Control of Physically-based Simulated Walking

Joseph F. Laszlo
Michiel van de Panne
Eugene Fiume

Department of Computer Science
University of Toronto[1]

Abstract

Walking is a deceptively simple behaviour that is surprisingly difficult to model. Motion capture and other kinematic animation techniques ignore the dynamics and therefore miss modelling important motion characteristics. Physically-based simulations provide a more general model for the motion, but requires detailed computation of the necessary muscle actions to control a walk.

This work describes a technique which adds closed-loop control to open-loop motions. The open loop control is provided by the animator and treats walking much like the repetitive motion of a wind-up toy. We then use *limit cycle control* to automate the calculation of the control adjustments required to maintain balance. The walking direction and speed of the character remain under the animator's control.

We have experimented with a dynamic human model as well as an imaginary robotic bird, controlling both walking and running. Our plans for the future include improving the fluidity of the motion as well as teaching our character how to perform a variety of other motions.

1 Introduction

Walking is a good example of something which seems trivial in real life bit is difficult to model or simulate. We easily forget the time we ourselves have spent as toddlers learning how to walk, run, and maintain balance. Yet, it is not unreasonable to hope to create models of such motions for use in computer graphics.

Two approaches are commonly used for modelling walking motions for computer animation. The first records data from walking motions and replays them using the desired animated models. A good survey of recent techniques for making flexible use of motion capture data can be found in the work of [?, ?, ?, ?]. The advantages of motion capture are obvious: the immediate generation of realistic human motions. However, motion capture does not provide the understanding necessary for creating more general walking motions, especially when conditions are unpredictable, when new motions need to be generated, or when dealing with non-human characters.

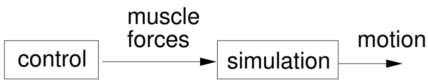

Figure 1: Physically-based animation.

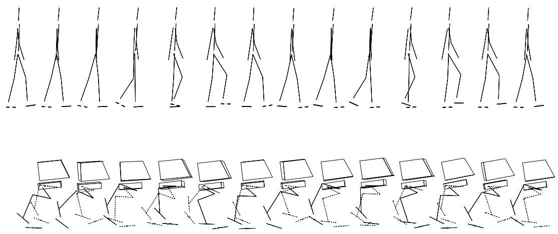

Figure 2: A 3-D dynamic walk for a human model and a running robot.

Physically-based methods provide a second approach for modelling walking motions. These methods apply the basic laws of physics to a mechanical simulation of the human body in order to generate a walking motion. They also require knowledge of the muscle forces which are ultimately responsible for originating the motion, as shown in Figure 1. Unfortunately, determining how muscles should act in order to achieve a natural, stable walking or running gait, is not a simple problem. The work presented here offers one potential solution. It provides a method for

[1] {jflaszlo|van|elf}dgp.utoronto.ca
`http://www.dgp.utoronto.ca`

We gratefully acknowledge the financial support of ITRC (Ontario) and NSERC (Canada).

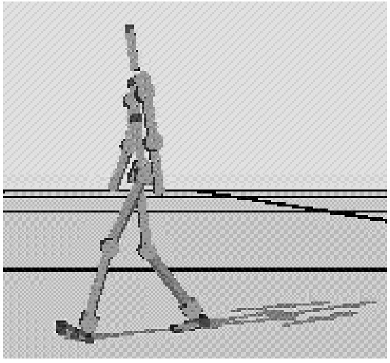

Figure 3: The human model.

calculating the necessary muscle control for a reasonably complete 3D human body model having average physical dimensions and mass. Figure 2 shows example walking and running motions generated using the proposed control technique.

2 Previous Work

The study of walking has long been a topic of fascination for those interested in animation and control. In animation, many parameterized kinematic solutions have been proposed[?, ?, ?, ?]. Walking motions are modelled in these methods by extracting key kinematic relationships from real walking data and using these as a basis for reconstructing new instances of walking. In certain cases, these kinematic models can be mixed in part with dynamic models to lend additional realism[?]. Kinematic methods such as these are flexible in that they can be used to provide an animator with as much (or as little) control over motion details as desired. For example, an animator can directly specify the stride length and hip height for the walking gait if desired.

The use of physically-based methods in animation has been more recent and thus far see little use in commercial animation production. Nonetheless, they do offer certain advantages, primarily because the motion is defined implicitly by the the application of physics rather than explicitly as with kinematic methods. The challenge lies in creating appropriate controlling functions which generate the muscle forces to create the desired type of movement. The methods for performing the simulations themselves are relatively well known and are not discussed here. In our work, we make use of a commercially available simulation package which generates the simulation code directly from a physical description[?].

There are relatively few examples of 3D dynamic simulation of human walking or running. One of these uses a type of underconstrained inverse dynamics together with a hand-tuned controller to generate stable 3D walks[?]. A second good example is that of a human runner[?]. In this case, the controller is a derivative of a control strategy proposed by Raibert[?] for one-legged hopping robots. This controller makes use of an elegant decomposition of the control task into three independant control subtasks. However, the decomposition relies on the fact that the legs are relatively light-weight with respect to the body.

