Goal-Directed, Dynamic Animation of Bipedal Locomotion

by

Armin W. Bruderlin

Diplom (FH) Allgemeine Informatik,
Fachhochschule Furtwangen (W-Germany), 1984

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
December 1988

Approval

Name: Armin W. Bruderlin

Degree: Master of Science

Title of Thesis: Goal-Directed, Dynamic Animation of Bipedal Locomotion

Examining Commitee:

Dr. Brian V. Funt, Chairman

Dr. Thomas W. Calvert
Senior Supervisor

Dr. Binay K. Bhattacharya

Dr. Arthur E. Chapman
School of Kinesiology
Simon Fraser University
External Examiner

21 November 1988
Date Approved

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

Goal-Directed, Dynamic Animation of Bipedal Locomotion.

Author:

(signature)

Armin Bruderlin

(name)

5 December 1988

(date)

Abstract

Since the advent of three dimensional computer animation, motion control for articulated bodies such as humans has been a central problem. Two recent trends are most promising. One goes toward high-level, goal-directed control, reducing the amount of detail necessary to define a motion; the second trend is to apply dynamic analysis to the motion control process, leading to more realism in movements.

In this thesis, a hybrid approach between goal-directed and dynamic control to animate bipedal locomotion is presented. Internal knowledge about the locomotion cycle determines the forces and torques that drive the dynamic model of the legs to produce a natural animation. The KLAW (Keyframe-Less Animation of Walking) system can generate a variety of human walks with very little effort, depending on a few specifications, such as desired velocity or step length.

KLAW should be a useful tool for computer animators. Also, since the motion of the legs is based on dynamic simulation, results from KLAW could be helpful in the design and control of walking robots, or in the analysis of human movements in biomechanics and sports.

To an amazing person

"But the creative principle resides in mathematics. In a certain sense, therefore, I hold it true that pure thought can grasp reality, as the ancients dreamed."

Albert Einstein

Acknowledgements

Due to the nature of the problem, this research required a considerable amount of experimentation. Progress was characterized by much trial and error; innumerable times I went through the simulation - evaluation - modification - cycle before reasonable results were obtained. I am grateful to all my friends, who were willing to endure me during this period.

In particular, I would like to thank my supervisor, Dr. Thomas W. Calvert, for his guidance, support and patience throughout the work on this thesis. I am also indebted to Scott Selbie, who spent many hours introducing me to the theory of analytical dynamics and was helpful in resolving dynamic obstacles thereafter. I sincerely thank Sanjeev Mahajan for his many quick suggestions to geometrical and numerical problems. Finally, I am grateful to Severin Gaudet, whose invaluable advice particularly during the inplementation phase of KLAW is very much appreciated.

Table of Contents

Approval ii
Abstract iii
iv
Acknowledgements v
Table of Contents vi
1. Introduction 1
1.1. History of Motion Control 1
1.2. Proposed Work 2
1.3. Motivation 3
2. Overview 5
2.1. Motion Control in Animation Systems 5
2.2. The Control of Legged Locomotion 13
3. Theory 19
3.1. The Problem 19
3.2. The Basic Approach 20
3.3. The Refined Approach 24
4. Low Level Control Principles 28
4.1. Multilink Structure 28
4.2. Lagrange's Equations 30
4.3. Integrating the Equations of Motion 33
4.4. External Constraints 35
4.5. Interpolation of the Generalized Forces 37
5. High Level Control Concepts 39
5.1. Locomotion Cycle Characteristics 39
5.2. Symmetry of Steps 43
5.3. Determinants of Gait 46
5.4. Virtual Leg 51
5.5. Upper Body Angles 57
6. Low Level Control Details 59
6.1. Stance Phase 61
6.2. Swing Phase 64
6.2.1. SWING1 64
6.2.2. SWING2 69
6.2.3. SWING3 70
6.2.4. Starting and Stopping 71
7. Results 72
8. Conclusions 83

Appendix A. Terminology 86
Appendix B. Body Data 94
B.1. Anthropometric Data 94
B.2. Human Walking Figure Data 96
Appendix C. Derivation of Equations of Motion 98
C.1. Stance Phase 99
C.2. Swing Phase 102
C.3. Collisions 105
C.3.1. Heel Strike 105
C.3.2. Locking of the Knee 106
C.4. Spring and Damping Constants 108
Appendix D. Locomotion Parameters 109
Appendix E. Locomotion Attributes 111
Appendix F. Photographic Images 114
References 122

List of Tables

Table B-1: Anthropometric values of lower body segments.

List of Figures

Figure 3-1: Layers of coordination in articulated motion. 20
Figure 3-2: Model of articulated motion control. 22
Figure 3-3: Schematic diagram of the control system for legged locomotion. 26
Figure 4-1: Examples of multilink structures. 29
Figure 5-1: Locomotion cycles for bipedal walking and running. 41
Figure 5-2: Symmetry of compass gait for different step lengths. 43
Figure 5-3: Dynamic model at heel strike: the swing leg is extended, a foot has been added kinematically; θ_3 is assumed to be negative, all other angles are positive. 45
Figure 5-4: Exaggerated movements of the center of mass during one stride; different line spacing indicates a change in velocity (lines closer together mean deceleration, larger line spacing acceleration). 47
Figure 5-5: Approximations of kinetic, potential and total energy of the HAT for one walking step at average walking speed (adapted from [Inman 81]). The total energy is computed with the minimum potential energy level assumed at zero. The ordinate values in parentheses are for the potential energy. 48
Figure 5-6: Exchange of kinetic and potential energy of a pendulum (neglecting friction). The total energy is constant, thus the system is conservative. 48
Figure 5-7: Superposition of a leg over the dynamic stance leg model. The size of the mid-foot (l_{12}) and toe (l_{13}) are exaggerated; the location of the hip (H) comes from the dynamic simulation, for which $ankle_1$ is assumed to be fixed at (x_a, y_a). The tip of the toe (T) maintains ground contact during the whole stance phase at position $(x_a + l_5 + l_6, 0)$; l_{12} and l_{13} are simply defined as $\sqrt{l_5^2 + l_7^2}$ and $\sqrt{l_6^2 + (l_9 - l_7)^2}$, respectively (see appendix B.1). 52
Figure 5-8: Virtual leg concept for stance leg; the proportions of the foot (l_{12} and l_{13}) are exaggerated. 54
Figure 6-1: Illustration of swing phase (upper body is ignored); the proportions of the kinematic foot are exaggerated. The same legend applies to the leg angles and segments as in figure 5-7. The SWING1 phase goes from t_0 (time of toe-off) to t_1, SWING2 from t_1 to t_2 and SWING3 from t_2 to t_3 (time of heel strike). 65
Figure 7-1: Display of a walking figure. 72
Figure 7-2: Comparison of hip angles during a locomotion cycle. Left: angles as calculated by KLAW; right: angles obtained from a walking human subject [Winter 79]. 73
Figure 7-3: Comparison of knee angles during a locomotion cycle. Left: angles as calculated by KLAW; right: angles obtained from a walking human subject [Winter 79]. 74
Figure 7-4: Comparison of ankle angles during a locomotion cycle. Left: angles as calculated by KLAW; right: angles obtained from a walking human subject [Winter 79]. 74
Figure 7-5: Dynamic hip angle θ_1 for walking cycle W. 75
Figure 7-6: Upper body angle θ_2 (sagittal plane) for walking cycle W; a positive value corresponds to a forward tipping of the body. 76
Figure 7-7: Leg force F_ω and torque F_{θ_1} during stance for walking cycle W. 77
Figure 7-8: Hip torque F_{θ_3} and knee torque F_{θ_4} during swing for walking cycle W. 78
Figure 7-9: Motion of leg for two complete walking sequences at different speeds (only one leg is 79

displayed). Both walks cover about the same distance from the start to the end. Top: 2 km/h, 204 frames, 4 walking cycles (2nd and 3rd cycles are rhythmic). Bottom: 5 km/h, 108 frames, 3 walking cycles (2nd cycle is rhythmic).

Figure 7-10: Hip, knee, ankle and meta angles for a complete walking sequence (including starting and stopping) of 108 frames, which corresponds to the walk of figure 7-9, bottom. 80

Figure 7-11: Hip and knee angles for 3 walks with the same speed, $v = 5\ km/h$, but different step lengths ($sl_{5-1} = 0.5\ m$, $sl_{5-0} = 0.77\ m$ and $sl_{5-2} = 1.05\ m$) and different step frequencies ($sf_{5-1} = 166.7\ steps/min$, $sf_{5-0} = 107.5\ steps/min$ and $sf_{5-2} = 79.4\ steps/min$). The number of frames for each walking cycle w are: $w_{5-1} = 23$, $w_{5-0} = 34$ and $w_{5-2} = 46$. Walk 5–0 is a "natural" walk. 81

Figure 7-12: Hip and knee angles of 2 walks with the same locomotion parameters ($v = 3\ km/h$, $sl = 0.6\ m$ and $sf = 83.3\ steps/min$), but a different value for the `pelvic_list_factor` locomotion attribute: $\texttt{pelvic\_list\_factor}_{3-1} = 1$, $\texttt{pelvic\_list\_factor}_{3-3} = 3$. A rhythmic walking cycle is assumed. 82

Figure B-1: Indices for anthropometric values of lower body segments. 94

Figure C-1: Dynamic models for simulating different phases in legged locomotion; θ_3 is assumed to be negative in the given configuration, all other angles are positive. 98

Figure F-1: Legs shown at heel-strike of a walking sequence (frame 120). The data for this walk are based on real human subjects and were collected by Winter [Winter 79] utilizing film recording techniques. The foot goes slightly through the ground due to the fact that the exact anatomical data of the subjects were not specified and had to be approximated; the walking speed is $v = 5\ km/h$, the step length $sl = 0.79\ m$ and the step frequency of $sf = 107\ steps/min$; no data was supplied for the upper body angles. 114

Figure F-2: Illustration of a walking sequence at heel-strike (frame 103), generated by the walking algorithm. Once the rhythmic phase is entered (after one step), the leg patterns come very close to a real walk (compare to figure F-1). The locomotion parameters are $v = 5\ km/h$, $sl = 0.77\ m$ and $sf = 107.5\ steps/min$; only v was specified, sl and sf were chosen by the system (see algorithm in appendix D). 115

Figure F-3: Illustration of a walking sequence at heel-strike (frame 121), generated by the walking algorithm. The locomotion parameters are $v = 5\ km/h$, $sl = 1.05\ m$ and $sf = 79.4\ steps/min$; v and sl were specified, sf was chosen by the system. Although the walking speed is the same as for the walk in figure F-2, the leg patterns show significant differences; sl approaches a maximum value ($sl_{max} = 1.08\ m$). 116

