of mass

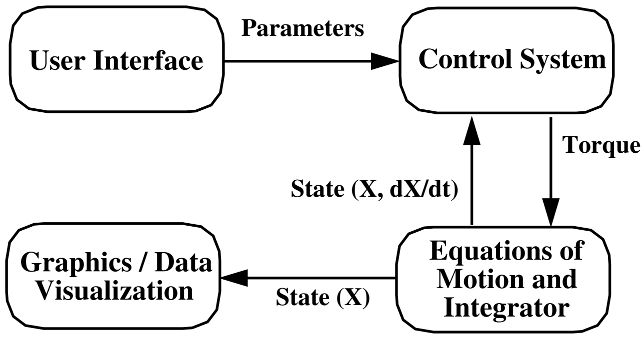


Figure 3: A diagram of the animation system used for the dynamic simulation of a human. The user controls the animation by providing parameters for the control system, such as the amount of twist during liftoff, forward velocity of the torso, or the height of a jump. The control system uses this information to compute the torque that should be applied at each joint. The equations of motion are used to compute the acceleration for a given set of joint torques and state of the system. The velocity and position at the next time step are computed by integrating the acceleration. The system state is then used to draw the graphical image and provide feedback for the control system.

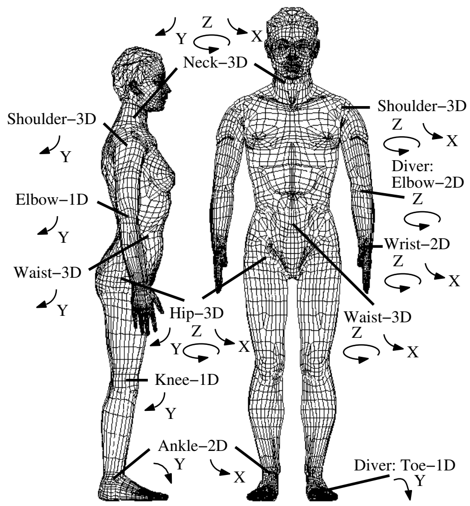


Figure 4: The controlled degrees of freedom of the human model. The number of degrees of freedom are shown at each joint. The direction of each arrow indicates the positive direction of rotation for each degree of freedom. The diver has two more degrees of freedom than the gymnast, one in the elbow and one in the foot. The polygonal models were purchased from Viewpoint Datalabs.

Link	COM to Proximal (x, y, z m)			COM to Distal (x, y, z m)		
Trunk to Head				0.012	0.000	0.320
Trunk to Pelvis				0.012	0.000	-0.220
Trunk to Arm				-0.048	$\pm$0.164	0.120
Head	-0.009	0.000	-0.064			
Pelvis	0.023	0.000	0.103	0.005	$\pm$0.098	-0.110
Thigh	0.024	$\pm$0.006	0.120	-0.052	$\pm$0.019	-0.210
Shank	0.005	$\pm$0.019	0.165	-0.002	$\pm$0.009	-0.250
Foot	-0.046	$\pm$0.009	0.048			
Arm	-0.0002	$\pm$0.056	0.120	-0.005	$\pm$0.036	-0.170
Forearm	-0.025	$\pm$0.007	0.090	0.012	$\pm$0.014	-0.110
Hand	-0.026	0.000	0.085			

Table 1: The distance from the center of mass of each link to the proximal and distal joints for the male model in x, y, and z. The positive distance along the y-axis refers to a location on the left side of the body; a negative distance refers to the right side. The z-axis is vertical and the x-axis is positive in the direction that the model is facing.

Link	COM to Proximal (x, y, z m)			COM to Distal (x, y, z m)		
Trunk to Head				0.003	0.000	0.280
Trunk to Pelvis				0.001	0.000	-0.190
Trunk to Arm				-0.027	$\pm$0.131	0.122
Head	-0.008	0.000	-0.076			
Pelvis	0.035	0.000	0.115	-0.016	$\pm$0.076	-0.090
Thigh	0.002	$\pm$0.006	0.130	-0.008	$\pm$0.007	-0.230
Shank	0.016	$\pm$0.006	0.115	-0.020	$\pm$0.014	-0.204
Foot	-0.051	$\pm$0.006	0.041			
Arm	0.006	$\pm$0.037	0.108	0.002	$\pm$0.027	-0.170
Forearm	-0.015	$\pm$0.003	0.088	0.025	$\pm$0.006	-0.123
Hand	-0.013	0.000	0.064			

Table 2: The distance from the center of mass of each link to the proximal and distal joints for the female model. The coordinate system is the same for both the male and female models.

Link	Density (g/cm^3)	Mass (kg)	Measured Mass (kg)	Moment of Inertia (x, y, z kgm^2)			Measured MOI (y kgm^2)
Trunk	1.008	29.27	29.51	0.73	0.63	0.32	-
Head	1.170	5.89	5.12	0.03	0.033	0.023	0.0320
Pelvis	1.029	16.61	16.32	0.23	0.18	0.16	0.2440
Thigh	1.040	8.35	6.09	0.15	0.16	0.025	0.1220
Shank	1.078	4.16	2.85	0.055	0.056	0.007	0.0560
Foot	1.066	1.34	0.88	0.0018	0.0075	0.007	0.0040
Arm	1.067	2.79	1.63	0.025	0.025	0.005	0.0160
Forearm	1.101	1.21	0.95	0.005	0.0054	0.0012	0.0074
Hand	1.069	0.55	0.38	0.0016	0.002	0.0005	0.0003

Table 3: Parameters of the rigid body model of a male human. The moment of inertia is computed about the center of mass of each link. The densities and measured masses are given in Dempster and Gaughran [12]. The measured moments of inertia are from subject number 15250 [11]. Moment of inertia data was not available for a body segment shaped like the trunk used in our model.

Link	Density (g/cm^3)	Mass (kg)	Moment of Inertia (x, y, z kgm^2)		
Trunk	1.008	19.81	0.39	0.36	0.152
Head	1.170	4.22	0.016	0.019	0.013
Pelvis	1.029	14.10	0.179	0.146	0.115
Thigh	1.040	6.57	0.099	0.101	0.017
Shank	1.078	2.76	0.0275	0.028	0.0037
Foot	1.066	0.75	0.0007	0.0029	0.0027
Arm	1.067	1.91	0.017	0.017	0.0025
Forearm	1.101	1.21	0.0038	0.004	0.0006
Hand	1.069	0.26	0.0004	0.0006	0.0001

Table 4: Parameters of the rigid body model of a female human. The male densities are used as an approximation for the female densities.

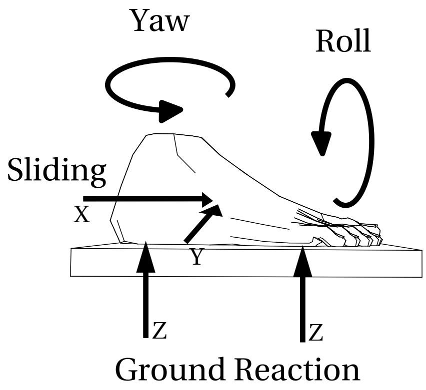


Figure 5: Six constraints are used for each foot. Two ground reaction force constraints prevent the foot from penetrating the ground. Two constraints prevent the foot from sliding on the plane. Two more constraints are used to prevent the foot from rolling or yawing about its center of mass.

data to calculate the mass and moments of inertia for the corresponding body segment [12, 11]. The densities and mass properties of each body segment are given in tables 3 and 4. One of the simplifying assumptions in this model is that the density of each body segment is uniform. In general this assumption is not correct. For example, the torso has a lower density in the lung tissue than in the muscle tissue. However, the mass and moments of inertia of the body segments compare favorably with the anatomical literature, as seen in table 3.

The equations of motion were generated from the dynamic parameters of the model with a commercially available package called SD/FAST. The package generates C subroutines for the equations of motion using a variant of Kane's method and a symbolic simplification phase [46]. The subroutines determine the accelerations, velocities, and positions of each body segment at each time step given the applied forces and torques. We use a fixed step size, fourth-order Runga-Kutta integrator to advance the simulation forward in time.

Control systems are used to derive the torques that are applied at each axis of each joint. Our current control system uses a torque source at each joint to generate motion. A more accurate biological model would use a muscle model to compute the forces at the muscle attachment points on the skeleton.

The interactions between body segments and the environment are modeled using constraints. Interaction of the feet with the ground use six constraints for each foot: two keep the metatarsus and heel above the surface of the ground, two prevent the foot from sliding on the ground, and two prevent the foot from rolling or twisting about the center of mass of the foot. The linear and rotational acceleration of the contact point of the foot with respect to the ground is the constraint error. The penetration of the foot into the ground and the velocity of the foot relative to the ground stabilize the constraint error [3]. The forces computed with the constraint matrix are applied to the foot at the point or points of contact to prevent the foot from penetrating or slipping on the ground. Friction is infinite in our model. Torques are applied about the center of mass of the foot to prevent rolling or twisting (figure 5). To allow the feet to leave the ground, the force is applied only when the foot has penetrated the ground and its vertical velocity is negative. Contact between the hands and the environment are modeled in the same manner as the feet with contact points at the base of the palm and the tips of the fingers. Constraints are also used to prevent the joints from exceeding user-specified minimum and maximum angles. The joint limit constraints used in our models were taken from Heck, Hendryson, and Rowe [18].

4 Control Systems

Although the details of the control systems for diving and vaulting differ, they are designed from a common toolbox of methods. State machines are used at the highest level to determine appropriate control actions based on the phase of the activity. Synergies are used to actuate multiple joints to achieve a single goal. Inverse kinematics is used to compute the joint angles that would place end effectors at desired locations. Proportional-derivative servos are used at the lowest level to compute the torques required to move joints to the desired locations.

Platform diving and gymnastic vaulting involve all of the basic skills (leaping, landing, tumbling, and balancing) that I propose to develop in this research. The next two sections describe the current versions of the control systems for diving and vaulting to provide a foundation for the proposed work and to point out areas for further improvement.