A variety of algorithms proposed for walking robots can also potentially be applied to animation[?, ?, ?, ?, ?]. Typically, however, these control solutions are specific to the particular robot configuration in question, and therefore they do not necessarily generalize well to figures having human dimensions. Because our work is based on using limit cycles to effect control, the work presented in [?, ?] is also of particular interest. It demonstrates the feasibility of building passively-powered walking robots, which derive their energy from walking down shallow inclines.

Another interesting line of research has been the automated synthesis of controllers, typically using search and optimization techniques. A large family of controllers can be defined by using a fixed architecture with a set of to-be-determined parameters. Optimization techniques can make successive improvements to a given controller by making small changes to the current set of chosen parameters and evaluating the outcome of these changes using simulations. The work of [?, ?, ?, ?, ?] is illustrative of this type of approach. While the results indicate a successful ability to generate locomotion control strategies, the optimization techniques are not adept at finding control solutions for finely tuned unstable motions such as human walking.

One last approach uses a 'learning' approach similar to that commonly used for teaching toddlers to walk and athletes to learn gymnastics. A child learning to walk or skate can often make use of a chair to help support themselves. Athletes attempting gymnastic manoevres also make use of harnesses which lend external support to a movement. In [?], a simulated *hand of God* is introduced into dynamic simulations of walking in order to make the learning problem easier. The goal is to have the walking motion gradually reduce its reliance upon the external support, thereby eventually arriving at the control for a delicate task (balanced walking). Experience has indicated, however, that this technique is not always successful and is computationally prohibitive at present. It appears that yet another different approach is required in order to simulate balanced human walking.

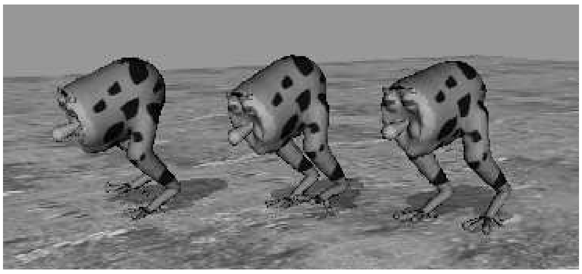

Figure 4: A simulated windup toy: the monster

3 Simulated Wind-up Toys

Our work begins with the basic recognition that walking and running are periodic motions. As such, the basic actions taken are repetitive. We shall use the simulated equivalent of a wind-up toy as the basic control action[?]. This component of the control is *open loop* – the virtual wind-up toy is oblivious to the surrounding world. For example, if it falls over, it will nevertheless keep its legs performing the same cyclical behaviour. The walking monster creature shown in Figure 4 is an example of the type of motion that can result from applying a cyclic control function. It is worthwhile noting that because of its small feet, the monster is in fact very vulnerable to falling over at the slightest perturbation.

The cyclic controllers are implemented using finite state machines (FSMs) and proportional-derivative (PD) controllers. This is a commonly used type of control in both animation and robotics[?, ?, ?]. An example FSM is shown in Figure 5. The operation of the finite state machine is as follows. Each state defines a set of desired joint angles, thereby defining a desired *pose* for the figure. The illustration accompanying each state shows these desired poses for a walking controller.

The muscle forces for the simulated figure are represented as a set of control torques at the joints. A PD controller calculates a torque according to

$$\tau = k_p(\theta_d - \theta) - k_v\dot{\theta},$$

where θ_d is the desired joint angle, θ is the actual joint angle, and k_p and k_v are gain constants which serve to define the strength of a joint. The torque acts as a virtual spring-and-damper pair which pulls the joint towards the desired angle, as shown in Figure 6.

Two types of state transitions are used in the finite state machine. Timed-state transitions pass onto the next state after resting in the current state for a fixed amount of time. States 1 and 4 have sensor-based transitions, which pass onto the next state upon the new stance foot striking the ground. These transitions

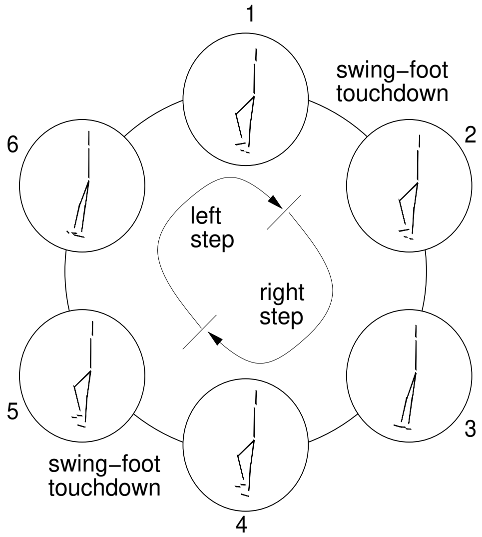

Figure 5: Finite state machine employed for walking.

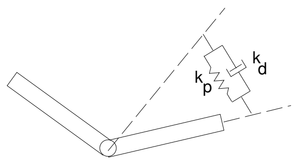

Figure 6: Proportional derivative joint control.

perform the function of keeping the state machine and the motion synchronized.

The particular state machine shown in Figure 5 has two unique poses, those defining states 2 and 3. State 1 is a copy of state 2, and states 4–6 are symmetrical copies of 1–3. While two unique poses per step (four per stride) is sufficient for the purpose of obtaining a reasonable walking gait, the addition of more poses would likely help to achieve a more natural gait.