Figure F-4: Illustration of a walking sequence at heel-strike (frame 108), generated by the walking algorithm. The locomotion parameters are $v = 5\ km/h$, $sl = 0.50\ m$ and $sf = 166.7\ steps/min$; v and sl were specified, sf was chosen by the system. Although the walking speed is the same as for the walk in figure F-2 and F-3, the leg patterns show significant differences. sf approaches a maximum value ($sf_{max} = 182\ steps/min$). 117

Figure F-5: Illustration of pelvic rotation in the transverse plane (a top view is assumed) for the walk shown in figure F-2. Top: heel-strike of left leg at frame 86. Middle: mid-stance of left leg at frame 93. Bottom: heel-strike of right leg at frame 103. 118

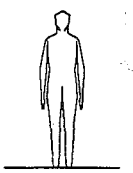
Figure F-6: Illustration of lateral displacement of the body for the walk shown in figure F-2. The body is shifted over the right leg (see arrow) at mid-stance for the right leg (frame 113). - next page - Top: at heel-strike (frame 120), the body is centered between the legs. Bottom: at mid-stance for the left leg (frame 130), the body is shifted over the left leg. 119

Figure F-7: Illustration of pelvic list in the coronal plane. Top: natural pelvic list at toe-off (frame 106) for the walk shown in figure F-2. Bottom: accentuated pelvic list at toe-off (frame 126) for the walk shown in figure F-3. 121

Chapter 1
Introduction

Three-dimensional computer animation[1,A] can be described as the specification and display of moving objects. Generally, this process is decomposed into object modeling, motion specification and image rendering. Although all three phases contribute to making a computer animated film sequence, the second one deserves to be considered the essence of animation, for without it, the problem reduces to just generating still images. Whereas the animator specifies his idea of a motion, the computer has to translate it into the actual positions and orientations for each time step. This aspect of an animation is termed *motion control.*

1.1. History of Motion Control



In the early days, when the objects to be moved were just independent, rigid bodies[A] like boxes or abstract symbols, motion control was straightforward and the paths of "flying logos" could be fairly readily expressed by simple concatenations of matrix transformations. Quite often, though, special rendering effects were more spectacular than the movements of these objects in a scene.

As the complexity of the objects and their potential movements has increased, motion control has grown to become a principal issue. The animation of articulated bodies[A] such as humans and animals has been especially challenging. A body is represented by a hierarchical structure of rotational joints where each joint has up to three degrees of freedom[A] [Calvert 88, Zeltzer 82a]. The human body, for example, possesses over 200 degrees of freedom (DOF) and is capable of such complex movements, that ongoing research is still trying to measure, analyze and represent it. In computer animation, this is often referred to as the DOF problem [Zeltzer 85], which indicates the non-trivial task of coordinating and controlling the limbs of an articulated body to achieve a desired motion out of the vast range of possibilities.

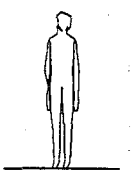
In practice, a motion control system should offer some easy, natural means of specifying a motion and then should generate a realistic animation. Most current animation systems show some kind of a

[1]Throughout this paper, superscripts of capitalized letters are indices to the corresponding Appendix.

trade-off between these two demands; for example, in traditional keyframing [Sturman 86a], the quality of a motion is usually directly proportional to the number of key positions that are specified. Although the computer calculates the in-between frames, the animator is still left to work out a lot of non-creative, tedious detail (i.e. all the joint rotations) at each key frame. In particular if the desired movements are complicated, the animator, rather than the system, does motion control.

In an effort to alleviate the excessive amount of specification for character animation, a tendency towards higher level motion control [Csuri 81, Drewery 86, Zeltzer 83, Zeltzer 85] has emerged. By incorporating knowledge, standard actions or tasks like grasping or jumping are automated and visible to the user only as parameterized modules. The global coordination of a motion is now done by the computer. To achieve a realistic execution of a primitive movement (swinging of an arm, jumping of the body as a whole), dynamics[A] can be applied to the motion control process [Armstrong 85, Calvert 82, Girard 85, Isaacs 87, Wilhelms 85]. By simulating the real world, objects move as they should move, according to the laws of physics. The major drawback with dynamic analysis for computer animation has been that one has to specify a motion in terms of forces and torques which is neither intuitive nor easy.

The compromise which must be resolved is between the need to use dynamics for realistic movements, whereas for a convenient, userfriendly specification, a high-level, task-oriented approach is necessary. The fundamental goal of this research is to merge ideas from both techniques in order to come to terms with one of the mechanically most intricate actions that an articulated body is capable of performing, legged locomotion.



1.2. Proposed Work

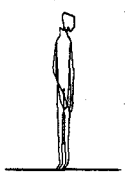
This thesis introduces a control method for locomotion of legged figures, with particular emphasis on humans. The title *Goal-Directed, Dynamic Animation of Bipedal Locomotion* really indicates the interdisciplinary character of the problem: the application is *computer animation*, thus the main goal is to calculate a body reference position, orientation and all the joint angles over time.

The term *goal-directed* is borrowed from Artificial Intelligence or Robotics. In this context, it stands for a high-level control, where the system, rather than the user is an expert on each of the various gaits[A] of locomotion. The system is told *what* to do and not *how* to do it. Goal-directed, as used here, does not address issues like changing directions, path planning or collision detection [Lozano 79]. Algorithms for these problems could be implemented on top of our control.

Dynamics of legged systems originated in Biomechanics and Robotics, and, as mentioned above, recently found applications in computer animation. Although in all three cases dynamics denotes the simulation[A] of the real world, where forces and torques govern the motion of masses, there are subtle differences in the rationale behind their usage. In Biomechanics [Winter 79], dynamic models of the legs are studied to obtain information about the exact muscular forces that are applied during the various phases of a locomotion cycle. In Robotics [Raibert 86a], dynamics is inherent in actual machines that walk or hop. To solve problems such as balance and the control of external forces, the mechanics of bipedal robots and their underlying dynamic models have to be kept simple with the effect that resulting locomotions look rather stiff and not human at all. But natural, realistic movements are exactly the salient ambition in computer animation of articulated bodies.

The approach taken here is to think of dynamics as a sort of low-level control to produce the generic locomotion pattern, which is visually upgraded by some kinematic "cosmetics". The dynamics are regulated by higher levels of control, in that the proper forces and torques to generate a desired locomotion are calculated as a result of a stepwise decomposition of a task (e.g. walk at speed v_{des}).

The system, which is named KLAW (Keyframe-Less Animation of Walking), has been implemented according to these principles. KLAW generates human walks that meet certain *walking parameters* like desired forward velocity, step length or stride width, which are conveniently specified by the user. A walking sequence includes starting from a resting position, acceleration, rhythmic phase, deceleration and coming to a full stop.



It should be understood, that although the system in its current state of development is only able to produce bipedal walks, this approach can easily be applied to other gaits[A] as well as to figures with more than two legs.

1.3. Motivation

Before the era of computer animation, Disney animators had already recognized the importance of realistic reproduction of motion and their great success was based on the long hours they devoted to studying and observing the movements of humans and animals [Zeltzer 85]. The human eye seems to be very sensitive to "errors" and gets disturbed by irregularities in standard actions that it perceives every day.

The natural animation of legged locomotion has always been a problem because in mechanical

terms, it is extremely difficult to capture [Miller 75] even though it appears to be an easy skill of regular and periodic nature. Quite frequently, the issue is avoided by just showing the upper body of a figure during walking or running. Existing attempts to really animate gaits look rather crude and angular, and the animator is overtaxed by the burden of necessary specification. Often, the motion of the foot lacks detail, and the timing for the different phases like stance or swing is incorrect; in fact, in most cases, the whole figure appears to be weightless or tends to move like a puppet pulled by strings.

The best results for the animation of human locomotion are currently obtained through recording or rotoscoping techniques such as television, multiple exposure or optoelectric methods [Winter 79]. To remove the inherent noise from the raw data, smoothing and filtering have to be applied, which makes these approaches expensive and time consuming. Also, they are quite inflexible, since to obtain desirable variations of a locomotion sequence, the whole procedure has to be repeated each time.

The above difficulties became the major incentive for developing a control system that effortlessly and fairly naturally animates the locomotion of legged figures. In addition, a completely different aspect influenced the design and structure of KLAW. Motivated by research on how real living beings control walking [Pearson 76], our approach tries to incorporate results from Neurophysiology: walking appears to be an automated activity governed by the subconsciousness. The only mindful action we take is to have an idea on how fast we want to walk at any time or, what type of walk (or gait) we prefer. This is exactly the input to KLAW. The muscular torques required to move the legs in order to produce a desired walk are generated completely internally (one never thinks in terms of forces or torques). The actual motions of the legs are executed by the autonomous *motor programs* [Csuri 81, Zeltzer 82b], each of which controls a certain group of muscles and joints (*synergies*). In KLAW, rules or internal knowledge about locomotion shape the motor programs, which will in turn determine the proper forces and torques; the synergies are essentially the low-level systems of differential equations which take those forces and torques to produce a motion. Section 3.2 describes this analogy in more detail.

In the first part of the next chapter, a discussion of how current animation systems tackle the general problem of motion control is given. This is followed by a summary of the most significant results on legged locomotion from the areas of Robotics and Biomechanics. The basic approach taken here is outlined in chapter 3 and the main concepts are illustrated. In chapter 4, 5 and 6, the control algorithms are described step by step and implementation issues of KLAW are addressed. Chapter 7 continues with the presentation and evaluation of results gathered from various test runs with KLAW. A comparison with walking data obtained from film is carried out. Finally, chapter 8 concludes with a final evaluation of our approach and suggestions for future work.

Chapter 2
Overview

2.1. Motion Control in Animation Systems

An animation system has to supply the user with tools to generate sequences of motion. Motion control is the specification of the frame to frame changes in an animation that create the illusion of an action [Sturman 86b]. A computer animation system must be able to generate and record the changes in motion of an object which the user requires. We agree with Wilhelms [Wilhelms 86], that this process is still in its infancy and that it is difficult to design for 3-D motion because there is a wide range of possible movements plus an immense amount of information that has to be specified.

In designing an animation system there is a trade-off between avoiding constraints on the imagination of the animator (i.e. giving him complete control over the motion) and avoiding the need for him or her to define all the details of motion. None of the control techniques below should be considered as optimal, for they depend largely on the application and users for whom they are intended. In entertainment, communication comes before realism: traditional animation skills become important [vanBaerle 86]; the characters should often look funny, elastic motion is exaggerated, major actions are preceded by anticipating movements and scene composition is a central issue, etc. For animation in science, the major goal is the simulation of what really happens; movements should look the same as they do in real life.

