

Goal-Directed, Dynamic Animation of Human Walking

Armin Bruderlin
Thomas W. Calvert

School of Computing Science
Simon Fraser University
Burnaby, British Columbia, Canada V5A 1S6

ABSTRACT

This paper presents a hybrid approach to the animation of human locomotion which combines goal-directed and dynamic motion control. Knowledge about a locomotion cycle is incorporated into a hierarchical control process. The desired locomotion is conveniently specified at the top level as a task (e.g. walk at speed v), which is then decomposed by application of the concepts of *step symmetry* and *state-phase-timings*. As a result of this decomposition, the forces and torques that drive the dynamic model of the legs are determined by numerical approximation techniques. Rather than relying on a general dynamic model, the equations of motion of the legs are tailored to locomotion and analytically constrained to allow for only a specific range of movements. The dynamics of the legs produce a generic, natural locomotion pattern which is visually upgraded by some kinematic "cosmetics" derived from such principles as *virtual leg* and *determinants of gait*. A system has been implemented based on these principles and has shown that when a few parameters, such as velocity, step length and step frequency are specified, a wide variety of human walks can be generated in almost real-time.

CR Categories and Subject Descriptors: I.3.7: [Computer Graphics]: Three-Dimensional Graphics and Realism - Animation; G.1.7: [Numerical Analysis]: Ordinary Differential Equations.

Keywords: Animation, goal-directed animation, human figure animation, dynamics, kinematics, inverse kinematics.

1. INTRODUCTION

The specification and control of motion in human figure animation has always been a challenge, but two recent trends promise to relieve the tedious work of the animator. One involves high-level, goal-directed control, which reduces the amount of detail necessary to define a motion; the second involves applying dynamic analysis to the motion control process, leading to more realism in movements.

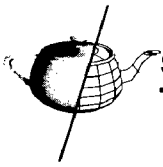
In traditional keyframing [13], the quality of a motion is usually directly proportional to the number of key positions specified. If the desired movements are complicated, the animator, rather than the system, does motion control. It has been recognized that if the

excessive amount of specification for character animation is to be reduced, higher level motion control is necessary [2, 7, 17]. At the lowest level, all movements are expressed by joint rotations over time, but these joint rotations must be coordinated within a limb, between limbs and are subject to the interaction of the whole figure with its environment. By incorporating knowledge or rules about these inter-relationships, tasks like grasping or jumping can be automated and presented to the user as parameterized goals. In such a goal-directed system, the global coordination of a motion is done by the computer. However, movements are still executed kinematically at the lowest level, and the impact of physical laws such as gravity or collisions on the motion process are ignored. To achieve realistic and natural movements, dynamic analysis must be applied as a motion control technique. By simulating the real world, objects move as they should move, according to physical laws. The drawback is that the animator has to specify motion in terms of forces and torques; this is neither intuitive nor easy, and it is complicated by the computationally expensive character of this approach. In the past, simulation of human figures concentrated on simple, elementary movements not involving coordination between several limbs (e.g. raising an arm or dropping an arm under the influence of gravity). By combining a goal-directed higher level control with dynamic simulation of motion, a system can be developed for economic and realistic animation of many co-ordinated human movements.

This paper introduces such a method for the purpose of animating human locomotion. To this end, we have implemented the KLA (Keyframe-Less Animation of Walking) system to animate walking. Dynamic simulation provides the low-level control; a dynamic walking model, inspired by research in robotics [12] and biomechanics [4] produces a generic walking pattern from different sets of analytically constrained equations of motion which are applied as appropriate to the current state of the locomotion. Kinematic algorithms are applied to calculate all the body angles from the motion of the dynamic model. The dynamics, in turn, are regulated by a higher level control; the proper forces and torques which generate a desired locomotion are calculated as a result of a stepwise decomposition of a few walking parameters specified by the user. Thus, motion is specified conveniently and realistic animations are obtained based on the dynamic equations of motion without explicit specification of forces and torques.

The goal-oriented approach used in this paper builds on the work of Zeltzer [16], who developed a task-oriented system to animate human locomotion; although tasks like walking and jumping were implemented, the calculation of the joint angles were done kinematically and based on interpolation methods and clinical data. Thus it was not possible to easily realize variations in locomotion by changing step length or speed. The general

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to publish, requires a fee and/or specific permission.



approach to dynamic analysis which we have adopted is based on the work of Wilhelms [15], Armstrong [1] and others [3, 11]. Perhaps the most comprehensive approach is that of Wilhelms, who produced *Virya*, a dynamic system for the animation of human figures which also allows for kinematic and hybrid kinematic-dynamic motion specification. In the dynamic mode, however, forces and torques have to be input in order to achieve a motion. Badler *et al.* [2] have been developing kinematic techniques to animate human figures. They have proposed a higher level of control, where goals such as reaching for a certain position can be defined and the joint angles are found using inverse kinematic algorithms. They also investigated dynamic and kinematic animation of specific tasks (e.g. movement in a space vehicle). A system to animate legged figures was developed by Girard and Maciejewski [8]; dynamic control was applied to the body as a whole, and the legs were specified kinematically. The problem of constraining a foot to be on the ground during its support phase was formulated as an inverse kinematics problem and solved by means of a pseudoinverse jacobian.

2. THE KLAU SYSTEM

2.1 Overview

Legged locomotion describes an intricate activity where body translation results from rotational movements in the lower limbs; problems such as coordination between the legs, proper timing of the individual leg motions and balancing of the upper body have to be addressed. It is clear that humans and other animals, however, are able to walk effortlessly without conscious thought. This is because they are inherently goal-directed [17]. Rather than thinking in terms of forces and torques, humans walk with certain goals like speed or step length in mind — thus, a hierarchical control scheme is well suited to animate human locomotion. Figure 1 gives a structural outline of the KLAU system. The animator specifies a desired walk with up to 3 fundamental *locomotion parameters* which largely determine the pattern of motion and gait: forward velocity (v), step length (sl) and step frequency (sf). A major concern in constructing a goal-directed system has been the degree to which a task should be parameterized. The animator should have access to a simple, yet flexible set of movement commands that can generate a variety of instances of a task. In KLAU, therefore, in addition to the 3 locomotion parameters, up to 28 *locomotion attributes* [5] may also be specified which individualize the locomotion. The default values of these attributes may be modified by the animator. Examples are: lateral distance between the feet, toe clearance during swing, and maximum rotation and list of the pelvis. After parameter specification, the system computes and outputs the body angles as functions of time — these drive the animation of a human figure.