4.1 Control of Diving

A diver can leave the platform and enter the water in one of four ways. Figure 6 illustrates each of the four different takeoff methods. In forward dives, the athlete faces the water and leaves the platform rotating towards the water. In backward dives, the athlete faces away from the water and his back rotates towards the water. In inward dives, the athlete faces away from the water but his face rotates towards the water. In reverse dives, the athlete faces the water but his back rotates towards the water. After the diver leaves the platform, he performs a number of somersaults and

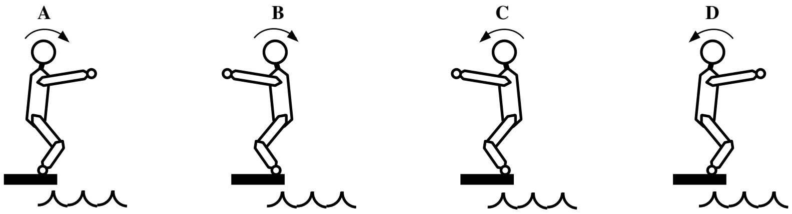


Figure 6: The four takeoff methods for platform dives. The arrow indicates the direction of rotation of the torso at takeoff. A: Forward; B: Backward; C: Inward; D: Reverse.

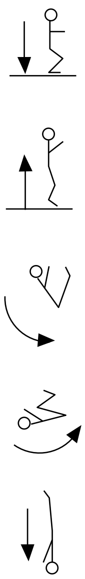
twists. The three most common positions are pike, tuck, and layout. In the pike, the athlete bends at the waist and in the tuck he bends at both the waist and the knees. In the layout position the diver performs the maneuvers with both the waist and knees straight.

To perform a platform dive, the athlete must take different actions during each phase of the dive. We reproduce this behavior in simulation by using a state machine to select the type of control needed for each phase: *compression*, *decompression*, *flight1*, *flight2*, and *entry*. The control laws for the three simulated dives are presented in table 5. In the *compression* phase, the simulated diver bends his knees, hips, and ankles in preparation for the dive. In the *decompression* phase, the diver straightens his hips and knees, while pushing off the platform with his ankles. During *decompression* the diver swings his arms to generate the appropriate angular velocity for a twisting or somersaulting dive. In *flight1* and *flight2*, the diver tucks, twists, or pikes to perform the diving maneuver. During the *entry* phase the diver straightens, puts his arms over his head, and enters the water. Although the basic phases are common to all dives, different dives require different control actions at each phase. For example, during *flight1*, a twisting diver uses control actions to twist at the waist, but a somersaulting diver bends at the waist instead.

Multiple levels of control are active during each dive. At the highest level, the state machine selects the phase of the dive to be performed. Once a phase has been selected, middle level control systems such as balancing or leaping calculate new desired locations for body segments. The desired values computed by the middle level control systems include the speed at which to raise the arms, how much to bend the knees in preparing for the jump, and the orientation of the torso. The lowest level of control takes the desired joint locations and positions the body segments using a proportional-derivative servo:

$$\tau = k_p(\phi_d - \phi) - k_v \dot{\phi} \tag{1}$$

where τ is the torque computed for the joint, ϕ is the joint location, ϕ_d is the desired joint location,



State	Inward $1\frac{1}{2}$ somersault	Reverse $3\frac{1}{2}$ somersault	Backward $1\frac{1}{2}$ twist
Compression	Prepare for jump: Bend at knees $+y$ Bend at hips $-y$	Prepare for jump: Bend at knees $+y$ Bend at hips $-y$ Swing arms behind back	Prepare for jump: Bend at knees $+y$ Bend at hips $-y$ Swing arms behind back
Decompression	Jump from platform: Bend at waist $-y$ Straighten hips Straighten knees Swing arms down Extend ankles	Jump from platform: Bend at waist $+y$ Straighten hips Straighten knees slightly Swing arms over head Extend ankles	Jump from platform: Bend at waist $+y$ Twist at waist $+z$ Straighten hips Straighten knees Swing arms forward Extend ankles
Flight1	Perform pike: Bend at hips $-y$	Perform tuck: Bend at hips $-y$ Bend at waist $-y$ Bend at knees $+y$ Bring arms down to knees	Perform twist: Untwist at waist $-z$ Bend at waist $+x$ Bring left arm over head Bring right arm across chest
Flight2		Perform tight tuck: Constrain hands to knees	Perform pike: Bring both arms to sides Unbend at waist $-x$ Bend at waist $-y$ Bend at hips $-y$
Entry	Prepare to enter water: Straighten hips Swing arms over head	Prepare to enter water: Straighten hips Straighten knees Straighten waist Swing arms over head	Prepare to enter water: Straighten hips Straighten waist Swing arms over head

Table 5: The state machine determines the control laws that are in effect at each phase of the dive.

and $\dot{\phi}$ is the joint rotation rate. The gains of the proportional-derivative servo, k_p and k_v, are chosen empirically for each joint (table 6). A smooth trajectory is used to take the desired value from its current value to a newly computed desired value because eliminating step changes in the desired values reduces the jerkiness of the generated motion.

4.1.1 Balancing

The balance controller prevents the diver from falling down before the start of the dive. Changes in the ankle and hip angles compensate for disturbances to the location of the center of mass caused by motion of the upper body and arms. This controller is active whenever the feet are in contact with the platform.

The balance controller moves the projection of the center of mass of the diver on the ground, (C_x, C_y), towards the desired center of mass, (C_{x_d}, C_{y_d}). For balance, the desired center of mass is located in the center of the polygon formed by the four contact points of the feet (the support polygon). The desired angles for the ankle are

$$ankle_{x_d} = -(k_{ax}(C_{x_d} - C_x) - \dot{k}_{ax}\dot{C}_x) \tag{2}$$

Joint	Proportional Gain	Derivative Gain
Neck (x)	30	3
(y)	60	6
(z)	30	3
Waist (x)	450	30
(y)	500	50
(z)	450	30
Hip (flight, x)	200	25
(stance, x)	400	45
(y)	500	30
(z)	40	4
Knee (y)	400	40
Ankle (flight, x)	20	0.3
(flight, y)	60	3
(stance, x)	200	3
(stance, y)	600	30
Metatarsal (flight, y)	1	0.01
(stance, y)	20	0.1
Shoulder (x)	60	6
(y)	60	6
(z)	30	3
Elbow (y)	40	4
Wrist (x)	10	1
(z)	1	0.1

Table 6: Gains used for the proportional-derivative servos on each joint for the diver simulation. The gains depend on whether the system is in flight or stance for some joints.

$$ankle_{y_d} = k_{ay}(C_{y_d} - C_y) - \dot{k}_{ay}\dot{C}_y \tag{3}$$

where k_{ax}, k_{ay}, $\dot{k}_{ax}$, and $\dot{k}_{ay}$ are gains for the proportional-derivative controller. The same equations are used for both the left and right ankles. Similar equations are used to compute the desired hip angles, the only difference being the sign and the gains

$$hip_{x_d} = k_{hx}(C_{x_d} - C_x) - \dot{k}_{hx}\dot{C}_x \tag{4}$$

$$hip_{y_d} = -(k_{hy}(C_{y_d} - C_y) - \dot{k}_{hy}\dot{C}_y) \tag{5}$$

The hips and ankles act in concert to control the left/right position of the center of mass. Using the ankles alone would cause one foot to lift off the ground if the diver leaned too far to the side of the center of support.

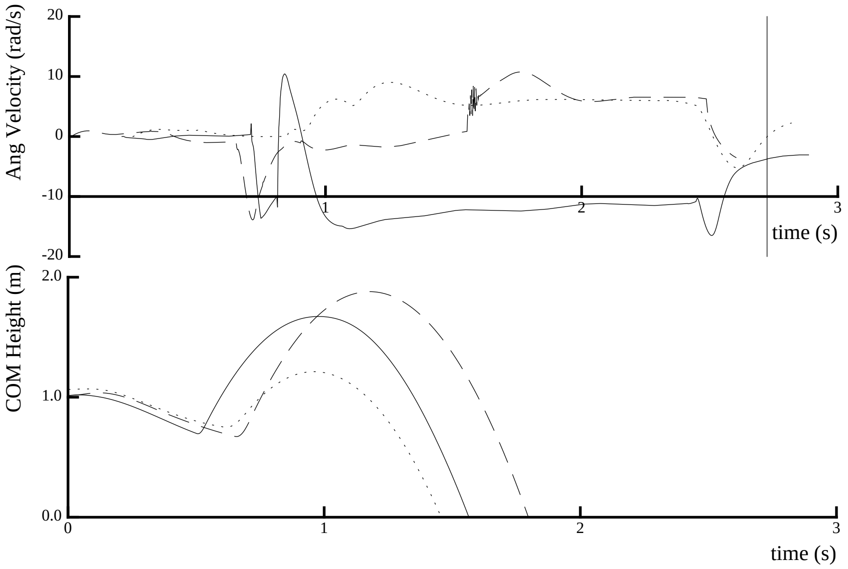


Figure 7: Graphs of the angular velocity about the somersaulting axis and the trajectory of the center of mass for three dives. The reverse $3\frac{1}{2}$ somersault tuck is represented by a solid line, the inward $1\frac{1}{2}$ somersault pike by a dotted line, and the backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist by a dashed line. The vertical line shows the point in time that the center of mass (COM) of the diver entered the water. The COM graph is truncated when the COM falls below the platform to allow the data to be scaled appropriately.

Some maneuvers, like the reverse somersault, require the desired center of mass to fall behind the support polygon. The ankles and hips pitch the upper body backwards and move the center of mass towards the desired offset location. When the center of mass falls behind the center of the support polygon the ankles alone are not effective because the metatarsus leaves the ground as torques are applied to move the center of mass forward.