As vanBaerle [vanBaerle 86] pointed out, from the viewpoint of the animator, all current animation systems have drawbacks that limit the animation process in one way or another. To get the desired motion is also time-consuming; for instance, it took 14 months to produce the 13 minute film *Dream Flight* [Magnenat 83].

An extensive bibliography on current systems is given in a paper by Magnenat-Thalmann [Magnenat 85]. Attempts have also been made to classify the different approaches to motion control [Wilhelms 86, Zeltzer 85, Forest 86]. Depending on the amount of specification needed to define motions or, conversely, the amount of knowledge the system has about generic types of movements, they can be placed

on a scale from *low-level* to *high-level* control. Alternatively, one could partition the techniques according to whether they are purely *kinematic*[A] or use dynamic analysis (*dynamics*[A]).

In the following, two classes of control mechanisms are distinguished: *interactive* (visually-driven) and *scripted* (language-driven or algorithmic). Interactive motion control means that the description of a movement causes immediate realtime feedback on the screen. Thus the animator is able to obtain a quick idea of a motion, possibly modify it and proceed with the specification process.

Keyframing is the oldest such method and is still used by most of today's commercially available systems. In the original 2-D animation systems [Catmull 78], key-positions for a motion sequence (usually 5 frames apart depending on the complexity of the movements) were defined by the user and the computer interpolated the in-between frames using, for example, linear, quadratic or cubic splines. With the advent of 3-D, this method was adopted into systems like *Body* [Ridsdale 86] or *BBOB* [Sturman 86a]. Now the model, motion specification and interpolation are expressed in 3-D. In order to move a character or parts of it, control devices like joysticks are used, which permit the interactive change of the transformation matrices at the joints. Since a higher control is not present in these systems, unrealistic and impossible movements can result.

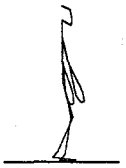
Another interactive technique, *parameterized-keyframed animation*, minimizes the above problems through a slightly higher level of control. Objects, as well as movements are parameterized. The representation of an object contains information on how it may move, thus its DOF and limits of the motion are defined implicitly. In *EM* [Hanrahan 85], parameters determine and possibly constrain rotations by imposing bounds on them. The animator has an interactive control over the parameters; by changing their values (i.e. joint angles), he can specify a motion, or better, a new key frame. The in-between frames are generated by interpolating the transformation parameters.

Keyframe animation generally has the disadvantage of providing only a very low-level of control. For instance, to make a figure bend, the torso is rotated forward, but at the same time, both legs may rotate back off the floor and have to be adjusted manually. There is also no implicit knowledge of balance, other objects in the environment, and the like. But on the other hand, the animator possesses total control which is invaluable for expressing complicated movements.

A third interactive motion control technique is applied in *Virya* [Wilhelms 86]. Control functions specify the motion for each DOF (when operated in *pure kinematics* mode). Since one control function is

stored for each DOF, changes in motion can be easily made by manipulating individual functions. The major drawback with control functions is the difficulty imagining or visualizing the resulting motion in the animated world.

In scripted animation, the motion is described as a formal script by the user and interpreted by the computer in a batch-type manner to produce the animation. Systems like *ASAS* [Reynolds 82] or *MIRA* [Magnenat 83] offer high-level languages to express motion. These allow coordination and interactions of objects. *ASAS* is based on LISP and includes graphical objects and operators. *MIRA*, which is an extension of PASCAL, supports 3-D vector arithmetic, graphic statements, standard functions and procedures as well as viewing and image transformations. *MIRA* further permits the definition of parameterized, 3-D graphical abstract data types for static objects (figures) or animated figures (animated basic type, actor type). Graphical variables of animated data types can be animated by specifying start and end values, a lifetime and a function that defines how values vary with time. The idea of data types which incorporate animation is fundamental also to *ASAS*, where a graphical entity represents an actor with a given role to play. The ideal characteristics of a language to specify human movement have been discussed by Calvert [Calvert 88].



Zeltzer [Zeltzer 85] has described this method of using scripts as *animator-level animation*, meaning that the animator programs the motion. For human-like characters, the algorithmic formulation of complex movements tends to be fraught with problems.

If motion control is achieved at a very high level [Zeltzer 83, Drewery 86, Csuri 81], animation commands are expressed in more general terms much like natural language (e.g. "walk to the door"). This is, what Zeltzer [Zeltzer 85] denotes as *task-level (or goal-directed) animation* to emphasize that the animator only states general tasks like "walking" or "grasping", which the system then transforms internally using an intelligent hierarchical procedure to generate the low-level primitives needed to draw the frames. The system is told what to do and not how to do it. If the motion to be expressed is very regular or periodic like walking, this method can be readily applied and relieves the animator by filling in the details using default values.

Zeltzer designed *SA*(Skeleton Animation System), where the internal hierarchical procedure to obtain the movement instructions from the task specification is implemented in 3 layers [Zeltzer 82b]: at the top-level, a *task manager* isolates the motion verbs , like "walk", from the task and assigns it to a corresponding skill *s*, which is internally represented as an intelligent data structure, a frame[A]. Attached to it are slots that point to other skills which, under certain conditions, might have to be executed first before *s* can

be satisfied. This inherits the idea of potentiation and depotentiation [Zeltzer 83], whereupon one skill may enable or disable other behavioral units and the extent to which this is done reflects the richness and flexibility of behavior. As an example, the skill for walking might have a slot for "stand up", which is potentiated each time a walk is requested. If the figure is already standing, it starts to walk; if it is sitting on the ground, then the stand-up skill becomes active first, and in turn might activate, or at least potentiate other skills. At the middle level, the skills now get executed by corresponding *motor programs*, which invoke a set of *local motor programs* (LMPs) on the lowest level. For walking, the LMPs could be left swing, right swing, left stance, right stance. All motor programs are implemented as finite state machines with the input alphabet consisting of feedback signals (events) from the movement process. For example, the FSM for the left swing phase in walking would go into its final state if the event "heel-strike" is signaled. The control is then returned to the FSM of the walk-controller. This approach supports *adaptive motion* [Zeltzer 85] yielding an intelligent system incorporating information about the environment, collision detection and obstacle avoidance [Lozano 79].

There are other motion control techniques that do not quite fit into the proposed classification scheme. Badler's [Badler 79] research initially focused on notation systems like Labanotation, whose prime goal is the recording of movement. He proposed a computerized version of Labanotation and designed an architecture [Badler 80] consisting of one processor for each joint of the body. These joint-processors execute parallel programs which contain descriptions of motion expressed in Labanotation-primitives (directions, rotations, facing, shape and contact). A progression-processor is responsible for the center of gravity and is a monitor for scheduling. The monitor further maintains the body data base, supervises possible contact events and passes the joint positions to the graphics output. Although the architecture is quite notable, this approach is most useful to experts in Labanotation who can describe a dance sequence in this complicated notation.

In recent years, Badler *et al.* have been developing *TEMPUS*, a system to animate human figures. In order to obtain different key positions of a figure, the user is no longer required to define all the joint angles. Instead, positioning is done by specifying multiple constraints, or goals, and the system calculates the joint angles by applying an *inverse kinematics*[A] method to meet these constraints [Badler 87]. A goal has two parts: the desired location of a limb, and a weight which indicates the relative importance of the constraint or the degree to which the constraint should be enforced. For example, the user may define the locations for the left (G_1) and right (G_2) hand as two constraints, the former with a weight of 40, the latter with a weight of 10. If both goals are attainable, i.e. both hands can reach their desired locations, the system calculates the angles for the joints between the two hands, ignoring the weights. If the specified locations are too far apart to be reached by the hands, the goals are approximated the following way: since the weight of the left hand is

4 times bigger than the one of the right hand, the distance between the left hand and G_1 will be 4 times shorter than the distance between the right hand and G_2. Besides this kinematic algorithm, which supports keyframing, it has been an ongoing process to embed ideas from other techniques like dynamics (see below) or task-oriented control into *TEMPUS*. As Badler notes [Badler 88], no single technique is generally superior for articulated motion control.

There are other approaches that use combinations of motion control methods. Forest [Forest 86] applies a parameterized-keyframed concept which relies on ideas from algorithmic animation. A keyframe-based subactor, which is a variable of type subactor and defined similarly to an actor in *MIRA*, is dependent on some parameters, whose values change from keyframe to keyframe. Usually, interpolation is applied to get parameter-values for in-between frames, but if a law (algorithm) is specified for the behavior of a parameter, interpolation is ignored and the values yielded by the law are used. This is particularly useful because, for instance, the laws of dynamics (see below) could be implemented to calculate the values of certain joint-angles when needed, thus resulting in a more realistic motion.

Another animation system, *Twixt* [Gomez 85], is also logically situated somewhere between a keyframe and a scripted system. It is based on *event-driven animation*, which is more of a concept than a particular technique for motion control, because it considers all the different aspects of making animation: control values (of possibly different types) for position, rotation, scaling, color, transparency etc., together with the time when they are used, define *events*. Control values are the input to associated *display functions*, which output new values contributing to the picture. A *track* is a time-sorted list of events describing the activity of a particular display function F. Events are stored only if an input value for F changes, then interpolation is applied between different values. Tracks are treated as abstract objects which can be linked together and transformed. For example, one could transform the track for the position of object A into object B's position track (this is possible since both are of the same type) and multiply it by -1. The resulting animation would show B exactly mirroring A. Due to this manipulation on the track-level, *Twixt* is not only a keyframe system, but inherits some higher algorithmic control. In terms of user-friendliness, the definition of display functions seems rather abstract. Much of this approach is incorporated into commercially available systems, such as those of Vertigo and Wavefront.

All of the above mentioned motion control techniques are built on a *kinematic* model of the objects. Motion is generated solely by the determination of positions over time neglecting the forces and torques that actually cause the motion. Thus movements often produce a somewhat unrealistic appearance [Girard 85]. A *dynamic* analysis describes a system with the underlying physical laws of motion. Dynamics has appeal as an

alternative method for motion control, since the behavior (movement) of an object is totally defined by its equations of motion[A]. The main advantage over kinematic systems is that the motion is bound to look natural since bodies move under the influence of forces and torques (such as muscle torques) or gravity in a way that depends on the actual physical characteristics such as mass, friction etc. An object responds naturally to collisions like heel-strike in walking. Also, a motion which is caused in one part of an articulated body will automatically affect other body parts.