2.2 Levels of Control

Knowledge about a locomotion cycle occurs at three levels: the conceptual abstraction (high-level), the gait refinements (middle-level) and the physical abstraction (low-level). The conceptual level contains a few gait-specific rules or laws. These are utilized to transform the locomotion parameters into *step constraints* which are fed to the low-level control to "guide" the dynamic simulation of the legs. The middle-level control is responsible for the coordination of the motion and functionally operates much like a finite state machine as suggested by Zeltzer [16]. For instance, upon a "heel-strike" event, the state "single support" is changed to "double support". Hierarchically, the middle-level control manifests a stepwise reduction in the number of degrees of freedom along with a decrease in the levels of coordination (e.g. the single support state of a walk consists of a stance and a swing

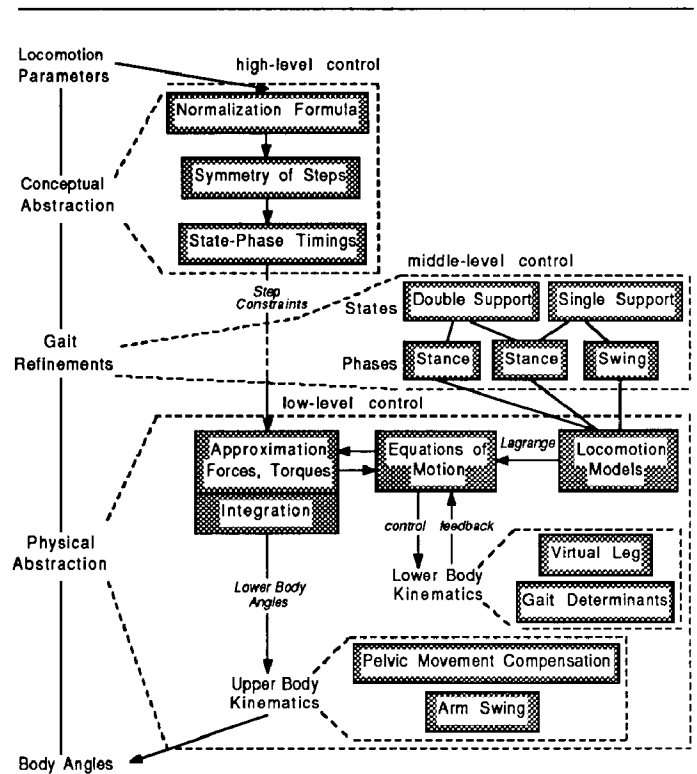


Figure 1: Schematic diagram of the control hierarchy in KLAU.

phase). The bottom level is represented by sets of specialized equations of motion; in fact, the phases are divided into subphases in which the equations are further constrained analytically. The low-level control uses the *step constraints* (which essentially are the durations and final leg angles for the stance and swing phases of a locomotion step) as conditions for a special kind of a boundary value problem. That is, the equations of motion are solved by approximating the forces and torques until the constraints are satisfied. For example, the simulation of the swing leg for the current step is repeated by varying the joint-torques until it swings forward in the exact time required and heel-strike occurs with the desired hip and knee angles. In practice, this process converges quite quickly.

2.3 Dynamic Model

A principal objective is to keep the dynamics simple, otherwise the internal calculation of the forces and torques becomes infeasible. As shown in figure 2, the swing leg is represented by two segments. The stance leg supports the upper body and is implemented as a length-changing telescopic segment which simulates knee flexion in the early part and plantar flexion of the ankle in the latter part of the stance phase (as explained later, for animation, a full leg with knee, ankle and metatarsal joints are superimposed). This approach is chosen since a linear force along the leg axis is much easier to control than additional torques at the leg joints.

The segment masses are assumed to be constant and the segments to be symmetrical. The latter implies that the principal axes of inertia are identical to the anatomical axes of rotation, and therefore the products of inertia are zero. Thus, the distribution of mass is solely defined by the moments of inertia which are calculated as described in appendix B. This simplification is justified for dynamic analysis in computer animation, since it has

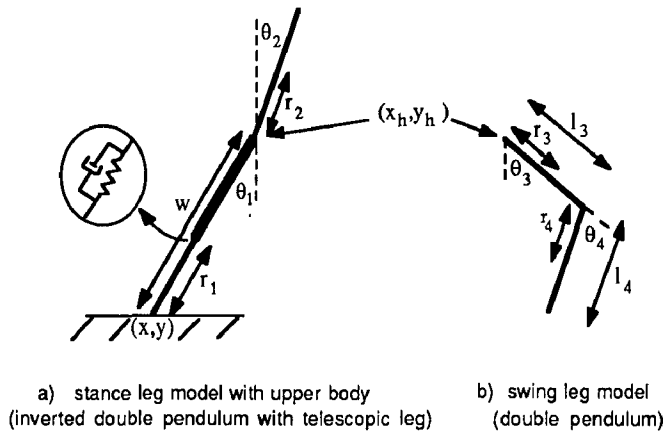


Figure 2: Dynamic models for the different phases in locomotion. θ_3 is negative in this configuration, all other angles are positive; see appendix B for anthropometric values.

no significant effect on the motion. The equations of motion are derived by the method of Lagrange [14] as shown in appendix A.1 and A.2. The ground constraint for the stance phase is implemented as an analytical constraint; assuming that the "dynamic foot" does not move during stance, the two degrees of freedom, x, y , are removed. In this way, there are a total of only 5 degrees of freedom ($w, \theta_1, \theta_2, \theta_3, \theta_4$) and consequently 5 second order, nonlinear equations of motion. The equations are solved by an A-stable, standard numerical integration method [9] which has produced numerically stable results for this problem.

2.4 Control Principles

The execution of the different components in KLAWE is based on four assumptions or principles:

1. The control hierarchy as illustrated in figure 1 is applied to each step of a walking sequence where a step is defined as the double plus the single support state (see also figure 3). While the high-level concepts are executed before the impending step, the low-level motion control takes place during the step. Thus, KLAWE is able to adapt to changes in the locomotion parameters from step to step, i.e. accelerations and decelerations in the motion are possible with a granularity of one step.

2. Lower body dynamics and kinematics must be executed simultaneously. The dynamic simulation produces the generic locomotion pattern which is visually upgraded by kinematic measures. As explained in section 4, a human leg is superimposed onto the telescopic stance leg according to the *virtual leg principle*, and *gait determinants* like pelvic rotation or list get injected into the one-hip dynamic model. In a sense, the equations of motion guide the lower body kinematics, but the kinematic computations may, in turn, affect the dynamics. For instance, the simulation of the swing leg, where the foot is assumed to be locked, has to take into account the updated position of the heel as a result of the kinematic foot rotation, in order to achieve heel-strike properly at the end of a step. Similarly, the kinematic pelvic rotation can actually lower the hip during the swing phase, which might "force" the dynamic leg to increase its hip torque to avoid stubbing its toe. Though considerable kinematic "cosmetics" are applied, the dynamics are the very heart of the control for they guarantee natural looking rotational movements of the legs.

3. It is assumed that the upper body follows or depends on the lower body movements. Whereas the dynamic model accounts only for a natural forward and backward motion of the upper body (θ_2), the angles of the arms as well as the rotations in the shoulder and spine which compensate for pelvic movements are expressed as functions of the corresponding angles in the lower body. The arms, for example, swing forward with the opposite legs. Thus, these angles are calculated after the dynamic simulation.

4. The last assumption concerns the dynamic model discussed above: the simulations for the stance and swing phases are separated which greatly simplifies the control as well as the numerical integration process. The rationale is that the stance leg model constitutes the major propulsive element in bipedal locomotion. It supports the body and influences the swing leg by its hip motion. On the other hand, the swing leg has little or no effect on the stance leg and the upper body. Of course, this is not completely true in real human walking, but it can be justified by the fact that the mass of the leg is small compared to the total mass of the body (approx. 16%). Hence, the swing leg does not change the inertia of the body significantly unless the motion during swing happens very suddenly, which is hardly the case for a moderate walk. Therefore, for each step, the simulation of the stance phase is executed first followed by the swing phase dynamics which incorporate the position of the hip (x_h, y_h) from the stance phase.