4.1.2 Leaping

A number of criteria must be met to produce a successful leap from the platform. The height of the jump determines the amount of time the simulation has to perform the aerial maneuver, therefore, the height of the jump must match the angular velocity to produce the desired number of rotations. To prevent the diver from colliding with the platform, the leaping controller must generate enough forward velocity to enable the body to clear the platform. Figure 7 shows the angular velocity and the leaping height for three simulated dives.

The leaping controller is composed of two phases, *compression* and *decompression*. In the *compression* phase, the simulation servos the knees to a specified crouching height using the kinematics of the legs:

$$knee_{y_d} = \pi - acos\left(\frac{ul^2 + ll^2 - h_d^2}{2\ ul\ ll}\right) \tag{6}$$

where $knee_{y_d}$ is the desired knee angle, ul is the length of the upper leg, ll is the length of the lower leg, and h_d is the desired crouch height. The hip is used to keep the upper body vertical and the waist is used to keep the center of mass over the support polygon:

$$hip_{y_d} = -acos\left(\frac{ul^2 - ll^2 + h_d^2}{2\ ul\ h_d}\right) \tag{7}$$

$$waist_{y_d} = \frac{1}{2} hip_{y_d}. \tag{8}$$

The ankles are used to keep the foot flat on the ground.

The *decompression* phase is an open-loop controller that servos the joints to specified positions. Each dive uses different parameters to produce the amount of angular velocity and height required for the aerial maneuver used in that dive. The correct positions, velocities, and proportional-derivative servo gains for each joint must be determined empirically to generate a successful leap from the platform. As the controller rapidly moves all of the joints in the leg, the reaction forces from the platform cause the diver to leave the platform and enter the *flight* phase.

4.1.3 Tumbling

When the simulation enters the *flight* phase, the open-loop aerial controller performs the specified diving maneuver by selecting the desired joint positions for each phase of the maneuver. Transitions from the *flight1* phase to the *flight2* phase occur at different times for each dive. The desired positions are provided for each joint over time, as well as the time at which the controller transitions from the *flight1* phase to the *flight2* phase.

The aerial controller for the inward $1\frac{1}{2}$ somersault pike performs a simple somersault maneuver. As the diver falls he enters a piked position and his angular velocity about the somersaulting axis increases because angular momentum is conserved during flight.

The aerial controller for the reverse $3\frac{1}{2}$ somersault tuck is similar to the one for the inward $1\frac{1}{2}$ somersault pike but the required angular velocities are much larger. To generate the large velocities, the diver generates more angular momentum during the *decompression* phase and enters a tight tuck position. Constraints are used to place the hands on the knees and maintain the tight tuck position. Without the constraints, centrifugal force pulls the diver out of the tight tuck position.

In the backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist the diver performs a $\frac{1}{2}$ twist about the vertical axis as he leaves the platform followed by a $1\frac{1}{2}$ piked somersault about the y-axis before entering the water.

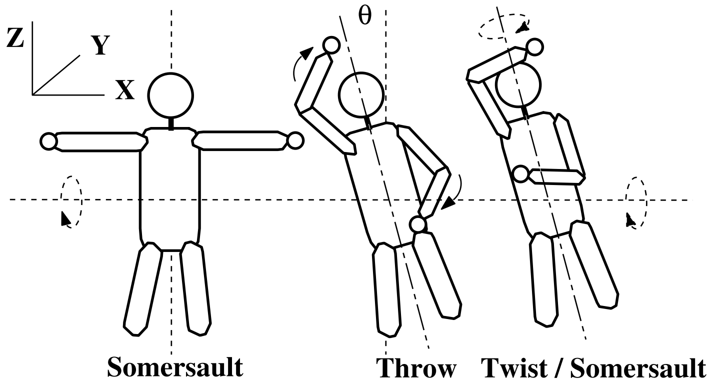


Figure 8: As a somersaulting diver brings one arm over the head and one arm to the side, a rotation about the y-axis is induced. If the diver had nonzero angular momentum, a twist would be induced about the z-axis in order to conserve angular momentum. Because of the rotation by θ, the body-local x-axis is no longer aligned with the angular momentum axis and the body begins to twist.

The aerial controller must transfer angular velocity about the somersaulting axis to angular velocity about the twisting axis. The technique used to make the transfer is presented in Frohlich [13]. A diver having nonzero angular momentum and performing a somersault about the x-axis can initiate a torque-free twist by throwing the left arm down and the right arm up (figure 8). This motion results in a counterclockwise rotation of the diver about the y-axis. This rotation causes a twisting motion about the z-axis because the diver's principal axes of rotation are no longer aligned with their angular momentum axis, and the angular velocity vector now has components along both the x and z-axes.

4.1.4 Results

We simulated three dives using this control framework. Figure 9 shows images from the three simulated dives and a human dive. Of the three, the inward $1\frac{1}{2}$ somersault pike (difficulty rating 2.0 [35]) is the least difficult dive to control because the somersault does not require a large angular velocity and the diver has a large window of time to prepare for entry into the water. The angular velocity for the inward $1\frac{1}{2}$ somersault pike rises slowly, remains nearly constant for the duration of flight, and then drops when the diver untucks (figure 7).

The reverse $3\frac{1}{2}$ somersault tuck (difficulty rating 3.4 [35]) is more difficult to control than the inward $1\frac{1}{2}$ somersault pike because the required angular velocity is larger and a very tight tuck is

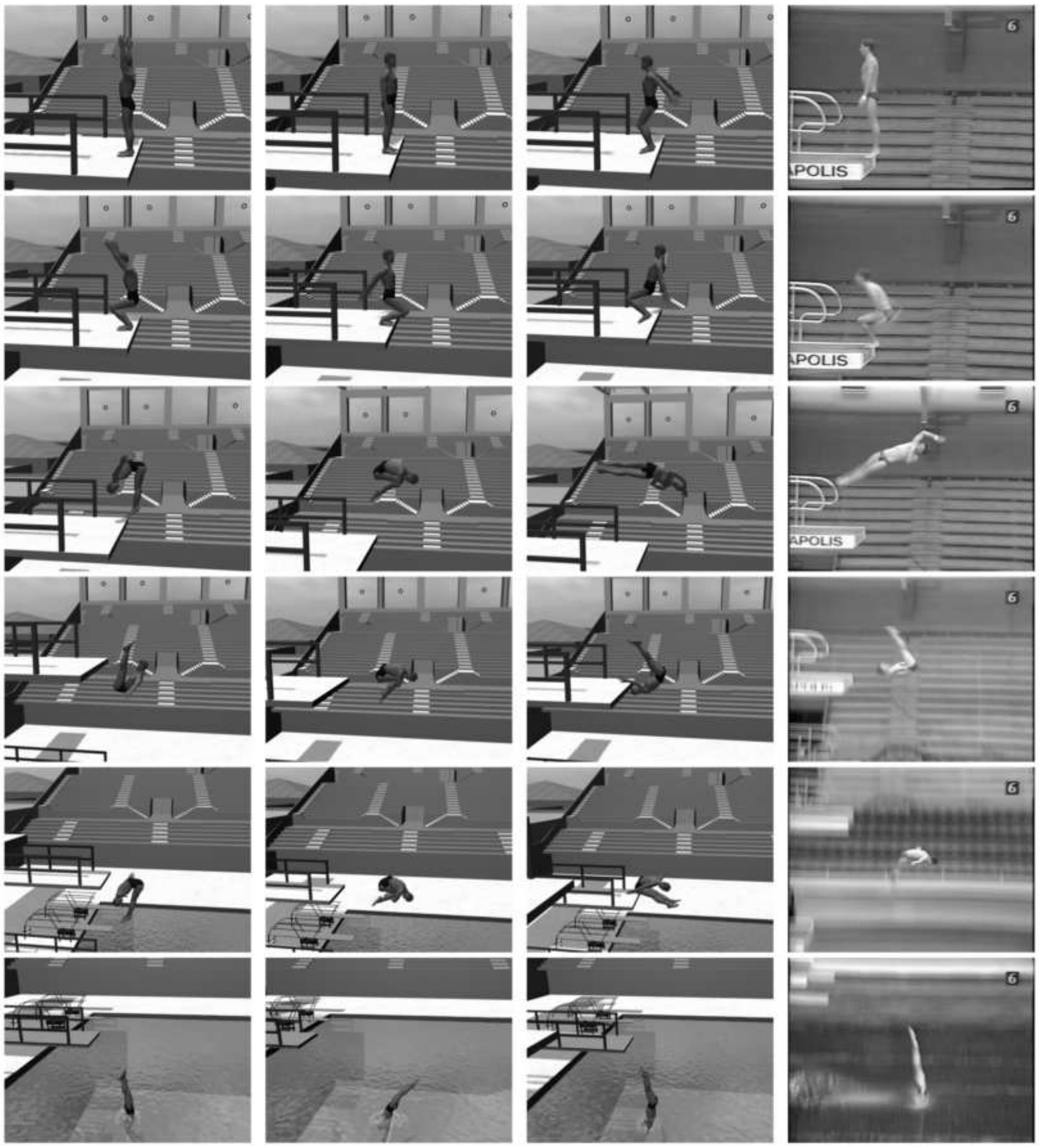


Figure 9: A sequence of images from the three simulated dives. The inward $1\frac{1}{2}$ somersault pike is on the left and the reverse $3\frac{1}{2}$ somersault tuck is shown in the second column. The third and fourth columns show the backward $1\frac{1}{2}$ somersault pike with $\frac{1}{2}$ twist as performed by the simulated diver and by an Olympic athlete. Each image is separated by 0.5 s and time increases from the top to the bottom image in each column. In the reverse $3\frac{1}{2}$ somersault tuck, the diver is at the beginning of each somersault in frames 3, 4, and 5.