The selection of the appropriate parameters involves experimentation by the animator. The FSM designed in this way need only generate a reasonable open-loop motion. Figure 7 shows an example walk obtained using the open-loop FSM alone. As will be seen shortly, this open-loop motion will be automatically stabilized through the use of limit cycle control.

4 Limit Cycle Control

The virtual wind-up toys described in the previous section form the point of departure for the limit cycle control method, which is also described in [?, ?].

A typical mechanical wind-up toy can be thought of as applying a periodic control function to achieve a stable periodic motion or *limit cycle*. Plotting the

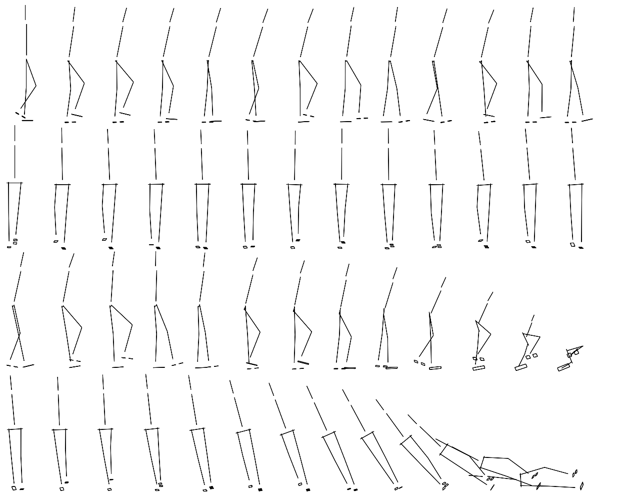

Figure 7: Walking with open-loop control (front/side views.

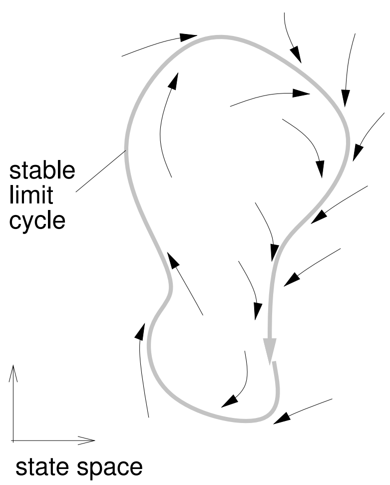

Figure 8: A stable limit cycle.

motion of the two of the state-variables over time, for example two joint angles, produces a *phase diagram* of the limit cycle, an abstraction of which is shown in Figure 8. For mechanical wind-up toys, the passive interaction between the environment and the toy is sufficient to push the toy back onto the limit cycle.

The passive stability of wind-up toys is not sufficient to control walking and running, however. The goal of limit cycle control is to add the necessary control to always return a motion to a fixed periodic cycle. This will be accomplished by making small modifications to the periodic control actions. The idea of actively stabilizing a periodic motion exists in a variety of forms in the control literature[?], but it has not been applied to systems of the type of complexity which animators might require.

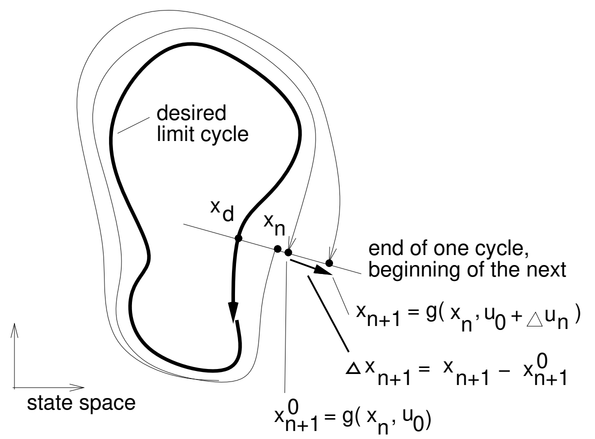

Figure 9: A limit cycle as a discrete dynamical system.

Limit-cycle (LC) control is a means to achieve the kind of stable limit cycle shown in Figure 8 using active control. There are several possible ways to accomplish this. The first is to create a control system which continuously drives the system state back towards the desired limit cycle, much like what is shown in Figure 8. This can be a difficult task, however, as the system dynamics can vary significantly in different regions of the limit cycle. The approach which we choose here is to avoid dealing with such complexities by representing the motion using a well-behaved *discrete* dynamical system.

In general, an action like walking can be thought of as being defined by a set of continuous non-linear differential equations defined by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}),$$

where $\mathbf{x}$ is the system state, $\dot{\mathbf{x}}$ its time derivative, and $\mathbf{u}$ is the control input. The simplified system which we shall deal with is defined by the following discrete dynamical system,

$$\mathbf{x}_{n+1} = g(\mathbf{x}_n, \mathbf{u}_0 + \Delta\mathbf{u}_n),$$

where $\mathbf{x}_n$ represents the initial state of the n^{th} periodic limit cycle, $\mathbf{x}_{n+1}$ is the state on the next cycle, $\mathbf{u}_0$ is the periodic open-loop control, and $\Delta\mathbf{u}_n$ is an applied control perturbation. The purpose of $\Delta\mathbf{u}_n$ will be to drive the sequence $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i$ to a desired value, $\mathbf{x}_d$. Figure 9 depicts an abstraction of the discrete dynamical system.