3. HIGH-LEVEL CONCEPTS

This section gives a discussion of the high-level control module whose task is to transform the 3 locomotion parameters v, sl and sf into the step constraints for the low-level control. At least one locomotion parameter (e.g. desired velocity) has to be input by the animator. If all 3 of the parameters are not specified, the system completes the parameters using a *normalization formula* (the parameters are also checked at this point to ensure that they are within anatomical limits defined by *locomotion attributes*, e.g. $sf_{max} = 182 \text{ steps/min}$). Once the locomotion parameters are accepted and specified, the step frequency and the *state-phase-timings* are applied to determine the durations of the stance and swing phases, and the step length is used with the *symmetry of steps* concept to compute the final conditions for each phase of the current step.

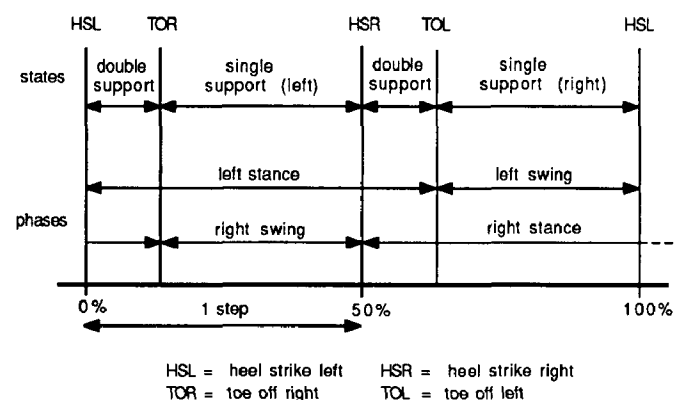
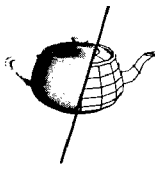


Figure 3: Locomotion cycle for bipedal walking.

Since rhythmic locomotion is just a series of recurring movements with the natural period of one stride (the locomotion



cycle), it is sufficient to compute movements for one cycle. The human walking cycle has been thoroughly studied — see Inman [10], for example. For bipedal walking, a locomotion cycle consists of two steps. As long as a symmetric gait is assumed where the left and right leg perform the same movements, just shifted in time, the principal unit of locomotion can be reduced to one step (see figure 3). Walking is possible at a wide variety of combinations of sl and sf ($v = sl \cdot sf$). However, a person, when asked to walk at a particular velocity, is most likely to choose parameters which minimize energy expenditure. This observation is expressed in the experimentally derived equations [10], called *normalizing formulae*, which show a linear relationship between sl and sf , where sl and *body_height* are measured in *m*, and sf in *steps/min* :

$$\frac{sl}{sf \cdot \text{body_height}} = 0.004 \quad (1)$$

$$\Leftrightarrow sf^2 = \frac{v}{0.004 \cdot \text{body_height}}, \text{ because of } sl = \frac{v}{sf}.$$

The *body_height* normalizes the equation. It indirectly represents the length of the legs, which has an effect on the preferred step length. Based on equation (1), the locomotion parameters are now checked and supplemented if at least one is specified. For instance, if a velocity is defined, a "natural" step length and step frequency are calculated; in the case where a velocity and a step length are specified a more angular motion might result if the step length deviates significantly from the "natural" one (see also figure 10).

The step frequency is the input to the *state-phase-timings* calculation. A walking cycle consists of two steps (figure 3), each step having one double support state (ds) where both feet are on the ground and one single support state where one foot is off the ground. With respect to timing of the individual legs the following holds, assuming t to denote a duration:

$$t_{step} = t_{stance} - t_{ds} \text{ and } t_{step} = t_{swing} + t_{ds}. \quad (2)$$

Experimental data [10] suggest that in human walking there is an approximately linear relationship between the step frequency and the duration of the double support state as a percentage of a cycle, i.e. the duration of the double support state decreases with increasing step frequency. As t_{ds} vanishes, walking becomes running. Based on results from different experiments, t_{ds} can be described in terms of sf and t_{cycle} :

$$t_{ds} = (-0.16 \cdot sf + 29.08) \cdot t_{cycle} / 100.$$

Since sf is known as one of the locomotion parameters, and because of

$$t_{cycle} = 2 \cdot t_{step} = \frac{2}{sf},$$

t_{ds} can be determined, and consequently the values for t_{stance} and t_{swing} are obtained from equation (2). It should be noted that the length of the stance phase is greater than a step, i.e. $t_{stance} = t_{step} + t_{ds}$ (see figure 3). To simplify implementation and to satisfy the step-oriented control principle (assumption 1.), the stance phase of a leg is only simulated for the duration of a step and at heel strike, when the stance phase of the leading leg starts, the continuing stance phase of the hind leg is completed kinematically.

Step symmetry is based on a compass gait (figure 4), and means simply that at heel strike, provided that both legs are of the same

length, the angles of the legs measured from the vertical are identical. That is $\theta_1 = \theta_3$ at times t_1 , t_2 and t_3 . Further, θ_1 and θ_3 depend only on the step length sl . Most importantly, this remains true when the body is accelerating or decelerating, indicated in figure 4 by the increased step length at time t_3 (i.e. the body accelerated from t_2 to t_3).

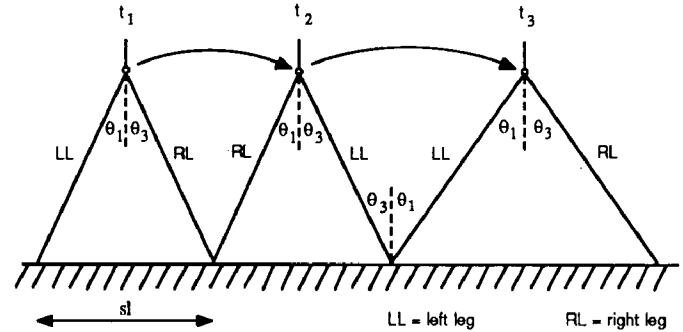


Figure 4: Symmetry of compass gait for different step lengths.

This principle is now adapted to the model in figure 5 to determine w , θ_1 and θ_3 at the end of a step (heel strike) utilizing the current step length sl . Although the actual step configuration at heel strike is no longer symmetric because of the introduction of a kinematic foot for the swing leg, the basic idea can still be applied. We just imagine the symmetric step situation when the foot is flat on the ground some time after impact (illustrated by the dashed line) and calculate "back in time". For this purpose, the step length sl is measured between *ankle*₁ and *ankle*₂. The effect of the foot at heel strike is that the absolute value of θ_3 is smaller than it would be without a foot (also, $\theta_3 < \theta_1$). In addition, the foot raises the position of the hip at impact, which has to be compensated by lengthening the telescopic stance leg beyond its initial length, i.e. $w > l_1$. The origin for the simulation of the stance leg stays fixed at *ankle*₁, which is at a distance l_9 above the ground.