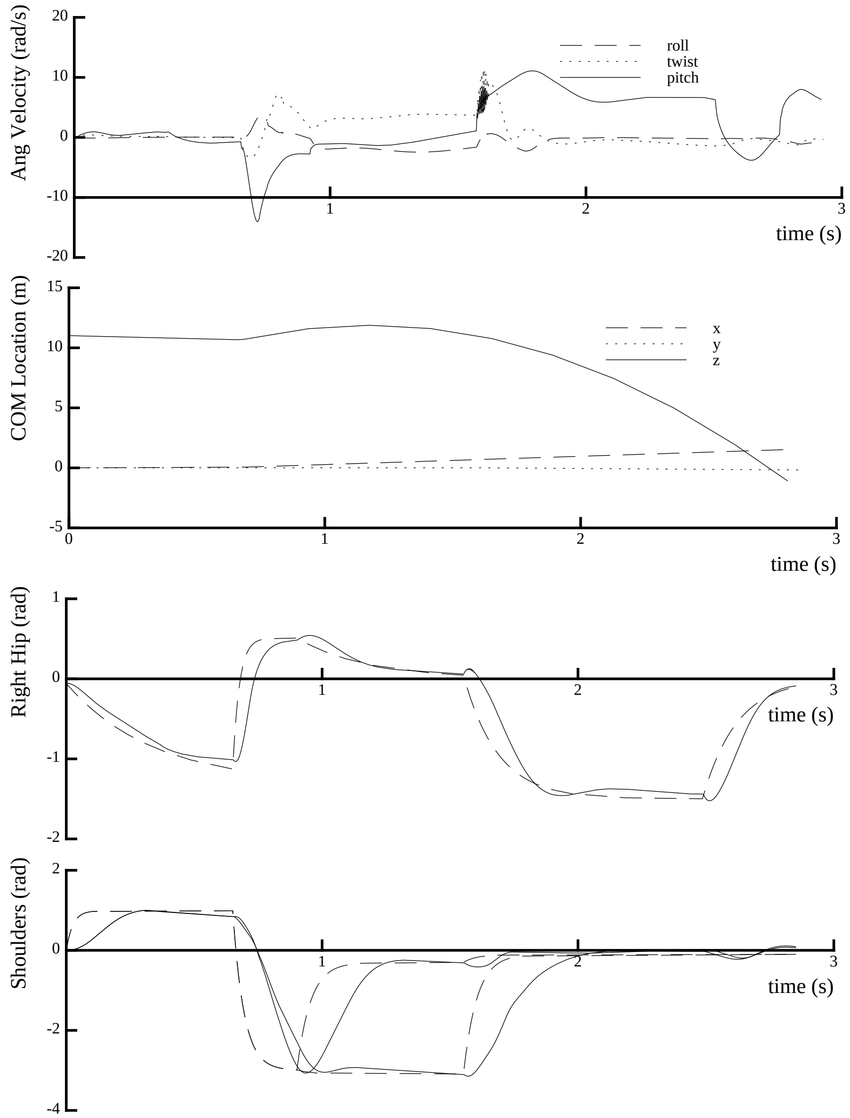


Figure 10: Graphs of the location of angular velocity, center of mass, and selected joint motion for the twisting dive. The dashed lines in the lower two graphs represent the desired location of a joint, and the solid lines represent the actual location. The left hip follows the same trajectory as the right hip.

required during the flight phase to maintain the angular velocity. A graph of the angular velocity about the somersaulting axis for this dive is shown in figure 7. The rotation rate is much higher and is in the opposite direction of the inward and backward dive.

The backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist (difficulty rating 2.1 [35]) was the most difficult dive to control because it involves rotations about multiple axes. However, this dive is rated as just slightly more difficult than the inward $1\frac{1}{2}$ somersault pike for human divers. The backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist required the generation of twisting torque at liftoff because the torque-free method, described in the aerial control section, did not produce enough angular velocity about the twisting axis. The angular velocities for this dive are shown in figure 10.

4.2 Control of Vaulting

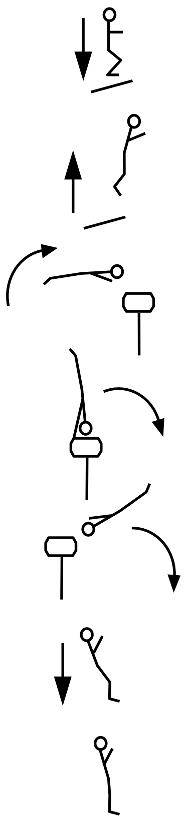
To perform a vault, the gymnast uses a springboard to launch herself toward the horse, pushes off the horse with her hands, and lands on her feet on the other side of the horse. While airborne after leaving the horse, the gymnast performs a combination of somersaults or twists. In the handspring vault, the gymnast performs a full somersault over the horse while keeping her body extended in a layout position. In the arabian vault, the gymnast performs a $\frac{1}{2}$ twist piked and lands facing the opposite direction.

The handspring and arabian vaults are structured by a state machine with seven phases: *compression*, *decompression*, *flight1*, *horse*, *flight2*, *landing* and *standing*. The control actions required during each of the seven phases are presented in table 7. The simulation of each vault begins in a flight state just before touchdown on the springboard. The initial conditions were estimated from video footage and confirmed with data measured from gymnasts [50] (forward velocity is 6.75 m/s and the height of the center of mass is 0.9 m).

The vaulting simulation uses the same control hierarchy as the diving simulation. A state machine selects the appropriate phase for the vault and the middle level controllers calculate new desired angles for the joints. The lowest level of control uses proportional-derivative servos to compute the torque required to generate the desired motion. The next four sections will discuss the implementation of the middle level controllers for vaulting: leaping, tumbling, landing, and balancing.

4.2.1 Leaping

The leaping controller has two phases, *compression* and *decompression*. During *compression*, the simulated gymnast lands on a springboard that is supported by a linear spring and damper (figure 11). Using a technique called *blocking*, the control system positions the hips forward before



State	Handspring	Arabian
Compression	Place feet in front of COM: Bend knees $+y$ Bend hips $-y$ Bend waist $-y$ Compress springboard	Same as handspring
Decompression	Jump from springboard: Decompress springboard Straighten knees Straighten hips Straighten waist Extend ankles	Same as handspring
Flight1	Move hands to vault: Swing arms upward $-y$ Bend at hips slightly $+y$ Bend neck upwards $-y$	Same as handspring
Horse	Initiate somersault: Keep wrists flat Passively swing over vault	Initiate twist: Keep wrists flat Passively swing over vault Pike at hips $-y$ Pike at waist $-y$ Twist at hips $-z$ Twist at waist $-z$
Flight2	Prepare to land: Extend legs for wide stance $\pm x$	Prepare to land: Extend legs for wide stance $\pm x$ Unpike hips $+y$ Unpike waist $+y$ Untwist hips $+z$ Untwist waist $+z$
Landing	Recover balance: Straighten neck Bend at knees $+y$ Bend at hips $-y$ Bend at waist $-y$	Same as handspring
Standing	Stand up: Raise arms over head Straighten knees Straighten hips Straighten waist	Same as handspring

Table 7: The state machine determines the control laws that are in effect at each phase of the vault.

touchdown on the springboard so that much of the horizontal velocity at touchdown is transformed into rotational and vertical velocity at liftoff.

When the springboard reaches maximum deflection, the simulation enters the *decompression* phase. In this phase the control system extends the knees, pushing on the springboard, and adds energy to the system. As the springboard rebounds, the simulation extends the leg muscles rapidly and is launched into the air. When the simulated gymnast's feet leave the springboard the *flight1* phase begins.

The leaping control system is based on the diver's, but is different in a couple of ways. The gymnast's controller does not require as much leg compression as the diver's because the gymnast leaps from a compressible springboard. The diver is standing motionless on the end of the platform–

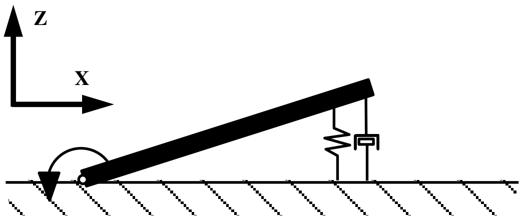


Figure 11: The springboard is modeled as a rigid board supported by a linear spring and damper. The mass of the board is 9.43 kg. The moment of inertia about the center of mass of the board is (0.2927, 1.169, 1.461) kgm^2. The spring constant is 80000 and the damping constant is 500.

the leaping controller must generate the forward and vertical velocity needed for the dive. The gymnast has forward velocity before contacting the springboard and the controller must convert the forward velocity to vertical velocity.

4.2.2 Tumbling

During the *flight1* phase, the control system prepares to put the gymnast's hands on the horse by positioning her hands on the line between the shoulders and the desired hand position on the vault:

$$\gamma_{\mathrm{y_d}} = \lambda_{\mathrm{y}} - \phi \tag{9}$$

where $\gamma_{\mathrm{y_d}}$ is the desired shoulder angle relative to the body, λ_{y} is the angle between horizontal and a vector from the shoulder to the desired hand position on the vault, and ϕ is the pitch angle of the body with respect to horizontal (figure 12). Because the shoulders are moving toward the vault during flight, this control law reduces the relative velocity between the hands and the horse. This technique is termed ground speed matching in the biomechanical literature. The wrists are servoed to ensure the hands hit palm down and parallel to the surface of the horse

$$\phi_{\mathrm{y_d}} = \arccos(V_{\mathrm{arm}} \cdot V_{\mathrm{x}})$$

where $\phi_{\mathrm{y_d}}$ is the desired angle for the wrist, V_{arm} is a vector from the palm to the shoulder, and V_{x} is a vector along the positive x-axis.