The goal of the cycle discretization is to obtain resulting dynamics which are better-behaved than the original continuous-time dynamics, and might therefore be subject to linear approximation. We shall shortly present experimental data that shows this is

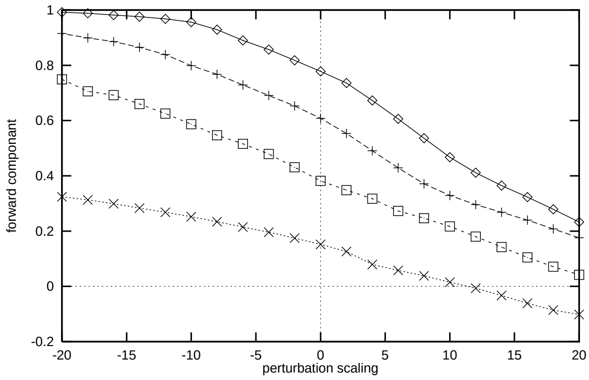

Figure 10: Linearity of perturbation control.

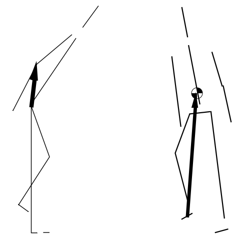

Figure 11: Regulation variables for use in walking. From left to right: (a) Up vector, (b) swing-COM vector.

typically the case. For small control perturbations, $\Delta\mathbf{u}_n$, that will be applied on each cycle, the resulting change in state, $\Delta\mathbf{x}_{n+1}$, has a first order approximation of $\Delta\mathbf{x}_{n+1} = J\Delta\mathbf{u}_n$, where J is a Jacobian relating the change in state after one cycle to the control perturbation. A first order approximation of the change of state makes it possible to calculate the control perturbation required to bring a system back onto a desired limit cycle, namely $\Delta\mathbf{u}_n = J^{-1}\Delta\mathbf{x}_{n+1}$, where the desired $\Delta\mathbf{x}_{n+1}$ is calculated as $\Delta\mathbf{x}_{n+1} = \mathbf{x}_d - \mathbf{x}^0_{n+1}$, $\mathbf{x}^0_{n+1}$ being the state achieved when the nominal control, u_0, is applied.

How well does this first order approximation hold? Figure 10 provides experimental evidence that a linear model is a sufficient for modelling the effect of control perturbations over a complete cycle of motion, despite the occurrence of discontinuous events such as footfalls during the cycle itself.

The y-axis on this graph is a variable we would like to be able to control, namely some aspect of the system state. In this particular example, it is a measure of the forward pitch of the body of a simulated human model performing a walking movement, as measured at the end of a single step. The x-axis gives the magnitude of an applied control perturbation, in this case an alteration of the hip pitch angle during a particular part of the stride. The graph thus tells us what type of forward pitch our simulated human body will have after having taken one step, using different variations of applied control. Each line shows an example of this relationship for a different step. It is evident in this figure that we can alter the forward pitch of the body as we desire and that a linear approximation is a reasonable model for the effect of a control perturbation.

The effects of control perturbations depend on the initial state, $\mathbf{x}_n$. This is clear from Figure 10, which shows that the relationship between control perturbations and changes in state varies for each step. What this means in practice is that for each cycle of the motion, we need to recompute the Jacobian J, which defines the relationship between applied control perturbations and the resulting changes in state after one cycle.

4.1 Working with a Reduced State

The limit cycle control method as described thus far is difficult to implement for several reasons. One of these is the practical problem of calculating the full Jacobian. Another is that of choosing a suitable set of control perturbations, $\Delta\mathbf{u}_n$. As a result, we choose to work with a dynamical system of reduced dimension, dealing only with a small number of *regulation variables* and an equivalently small number of control perturbations. An implicit assumption will be that if the regulation variables are controlled to follow a limit cycle, so will the motion as a whole. It should be noted that *reduced-order models* have also been explored in the control literature[?, ?].

We now present the specific choices of regulation variables which we have chosen to use for the walking simulations. Figure 11 shows two possible choices for sets of regulations variables, each based upon the definition of a particular type of vector. Both of these choices work, although they provide the resulting motions with different characteristics.

The simplest of the two is the *up-vector*, so we begin by explaining this choice. The up-vector is a fixed vector of unit length, defined in the coordinate frame of the pelvis, which can be used to measure the forward lean and sideways tilt of the pelvis. The regulation variables are the forward and lateral components of the up-vector, i.e., the projection of the up-vector onto a horizontal ground plane. Controlling the values of these two scalar variables using LC control is sufficient to yield a balanced, fully three-dimensional human walking motion.

Other choices of regulation variable are also possible. The swing-center-of-mass (swing-COM) defines

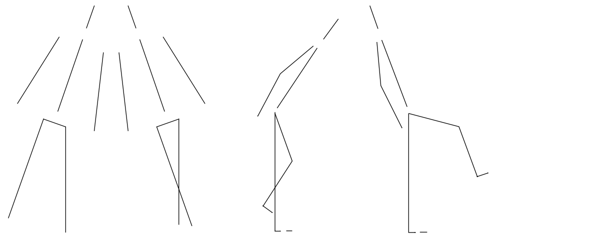

Figure 12: Control perturbations used for walking.

a vector from the swing foot to the center-of-mass, and thus measures where the center-of-mass will lie with respect to the future point of support. Once again, the two scalar components of the projection of the vector onto the ground plane form the set of regulation variables. In the presentation of the results, we primarily demonstrate the use of the up vector.