For the following calculations, it is assumed that the ground is at zero height and θ_5 at impact is specified as one of the *locomotion attributes*. Given l_1, l_8, l_9, l_{11} and $\cos(\omega_8) = \frac{l_8}{l_{11}}$, the application of the cosine law yields

$$r^2 = l_1^2 + l_{11}^2 - 2 l_1 l_{11} \cos(\theta_5 + \omega_8), \text{ and}$$

$$\omega_3 = \cos^{-1} \left(\frac{l_1^2 + r^2 - l_{11}^2}{2 l_1 r} \right).$$

Since

$$\text{ankle}_1 = (x_a, y_a) = (x_{nh'} + l_8, l_9)$$

where $x_{nh'}$ is x_{nh} from previous step, and

$$\text{heel} = (x_{nh}, y_{nh}) = (x_a + sl - l_8, 0)$$

$$\text{hip} = (x_h, y_h) = (x_a + \frac{sl}{2}, \sqrt{r^2 - (x_{nh} - x_h)^2}),$$

it follows that

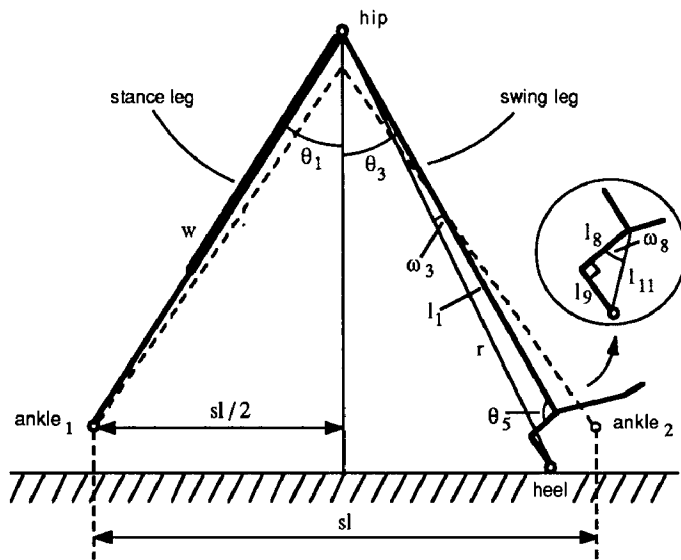


Figure 5: Dynamic model at heel strike: the swing leg is extended, a foot has been added kinematically, the upper body is ignored; θ_3 is assumed to be negative, all other angles are positive.

$$\begin{aligned}
 w &= \sqrt{(y_h - y_a)^2 + \left(\frac{sl}{2}\right)^2} \\
 \theta_1 &= \sin^{-1}\left(\frac{sl}{2w}\right) \\
 \theta_3 &= -\omega_3 - \sin^{-1}\left(\frac{x_m - x_h}{r}\right). \quad (3)
 \end{aligned}$$

These are the 3 final conditions for a locomotion step; together with the durations for the leg phases they form the *step constraints* that govern the execution of the low-level control and determine the motion by leading to the internal calculation of the applied forces and torques.

4. LOW-LEVEL CONTROL

The low-level control generates the actual motions by application of a mixture of dynamic and kinematic algorithms. The essence of this control is explained in this section and a full discussion can be found elsewhere [5]. The stance and swing phases of a locomotion cycle are examined separately. Although the dynamics are subject to the *step constraints* which force the execution of a particular step length and step frequency, they need to be guided to produce desired motions during a phase by applying rules about walking directly at this low level. For instance, regardless of the stiffness of the leg spring, the hip of the stance leg model must never be allowed to drop below its minimal value at heel strike in order to maintain some kind of a sinusoidal motion pattern; similarly, the swinging leg does not just swing forward to reach the final hip angle at impact, but the motion has to be timed appropriately throughout the swing to make it look real. For this purpose, additional restrictions are imposed in two ways: each phase is divided into a number of *subphases*, where the equations of motion are "fine-tuned" to further suit bipedal walking, and the trajectories of the applied forces and torques are expressed as specific functions of time. The dynamics account

not only for natural movements within a phase, but they also provide continuity across phases. For example, heel strike which occurs between the swing and the stance phase is treated as a collision, whereby the new initial conditions for the stance phase are calculated by the conservation of linear and angular momentum. Although the dynamically simulated motion appears natural in terms of timing and continuity, it does not show human characteristics. To humanize the movements, kinematic algorithms based on the principles of the *virtual leg* and the *determinants of gait* are integrated into the dynamic motion control process as shown below.

4.1 Calculation of Forces and Torques

The problem of finding the forces and torques to meet the desired constraints for each stance and swing phase is formally expressed as follows, assuming the matrix representation of the equations of motion as set out in appendix A:

$$\begin{aligned}
 A \ddot{q} &= B(q, \dot{q}, F_q), \quad \text{subject to} \\
 q(t_0) &= \alpha, \quad q(t_e) = \beta \quad \text{and} \quad t_0 \leq t \leq t_e.
 \end{aligned}$$

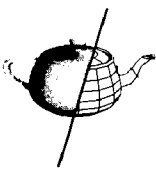
The generalized forces F_q are now the independent variables and the objective is to find the proper forces or torques such that, given the initial conditions $q(t_0)$, the system reaches the final conditions $q(t_e)$ in exactly time t_e . This is a classical *root-finding problem* where the roots F_q are approximated by numerical techniques.

As an example, consider the stance phase: the initial conditions come from the end of the preceding swing phase and the collision laws at heel strike. The final conditions are the hip angle θ_1 and the length of the leg w at time t_{step} as calculated from equations (3) and (2). The equations of motion are now iteratively integrated over the duration t_{step} by modifying the leg torque F_{θ_1} and the leg axis force F_w on each iteration until the final conditions are met.

The approximation of F_q is performed in two stages. First, the Bisection method computes a reasonable approximation which is then refined by the Secant method. This technique was employed because the Secant method converges fast, but needs a good first approximation [6]. A solution to F_q is usually obtained within a few iterations (between 6 and 10). Once the rhythmic phase of a locomotion sequence is reached, i.e. the forward velocity of the body as a whole is fairly constant, the algorithm converges even faster since the F_q profiles from one step are carried over to initialize the next.