During the *horse* phase the gymnast generates the angular and vertical velocities required for the maneuver. No torque is applied at the shoulders or wrists, but the gymnast's angular and forward velocity carry her over the horse. During a handspring vault, the gymnast maintains a

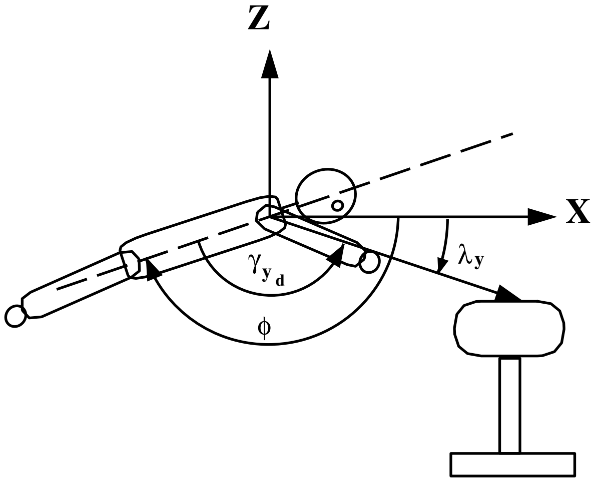


Figure 12: The control system performs ground speed matching to ensure that the gymnast's hands will contact the horse at the desired location. The shoulder angle that will allow this is γ_{y_d}.

layout position. During an arabian vault, the gymnast begins to perform a maneuver for rotation about the twist axis. To initiate this maneuver, the gymnast enters a pike position before her hands leave the horse. When the hands leave the horse, the *flight2* phase begins.

During the *flight2* phase, the control system completes the specified maneuver and prepares to land. For the handspring vault, the control system simply maintains the layout position. For the arabian vault, the control system performs a $\frac{1}{2}$ twist by moving the limbs in a specified sequence to perform a "swivel hips" maneuver (figure 13). The "swivel hips" maneuver allows the gymnast to rotate the body 180° about the z-axis even when the gymnast has little angular momentum. Often gymnasts will swing their arms, much like divers, to generate more twist, but our control system did not use this strategy. All of the twisting velocity for the arabian vault was generated with the "swivel hips" maneuver.

As the gymnast nears the mat the control system prepares for landing. The feet are spread slightly to give a larger area of support. The angle of approach between the legs and the ground is adjusted to place the feet so the forward velocity of the gymnast is near zero when the center of mass moves into the center of the support polygon.

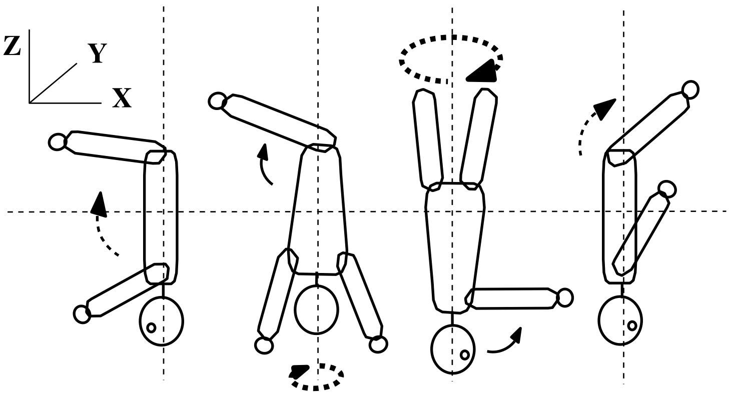


Figure 13: The gymnast performs a one half twist using a "swivel hips" maneuver. As the gymnast leaves the horse she rotates her upper torso as far as possible while unpiking the lower torso. As the legs unpike, the body performs the twist about the z-axis. The gymnast then "swivels" her hips and ends the maneuver by pulling her arms down while returning to a piked position.

4.2.3 Landing

When the feet hit the ground, the *landing* phase begins. The landing control system must remove the horizontal and rotational energy required for the maneuver and establish an upright, balanced position. The landing controller for the handspring vault is similar to the controller for the arabian except the orientation of the landing is reversed.

During landing, the knees and waist absorb most of the energy. The knees are bent and the damping gains in the knees are increased substantially (table 8). At touchdown, the gymnast is not in a balanced state because the projection of the center of mass is outside the support polygon but the forward velocity of the gymnast causes the center of mass to move towards the center of the support polygon. If the legs were positioned correctly at touchdown the forward velocity will be near zero when the center of mass is over the center of the support polygon. Feedback control is used to servo the ankles to adjust the forward velocity of the center of mass as it moves forward.

4.2.4 Balancing

When the gymnast's forward velocity is nearly zero the *standing* phase begins. The balance controller straightens the hips and knees and causes the gymnast to stand vertically. The balance controller is the same as the diver's, except the mass and moments of inertia of the female model require different gains for the proportional-derivative controllers. This controller not only allows the gymnast to stand

Joint	Proportional Gain	Derivative Gain
Neck (x)	23	2
(y)	23	2
(z)	23	2
Waist (x)	300	25
(y)	340	40
(z)	300	25
(landing, x)	300	50
(landing, y)	340	80
(landing, z)	300	50
Hip (flight, x)	133	13
(stance, x)	260	26
(landing, x)	260	50
(y)	340	20
(z)	60	3
Knee (flight, y)	300	5
(stance, y)	340	17
(landing, y)	200	35
Ankle (flight, x)	75	0.25
(flight, y)	75	2
(stance, x)	200	2.3
(stance, y)	400	20
Shoulder (flight, x)	45	1.5
(flight, y)	45	1.5
(flight, z)	45	1.5
(horse, x)	100	1
(horse, y)	100	0.001
(horse, z)	100	1
Elbow (flight, y)	30	2
(horse, y)	200	10
Wrist (flight, x)	3	0.3
(flight, z)	0.3	0.01
(horse, x)	5	0.01
(horse, z)	18	0.9

Table 8: Gains used for the proportional-derivative servos on each joint. The gains depend on whether the system is in flight, stance, landing on a mat, or pushing off the horse.

Variables	Human			Simulation
	Mean	Min	Max	
Mass (kg)	47.96	35.5	64.0	64.3
Height (m)	1.55	1.39	1.66	1.64
Board contact (s)	0.137	0.11	0.15	0.105
First flight (s)	0.235	0.14	0.30	0.156
Horse contact (s)	0.245	0.19	0.30	0.265
Second flight (s)	0.639	0.50	0.78	0.632
Horizontal velocity (m/s)				
Board liftoff	4.61	3.97	5.26	4.01
Horse touchdown	4.61	3.97	5.26	4.01
Horse liftoff	3.11	2.48	3.83	2.83
Vertical velocity (m/s)				
Board liftoff	3.34	2.98	3.87	3.81
Horse touchdown	1.26	0.74	2.39	2.13
Horse liftoff	1.46	0.56	2.47	1.10

Table 9: Velocities, contact times, and forces for a simulated gymnast and human data measured by Takei [50]. The human data was averaged from 24 subjects. This data shows that the simulation falls within the range of average values for athletes performing this maneuver.

after landing, but also compensates for disturbances resulting from the motion of other parts of the body while she is standing. Balance is also maintained as the simulated gymnast throws her arms back in a gesture of success at the end of the vault.

4.2.5 Results

Comparison of recorded biomechanical data and data collected from the handspring simulation was used in evaluating the success of the motion generated by the simulation. Table 9 compares data from the handspring vault performed by 24 gymnasts in the 1987 Pan American Games to our simulated handspring vault [50]. Most of the simulated data fell within the range of the measured data.

A handspring vault is easier for gymnasts to perform than the arabian vault. We found the handspring vault easier to control than the arabian because it is easier to recover balance when the twisting angular velocities are near zero.

Subjective comparisons were made between the simulated arabian vault and video footage of an actual arabian vault (figure 14). The largest difference between the simulation and actual performance was the motion of the arms during the *flight2* phase. In the actual performance, the gymnasts

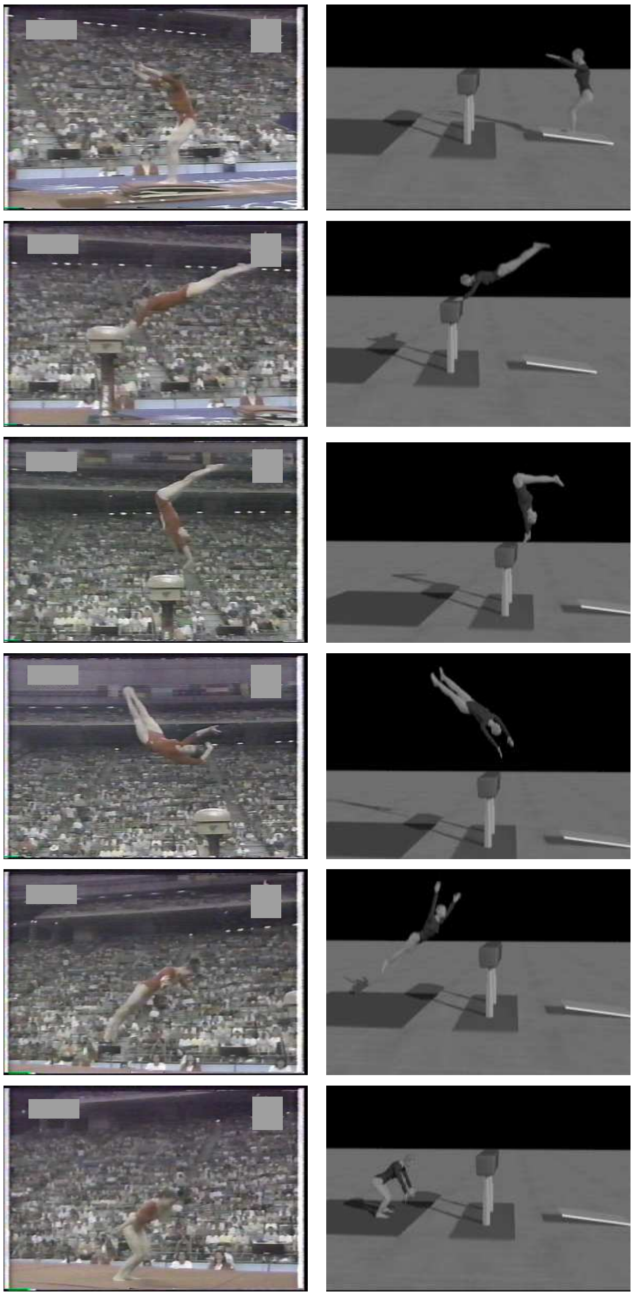


Figure 14: Comparison of an arabian vault from an Olympic performance and a simulated performance.

pulled their arms towards their torso, as a diver does during a twisting dive. In our simulation, the gymnast holds her arms out over the top of her head because additional twist was not needed. If the simulation did not perform a tight pike and used the arms to generate the additional twist, it would more closely resemble the actual performance.