4.2 Choosing Control Perturbations

For two regulation variables, as is the case for our walking example, we shall require two appropriately-chosen control perturbations in order to yield a well-formed Jacobian J. The control perturbations we work with for human walking are twofold. First, one can use changes to the stance-hip pitch and roll, effected over a particular portion of the walk cycle. Figure 12 illustrates the effects of these perturbations in an exaggerated fashion. The two figures on the left demonstrate stance-hip roll, while those on the right demonstrate stance-hip pitch. Second, the use of alterations to swing-hip pitch and roll can equally well be used as a suitable pair of control perturbations. In this case, the resulting control can be thought of as a type of foot-placement strategy. As with the choice of regulation variables, the choice of control perturbations is not unique, although they must be chosen to span the space of desired changes to the regulation variables.

4.3 Implementation

There remain several decisions to take in implementing limiting cycle control. The first of these is the choice of a target or 'desired' state for the discrete dynamical system. As indicated in Figure 9, the final nature of the limit cycle is defined by the target state to be achieved at the end of the each cycle. LC control is not successful for all choices of target values. Target values should be similar to the values that can be observed during the first few steps of an open-loop motion. This ensures that the generated limit cycle is close the unstable open-loop limit cycle, thereby lim-

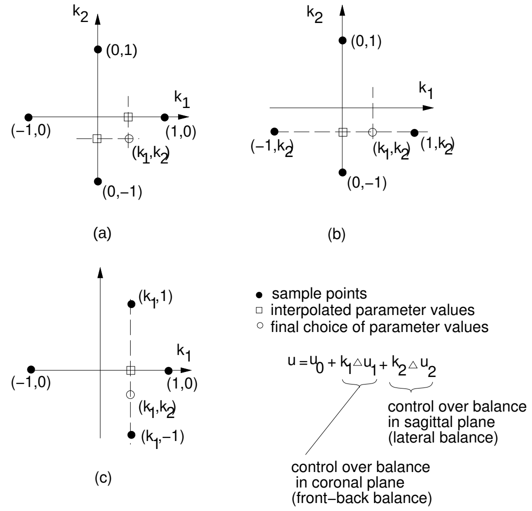

Figure 13: Sampling the effects of control perturbations: (a) superposition; (b) lateral then frontal; (c) frontal then lateral.

iting the LC control to having to perform relatively small control corrections.

The power of LC control lies in being able to predict the change in values of the regulation variables with respect to the applied control perturbations. Using a linear model for this allows us to easily predict the required perturbation. For the walking model, we choose to test several linear interpolation methods, instead of directly evaluating a Jacobian. Figure 13 shows three sampling strategies that can be used to construct the desired linear model.

All of the suggested sampling methods require the use of four sample points, and therefore four simulations of the current step, each with a unique set of control perturbations. The data from these four simulations are then used to estimate the necessary control perturbations to achieve the desired target state. A fifth and final simulation is then applied with the correct control values, thus finallly advancing the current walking simulation by one step.

The chosen pair of control perturbations for human walking, namely alterations to the desired hip pitch and roll angles, were designed to allow for more-or-less independent control of each of the regulation variables. The hip roll is effectively used to provide balance in the coronal plane, while the hip pitch provides balance in the sagittal plane. In effect, this corresponds to only requiring the use of the diagonal elements of the Jacobian.

Figure 14 illustrates the linear interpolation scheme, which is applied twice, once for coronal balance (front-back), and once for sagittal balance (side-

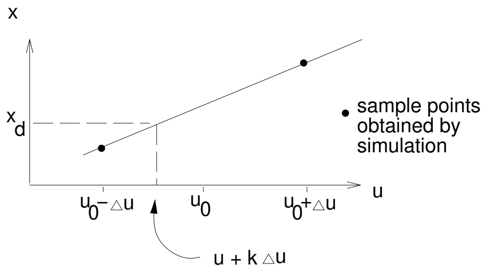

Figure 14: Interpolating for a desired control perturbation.

to-side). The different sampling techniques shown in Figure 13 represent variations in the order in which these two interpolations are carried out. A first approach is to carry out the two interpolations independently of each other. Another choice is to use the known results of the lateral balance in performing the front-back balance sampling and calculation. Lastly, we could choose to perform front-back balance followed by lateral balance. All three of the schemes generate suitable control of walking gaits. As might be expected, the superposition model performs worse in some circumstances because the control perturbations do not truly yield completely independent control of each of the regulation variables.

5 Model Descriptions

The control of a 3D dynamic human walking simulation is our primary example for illustrating the use of limit cycle control. It is typical of motions for which the open-loop control actions are relatively easy to construct and can thus benefit immediately from LC control in order to "close the loop." It is also a challenging type of figure to animate because of its complexity and instability, as well as the lack of well-known general solutions for this problem.