4.2 Stance Phase

During stance the upper body is balanced by the torque F_{θ_2} the magnitude of which is determined by a simple spring and damping model. The leg torque F_{θ_1} is calculated by the approximation procedure described above to satisfy the hip angle θ_1 at the end of the step. Since experiments on human subjects utilizing electromyography and force plates [10] have shown that a significant torque at the hip occurs only just after heel strike and lasts for about 20% of the cycle time, F_{θ_1} is applied as a step input torque which is turned off at $0.4 \cdot t_{step}$. The leg axis force F_w is approximated such that the telescopic leg w is extended to its desired value at the end of the step. The force profile of F_w is expressed as a spring and damper model of the form



$$F_w = k_w (l_1 + pa - w) - v_w \dot{w}, \quad (4)$$

where k_w and v_w are spring and damping constants, respectively; l_1 is the unloaded length of the leg, w is the current leg length and \dot{w} the velocity along the leg axis. A *position actuator*, pa (initially zero), actively controls the magnitude of the force; pa must be chosen such that the hip of the stance model prescribes a vertical sinusoidal curve typical in human walking [10]. The hip is therefore constantly monitored during stance: if it drops too low, pa is increased and if the leg extends too much, the value of pa is reduced. However, this method might cause the telescopic leg to become too long (particularly at a low walking speed where the leg does not shorten much after heel strike); if $w > l_1$ (which simulates a plantar flexion of the ankle) occurs too early in the stance phase an unnatural leg motion results. To prevent this, the stance leg is locked as soon as it reaches a critical length which is either l_1 in the early part or the desired length as calculated by equation (3) towards the end of the stance phase. For this purpose the dynamics of the stance phase are divided into *subphases*: one in which the equations of motion apply as defined in appendix A.1 where the leg is represented by a telescopic inverted pendulum, and another phase in which the leg is defined as a rigid inverted pendulum, expressed by the following modifications to the matrix form of the equations:

$$a_{1,1} = 1, \quad b_1 = a_{1,3} = a_{3,1} = 0 \quad \text{and} \quad \dot{w} = 0.$$

The subphases are coordinated by the middle-level finite state machine which basically switches between the different sets of equations of motion upon signaling of such events "leg too long". If the leg is locked because $w > l_1$, the lock is removed as soon as the event $\theta_1 > 0$ (leg passes through vertical) occurs, and an increase in pa extends the leg to the desired length at the end of the step.

The *virtual leg* principle describes the procedure which is applied to superimpose a human leg onto the telescopic stance leg (figure 6) at each time step during the simulation. Unfortunately, the number of possible configurations is infinite, i.e. a unique solution does not exist for the orientation of the segments from the hip (H) to the tip of the toe (T). This is a typical *inverse kinematics* problem, where the proximal (H) and distal (T) endpoints are given (H is known from the simulation, T is fixed during stance) and the task is to find the angles of the kinematic chain spanned between these endpoints. At least two of the four angles ($\theta_3, \dots, \theta_6$) must be known to fully specify a particular configuration. The information to calculate all leg angles is supplied by rules about the motion of the foot during the stance phase. These rules express a *normal* period just after heel-strike where the foot rotates around the heel until it is flat on the ground. The *normal* period ends when the ankle angle θ_5 reaches a limiting value, at which time the heel begins to come off the ground (*heel-off* period), i.e. the mid-foot rotates around M during this period with radius l_{12} . As soon as θ_6 reaches a limit, the *meta-off* period is entered where the whole foot rotates around T until the end of the stance phase (toe-off). Once θ_5 and θ_6 are known the hip and knee angles (θ_3, θ_4) are determined by simple trigonometric calculations.

The *determinants of gait* [10] mainly describe the movements of the pelvis during locomotion and play a major role in bestowing human appeal to the motion. Pelvic rotation (transverse plane), pelvic list (coronal plane) and a lateral displacement of the body — the body weaves slightly from side to side following the weight-bearing leg — have been implemented. By introducing a pelvis, the kinematics of the determinants basically add a second hip to the locomotion model and must therefore be applied after

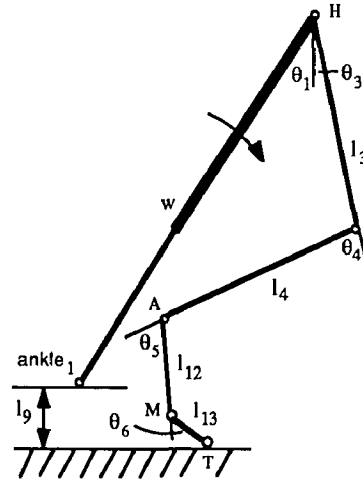


Figure 6: Superposition of a leg over the dynamic stance leg model. The proportions of the foot (l_{12}, l_{13}) are exaggerated.

the the stance phase simulation but before the simulation of the swing leg which uses the new position of this hip. The rotation (list) of the pelvis is a maximum (minimum) at heel-strike and a minimum (maximum) at mid-step, whereas the lateral displacement is a maximum shortly after toe-off and a minimum at heel-strike. These boundary values are specified as *locomotion attributes* and linear interpolation is applied to obtain all the intermediate angles. A linear interpolation is justified since the absolute displacements produced by the determinants are rather small.

4.3 Swing Phase

As with the stance phase, the simulation of the swing phase is broken up into subphases in order to achieve a natural movement of the leg. Three subphases are distinguished — they are illustrated in figure 7. During *swing1* (from t_0 to t_1) the ankle is constrained to move along the curve P until the toe is exactly under the knee. At the same time, the hip angle reaches a maximum, which is the desired value for heel strike as calculated by the symmetry of step concept. The *swing2* subphase lasts from t_1 to t_2 and is characterized by a rapid extension of the knee joint while the hip angle stays fairly constant. After the knee is fully extended at time t_2 , a small moment at the hip forces the heel onto the ground during *swing3* to bring about heel-strike (at t_3). Whereas in the stance phase the subphases are triggered by events, the duration of each swing subphase is known *a priori*. Based on experimental data [4, 10], the end for *swing1* occurs at about 50 % of the time for the swing, which means that after half the swing time, the thigh of the swing leg has reached its desired orientation for heel strike. The end of *swing2*, marked by the straightening of the leg, takes place about 85 % into the swing, and the end of *swing3* coincides with the end for the swing phase. Because the time for the swing t_{swing} of the current step is known from the state-phase timing concept, the durations for the subphases can be readily determined. A foot is added kinematically to the model whereby the ankle and metatarsal angles are interpolated between their values at toe-off and heel-strike (the former are known from the kinematic *meta-off* period, the latter are specified as *locomotion attributes*).

During the *swing1* phase the hip torque F_{θ_3} is expressed as a

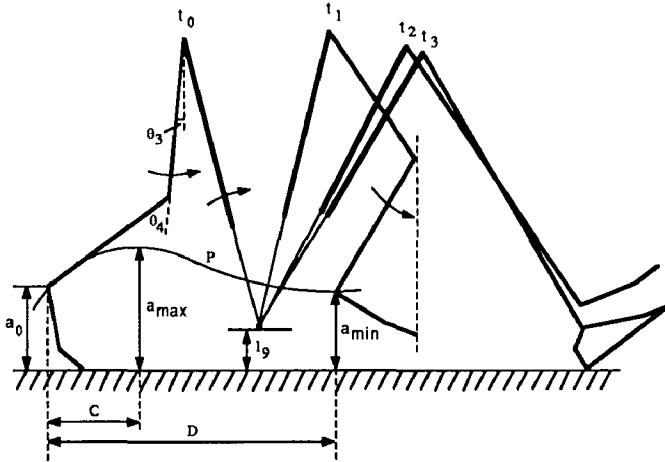


Figure 7: Illustration of swing phase; the kinematic foot proportions are exaggerated, the upper body is ignored.