Wilhelms [Wilhelms 85] and Armstrong [Armstrong 85] used this approach for the simulation of the human body. The former developed *Deva*, where the equations of motion are formulated (using the Gibbs formula) for a 15 segment human model. Armstrong derives the equations of motion by Newtonian formulation, and specifies 6 equations (actually three pairs) for torques and forces at each link and for the relation between accelerations at the parent and son nodes.

A major problem with dynamics of this complexity is that there are no analytical solutions and as a result of the many DOF, the system of nonlinear differential equations becomes rather large. Although some effective recursive numerical methods [Armstrong 85] exist, the cost of computation is high and numerical instabilities can occur.

Probably the biggest disadvantage is that the forces and torques must be specified as input to initiate and guide a motion. Whereas with kinematics, movements are defined by positions in space and the animator has to make adjustments until the motion between these positions looks right, with dynamics, motions look perfectly real, but the animator has to experiment until he gets the one he desires.

This somewhat limits the usability and practicality of pure dynamics for the purpose of animation. Attempts have been made to get around the force-torque specification problem, or in general, to simplify a full-blown dynamic simulation in one way or another, by applying some mixture of dynamics and kinematics concurrently.

In *Virya* [Wilhelms 86], as already mentioned above, control functions define the movements of either translational or rotational joints over time, when operated in *kinematic mode*. *Virya* also supports a *dynamic mode* where these control functions represent forces for sliding DOF or torques for revolute DOF. In this mode, one is faced with exactly the above difficulty of having to non-intuitively specify motion in terms of forces and torques. *Virya* therefore offers a *hybrid k-d mode*, in which the control functions describe translations or rotations of a joint as in kinematic mode. These descriptions are then taken to estimate the

forces or torques for the corresponding DOF that will reproduce the specified motion when input to the underlying equations of motion for that joint. However, the application of the hybrid k-d mode is rather restricted: since the calculated forces and torques are only an approximation obtained by a stripped-down inverse dynamics[A] procedure, the dynamically generated motion will not be exactly the same as its kinematic description in the control functions. In addition, the motion for a joint has to be defined at each time step first (as a control function), which is basically all that is needed in computer animation, so the subsequent dynamic simulation is of little immediate contribution. Nevertheless, it is fair to say that acquiring knowledge about forces and torques is helpful in shaping the control functions in dynamic mode. Also, applying dynamics might still be useful if the motion of certain joints is known and the remaining DOF obey the laws of physics.

Another approach, based on dynamics enriched with some kinematic aspects, is shown in *DYNAMO* [Isaacs 87]. At each time step during a dynamic simulation, *behavior functions* describing the desired forces or motion of an object are executed first. Their output is then used to solve the dynamic equations. For example, the behavior function for damping of a joint accepts as input the angular velocities at each time step and calculates the decelerating torque. *Kinematic constraints* can be specified in terms of accelerations for some DOF for every time step, and this reduces the number of equations much like the control functions in *Virya* during hybrid k-d mode. *DYNAMO* also supports *inverse dynamics* in case the motion for a DOF is known, but not the forces. Through the use of behavior functions the system is able to give elementary movements like a swing or a kick a realistic appearance.

A step in the opposite direction has been undertaken by Girard and Maciejewski [Girard 85]. They designed *PODA*, a system to animate legged figures, which is implemented using kinematic techniques extended by a few very basic dynamic ideas. The dynamics became necessary to ameliorate the typical visual kinematic side-effects when animating locomotion, such as bodies looking as if they were suspended from strings dragging their feet behind them. The dynamics in *PODA* define the motion of a body as a whole. The vertical and horizontal control are treated separately. In the vertical case, the animator has to supply the system with an upward force for each leg, which will cause an acceleration depending on which current phase the leg is in. The trajectory followed by the body is just the sum of all the individual leg accelerations. The horizontal path of the figure is specified by the animator as a cubic spline. The summation of all the horizontal leg forces that are input to the system will yield an acceleration, which determines the position of the body on the spline. The legs are moved kinematically. The animator is able to design different gaits for figures with variable number of legs by assigning values to parameters like duty factor[A], relative phase[A], the duration that the leg is on the ground and in the air during each cycle, etc. A step is specified by the trajectory

of the feet through key positions. The coordination of the legs is done by the system. To make sure that a foot stays on the ground during a support phase, the inverse kinematics problem has to be solved, which is done by generating the Jacobian and calculating its pseudoinverse.

Although results from *PODA* are quite impressive, we believe that a system for animating legged locomotion should not only coordinate the legs but also execute the quite intricate movements of the legs autonomously without the user having to specify positional data over time. The simulation of the legs is an appealing thought, since the control is shifted to the underlying physics. But this brings along another problem as mentioned above. Since we want the system to behave in a certain way, it becomes non-conservative[A] and knowledge about external forces and torques is required to produce a desired motion. As long as one can not conveniently determine the inputs to a dynamic system, dynamics remains just a toy in computer animation. The fundamental difficulty is not to derive the equations of motion, it is to control them. One way in which dynamics can succeed is to make it more application-oriented, i.e. to have different dynamic models for different, specific motions rather than striving for a universal model of a whole figure with its many DOF. At the same time, knowledge about these specific movements could "guide" the dynamics.

In this thesis, the control of legged locomotion is investigated. A higher level control is suggested, where information and rules about the locomotion cycle lead to the determination of the forces and torques required to move the legs. The dynamic model is tailored for simulating a gait. Studies on legged locomotion, which are presented in the next section, have inspired this approach.

In summarizing the motion control techniques described above it should be noted that none can be called best or worst *per se*, although there is a tendency away from keyframing towards higher levels of control. It is questionable, however, whether there will ever be a system that provides a high level, task-oriented control for all of the types of movements which complex articulated figures such as humans are capable of performing. That is why keyframing will almost certainly survive as a means to specify complicated, individual and detailed motions that can only be "hand-coded".

2.2. The Control of Legged Locomotion

The scientific analysis of legged locomotion began in 1872 with Eadweard Muybridge. By electronically triggering a series of cameras along a horsetrack in California, he was able to prove that there are phases during the trotting of a horse, where all 4 feet are off the ground. Subsequently, Muybridge extended his studies of gaits and postures to other mammals including humans. His extensive photographic documents are published in several volumes [Muybridge 70].

Since then, research on different areas of science has contributed to a better understanding of legged systems. A lot of investigation has been done on human walking [Dagg 77, Inman 81], and at least conceptually and kinematically, it is well understood. Most research is of an analytical nature and little has been done to actually assemble or synthesize a whole walking sequence. In the following, some results from Neuro-Science, Zoology, Biomechanics and Robotics are briefly discussed keeping in mind our main goal, the animation of bipedal locomotion.

In *Neuro-Science*, the control of walking, running, etc. are investigated. Early in this century, C.S. Sherrington and T.G. Brown defined 2 concepts that describe animal locomotion [Pearson 76]. They observed - independently from each other - a basic *rhythm* (pattern), which is generated entirely within the spinal cord, thus somewhat subconsciously, and *reflex actions*, that reinforce this rhythm. The reflexes, caused by sensory signals during one step, elicit the next part of the cycle. Although most of the research was done on cats and cockroaches, it is widely accepted that these concepts are applicable to humans as well. A mathematical model of a possible structure and operation of the spinal pattern generators was developed at SFU by Patla [Patla 85].

Locomotion in *Zoology* involves studies of the gaits of animals. Alexander [Alexander 84] classifies gaits and identifies some of their important features. The 2 major gaits for bipeds are *walk*, where at least one foot is on the ground at all times, and *run*, which contains phases during a stride[A] when both feet are in the air. Alexander also examines the effects of body-size on motion and he shows that mammals (including humans) of different height move *dynamically similarly*, if their *Froud numbers* $\frac{u^2}{gh}$ are equal (here u is the speed of travel, g is the acceleration of free fall, h is the leg-length). Two systems behave dynamically similarly, if the motion of one can be made identical to the motion of the other by multiplying all forces, time intervals and linear dimensions by a constant (e.g. 2 pendulums of different length move dynamically similarly). A practical result from this hypothesis is that animals of different sizes use the same gait when traveling with equal Froud numbers.

The *conservation of energy* in mammalian walking is another important principle. Mammals naturally choose a cadence as to more or less minimize energy cost at any walking speed (see below). Alexander derives formulas for the horizontal and vertical forces exerted in human walking and running on the basis of this principle. Research on humans [Inman 81] has indicated that any variation in the individual, natural walking patterns resulting from alterations of step length and step frequency at a given speed, increase the required energy per unit distance traversed. The total energy expenditure for a person during walking is shown to be directly proportional to the square of the forward velocity. Energy per unit distance walked per unit body weight is also investigated [Inman 81]; an adult human seems to be walking most "efficiently" at a speed close to 80 m/min in the sense that energy cost per meter traveled per kilogram of body weight is minimal at this speed. In this context, it should be mentioned that in terms of total energy expenditure, running is comparable to a bouncing ball, whereas walking resembles more a rolling egg. In fact, energy in walking is almost at a constant level, since kinetic and potential energy rise and fall in opposite phases.

The *Biomechanics* of locomotion can be defined [Winter 79] as the measurement, description, analysis and assessment of locomotion. The measurements involve recording the movements (by filming or other means), recording of the electrical signals of muscular activities (electromyography), and the use of force plates to get ground reaction forces. The description and analysis of locomotion require a dynamic model of the body-system, defined as a set of equations of motion for each DOF. The equations can be solved in 2 ways: the forces of the system may be given and the resulting motion is determined (*forward dynamics problem*), or the motion may be specified with the objective of finding the forces to produce this motion (*inverse dynamics problem*). Since measuring forces is usually fraught with inaccuracy, the inverse problem is solved first. The calculated forces can be used to test the model by solving the direct problem yielding kinematic data, which can be compared to the original motion.

Under certain circumstances, when the body forms one or more closed loops with itself or an external system, the initial inverse dynamics problem becomes difficult to solve. This is known as the *closed loop problem* [Vaughan 82a]. The most common example of this problem is the double support phase of the walk cycle: in a simplified model in 2-D, where we assume 2 stiff legs jointed at the hip, they form a triangle (closed loop) with the ground; only one angle is needed to totally specify the system, thus only one equation of motion results, which permits calculation of the magnitude of one force. However, individual ground reaction forces and a hip-torque may arise that make the system *underdetermined*. Vaughan [Vaughan 82a] classified common closed loop problems for the human body according to the degree of over-(under)determinism and the number of extremities which cause closed loops. He [Vaughan 82b] also

derived a general solution for this problem [Vaughan 82b], formulated as an optimization problem. Horizontal and vertical reaction forces were predicted reasonably well, although their points of application could not be determined exactly. (A similar type of problem occurs in inverse kinematics, when redundant DOF are present: there is usually an infinite number of solutions for the angles of the proximal joints given the position of the distal ends.)