5.1 The Human Model

The human model has a mass and inertia distribution comparable to those of a real human[?]. The model has 13 joints, as shown in Figure 15. The hip joints have three rotational DOF, while the ankle joints have two rotational DOF. The equations of motion are calculated and integrated using a commercially-available simulation package [?]. The ground is modelled using a penalty method. Stiff springs and dampers exert forces on a set of four points on the feet whenever they penetrate the ground. The ground model thus uses no artificial constraints to hold the foot in place, which

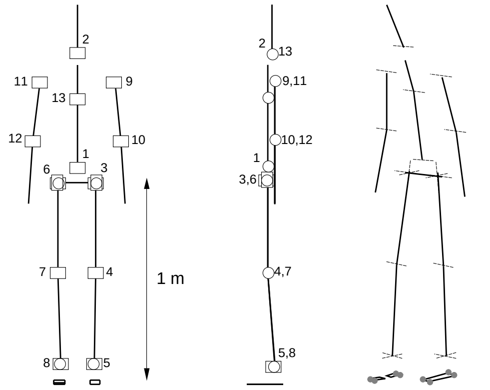

Figure 15: Construction of human model.

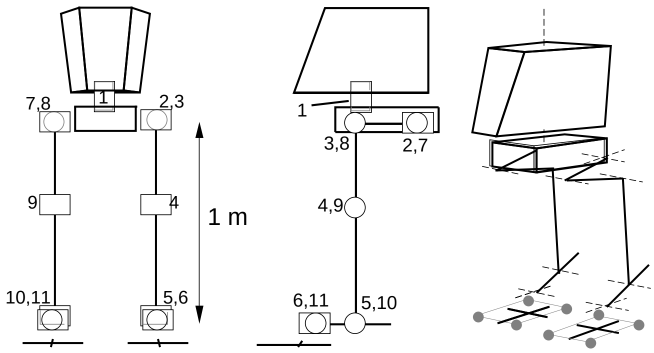

Figure 16: Construction of the robot model.

have in the past been used to simplify the simulation and control of human motion.

5.2 The Robot Model

As a second test case, we consider the robot shown in Figure 16. This figure has 11 joints and 15 rotational DOF. The lateral base of support is much wider than that of the human model, yielding a very different style of walk.

5.3 Torso Servo

In order to obtain improved performance of the limit-cycle control technique applied to the human model, a torso servo is given the job of keeping the body upright by controlling the bend at the waist. Without this torso servo, the walking motion exhibits excessive upper body motion. The torso servo consists of a PD controller applied to the one degree-of-freedom waist joint. The applied torque at this joint serves to force the torso to always remain upright with respect to the world coordinate frame. Note that the torso servoing does not prevent the biped from fall-

Figure 17: A falling motion illustrating the torso servo.

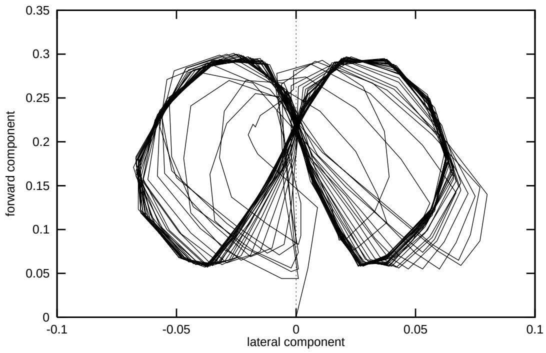

Figure 18: Regulation variable limit cycle.

ing because the legs must still ultimately provide for balanced support, as illustrated in Figure 17.

6 Results

We begin describing the results with an illustration of a typical limit cycle which is achieved, as shown in Figure 18. These figures are the real analog of the earlier abstraction shown in Figure 9. For walking, the limit cycle consists of two roughly symmetric halves. This occurs because a single cycle consists of a left step and a right step, each forming half of the complete limit cycle for a stride. The desired values for the regulation variables lie at the center of this diagram, at $x = 0, y = 0.25$. Perturbation control is applied on each step, thus forcing the limit cycle towards the desired point twice on each cycle. The startup phase of the motion is also evident from the figure, with the regulation variables (the forward and lateral components of the up vector) eventually being driven onto the desired limit cycle. The typical walk corresponding to this limit cycle is shown in Figure 2, which has the torso-servo enabled.

While LC control is a general method of adding balance to a walk, it does not by itself ensure a straight walk. Figure 19 illustrates different paths taken for different target values for the regulation variable controlling the desired forward lean. The figure is a top view of the walking motion, showing only the position of the hips, which are drawn using a line wider than the hips themselves in order to make the orientation of the body clearly visible during the walk. The real

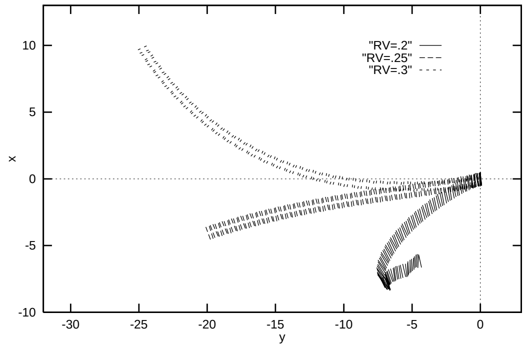

Figure 19: Dynamic walks using the up-vector.

scale is given by the axes, indicated in metres.