4.3 Transitions

Transitions between middle level controllers were used in the simulation of both diving and vaulting. The diver transitioned from a balance controller to a leaping controller and from a leaping controller to an aerial controller. The gymnast transitioned from a leaping controller to an aerial controller and from an aerial controller to a landing controller and finally to a balance controller.

The transitions were hand tuned for the three dives and two vaults. To produce a new dive or vault, the user would iteratively adjust control parameters in either the first or second controller until the final conditions of the first controller fell into the set of acceptable initial conditions for the second. Determining which control parameter to adjust and how to adjust it was often difficult, especially when a sequence of transitions were required. Adjusting parameters of one control system to perform a successful transition would sometimes break the preceding or following transitions. I plan to address this problem by developing control systems that perform robustly under a wider range of initial conditions and generate a wider range of final conditions. When the acceptable sets of conditions are larger, it will be easier to transition between controllers.

5 Proposed Research

In networked virtual environments, participants should be able to interact with one another as well as with simulated synthetic actors. One way to implement such a system is to build a library of control systems for a wide variety of actions and behaviors. The control systems should be general enough to apply to any size or type of synthetic actor. The control systems should generate motion that looks natural, making it difficult for the participants to distinguish between synthetic actors and other human users. When a synthetic actor changes from one behavior to another, the control system should provide smooth transitions. The desired behaviors for the synthetic actors could be automatically generated by high level behavioral programs or could be controlled by high level commands from the user.

I plan to work towards these goals by developing a small library of control algorithms for dynamically simulated synthetic actors. Dynamic simulation will generate physically correct motion and

I believe, based on previous results, that carefully designed control systems will generate natural-looking motion. The control systems I plan to develop will transition smoothly from one control system to another and I will perform a number of evaluations to measure the robustness of the control systems and the naturalness of the motion.

I will address four items while developing the control systems:

- A dynamic model of a human figure with enough degrees of freedom to exhibit expressive motion.
- Control systems for jumping, performing aerial maneuvers, landing, and balancing.
- Control systems for transitions from one behavior to another without manual tuning of the control system parameters.
- Evaluation methods to compare the generated motion with data obtained from human performances.

5.1 Dynamic Model

A dynamic model of the human body was presented in section 3 of this proposal. Most of the work on the model is complete, but there are a number of minor improvements to be made.

I plan to develop a dynamic model with properties that more accurately reflect data from the anatomical literature. The current model uses polygonal body segments that overlap at the joints and the mass and moments of inertia are computed from this polygonal model. Graphically the overlap is necessary to prevent gaps when the body segments move through their range of motion. However, the overlap introduces errors in the dynamic properties of the model because the mass of each joint is included in both the proximal and distal segments. I will remove the overlap at each joint in the version of the polygonal model used to compute the dynamic properties while continuing to use the overlapping model to display the graphics.

I plan to improve the collisions between the feet and the ground by using different constraints than those described in section 3. The current set of constraints does not allow accurate movement of the foot when it is in contact with the ground. For example, the roll constraint will prevent the simulation from standing on the side of the foot. Using four contact points per foot instead of two will remove the need for roll constraints about the center of mass of each foot. Realistic coefficients of friction should be added to the model to generate more realistic behavior during landing, otherwise, the simulations could perform landings that would be impossible on a real mat or floor.

The SD/Fast simulation package implements three-dimensional joints two different ways. The current simulation uses three-axis gimbal joints at the shoulder and hip, but the joint becomes gimbal locked when two of the three axes become aligned. The improved model will use ball joints and the joint location will be expressed as a quaternion, eliminating the gimbal lock problem.

I also plan to explore whether a more sophisticated numerical integrator will improve performance. Currently a Runga-Kutta fourth-order, fixed time-step integrator is used to move the simulation forward in time. I will implement an implicit integrator to take advantage of the improved accuracy of an adaptive, stiff method.

5.2 Control Systems

The control systems used in the vaulting and diving simulations are not very robust because small changes in the initial conditions require manual adjustment of control parameters. For example, if the forward velocity of the gymnast on the springboard is below 6.75 m/s, the user must adjust the desired hip angles during the *flight2* phase or the gymnast will land in a configuration outside the region where the landing controller can restore balance. Currently the adjustments must be made by hand but more robust control systems would make them automatically by computing the desired hip angles as a function of the forward velocity.

The control systems I plan to develop will operate under a wide range of initial conditions including the velocities and accelerations experienced during most athletic tasks. For example, gymnasts normally have forward velocities between 2.5 m/s and 3.0 m/s during the landing phase of a vault and divers leave the platform with an angular velocity between 15 rad/s and -15 rad/s. A wide range of acceptable initial conditions will allow the four proposed controllers to be used in sequence to generate more complex motion. For example, a standing broad jump could be produced by using the balance controller, the leaping controller (with desired forward and vertical velocities), the aerial controller, and then the landing controller.

The four controllers will be hand designed because the large number of degrees of freedom of the model would make it difficult to develop automatic controllers. I will derive equations that map the desired behavior to the joint torques required to generate the correct trajectories. The initial gains for the proportional-derivative servos will be tuned using a manual trial and error approach. Once a set of gains have been determined that generate acceptable performance, a simple, automated search algorithm will be used to fine tune the gains.

In keeping with our goal of producing natural-looking motion, the proposed controllers will incorporate biologically inspired constraints. The current muscle model is simply a torque source with joint limits and this simple model can produce unnatural joint trajectories. In biological systems, the position and velocity of a joint are factors that determine the maximum joint torque. I will add torque constraints based on joint position and velocity to make the torques more realistic. In the case of biarticulate muscles, which produce torque in two joints at the same time, the torque constraint would be based on the position of both joints. For example, as the knee flexes, the biarticulate gastrocnemius muscle generates a larger maximum torque at the ankle.

The feedback rate for the control system should be reduced. Currently the control systems are computed at the simulation time step (0.0027 s for the diver and 0.0017 s for the gymnast). In humans, the fastest postural response to the stretch stimulus is 0.07 s, about 25 times slower than the feedback rate of our simulations [15]. I plan to reduce the feedback rate used by the control systems to more closely mimic the human motor control system. Humans compensate for the low feedback rate with feedforward motor programs and I will add feedforward mechanisms to some stages of the control systems.

Early animators of two-dimensional figures realized that characters which remained absolutely motionless for a period of time appeared "lifeless" [51]. The same observation is true of human motion. Even when people remain still, small involuntary motions (like postural sway) are visually noticeable. Perlin uses noise in the joint trajectories of a kinematically animated, articulated figure to produce appealing character motion [40]. I propose to explore similar methods, but will integrate the noise into the desired trajectories, feedback sensors, and joint torques of the simulation. I will generate high frequency noise for cases where muscle fatigue comes into play and low frequency noise in cases where internal disturbances such as breathing affect the motion of other segments.

5.2.1 Balancing

I have developed a balance control system that allows the simulated gymnast and diver to stand in the presence of disturbances, but it is limited in a number of ways. Balance is maintained only if the feet are planted together side by side. The control system corrects for disturbances introduced by motion of the upper body, but large disturbances usually cause the control system to fail. For example, if the gymnast swings her arms upward with a velocity greater than 41.5 rad/s, the balance control system will fail to keep the center of mass within the support polygon.

The proposed control system will maintain balance for a wider range of initial conditions and internal and external disturbances. The control system will attempt to maintain balance for any configuration of foot placements on the ground and the feet will no longer need to be side by side. The control system will maintain balance even if the simulation is in a crouch or only one foot is making contact with the ground.

The user will be able to request a number of actions from the balance controller. A request to balance on one foot will cause the controller to shift the center of mass over that foot and lift the other off the ground. A request for a crouch will cause the controller to bend the knees to position the hips at the specified height above the ground. The user will also be able to specify motion for joints of the upper body while the balance controller compensates for the resulting disturbance by moving the waist, hips, and ankles to keep the center of mass over the support polygon.

Both feedforward and feedback mechanisms will be used to maintain balance. The torques in the feedback system will depend on the state of the system. The feedforward control system will use precomputed torques and will be triggered by changes in the state of the system. The feedforward mechanism will activate 70 ms after the joints of the lower limb exceed specified thresholds and will apply the precalculated torques in a proximodistal order–starting at the ankle, then the knee, and finally the hip. The order in which the torques are applied and the delay before they are applied mimics the muscle activation sequence in the human body [15]. Feedback from the location of the projected center of mass within the polygon of support, combined with the location of the head, will be used to make fine adjustments once the feedforward torques have been applied. User-specified motion for the upper body will elicit a response from the feedforward control system before the initiation of upper body movement to prevent disturbances that the control system cannot handle. Activating feedforward control before initiation of upper body movement has been observed in humans [8].

The balance control system will be assessed using many placements and orientations of the feet with respect to the body. External disturbances will be applied to the torso from eight compass directions to determine the largest disturbances that can be tolerated in normal standing, standing on one foot, and crouched standing. Table 10 shows the maximum force and velocity disturbances that can be applied to the center of mass of the torso with the current balance controller. I will also develop a set of tests similar to Nashner's translating platform, where EMG data and joint trajectories were recorded as the ground plane translates and rotates under the feet. I will compare the reaction time and trajectories produced by the balance controller to data from Nashner's subjects. Subjective tests will be used to assess the motion after noise has been introduced by comparing the motion to videos of human performance.