Solving the equations of motion for human systems is not the only difficulty. It is also difficult to choose a reliable model, which inherits all the characteristic factors of locomotion. It seems, even though bipedal walking and running are well observed and documented processes, that the abstraction and the formulation of proper equations of motion is troublesome. McMahon [McMahon 84] defined a *ballistic walking model.* He observed that human locomotion has quite a lot in common with the motion of a pendulum. Electromyographic measurements show muscular forces (torques) during the double support phase but very little muscle-activity in the swing phase. Therefore, the swing leg behaves much like a conservative jointed pendulum. By trial and error, McMahon determined the initial angular velocities of thigh and shank for various step-lengths. In the simple case, the model only simulates a compass gait (see section 5.3), but according to the author it can be extended to include additional gait-determinants. Since the model ignores the stance leg, which implies that the simulation is just done for about half a stride, a general statement about the validity of the approach for a total walking sequence must await further investigation.

Beckett and Chung [Beckett 68] also investigated the behavior of the swing leg of a walking step[A]. The motion of the hip follows a sine curve and is assumed to have constant velocity. Restraints are imposed on the movements of the swing leg according to 3 phases: in phase I, the toe is fixed to the ground and the foot rotates around it (this happens before the actual toe-off). In phase II (after toe-off), the ankle joint is locked and the toe moves along a predefined curve (which seems to be a very heavy restriction). Phase III is entered when the thigh reaches maximum flexion; then only the knee-joint is active until the leg is straight. A positive hip torque of constant magnitude is applied during the first 10% of a step, i.e. during phase I, to accelerate the leg and a negative hip torque is applied for the last 12% of the step to decelerate the motion of the thigh. The authors note that the torque at the knee plays a minor role during the swing phase in level walking. The knee is extended mainly through the negative hip torque at the end of the swing. Also, difficulties of obtaining smooth transitions between phase changes are reported, especially from phase II to III.

At this point, our main objective should be recalled: we are trying to find ideas from the above areas to implement bipedal locomotion for the purpose of computer animation. The last 2 examples, although

giving concrete formulations, address, as indicated, only part of a total locomotion cycle with only one leg involved and no upper body present. Hence the control problem of coordinating the different phases like stance and swing is avoided. Moreover, all simulation was done in 2-D. When extending the problem to 3-D, besides having to cope with more DOF, there is one particular issue that becomes non-trivial: *balance*.

In *Robotics*, where actual legged machines have been built, the problem of preventing the robot from falling over has been solved in different ways. Raibert presents an overview of legged robots and their justification [Raibert 86b]. Early walking machines all relied on four or more legs. The problem of maintaining balance becomes a *static* one; if at all times during locomotion the center of gravity (*cg*) falls within the convex hull spanned by the supporting legs, the figure can not fall over. For instance, in the quadrupedal case, if 3 feet are always on the ground, and the *cg* lies inside the "triangle", the figure is statically balanced (this is actually how 4-legged animals keep balance at normal walking speed). Bipedal (or even one-legged) machines have to be balanced *dynamically or actively* by "controlled tipping" (this is how most higher animals balance). Tipping is permitted for short periods of time but eventually it is compensated by a tipping in the opposite direction. Only a few machines have been constructed that are able to solve this task.

Miura and Shimoyama [Miura 84] developed a series of dynamically balanced, bipedal robots (BIPER-1,2,3,4,5). All are statically unstable, i.e. they have to move in order to not fall over. BIPER-3 has stiff legs and treats the stance leg as an inverted pendulum. A walking sequence is expressed as a series of inverted pendulum motions. The placement of the swing leg on the ground to compensate the tipping is determined by the tipping direction of the inverted pendulum. The movements of the legs are controlled by three motors, which allow each leg to swing sideways (pitch axis) or forward (roll axis). To lift a stiff leg off the ground, it is first moved along the pitch axis, and then the roll axis to compensate the tipping. The stiff leg movements look quite unnatural, but dynamic balance is guaranteed.

At SFU, Dr. Tad McGeer [McGeer 88] has taken a conceptually different approach in the design of a walking machine. He studied *passive* walking, where the natural dynamic characteristics of the model generate the walking cycle without requiring active controlling forces. Whereas in most other approaches the control is the major issue, McGeer's goal is to develop a basic unpowered machine that walks. Initially, this research was inspired by a bipedal toy which walks down shallow inclines by rocking from side to side to prevent the swing leg from stubbing its toe. No external control is necessary since the energy lost on impact (heel-strike) is compensated by the slope of the surface. In McGeer's robot, lateral rocking is precluded by the purely planar motion of two paired outer legs and one center leg. In order to clear the ground during

swing, the leg(s) must be shortened and subsequently lengthened during the stance phase. Thus, small retraction and extension forces are introduced which make the system not totally passive. However, these minor forces along the leg axis enable the robot to achieve a stable locomotion cycle on level terrain. The author claims that once passive walking is fully understood, it should be straightforward to add power to the machine.

Raibert [Raibert 84a, Raibert 84b] has concentrated his research on running rather than walking, and developed a one-legged hopping machine. (Hopping represents a special kind of running where all feet leave the ground at the same time. For systems with one leg, hopping is identical to running.)

This approach possesses 2 significant advantages over the previous bipedal walking method. First of all, the coordination of different legs becomes redundant since there is only one leg. Secondly, as Raibert argues, the control of the system turns out to be much easier, mainly because during the flight-phase (which does not occur in walking), the angular momentum is conserved. As a result, for instance, the foot can be positioned at a certain angle in the air, which influences the adjustment of body attitude and forward velocity during the stance interval. On the other hand, in the two-legged walking case, since at least one foot is always on the ground, generating a torque on the swing leg directly affects the stance leg and balance, which makes control difficult. Raibert decomposes the task into a *vertical* control to regulate hopping height and frequency, and a *horizontal* control for adjusting body attitude (balance) and forward velocity. The latter two variables in need of control are closely coupled and regulated by the following two *control actions*: foot placement and hip control. Foot placement, which is adjusted during flight, directly influences balance in that placing the foot forward of its neutral position (which is the center of the distance that the body travels during the support phase) will cause a backward tipping of the body, whereas placing it behind its neutral position will cause a forward tipping. Hip motion during stance phase, generated by a torque at the hip, influences the angular momentum and therefore is used to control forward velocity. For example, if no forward acceleration should occur, the angular momentum has to be kept constant (the horizontal components of all forces are zero), which means that the foot sweeps back at the same rate as the ground. The control actions have multiple effects on the system. That is why an alternative control algorithm works equally well: foot placement can be assigned to control forward velocity. Placing the foot forward (behind) its neutral position will decelerate (accelerate) the machine. If the foot is positioned directly on the neutral point, constant velocity is maintained. The hip torque applied during the support phase now controls the body attitude. These ideas are readily extensible to 3-D [Raibert 84b].

Although Raibert's control system is as ingenious as it is simple and has solved many problems in

the robotics world, its application to computer animation is limited, mainly because the objectives of the two applications are so different. Animators are not satisfied with "just" being able to control velocity -they want variations of walks and runs even for the same velocity with different step length and step frequency combinations, or a diversity in pelvic movement patterns. Moreover, the motion of Raibert's hopping machine does not look human or animal-like at all, since it is simplified to a minimal number of DOF. But this is exactly why Raibert succeeded in controlling it, and why he was able to make the machine move in any desired way.

The underlying dynamic model of the hopping machine is just a two-segmented pendulum, one segment for the upper body and a telescope segment for the leg. We adopted this model to simulate the stance phase. The results from the dynamic simulation provide the motion patterns, which are then visually improved at each time step by kinematic algorithms (e.g. a human foot is attached and the changing length of the telescope leg simulates knee flexion), which in turn affect the dynamics. The model of the swing leg is a double pendulum similar to Beckett's and Chung's [Beckett 68]. The exact realization is explained in chapter 6. Before discussing this, in the next chapter, the control algorithm is presented as a whole and the main concepts and formalisms are clarified.

Chapter 3
Theory

3.1. The Problem

The coordination and control of articulated motion is a complex procedure. Movements are entirely expressed by rotations around the joints of a body. However, each joint is an integral part of a multilink structure, or kinematic chain [Korein 82], which leads to a potentially large number of degrees of freedom and makes almost any kind of motion of an articulated figure difficult to capture. A particular challenge has been to animate legged locomotion, since rotational movements in the lower extremities must be coordinated in such a way as to achieve the desired overall translation of the body.

In a one-level control system, like keyframing [Sturman 86a], it is left to the animator to explicitly specify the proper joint rotations over the duration of a movement. That is, the animator has to determine which degrees of freedom are involved and how much they contribute to producing a desired motion. Thus, the specification process can become rather lengthy and intractable. Furthermore, the resulting animation of an articulated body is, quite often, only an approximation of the motion that the animator had in mind. These drawbacks arise because articulated motion is subject to anatomical, physical and environmental constraints that make a structural control necessary. If the system only supports one level, the animator has to estimate the impact of the constraints while constructing basic movement patterns.