decaying exponential function and numerically approximated such that the hip angle θ_3 reaches the desired value at time t_1 . At the same time, the knee joint is locked for the dynamic simulation; at each time increment, the knee angle θ_4 is updated by forcing the ankle onto the curve P and the new value for θ_4 is fed back to the simulation process. This measure makes an explicit calculation of the knee torque unnecessary, and simplifies the integration procedure. The equations of motion are reduced as follows, assuming the matrix form of appendix A.2:

$$a_{2,2} = 1, \quad b_2 = a_{1,2} = a_{2,1} = 0 \quad \text{and} \quad \dot{\theta}_4 = 0. \quad (5)$$

The curve P is represented by a 4th order polynomial $y(x) = \sum_{i=0}^4 A_i x^{4-i}$ where x and y are the coordinates of the ankle. The coefficients A_i are computed from the following 5 conditions:

$$\begin{aligned} y(0) &= a_0 \\ y(C) &= a_{\max} \\ \dot{y}(C) &= 0 \\ y(D) &= a_{\min} \\ \dot{y}(D) &= 0. \end{aligned}$$

The value for a_0 is computed at the end of the previous *meta-off* period. At a distance C , the ankle reaches the maximum height a_{\max} during swing. Based on observations [4, 10], C amounts to about 30% of the value for D . The latter, as well as a_{\min} , are derived from the hip position (known from stance phase), hip angle (known from step symmetry) and the fact that the toe is directly under the knee at t_1 . The value for a_{\max} has been chosen somewhat arbitrarily, but in such a way that the faster the walk, the smaller a_{\max} .

In the *swing2* phase the magnitude of the hip torque F_{θ_3} is calculated as a spring and damping model to hold the thigh in place, whereas the knee torque F_{θ_4} , whose profile is a decaying exponential function, is numerically approximated as to extend the leg at exactly time t_2 . However, the foot might intersect with the ground while the knee is extending. This could result from small spring and damping constants, by an accentuated pelvic list

or a short step length. A recovery algorithm increases the hip torque temporarily just enough for the foot to clear the ground. This is achieved by repeating the simulation of the *swing2* phase with an incrementally increased position actuator similar to equation 4 until the foot clears the ground. Finally, in the *swing3* phase the leg is extended with the knee joint locked, which involves analytically constraining the equations of motion as in equation (5). A hip torque is applied, chosen by the numerical approximation process such that heel-strike occurs at time t_3 which is exactly t_{swing} .

5. RESULTS

It has been demonstrated that the KLAU system can produce a wide variety of quite realistic human walks upon specification of only a few parameters. Besides a desired body height, the mass of the body and a simulation time, the animator needs to specify at least one of the locomotion parameters to obtain a walking sequence. Since the algorithm is step-oriented, changes in the locomotion parameters over time can be accounted for with a granularity of one step. This allows for accelerations and decelerations in the locomotion, and even the extreme cases of starting and stopping are possible as shown in figure 8.

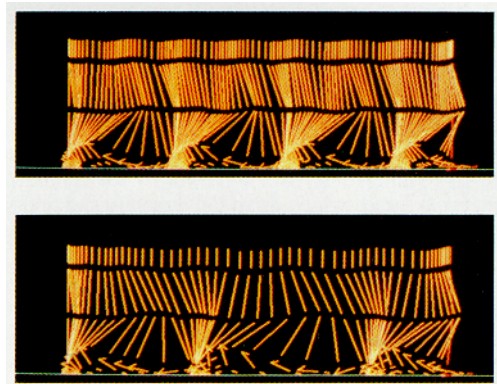


Figure 8: Motion of one leg for two complete walking sequences at different speeds; top: 2 km/h, 4 cycles, bottom: 5 km/h, 3 cycles.

The *locomotion attributes* are used to individualize a walk. For instance, by changing the amount of pelvic list as shown in figure 9, significant variations of a walk are generated even for the same locomotion parameters. Figure 10 illustrates the effects of different combinations of step length and step frequency for the same the walking velocity.

The system calculates a total of 56 angles for the 37 joints of the body model — 24 of these joints model the vertebrae in the spine — plus a position vector in space for each time step. Currently, these computations are performed not quite in real-time; for example, it took 37.3 sec of CPU time on a SUN3-50 computer with floating point processor to compute the motion of the 12 sec walk in figure 10, top. In practice, the *swing2* phase has been most expensive because of its recovery algorithm; although the duration of *swing2* is only about 35% of the time for the swing, the simulation usually takes more time than for the *swing1* and *swing3* phases combined. It is clear that real-time animation can be achieved by using a faster processor and by customizing the numerical integration routines.

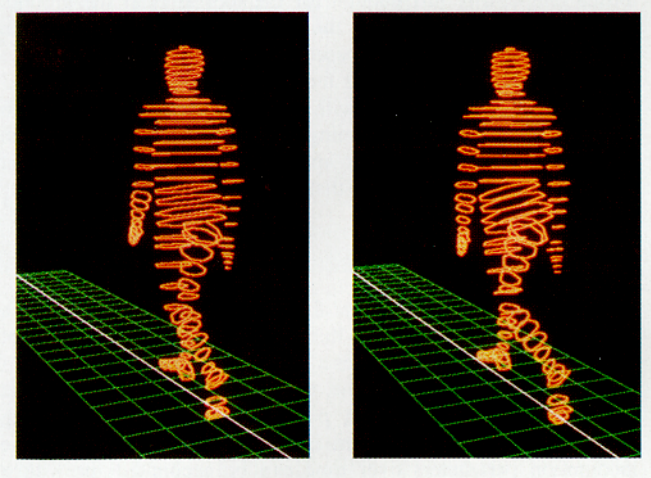
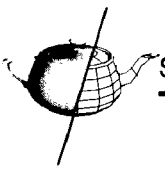


Figure 9: Illustration of pelvic list at toe-off.
Left: natural pelvic list, right: accentuated pelvic list.

6. CONCLUSIONS

A hybrid approach between goal-directed and dynamic motion control has been introduced to animate human walking. This meets two of the most important goals in human character animation — convenient specification and realistic motion production. The success of the approach demonstrates that it is feasible to build a knowledge base to guide the animation of a complex, co-ordinated human movement. It has been shown that the torques and forces which drive the dynamic simulation can be found automatically by iteratively approximating constraints defined by the knowledge base. We believe that the algorithm can be directly extended to other bipedal gaits and locomotion with more than two legs. Whereas running, for example, would require a modification of the dynamic model as well as the high-level concepts to account for the flight phase during which both legs are off the ground, additional legs are merely a coordination problem at the "state" level.

Since in practical terms, the dynamics could be regarded as an interpolation method between the key frames defined by some high level concepts (in this case, step symmetry and state-phase timings) we are currently investigating the possibility of building an entire animation system for articulated figures, where different classes of motions (locomotion, grasping, standing up, turning, etc.) are implemented as tasks which the animator can activate by a few motion parameters. The usefulness of such a system would greatly depend on the choice of parameters assigned to each task.

7. ACKNOWLEDGEMENTS

This work was supported in part by grants from the Social Sciences and Humanities Research Council of Canada and the Natural Sciences and Engineering Research Council of Canada. The authors acknowledge the contributions of Scott Selbie and Severin Gaudet.