Test	Male	Female
Horiz. Velocity (m/s)		
$+X$	0.28	0.30
$-X$	0.45	0.55
$\pm Y$	0.90	0.80
Horiz. Force (N)		
$+X$	80.0	60.0
$-X$	125.0	90.0
$\pm Y$	280.0	190.0

Table 10: The balance controllers for the male and female models were tested to determine robustness. The values in the table indicate the maximum horizontal velocities and maximum horizontal forces that could be applied to the center of mass of the torso without causing the model to fall over. The disturbances were applied in the four compass directions and the force was applied for 0.3s. The horizontal velocities are slower than a normal walk, and the largest impulse falls within human capacity to maintain balance.

5.2.2 Leaping

The leaping controller generates joint motion to push against the ground and propel the simulated human into the air. The existing control system requires that the user specify many unintuitive parameters, such as hip, knee, and ankle velocities and gains. Desired leaps are achieved by manually adjusting the parameters until the simulation produces the desired result. The angular velocity is controlled by moving the center of mass forward or backward of the center of the support polygon and by swinging the arms in the appropriate direction.

With the proposed controller, the user will specify a jump with a small number of intuitive parameters, such as the desired leaping height and the angular velocity of the body at that height. The control system will compute the torque needed at each joint to propel the body to the desired height. The desired angular velocity will be generated by swinging the arms and pitching the upper torso forward or backward.

I will develop a robust control system for leaping by determining the relationship between the rate of change of joints in the lower torso (ankles, knees, and hips) and the height and angular velocity of the jump. I will use the relationship between the velocity of the waist and hip angles and the angular velocity of the torso to control the angular velocity of the leaping maneuver. The relationships will be determined empirically.

Experiments will evaluate how well the leaping controller produces the desired height and angular velocity. The control system will be tuned to ensure that the simulation cannot leap higher than human athletes. The joint motion and joint torques in the lower leg will be compared against biomechanical data from subjects performing a maximum-height jump [38].

5.2.3 Tumbling

I have developed an aerial control system to produce twists, somersaults, and combinations of the two for diving and vaulting. The somersaults occur in a tucked, piked, or layout position. Transitions from somersaults to twists are performed with arm motion, hip motion, or a combination of the two. To control a somersault or twist, the user specifies the desired joint angles for a body position, along with an initial angular velocity large enough to perform the maneuver. When the desired number of rotations have been performed, the user specifies the required body position to end the maneuver.

The proposed control system will not require the specification of joint angles for body position. For somersaults the user will provide the desired position (tuck, pike, or layout). The user specifies either the number of rotations or the amount of time allotted to perform the maneuver. The control system will then compute how tight the tuck or pike must be in order to complete the desired maneuver. The user will use a similar specification for twists, but the control system will generate the twist with the hands, hips, or both.

I will develop a robust control system for tumbling by using the relationship between the moment of inertia and the angular velocity of a body in flight. For somersaults I will control the number of rotations by controlling the tightness of the tuck or pike. For example, if the desired position is a pike, the angular velocity can be controlled by changing the angles of the waist and hips. The layout position makes this type of control impossible and the success of the maneuver depends on the initial conditions.

I will evaluate the robustness of the aerial control system by measuring deviations from desired results as the control system performs pike, tuck, and layout somersaults. Similar tests will be used for the twists. I will explore whether accuracy can be improved by adjusting the configuration of the body segments throughout the maneuver to control the rotation rate. The stability of a layout somersault will be compared to the findings of Playter [41].

5.2.4 Landing

Landing on a flat surface requires reducing the horizontal, vertical, and angular velocities seen in free fall to those required to achieve balance. Energy is removed from the system during landing by the athlete's muscles and the damping effects of the landing surface.

The vaulting simulation uses a landing controller to transition from the *flight* phase following the vault to a balanced final state. The balance controller is activated once the landing controller has reduced the horizontal velocity of the center of mass below 0.1 m/s. The user specifies how much the knee should bend and how much damping is applied at the knee during the landing. Both parameters affect how much energy is removed from the system as the gymnast lands.

The proposed landing controller will allow landings for conditions in which the center of mass is moving in the same direction as the simulation is facing, as well as in the opposite direction, for any yaw orientation of the body, and for any placement of the feet. The horizontal velocity and angular velocity about the somersaulting axis will be used to determine the waist angle at landing so that the angle between the legs and the ground at touchdown lead to a successful landing. The control system will twist at the waist opposite to any angular velocity about the twisting axis to remove any twisting component during the landing. The control system will use the horizontal velocity and location of the center of mass relative to the support polygon to determine how much to bend the knees during landing. Arm movement will be used for fine control over the velocity of the center of mass when the simulation nears a vertical stance. The control system will take into account the effects of landing on different surfaces such as gymnastic mats and hardwood floors.

I will determine the maximum horizontal velocity of the center of mass for which the controller can produce valid initial conditions for the balance controller. The tests will include landings where the simulation is facing forwards and backwards and where the angular velocities about the somersaulting axis are both positive and negative. I will compare the simulated ground reaction forces to results from Özgüven and Berme to determine the validity of the ground collision model [36].

5.3 Transitions

One of the goals of the proposed research is transitions from one behavior to another without manual adjustment of control system parameters. Transitions will be accomplished by overlapping the initial and final state spaces of each controller. The key to the success of the transition will be the size of the state-space capture region of each controller, defined as a subset of all possible initial conditions for which the control system will successfully complete the behavior.

Experiments will be performed by combining the four different controllers with a variety of behaviors. For example, animation of a gymnastic floor routine would require the use of all four control systems. Evaluation of the transitions will be based on whether control parameters had to be manually adjusted to produce a successful transition.

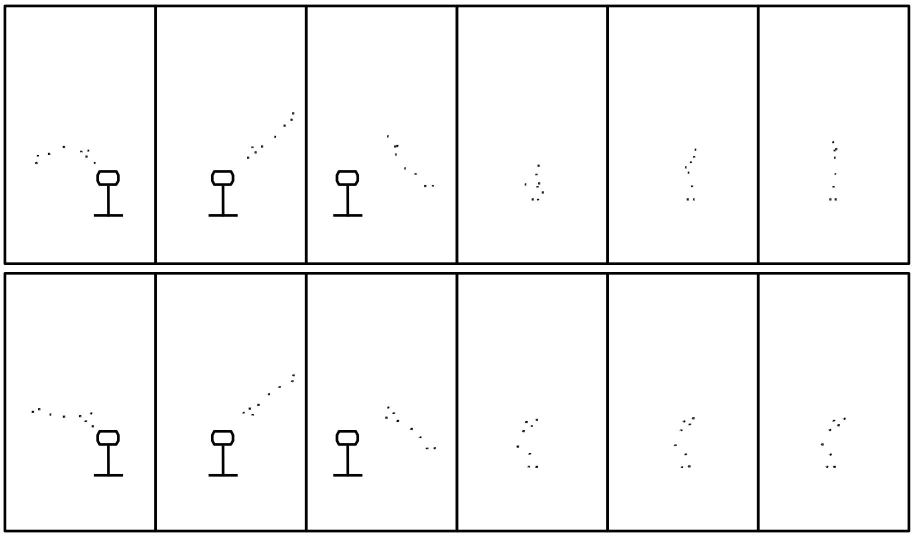


Figure 15: The black dots mark the joint locations on a gymnast as she performs a handspring vault. When a series of markers are digitized and played back over time the viewer normally perceives the motion as that of a human although the motion is hard to recognize in still images. The dots on the bottom represent a simulated gymnast and the ones on the top a human gymnast. The frames proceed from left to right and have been sampled every 0.66 s.

5.4 Evaluation of Results

To determine the naturalness of the simulations and control systems, I will compare the resulting motion with video, motion capture data, and recorded biomechanical data of human performances. One of the simplest evaluation methods is to compare human performance videos and graphical renderings of the simulated motion. Subjects can be asked to compare the motion but their comments will be subjective and the level of detail at which the subjects compare data will vary widely. I have informally used this technique for comparison of the simulated divers against Olympic athletes. Comments from the general public are favorable, however, comments from diving coaches and biomechanists point out many problems with the motion.

Another method of comparison composites motion capture data beside simulated data. Points are used because the motion capture data does not to look plausible when played back through a graphical model. Subjects will be asked which motion appears the most natural–the one on the left or the one on the right. Combinations in which both sides are simulated, both sides are motion capture data, and one side is simulated and the other motion capture will be used for this comparison. The results of the comparison will have less bias than the video comparison, however, the digitization

Activity	Human	Simulation	Don't Know
Diving	47%	20%	33%
Vaulting	62%	19%	19%
Running	83%	9%	8%
Bicycling	45%	22%	33%

Table 11: Subjects viewed two sets of digitized data, one human and one simulation, side by side and were asked which set of motion appeared more natural. The table gives the percentage of subjects who responded that the simulation or the human data was more natural. Some subjects were unable to detect a difference in the motion and responded that they did not know which set of data looked more natural. 394 subjects participated in the survey.

process may have removed some of the subtleties of the motion. Figure 15 is an example of a comparison between digitized data from a live gymnast and a simulated gymnast. Both sets of data were digitized by hand to reduce bias based on the noise introduced by the digitization technique.

We are currently using this evaluation method on the World Wide Web using MPEG-encoded video clips of four simulations developed in the Animation Lab. Table 11 gives preliminary results of the survey. There are a number of problems with the current survey. With MPEG video, the playback rate of the digitized data depends on the architecture of the viewer's machine. Presentation of the data at a rate slower or faster than 30 frames per second affects the viewer's perception of the motion. Some participants used an MPEG player that allowed the motion to be played in a loop while others did not. Viewing the motion multiple times could alter the viewer's perception of the motion. The use of dots to represent the joint locations was adequate for most subjects to recognize the motion but certainly does not convey the same level of detail as a full graphical rendering. Some of the data sets exhibited unique problems. The bicycling data was digitized from the side and was a highly constrained motion making it difficult to tell which motion was simulated. The platform diving data had a very large field of view that introduced noise into the data set. The human performance of the vault was taken from an amateur, but the simulated vault was modeled after an Olympic performance, so the landing style was quite different.