6.1 Speed Control

Control over the speed and direction of a walking gait is of obvious necessity to an animator. Because the stabilization (i.e., balance) of the walk is automated, it is a relatively simple matter to provide the necessary hooks to control these parameters, as well as other possible stylistic variations.

The speed of the walking gait could be controlled in several ways. We have chosen one of the simplest of these, namely noticing that the forward speed is a function of the target values chosen for the regulation values. As we choose limit cycles for which the pelvis and torso lean forward, the speed of the gait increases in a smooth and approximately linear fashion.

6.2 Direction Control

By changing the open-loop motion to include twisting motions in the hips, controllable turning motions can be achieved. Figure 20 shows the basic modifications necessary to the open-loop state machine in order to achieve controlled turning. The turning motion works best when torso servoing is applied, in that turns of tighter radius can be performed. Figure 21 shows turns obtained using scaled versions of the twisting motion and stabilized using LC control.

Once the turning radius can be controlled, it can be used to produce a path-following algorithm. Figure 22 shows a dynamic walk following a given trajectory. The algorithm makes use of a *target point* on the trajectory and chooses a turning rate proportional to the current error in direction. When the target point has almost been reached, its position is updated to be further along the desired path.

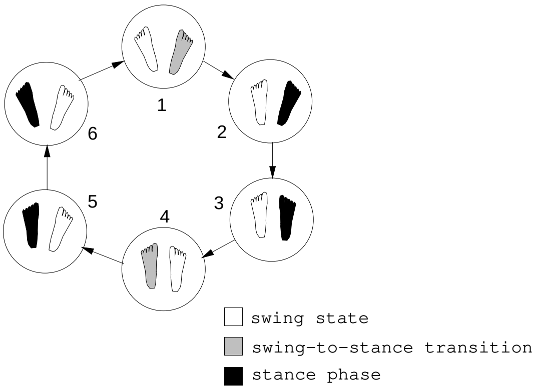

Figure 20: Modifications for turning during a walk.

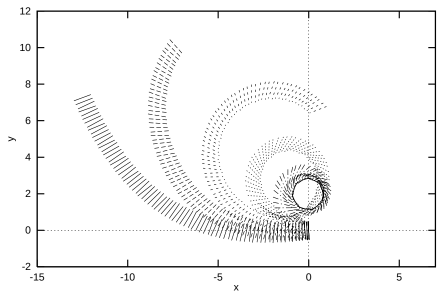

Figure 21: Turning motions for a dynamic walk.

6.3 Other Variations

Other walking styles can be implemented by tailoring the open-loop control as desired. Transitions between different motions can in many cases be performed by simple linear interpolation of the underlying open-loop control over a period of two or three steps. The LC control mechanism always remains the same. Note that because a dynamic simulation is always being used, ground constraints and other physics constraints are always fulfilled, something that

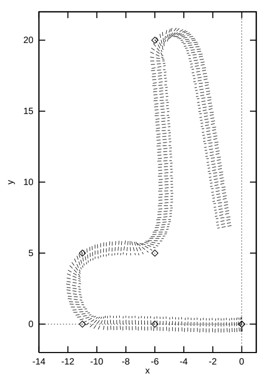

Figure 22: Path following behaviour.

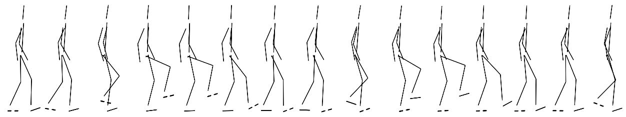

Figure 23: Stylistic variation on a walk.

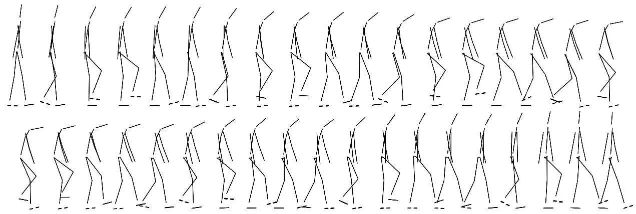

Figure 24: A ducking motion.

is not necessarily the case with direct kinematic interpolation of motion data, especially for large variations.

Figure 23 illustrates a stylistic variation on a walk with the knees being bent and lifted high on each step. Figure 24 shows a ducking motion obtained by transitioning into and out of a bent walk. We have also simulated walking motions into a strong wind, for which the automatic feedback provided by the LC control visibly alters the motion to lean into the wind. Speed and turning control can also be used to generate walks which side-step, walk in place, and walk backwards.

7 Conclusions

Tools for intelligent character animation are only in their infancy. Producing the tools of the future will involve solving some difficult problems which have their roots in artificial intelligence, biomechanics, robotics, and control. As with so many natural phenomena, simple motor skills such as walking can be much more challenging than they might appear. However, one can now easily imagine the time when we will have realistic simulated human characters capable of at least basic movement skills. Much interesting work remains to be done, but at the current rate of progress there is certainly cause for optimism.