APPENDICES

A. EQUATIONS OF MOTION

The equations of motion for the stance and swing phase below are based on figure 2 and were derived using the Lagrange method. The Lagrange equations for a system with n degrees of freedom can be written as

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_r} \right) - \frac{\partial L}{\partial q_r} = F_{q_r} \quad \text{where } r = 1, 2, \dots, n;$$

$$\begin{aligned} L &= \text{Lagrangian} = T - V, \\ T &= \text{Kinetic Energy}, \\ V &= \text{Potential Energy}, \\ q_r &= \text{Generalized Coordinate} \\ F_{q_r} &= \text{Generalized Force}. \end{aligned}$$

A.1 Stance Phase

For the stance phase, $n = 3$ and $q_r = [w, \theta_1, \theta_2]^T$. F_{θ_1} and F_{θ_2} represent torques, F_w a force along the stance leg axis. Since the foot remains fixed on the ground during stance, i.e. $\ddot{x} = \ddot{y} = \ddot{z} = \dot{x} = \dot{y} = \dot{z} = 0$, three equations of motion result. The meaning and values of anthropometric data are given in appendix B. In matrix form, $A \ddot{q} = B(q, \dot{q})$ where

$$\begin{aligned} A &= \text{inertia matrix } (n \times n \text{ matrix for } n \text{ degrees of freedom}) \\ q &= \text{solution vector} \\ B &= \text{vector of transient terms (including the } F_{\theta_r} \text{)}, \end{aligned}$$

the equations are written as

$$\begin{bmatrix} m_2 & 0 & -m_2 r_2 \sin(\theta_2 - \theta_1) \\ 0 & I_1 + m_1 r_1^2 + m_2 w^2 & m_2 r_2 w \cos(\theta_2 - \theta_1) \\ a_{1,3} & a_{2,3} & I_2 + m_2 r_2^2 \end{bmatrix} \begin{bmatrix} \ddot{w} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} =$$

$$\begin{bmatrix} F_w + m_2 w \dot{\theta}_1^2 + m_2 r_2 \dot{\theta}_2^2 \cos(\theta_2 - \theta_1) - m_2 g \cos \theta_1 \\ F_{\theta_1} - 2 m_2 w \dot{w} \dot{\theta}_1 + (m_1 r_1 + m_2 w) g \sin \theta_1 \\ \quad + m_2 r_2 w \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) \\ F_{\theta_2} + m_2 g r_2 \sin \theta_2 - 2 m_2 r_2 \dot{w} \dot{\theta}_1 \cos(\theta_2 - \theta_1) \\ \quad - m_2 r_2 w \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) \end{bmatrix}$$

A.2 Swing Phase

For the swing phase, $n = 2$ and $q_r = [\theta_3, \theta_4]^T$. F_{θ_3} and F_{θ_4} represent torques. The translational motion of the hip of the swing leg does not impose additional degrees of freedom to the system, since this motion is determined by the stance phase model as follows:

$$\begin{aligned} x_h &= x + w \sin \theta_1 & y_h &= y + w \cos \theta_1 \\ \dot{x}_h &= \dot{w} \sin \theta_1 + w \dot{\theta}_1 \cos \theta_1 & \dot{y}_h &= \dot{w} \cos \theta_1 - w \dot{\theta}_1 \sin \theta_1 \\ \ddot{x}_h &= \ddot{w} \sin \theta_1 + 2 \dot{w} \dot{\theta}_1 \cos \theta_1 & \ddot{y}_h &= \ddot{w} \cos \theta_1 - 2 \dot{w} \dot{\theta}_1 \sin \theta_1 \\ & \quad + w (\ddot{\theta}_1 \cos \theta_1 - \dot{\theta}_1^2 \sin \theta_1) & & \quad - w (\ddot{\theta}_1 \sin \theta_1 + \dot{\theta}_1^2 \cos \theta_1), \end{aligned}$$

where $\ddot{x} = \ddot{y} = \ddot{z} = \dot{x} = \dot{y} = \dot{z} = 0$.

In matrix form, the two equations of motion are

$$\begin{bmatrix} I_3 + m_3 r_3^2 + m_4 l_3^2 & I_4 + m_4 r_4^2 + m_4 l_3 r_4 \cos \theta_4 \\ + I_4 + m_4 r_4^2 + 2 m_4 l_3 r_4 \cos \theta_4 & \\ a_{1,2} & I_4 + m_4 r_4^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} = \begin{bmatrix} F_{\theta_3} + (m_3 r_3 + m_4 l_3) (\ddot{x}_h \cos \theta_3 - \ddot{y}_h \sin \theta_3) \\ + m_4 r_4 (\ddot{x}_h \cos (\theta_3 + \theta_4) - \ddot{y}_h \sin (\theta_3 + \theta_4)) \\ + m_4 l_3 r_4 \ddot{\theta}_4 (2 \ddot{\theta}_3 + \ddot{\theta}_4) \sin \theta_4 \\ - m_3 g r_3 \sin \theta_3 - m_4 g (l_3 \sin \theta_3 + r_4 \sin (\theta_3 + \theta_4)) \\ F_{\theta_4} + m_4 r_4 (\ddot{x}_h \cos (\theta_3 + \theta_4) - \ddot{y}_h \sin (\theta_3 + \theta_4)) \\ - m_4 l_3 r_4 \ddot{\theta}_3^2 \sin \theta_4 - m_4 g r_4 \sin (\theta_3 + \theta_4) \end{bmatrix}$$

$$\begin{aligned} m_3 &= 8 \text{ kg}, \\ l_3^{abs} &= 0.426 m \\ r_3^{abs} &= r_3 \cdot l_3^{abs} = 0.184 m, \\ \gamma_3^{abs} &= \gamma_3 \cdot l_3^{abs} = 0.138 m, \\ I_3 &= m_3 \cdot (l_3^{abs} \cdot \gamma_3^{abs})^2 = 0.0276 \text{ kg m}^2. \end{aligned}$$

B. ANTHROPOMETRIC DATA

Table B-1 lists the *relative* (lengths are fractions of body height, masses are fractions of body mass) anthropometric data of the segments used by the dynamics and lower body kinematics. l_i denote lengths, m_i masses, r_i distances to centers of mass and γ_i radii of gyration. It should be noted, that except for r_1 , which is measured from the distal end, the centers of mass are given from the proximal end of a segment. The radii of gyration are specified with respect to the center of mass:

segment	i	l_i	m_i	r_i	γ_i
pelvis	0	0.10059	-	-	-
leg	1	$l_3 + l_4$	$m_3 + m_4$	0.553	0.326
upper body	2	0.47	0.678	0.5	0.496
thigh	3	0.23669	0.1	0.433	0.323
shank	4	0.24556	0.061	0.606	0.416
mid foot	5	0.0858	-	-	-
toe	6	0.04734	-	-	-
hind foot	8	0.02959	-	-	-
ankle-footbase*	9	0.03846	-	-	-
heel-ankle*	11	0.04853	-	-	-
ankle-1st metatarsal*	12	0.08901	-	-	-
1st metatarsal-toe tip*	13	0.0496	-	-	-

Table B-1: Anthropometric values of lower body segments (the * indicates a distance rather than a segment length).