Another evaluation technique compares biomechanical data to simulation data. This method of comparison is less subjective than the video or motion capture comparisons. I have compared simulated and recorded data for the handspring vault and the results are presented in table 9 and I plan to complete additional comparisons using data available in the literature.

When simulation data is compared to biomechanical data a number of problems may arise. Simulations are models of systems with much less complexity than the human body and comparison

Tasks	Time
Improvements to dynamic simulation	1 month
Development of balance control system	2 months
Development of aerial control system	1 month
Development of leaping control system	2 months
Development of landing control system	2 months
Evaluation of controllers and motion	2 months
Writing thesis	6 months

Table 12: Time table for completion of proposed research. Total time of one and a half years to completion.

of data could be inaccurate if the data is affected by unmodeled phenomena. Because the dynamic properties of the simulated models are different from the properties of particular subjects presented in biomechanical literature, comparisons of data dependent on the dynamic properties of the model may be inconclusive.

6 Conclusion

One day synthetic actors may be used by animators in the production of movies or as virtual tour guides to show visitors around online museums. Many problems will have to be addressed before this futuristic view of synthetic actors becomes reality, including the development of techniques for generating realistic and natural-looking motion.

The contribution of this thesis will be the development of robust control systems for dynamically simulated, synthetic actors. If the synthetic actor's motion is to be compelling, each of the four proposed control systems will have to produce natural-looking motion. I will build control systems that achieve this by incorporating the appropriate biologically motivated control strategies. I will also develop control systems that are robust and have large state-space capture regions, allowing transitions between behaviors without manually tuning control parameters. I will also develop evaluation methods for determining whether or not the controllers generate motion resembling that of a human performing the same motion. Even though the research addresses the development of only a small subset of the library of behaviors required for autonomous synthetic actors, the techniques should simplify the creation of additional behaviors.

I plan to complete this research in the time frame given in table 12. I will first improve the dynamic model and the animation system. I expect to spend the first month developing each control system, and another month testing and expanding the state-space capture region. I will then evaluate how natural the motion appears.

References

[1] W. W. Armstrong, M. Green, and R. Lake. Near-real-time control of human figure models. *IEEE Computer Graphics and Applications*, 7(6):52–61, June 1987.

[2] N. I. Badler, C. B. Phillips, and B. L. Webber. *Simulating Humans: Computer Graphics Animation and Control.* Oxford University Press, New York, 1993. ISBN 0-19-507359-2.

[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. W. Calvert. Goal-directed, dynamic animation of human walking. In J. Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 233–242, July 1989.

[6] B. L. Caster and B. T. Bates. The assesment of mechanical and neuromuscular response strategies during landing. *Medicine and Science in Sports and Exercise*, 27(5):736–744, 1995.

[7] 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.

[8] P. J. Cordo and L. M. Nashner. Properties of postural adjustments associated with rapid arm movements. *Journal of Neurophysiology*, 47:287–302, 1982.

[9] L. S. Crawford and S. S. Sastry. Biological motor control approaches for a planar diver. In *1995 Conference on Decision and Control*, December 1995.

[10] J. F. Cremer and J. A. Stewart. The architecture of newton, a gerneral-purpose dynamics simulator. In *IEEE International Conference on Robotics and Automation*, pages 1806–1811, May 1989.

[11] W. T. Dempster. Space requirements of the seated operator. Technical Report WADC Technical Report 55-159, Wright Air Development Center, July 1955.

[12] 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.

[13] C. Frohlich. Do springboard divers violate angular momentum conservation? *American Journal of Physics*, 47:583–592, 1979.

[14] H. S. Gasser and A. V. Hill. The dynamics of muscular contractions. *Royal Society of London Proceedings*, 96:398–437, 1923.

[15] C. Ghez. *Neuroanatomy*, volume Unknown of *Unknown*, chapter 39, pages 596–607. Unknown, Unknown, unknown edition, Unknown 1990.

[16] M. Girard. Interactive design of 3D computer-animated legged animal motion. *IEEE Computer Graphics and Applications*, 7(6):39–51, June 1987.

[17] R. Grzeszczuk and D. Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. In R. Cook, editor, *Proceedings of SIGGRAPH '95 (Los Angeles, CA, August 6–11, 1995)*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22. ACM SIGGRAPH, ACM Press, August 1995. ISBN 0-201-84776-0.

[18] C. V. Heck, I. E. Hendryson, and C. R. Rowe. Pamphlet for the study of joint motion. Executive Committee of the American Academy of Orthopaedic Surgeons, 1964.

[19] J. K. Hodgins. Simulation of human running. In *IEEE International Conference on Robotics and Automation*, pages 1320–1325, May 1994.

[20] J. K. Hodgins. Biped gait transitions. In *IEEE International Conference on Robotics and Automation*, pages 2092–2097, May 1991.

[21] J. K. Hodgins and M. H. Raibert. Biped gymnastics. *International Journal of Robotics Research*, 9(2):115–132, 1990.

[22] J. K. Hodgins, P. K. Sweeney, and D. G. Lawrence. Generating natural-looking motion for computer animation. In *Proceedings of Graphics Interface '92*, pages 265–272, May 1992.

[23] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In R. Cook, editor, *Proceedings of SIGGRAPH '95 (Los Angeles, CA, August 6–11, 1995)*, Computer Graphics Proceedings, Annual Conference Series, pages 71–78. ACM SIGGRAPH, ACM Press, August 1995. ISBN 0-201-84776-0.

[24] P. M. Isaacs and M. F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 215–224, July 1987.

[25] H. Ko and N. I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics. In *Proceedings of Graphics Interface '93*, pages 9–16, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.

[26] P. Lee, S. Wei, J. Zhao, and N. I. Badler. Strength guided motion. In F. Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 253–262, August 1990.

[27] 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.

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

[29] Z. Liu, S. J. Gortler, and M. F. Cohen. Hierarchical spacetime control. In A. Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[30] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged locomotion. In F. Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 29–38, August 1990.

[31] N. Murthy and S. Keerthi. Optimal control of a somersaulting platform diver. In *IEEE International Conference on Robotics and Automation*, pages 1013–1018, May 1993.

[32] L. M. Nashner. Adapting reflexes controlling the human posture. *Experimental Brain Research*, 26:59–72, 1976.

[33] L. M. Nashner. Analysis of movement control in man using the movable platform. *Advances in Nuerology*, 39:607–619, 1983.

[34] 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.

[35] R. O'Brien. *Ron O'Brien's Diving for Gold.* Leisure Press, Champaign, Illinois, 1992.

[36] H. N. Özgüven and N. Berme. An experimental and analytical study of impact forces during human jumping. *Journal of Biomechanics*, 21(12):1061–1066, 1988.

[37] D. Pai. Programming anthropoid walking: Control and simulation. Technical Report 90-1178, Cornell Computer Science, 1990.

[38] M. G. Pandy, F. E. Zajac, M. G. Hoy, E. L. Topp, S. Tashman, P. J. Stevenson, C. T. Cady, E. Sim, and W. S. Levine. Sub–optimal control of a maximum–height, countermovement jump. In J. L. Stein, editor, *Modeling and Control Issues in Biomechanical Systems, 1988 ASME Winter Annual Meeting in Chicago*, volume 12, pages 27–44. ASME, New York, 1988.

[39] M. G. Pandy, F. E. Zajac, E. Sim, and W. S. Levine. An optimal model for maximum-height human jumping. *Journal of Biomechanics*, 23(12):1185–1198, 1990.

[40] K. Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, 1995.

[41] R. R. Playter. *Passive Dynamics in the Control of Gymnastic Maneuvers.* PhD thesis, Massachusetts Institute of Technology, September 1994.

[42] R. R. Playter and M. H. Raibert. Control of a biped somersault in 3-d. In *IEEE International Conference on Robotics and Automation*, pages 582–589, May 1992.

[43] R. R. Playter and M. H. Raibert. Control of a biped somersault in 3-d. In *International Symposium on Theory of Machines and Mechanisms*, Sept 1992.

[44] M. H. Raibert. *Legged Robots That Balance.* MIT Press, Cambridge, 1986.

[45] 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.

[46] 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.

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

[48] K. Sims. Evolving virtual creatures. In A. Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[49] A. J. Stewart and J. F. Cremer. Beyond keyframing: An algorithmic approach to animation. In *Proceedings of Graphics Interface '92*, pages 273–281, May 1992.

[50] Y. Takei. Techniques used by elite women gymnasts performing the handspring vault at the 1987 pan american games. *Medicine and Science in Sports and Exercise*, 6:29–55, 1990.

[51] F. Thomas and O. Johnston. *Disney Animation: The Illusion of Life.* Abbeville Press, New York, 1984.

[52] 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.

[53] 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.

[54] M. van de Panne, E. Fiume, and Z. G. Vranesic. A controller for the dyamic walk of a biped across variable terrain. In *Proceedings of the 31st Conference on Decision and Control*, pages 2668–2673, December 1992.

[55] A. J. van Soest, A. L. Schwab, M. F. Bobbert, and G. J. van Ingen Schenau. The influence of the biarticularity of the gastrocnemius muscle on vertical-jumping achievment. *Journal of Biomechanics*, 26(1):1–8, 1993.

[56] J. Wilhelms. Using dynamic analysis for realistic animation of articulated bodies. *IEEE Computer Graphics and Applications*, 7(6):12–27, June 1987.

[57] D. A. Winter, A. E. Patla, and J. S. Frank. Assessment of balance control in humans. *Journal of Biomechanics*, 21(12):1061–1066, 1988.

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

[59] W. L. Wooten and J. K. Hodgins. Simulation of human diving. In *Proceedings of Graphics Interface '95*, pages 1–9, Quebec City, Quebec, Canada, May 1995. Canadian Information Processing Society.

[60] M. R. Yeadon. The simulation of aerial movement-i,ii,iii. *Journal of Biomechanics*, 23(1):59–83, 1990.

[61] M. R. Yeadon, J. Athia, and F. D. Hales. The simulation of aerial movement-iv. *Journal of Biomechanics*, 23(1):85–89, 1990.