Two simplifications are at the heart of the control strategies described in this work. The first is that a linear predictive model can be applied to motions such as walking and running, despite their non-linearity. When the effect of a control perturbation is measured over a complete step, a linear model is in many cases reasonable. The second simplification is the use of a reduced state vector. It is often sufficient to work with the subset of the state vector which reflects changes in the motion of the majority of the body mass. The limit cycle control method resulting from these two simplifications is a general method of adding balancing actions to periodic motions. To the best of our knowledge, this is one of the first control methods for fully dynamic, unconstrained 3D simulation of human walking.

Much future work remains to be done. A first improvement would be an alternate means of determining the Jacobian for each step. At present, this involves carrying out four trial simulations to construct the predictive model (the Jacobian). A fifth simulation then simulates the final motion with the chosen control parameters. In an ideal system, the four predictive simulations could be eliminated. We believe that the Jacobian could itself be predicted using a model, which would result in true closed-loop control.

The walk controllers contain many parameters which need to be carefully chosen and tuned by hand. For a general-purpose animation tool, methods will be required to tune these parameters automatically. An interesting possibility is to make use of motion-capture data in this process. Lastly, realistic human models are needed to accompany the realistic models of their motion.

The search for general learning strategies for motion control needs to continue. How can we teach a simulated squirrel to jump gracefully from branch to branch? How do gazelles and mountain goats plan their motion? Character animation is an ideal context for exploring these kinds of questions.

References

[1] R. Boulic, N. M. Thalmann, and D. Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6:344–358, 1990.

[2] A. Bruderlin and T. W. Calvert. Goal-directed animation of human walking. *Proceedings of ACM SIGGRAPH*, 23(4):233–242, 1989.

[3] A. Bruderlin and T. W. Calvert. Interactive animation of personalized human locomotion. *Proceedings of Graphics Interface*, pages 17–23, 1993.

[4] A. Bruderlin and L. Williams. Motion signal processing. *Proceedings of Siggaph '95, ACM Computer Graphics*, pages 97–104, 1995.

[5] J. Auslander et al. Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics*, October 1995.

[6] J. K. Hodgins et al. Animating human athletics. *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pages 71–78, 1995.

[7] Vukobratovic et al. *Biped Locomotion: Dynamics, Stability, Control and Applications*. Springer Verlag, 1990.

[8] J. Furusho and M. Maubuchi. A theoretically motivated reduced order model for the control of dynamic biped locomotion. *Journal of Dynamic Systems, Measurement, and Control*, 109:155–163, 1987.

[9] J. Furusho and A. Sano. Sensor-based control of a nine-link robot. *The International Journal of Robotics Research*, 9(2):83–98, 1990.

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

[11] C. L. Golliday and H. Hemami. An approach to analyzing biped locomotion dynamics and designing robot locomotion controls. *IEEE Transactions on Automatic Control*, AC-22(6):963–972, 1977.

[12] R. Grzeszczuk and D. Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pages 63–70, 1995.

[13] H. M. Hmam and D. A. Lawrence. Robustness analysis of nonlinear biped control laws via singular perturbation theory. *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 2656–2661, 1992.

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

[15] R. Katoh and M. Mori. Control method of biped locomotion giving asymptotic stability of trajectory. *Automatica*, 20:405–414, 1984.

[16] H. Ko and N. I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics. *Proceedings of Graphics Interface '93*, pages 9–16, 1993.

[17] J. Laszlo. *Controlling Bipedal Locomotion for Computer Animation.* M.Sc. Thesis, Department of Computer Science, University of Toronto, 1996.

[18] J. Laszlo, M. van de Panne, and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. *Proceedings of SIGGRAPH '96*, pages 155–162, 1996.

[19] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[20] T. McGeer. Passive walking with knees. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1640–1645, 1990.

[21] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged locomotion. *Proceedings of SIGGRAPH '90*, pages 29–38, 1990.

[22] H. Miura and I. Shimoyama. Dynamic walk of a biped. *International Journal of Robotics Research*, pages 60–74, Summer 1984.

[23] J. T. Ngo and J. Marks. Spacetime constraints revisited. *Proceedings of SIGGRAPH '93*, pages 343–350, 1993.

[24] M. H. Raibert. *Legged Robots that Balance.* MIT Press, 1986.

[25] F. Saito, T. Fukuda, and F. Arai. Swing and locomotion control for a two-link brachiation robot. *IEEE Control Systems*, 14(1):5–12, 1994.

[26] K. Sims. Evolving virtual creatures. *Proceedings of SIGGRAPH '94, ACM Computer Graphics*, pages 15–22, 1994.

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

[28] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pages 91–96, 1995.

[29] M. van de Panne and E. Fiume. Sensor-actuator networks. *Proceedings of SIGGRAPH '93*, pages 335–342, 1993.

[30] M. van de Panne, R. Kim, and E. Fiume. Virtual wind-up toys for animation. *Proceedings of Graphics Interface '94*, pages 208–215, 1994.

[31] M. van de Panne and A. Lamouret. Guided optimization for balanced locomotion. *1995 Eurographics Workshop on Animation and Simulation*, 1995.

[32] A. Witkin and Z. Popovi'c. Motion warping. *Proceedings of SIGGRAPH '95*, pages 105–107, 1995.

[33] W. L. Wooten and J. K. Hodgins. Simulation of human diving. *Proceedings of Graphics Interface '95*, pages 1–9, 1995.