A better way is to adapt the control system to the structured nature of articulated motion, that is, to subdivide it into several hierarchical layers as illustrated in figure 3-1. At the top level, the interaction of a figure with its environment is coordinated. This aspect of motion control, termed *motion planning*, involves algorithms for finding paths and avoiding obstacles. Since motion planning on its own is a major research area in animation, vision, and robotics, it is not included in this work. The interested reader is referred to papers by Lozano-Perez [Lozano 79] and to Ridsdale's PhD thesis [Ridsdale 87]. The remaining three levels affect movements internal to a figure and are generically known as *joint-coordination* or motor control [Zeltzer 82b]. They address the coordination between the segments of a body, the coordination within a single segment, and at the bottom level, the actual rotations at each joint. If all of these issues are

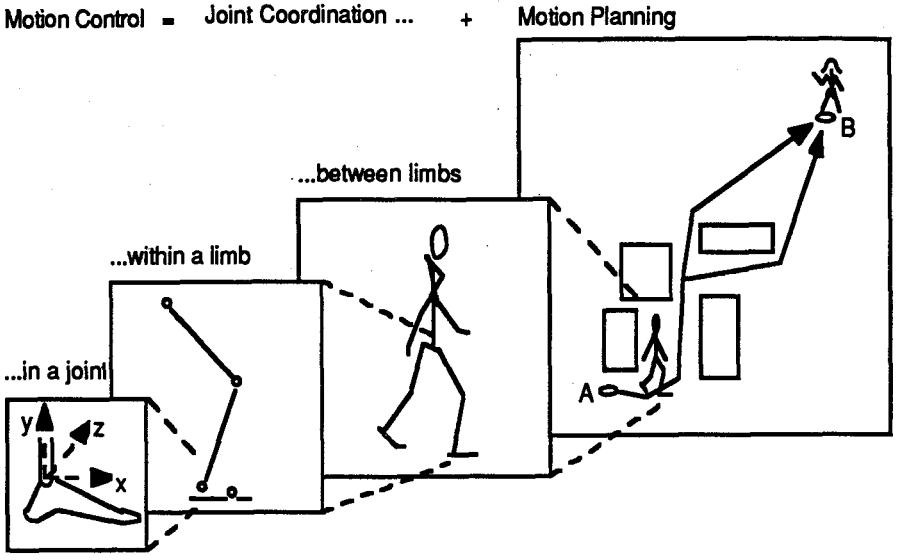
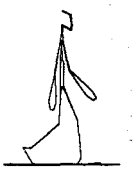


Figure 3-1: Layers of coordination in articulated motion.



incorporated into an animation system, one arrives at a high-level, goal-directed control. Zeltzer [Zeltzer 82b], as discussed in section 2.1, has been working on such an approach. The key idea is that the system "knows" about certain classes of motion and provides the animator with a set of movement commands or parameters which completely control the figure. For instance, the parameters for an action like walking could be forward velocity, step length, and direction of the walk. The animator no longer has to specify joint angles over time because the system, once initiated, is able to autonomously execute a desired motion. This type of control, as well as the way that knowledge about figure motion is integrated into the control process, parallels results from Neurophysiology on how living organisms manage to control complex articulated movements. We therefore take a closer look at a simple biological control system from which we derive a scheme for animating legged locomotion.

3.2. The Basic Approach

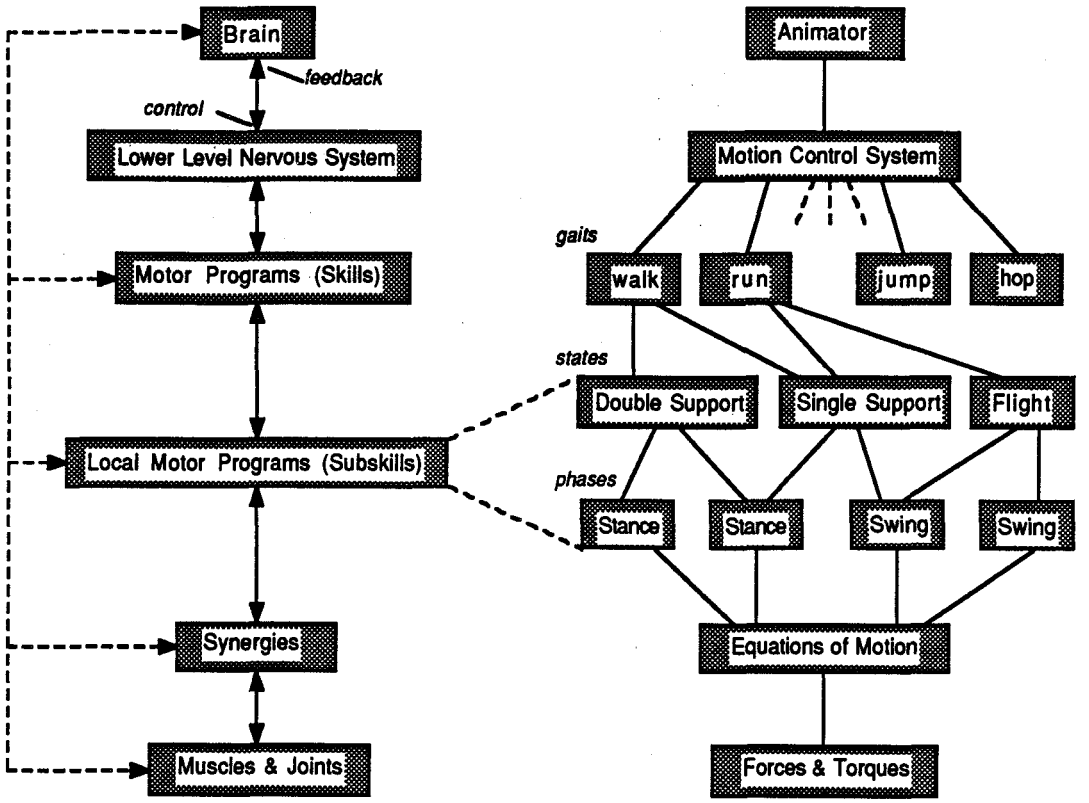
Biological movement systems are inherently goal-directed [Zeltzer 83]. They are able to constantly master complex sequences of articulated movements, seemingly effortless and partially governed by the subconscious. It appears that information about the situation or stage which an organism is in, and knowledge of the impending action reduce the number of degrees of freedom to only allow a specific motion to take place. For instance, if a cat sees a mouse, the angle of tilting and the turning of the head and neck tune

the spinal motor centers so that the brain only has to fire the command "jump" to make the cat jump in the right direction to catch the mouse [Greene 72]. The key notion is that higher vertebrates maintain a distributed set of hierarchical motor programs, with the effect that the whole system with many degrees of freedom gets decomposed into many subsystems, each with only a few degrees of freedom [Csuri 81]. An alternative view suggested by Greene [Greene 72] assumes that subsystems with many degrees of freedom are guided by higher level control systems which possess only a few degrees of freedom.

A simple motor control system is presented in figure 3-2 (a). The brain takes on a central role. During the learning period of a motion, it defines synergies, which are groups of cooperatively acting muscles (and joints) capable of performing a particular class of movements. At the same time, the brain also trains or determines the motor programs that control the synergies to execute a specific action. It is possible that different motor programs act upon the same synergy. For example, the synergy of the hip, knee and ankle joint, comprised mainly of the hamstrings, quadriceps and triceps muscle groups, is used by the motor program for kicking a ball as well as for walking. The motor control programs are organized hierarchically (and actually implemented as layers of local motor programs to indicate their increasingly local effect) by stepwise refinement of the control and reduction of the number of degrees of freedom for an action. Because of their interconnection, the (local) motor programs can be combined in various ways to suit the respective circumstances. This accounts for the immense flexibility in behavior and makes it unnecessary to store explicit movement descriptions.

Once a skill is acquired it becomes part of the "muscle memory". The lower level nervous system is able to autonomously carry out the motion issuing the proper motor programs. Movements now proceed without the total attention of the brain, which is only active to initialize and assign global motion parameters. The brain, therefore, behaves much like the pilot of an airplane, with the motor programs representing the autopilot. As a pilot, it supervises what happens by receiving constant feedback from all parties. If required, the brain may also interrupt the current process, modify synergies, reshuffle motor programs or just switch to another preprogrammed motion (this is indicated by the dashed line in figure 3-2 (a)). Because of the interrupt capability all biological systems are extremely adaptive to their environment. In figure 3-1, which explains the different layers of coordination, adaptive motion would be handled at the top level along with motion planning. The levels of internal joint coordination are absorbed by the motor programs.

This natural control concept is adapted to animate bipedal, legged locomotion using the approach in figure 3-2 (b). The animator assumes the position of the brain. His task is to initialize the desired motion. Since no information is kept on other objects in the scene, feedback at this level, and therefore adaptive



a) Motor Control Hierarchy

b) Animation of Legged Locomotion

Figure 3-2: Model of articulated motion control.

motion, is limited: the animator can just modify some parameters like forward velocity or step frequency; he should also be able to change the gait of a figure over the duration of an animation. (Note: this last point has not yet been implemented in KLAW, which controls bipedal walking.)

The final responsibility of the brain, namely to facilitate learning, is not included in the duties of the animator. The system already contains knowledge about locomotion, which is exactly the salient benefit of a goal-directed approach. This knowledge is incorporated at various levels: depending on the animator's specification, the control system selects a proper gait that is then decomposed by the (local) motor programs into its underlying components. Every gait is made up of the different states from a locomotion cycle. These states, in turn, are broken up into the phases which the individual legs happen to be in. Thus, a gradual reduction in the number of degrees of freedom, along with a decrease in the levels of coordination, is

achieved by the control system which parallels that of its natural, biological counterpart. In KLAW, the phases get further subdivided (STANCE1 to STANCE4 and SWING1 to SWING3) to restrict the range of possible movements for the dynamics and to adapt the motor system to special cases in locomotion (see chapter 6). At the bottom level, knowledge is represented by the dynamic equations of motion. Like the synergies, these equations are tailored to perform a specific application, and therefore, in a sense, are kept at a minimal number which proves to be advantageous when solving them (see section 4.5).

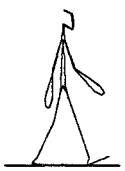
It is worthwhile to notice that the hierarchical illustration of figure 3-2 explains the decomposition of a task or skill quite well, but does not reveal all aspects of the control mechanism for locomotion. All gaits, once a constant forward progression has developed, are characterized by cyclic or rhythmic activities of the legs. Studies in Neurophysiology have led to the hypothesis of spinal *pattern generators* in vertebrates that are responsible for stimulating the proper muscles to drive the legs through their cycles [Pearson 76, Patla 85]. There also seems to be some means of *peripheral feedback*, whereby sensory signals can influence the output of the pattern generator such that, for example, an obstacle on the ground could trigger the swinging leg to be lifted higher than usual in order to clear the obstacle.

For computer animation, it has been suggested [Zeltzer 82b] that the pattern generators, which are represented by the local motor programs for the various phases of the legs, like left stance or right swing, could be modeled by finite state machines. A similar approach was taken here. In KLAW, the successive execution of the states for double support and single support maintains the cyclic locomotion pattern. A transition between states occurs when the time for the current state has been exceeded. Within a state, the different phases are executed concurrently: during double support, both legs are in their stance phases; in single support, one leg is on the ground while the other one swings forward. The bookkeeping of what leg is in which phase is done at the state level, and therefore, in a sense, by the finite state machine: one just imagines that there really are two states for double support, one where the right leg is first (leading), and one in which the right leg is the trailing leg. The single support state is split up as well, with the left leg in the swing phase and the right one in its stance, and vice versa (see also figure 5-1). This method appears to be flexible, and it is believed that it can be easily extended to control figures with more than two legs. Each additional leg could be accounted for at the phase level and the program for the correct sequencing would be provided by the finite state machine. The dynamic equations would stay the same, since all legs go through the same phases, just shifted in time. Also, by the same token, different gaits are merely a timing problem, as indicated in figure 3-2 (b) for walking and running. In both gaits, the left and right legs occupy stance and swing phases represented by the same sets of equations of motion. It is the relative timing, and the degree to which the phases of the two legs overlap, that determines a gait. As the time for double support (stance

phases overlap) decreases in walking and eventually vanishes due to an increase in step frequency, the gait changes to running. If the stance phases of the two legs move further apart, the time when both legs are in the air becomes larger (the swing phases overlap). Thus, although in walking there is a double support state, whereas running has a flight state, the individual legs still follow the stance-swing-stance cycle (see also section 5.1).