The *absolute* anthropometric data, including the moments of inertia I_i , are calculated by the system once the values for body height and body mass are specified. As an example, if the total body height is to be 1.8 m and the body mass 80 kg, the following values result for the thigh:

REFERENCES

- William W. Armstrong, Mark Green. The Dynamics of Articulated Rigid Bodies for Purposes of Animation. Graphics Interface '85, Proceedings, 1985, pp. 407-415.
- Norman I. Badler, Kamran H. Manoocherhri, Graham Walters. "Articulated Figure Positioning by Multiple Constraints". *IEEE Computer Graphics and Applications* 7, 6 (June 1987), 28-38.
- Ronen Barzel, Alan H. Barr. A Modeling System Based On Dynamic Constraints. SIGGRAPH '88, Proceedings, August, 1988, pp. 179-188.
- Royce Beckett, Kurng Chang. "An Evaluation of the Kinematics of Gait by Minimum Energy". *J. Biomechanics* 1 (1968), 147-159.
- Armin Bruderlin. Goal-Directed, Dynamic Animation of Bipedal Locomotion. Master Th., School of Computing Science, Simon Fraser University, 1988.
- Richard L. Burden. *Numerical Analysis*. Prindle, Weber & Schmidt, 1985.
- Thomas W. Calvert. The Challenge of Human Figure Animation. Graphics Interface '88, Proceedings, 1988, pp. 203-210.
- Michael Girard, Anthony A. Maciejewski. Computational Modeling for the Computer Animation of Legged Figures. ACM SIGGRAPH '85, Proceedings, July, 1985, pp. 263-270.
- Alan C. Hindmarsh. "LSODE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers". *ACM-SIGNUM Newsletter* 15, 4 (1980), 10-11.
- Verne T. Inman, Henry J. Ralston, Frank Todd. *Human Walking*. Williams & Wilkins, Baltimore, 1981.
- Paul M. Isaacs, Michael F. Cohen. "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics". *Computer Graphics* 21, 4 (July 1987), 215-224.
- Marc H. Raibert. "Legged Robots". *Communications of the ACM* 29, 6 (1986), 499-514.
- David Sturman. Interactive Keyframe Animation of 3-D Articulated Models. Graphics Interface '86, Tutorial on Computer Animation, 1986.
- Dare A. Wells. *Theory and Problems of Lagrangian Dynamics*. McGraw-Hill, New York, 1967.
- Jane Wilhelms. Virya - A Motion Control Editor for Kinematic and Dynamic Animation. Graphics Interface '86, Proceedings, 1986, pp. 141-146.
- David Zeltzer. "Motor Control Techniques for Figure Animation". *IEEE Computer Graphics and Applications* 2, 9 (1982), 53-59.
- David Zeltzer. Knowledge-Based Animation. ACM SIGGRAPH/SIGART, Workshop on Motion, 1983, pp. 187-192.

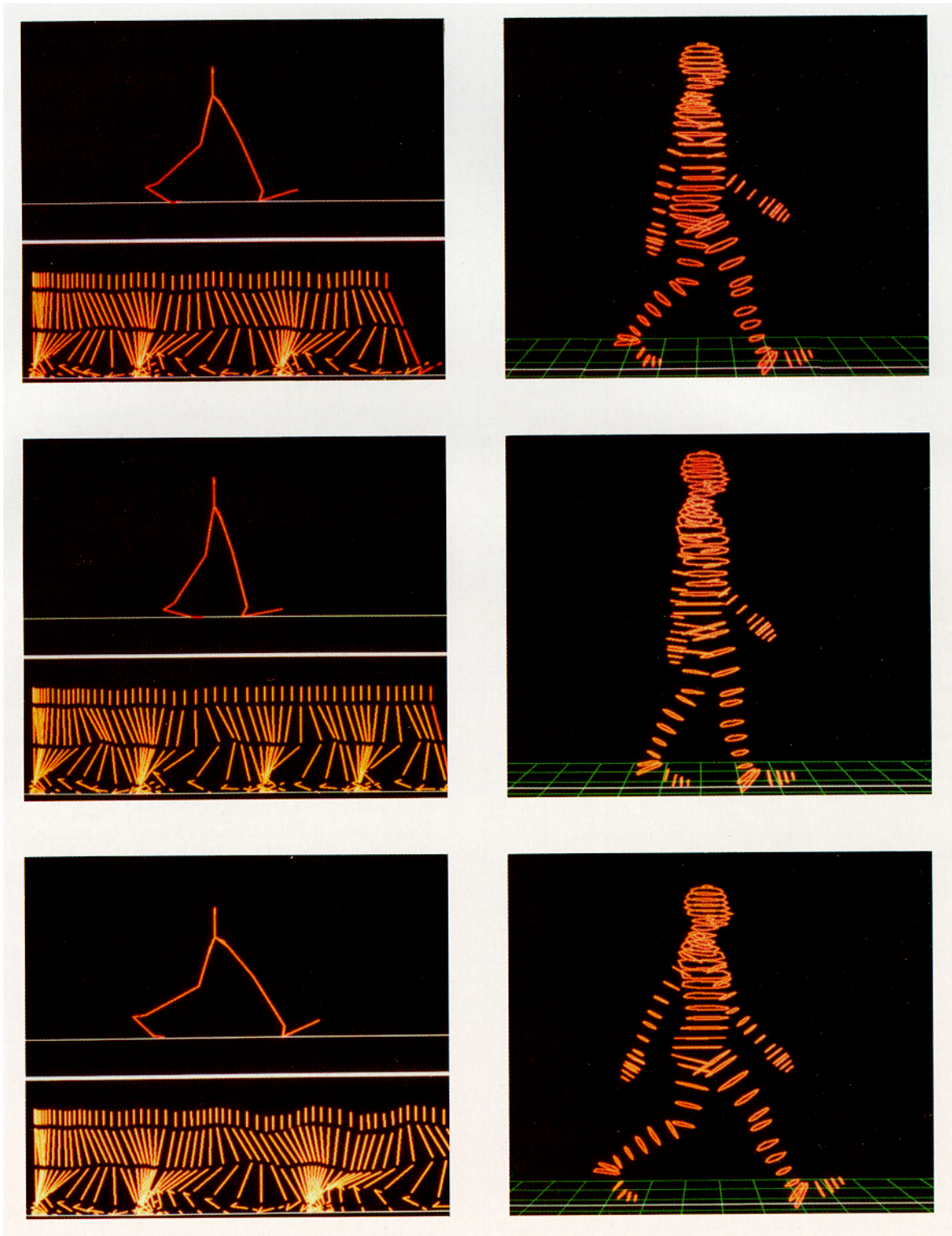
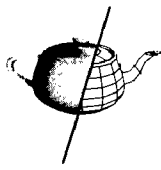


Figure 10: Heel-strike for 3 walking sequences at $v = 5 \text{ km/h}$.
Top: natural walk, only v was specified, $sl = 0.77 \text{ m}$ and $sf = 107.5 \text{ steps/min}$ were chosen by the system.
Middle: short step walk, v and $sl = 0.50 \text{ m}$ were specified, $sf = 166.7 \text{ steps/min}$ was chosen by the system.
Bottom: long step walk, v and $sl = 1.05 \text{ m}$ were specified, $sf = 79.4 \text{ steps/min}$ was chosen by the system.