The principle of sensory feedback, as introduced above, has been applied to the control structure in two places: at the state level and the phase level. A locomotion step is defined as one double support plus one single support state. Before each step, the deviations between the current and the desired status of the system determine the new configuration for the impending step. This might require a change in step length, step frequency, or the rate of acceleration or deceleration of the figure. Of course, this type of feedback becomes effective only when a change from the rhythmic pattern is expected, in particular at the beginning, during start from rest, and the end of a locomotion sequence, when coming to a halt. Feedback at the phase level is provided during the swing phase. If the figure was about to stub its toe or heel while trying to meet a desired specification, the leg is raised just enough to clear the ground. This is explained in more detail in chapter 6.

3.3. The Refined Approach



In the previous section, a rough outline of the hierarchically organized algorithm to animate the locomotion of legged figures was presented. The underlying idea is unique in that an attempt has been made to tie together a high-level, goal-directed control and dynamic analysis. The approach is different in principle from a pure dynamic method [Armstrong 85, Wilhelms 85], where the user is forced to specify movements nonintuitively in terms of forces and torques. It also goes beyond a straight goal-directed system like SA [Zeltzer 82b], in which, indeed, a task like walking gets decomposed step by step, but at the lowest level, the output, that is the actual joint angles over time, are obtained from clinical data and interpolation techniques. This is a bottleneck that makes it impossible to easily produce variations of a walk. The interpolations, in a sense, have here been replaced by the dynamic equations. The dynamics create some sort of an implicit control by autonomously performing elementary movements. The rationale behind the approach as indicated in figure 3-2 is really twofold. There has been an effort to imitate, to some extent, how living vertebrates control locomotion (hierarchical structure), and to simulate how the motion actually gets executed (dynamics).

Figure 3-3 gives a more detailed picture of the control algorithm. It should be regarded as a further iteration of figure 3-2, now showing the information that has to be given or that has to be calculated at each

level. At the top, the animator specifies certain parameters. Besides simulation time, body height, and body mass, there are three fundamental *locomotion parameters* that largely determine the patterns of motion and gait: the desired forward velocity (v_{des}), step length (sl_{des}), and step frequency (sf_{des}). Their relationship is expressed by the following simple equation:

$$v_{des} = sl_{des} \cdot sf_{des} \tag{3.1}$$

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 KLAW, therefore, in addition to the locomotion parameters, the user may specify up to 28 *locomotion attributes* which individualize the basic locomotion pattern. Among these attributes are a stride width factor (lateral distance between the feet), the amount of toe clearance during swing and the maximum rotation of the pelvis. A complete list and explanation is given in appendix E.

After the specification of the parameters, the system does all the necessary computations and outputs the body angles as a function of time; these can then be taken to drive an animation of a legged figure. The essence of the algorithm lies in its two levels of abstractions. Bearing the hierarchical structure in mind, these two abstraction processes could also be viewed as a top-down, stepwise application of high-level concepts and a bottom-up, physical construction of the necessary elements for legged locomotion; they are joined in the middle by the gait-state-phase reduction mechanism already discussed above.

The conceptual level contains a few gait-specific rules or laws. They are utilized to transform the locomotion parameters into some gait-phase constraints which are fed to the low-level control to "guide" the dynamic simulation. In more concrete terms, the high-level control box performs the following tasks: first, since at least one of v_{des}, sl_{des} and sf_{des} have to be specified by the animator, the system completes (if not all three parameters are input) and checks the locomotion parameters by the principle of the *normalization formula* and equation (3.1). At the same time, depending on sl_{des} and sf_{des}, the gait (walking or running) is determined. Subsequently, the application of the *symmetry of steps* concept and the evaluation of *state-phase timing* functions yield the times and final angles for the stance and swing phases (see chapter 5). This information is passed to the low-level control, where it forms the constraints for a special kind of boundary value problem [Burden 85], which is evaluated with an approach somewhat like a shooting method. That is, the equations of motion for the lower body are solved by interpolating the forces and torques until the constraints are satisfied. For example, the simulation of the swing leg 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 (see chapter 4).

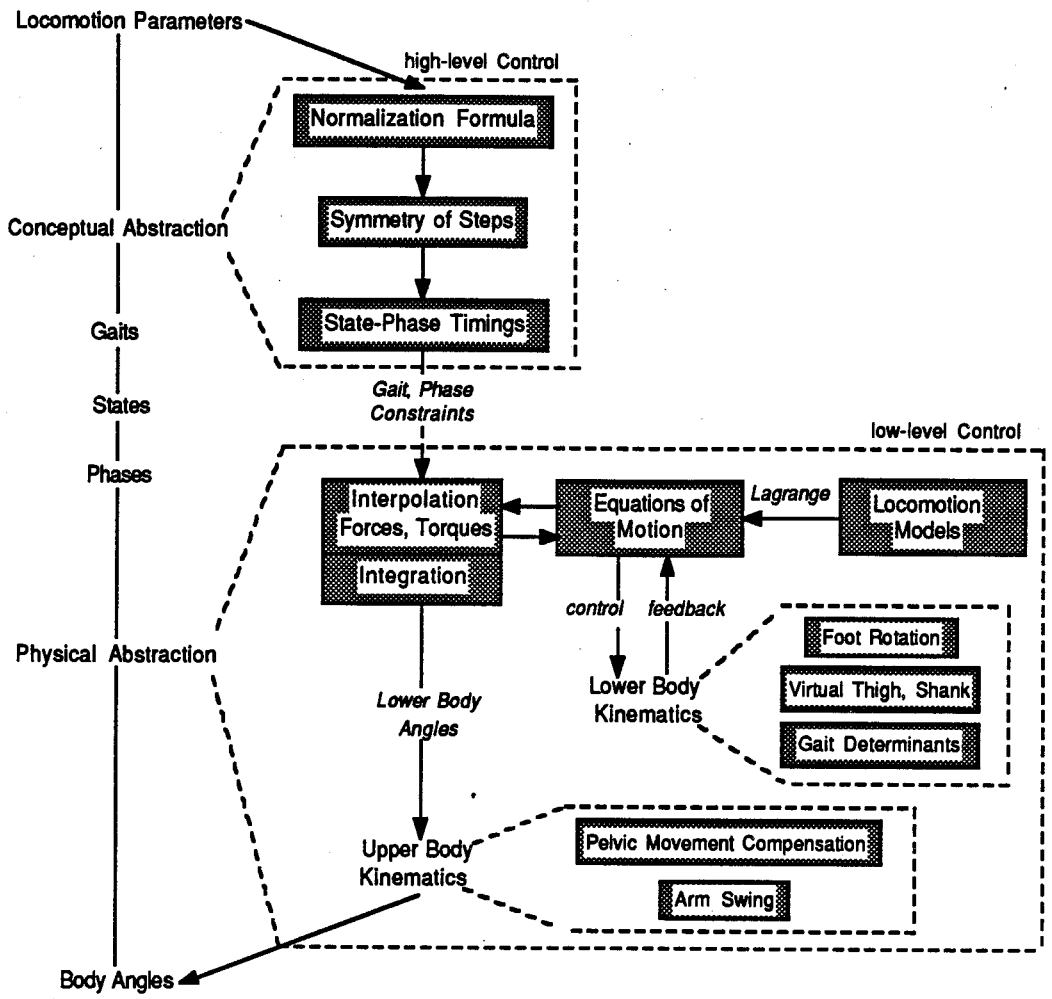


Figure 3-3: Schematic diagram of the control system for legged locomotion.

Although the implementation is quite complex, using efficient interpolation techniques, a solution is usually found in a few iterations. A principal objective is to keep the dynamics simple, otherwise this internal calculation of the forces and torques becomes infeasible. Since in legged locomotion, the activities in the legs play the dominant role and most of the motion is directed forward in the sagittal plane, a basic two-dimensional, dynamic model of the lower body (the upper body is represented by just a single segment) has been proven to be quite sufficient. Consequently, as long as there is at least one foot on the ground, the number of degrees of freedom amounts to five, and during the flight phase in running, the model is characterized by seven degrees of freedom (see appendix C.1).

The dynamic simulation produces the generic locomotion pattern which is visually upgraded by kinematic measures. The rotation of the foot is added to the dynamic legs, a thigh and shank are superimposed onto the telescope stance leg, and gait determinants like pelvic rotation or list get injected into the one-hip dynamic model. It is important to notice that, even though a considerable amount of kinematic "cosmetics" are applied, the dynamics are the very heart of the control for they guarantee the natural looking rotational movements of the legs plus smooth transitions between the phases and states. In a way, the equations of motion guide the lower body kinematics, but because both are executed simultaneously, the kinematic computations might, 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 resulting from the kinematic foot rotation in order to detect heel-strike properly at each time 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.

Once the angles are calculated for the joints of the lower body, the motion of the upper body is determined. This is done with the assumption that the upper body follows or depends on the lower body movements. The arms, for example, swing forward with the opposite legs. Thus, the angles of the arms, as well as the rotations in the shoulder and spine which compensate for the motion of the pelvis, are expressed as simple (at the moment linear) functions of the corresponding angles in the lower body. With this approach, a total of 56 angles for 37 joints of the body plus a position vector in space are computed for each time step (24 of these joints model the vertebrae in the spine; a complete list is given in appendix B.2).

One crucial aspect of the algorithm illustrated in figure 3-3 has not been mentioned in the above discussion: the whole top-bottom control structure is applied relative to a locomotion step. The constraints on the motion calculated by the high-level control are updated before each step. In this way, the system achieves autonomy in that it is able to adapt to changes in the locomotion parameters on a step to step basis. In other words, the granularity of the system is one step. (At the end of the previous section, this mechanism was indicated when discussing feedback at the state level).

In the following chapters, the elements in figure 3-3 are addressed in more detail. The next chapter introduces techniques related to dynamic analysis, in particular how the equations of motion are derived and solved. The concepts of legged locomotion like symmetry of steps, and the mechanics such as the determinants of gait are explained in chapter 5. Chapter 6, then, brings all these aspects together again while discussing implementational issues.

Chapter 4
Low Level Control Principles

4.1. Multilink Structure

For the analysis and composition of human movements, the body sections that generate a motion are usually modeled as a series of interconnected, rigid segments of constant length. Rotational joints represent the points of articulation between the segments. The orientation of each segment i is described by a homogeneous transformation matrix in 2-D as

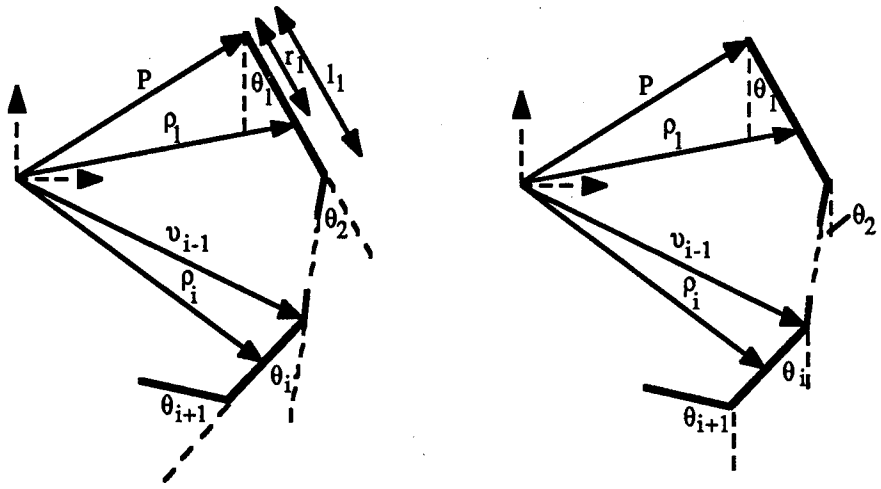
$$A_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The joint variables θ_i can be expressed in two ways: as *absolute* limb segment angles or as *relative* joint angles (figure 4-1). The former define the orientation of a segment independent from the other segments. This approach is used for the simulation of the stance phase (see appendix C.1) in order to specify the upper body angle with respect to the hip (which is the center of the whole body) rather than relative to the origin of the dynamic model, which is set to be the stance foot.

More commonly, however, the orientation of a segment within a multilink structure is specified relative to its parent in terms of joint angles. Each segment has its own local coordinate system (transformation matrix) with the property that changing any joint angle does not require a change in all the successive joint angles further down the chain. The absolute orientation of a segment i is calculated by $\prod_{j=1}^{i} A_j$.

To simulate articulated motion (as opposed to just animating it), the mass, center of mass, and the distribution of mass around the center of mass for each segment also have to be known. The 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[A] which are calculated as described in appendix B.1. It is submitted, that this simplification (i.e. to assume symmetry in segments) is justified for dynamic analysis in computer animation, since it has no significant effect on the motion.



a) joint angles b) limb segment angles

Figure 4-1: Examples of multilink structures.

Assume θ_i to be joint angles. The position vector ρ_i to the center of mass of segment i, as illustrated in figure 4-1 (a) is defined by

$$\rho_i = \upsilon_{i-1} - \left(\prod_{j=1}^{i} A_j\right)[0, r_i, 0]^T, \text{ where } \upsilon_{i-1} = P - \left(\sum_{j=1}^{i-1}\left(\prod_{k=1}^{j} A_k\right)[0, l_j, 0]^T\right) \quad (4.1)$$

and ρ_i = the proximal distance to the center of mass of segment i
l_i = length of segment i
P = the position vector connecting the body to the inertial (world) coordinate system
υ_{i-1} = the position vector to the distal end of segment $i-1$.

If the system is expressed by limb segment angles, ρ_i is calculated as follows:

$$\rho_i = \upsilon_{i-1} - A_i[0, r_i, 0]^T \quad \text{and} \quad \upsilon_{i-1} = P - \left(\sum_{j=1}^{i-1} A_j[0, l_j, 0]^T\right). \quad (4.2)$$

The ρ_i, including their derivatives, make up the translational motion of the centers of mass and are used in the process of deriving the equations by the variational principle.

4.2. Lagrange's Equations

The dynamic behavior of a physical system (in this case, of an articulated body) is defined by its equations of motion. In classical mechanics there are two fundamentally different approaches to derive these equations: *vectorial* mechanics and *analytical*, sometimes also referred to as *variational* mechanics. The former is a direct application of Newton's laws, which requires the individual components of a system to be analyzed in terms of vector quantities like force and momentum. The forces of constraint, such as joint forces, have to be explicitly calculated.

The analytical approach, on the other hand, provides a more general study of motion which is of higher mathematical abstraction. A system is considered as a whole by means of two scalar quantities: the kinetic energy and the work function. Constraints due to the structure of the system are determined implicitly, i.e. they are automatically included in the equations. The entire set of equations of motion is developed from a unified principle, where the calculus of variations is employed to find the stationary value of a certain *action* integral. Equation (4.3) below basically represents the solution or necessary condition for this integral to have a stationary value. Details about the theory of analytical mechanics are provided in [Wells 67] or [Meirovitch 70]. This mathematical approach is entirely independent of any particular reference system, and therefore it is possible to adapt the coordinates to the specific structure of a model.

The analytical principle has been formulated in various ways, most importantly by D'Alembert, Hamilton and Lagrange. Although they are all equivalent, in that they were derived from Newton's laws and the concept of virtual work [Wells 67], it was decided to use Lagrange's formulation because it seems to be most sophisticated, convenient and simplest to apply as a system gets more complex.

The Lagrange equations for a system with n degrees of freedom can be written as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_r}\right) - \frac{\partial L}{\partial q_r} = F_{q_r} \quad , \text{where} \quad r = 1, 2, \ldots, n \quad \text{and} \tag{4.3}$$

L = Lagrangian = $T - V$
T = Kinetic Energy
V = Potential Energy
q_r = Generalized Coordinate
F_{q_r} = Generalized Force .

By determining the kinetic and potential energies for a particular mechanical system and differentiating the Lagrangian according to equation (4.3), an ordinary, second order, nonlinear set of

differential equations is obtained that completely determines the motion of the system. There are exactly as many equations of motion as degrees of freedom. This coincides with the number of generalized coordinates q_r, which are defined as the minimum number of independent coordinates required to describe a system. The q_r can be angles, lengths, areas, etc., and there may exist more than one valid set of generalized coordinates for a given system. For the dynamic analysis of multilink chains, the joint variables θ_i are commonly chosen to be the generalized coordinates. Of course, in the simulation of the stance phase the length of the telescope leg changes and has to be accounted for by an additional generalized coordinate ω, as seen below.

The meaning of the generalized forces is directly dependent on the choice for the generalized coordinates. If q_r is a length, then F_{q_r} represents a force; if q_r is an angle, F_{q_r} stands for a torque. In general, a generalized force is a quantity such that the term $F_{q_r} \cdot \delta q_r$ (δq_r is an infinitesimal virtual displacement in q_r consistent with the system constraints) is the work done by the *applied* forces, while all other generalized coordinates are kept constant, i.e. only q_r varies. It is important to realize that the generalized forces only contain contributions of the applied forces, whereas forces implicit to the model (expressions like $m_2 \ddot{\omega}$, $m_2 \omega \dot{\theta}_1^2$ in the example below) are taken care of by the left side of equation (4.3); forces of constraint are not part of F_{q_r} either and, if necessary as in the case of nonholonomic constraints below, they have to be included explicitly.

As an example, the stance phase model is explained here briefly. A full illustration and derivation of the equations of motion is provided in appendix C.1 (the swing phase model is set out in C.2). The generalized coordinates are $q_r = \{ \omega, \theta_1, \theta_2 \}$. They are minimal, independent and sufficient to totally describe the system. The position vectors ρ_i of the centers of mass are expressed as functions of the generalized coordinates $\rho_i = \rho_i (q_1, q_2, \ldots, q_n)$ according to equations (4.1) and (4.2), rather than being generalized coordinates themselves, which would violate the minimum criterion. For instance, the cartesian coordinates x_2, y_2 of ρ_2 are calculated as follows,

$$x_2 = x_2 (\omega, \theta_1, \theta_2) = x + \omega \sin \theta_1 + r_2 \sin \theta_2$$

$$y_2 = y_2 (\omega, \theta_1, \theta_2) = y + \omega \cos \theta_1 + r_2 \cos \theta_2 , \quad \text{where } x, y \text{ are constants.}$$

The Lagrangian for the stance phase is given by

$$L = T - V = \tfrac{1}{2} m_1 \dot{\rho}_1^2 + \tfrac{1}{2} m_2 \dot{\rho}_2^2 + \tfrac{1}{2} I_1 \dot{\theta}_1^2 + \tfrac{1}{2} I_2 \dot{\theta}_2^2$$

$$- m_1 g (y + r_1 \cos \theta_1) - m_2 g (y + \omega \cos \theta_1 + r_2 \cos \theta_2) .$$

With this, differentiation of equation (4.3) yields a system of three equations of motion, where the generalized force F_ω is a force along the leg axis, F_{θ_1} and F_{θ_2} represent torques about the foot and the hip for the upper body, respectively:

$$F_\omega = m_2 \ddot{\omega} - m_2 \omega \dot{\theta}_1^2 - m_2 r_2 \ddot{\theta}_2 \sin(\theta_2 - \theta_1) - m_2 r_2 \dot{\theta}_2^2 \cos(\theta_2 - \theta_1) + m_2 g \cos\theta_1$$

$$F_{\theta_1} = (I_1 + m_1 r_1^2 + m_2 \omega^2)\ddot{\theta}_1 + 2 m_2 \omega \dot{\omega} \theta_1 - (m_1 r_1 + m_2 \omega) g \sin\theta_1$$
$$+ m_2 r_2 \omega \ddot{\theta}_2 \cos(\theta_2 - \theta_1) - m_2 r_2 \omega \dot{\theta}_2^2 \sin(\theta_2 - \theta_1)$$

$$F_{\theta_2} = -m_2 r_2 \ddot{\omega} \sin(\theta_2 - \theta_1) + m_2 r_2 \omega \ddot{\theta}_1 \cos(\theta_2 - \theta_1) + (I_2 + m_2 r_2^2)\ddot{\theta}_2$$
$$- m_2 g r_2 \sin\theta_2 + 2 m_2 r_2 \dot{\omega} \dot{\theta}_1 \cos(\theta_2 - \theta_1) + m_2 r_2 \omega \dot{\theta}_1^2 \sin(\theta_2 - \theta_1)$$

It should be noted that the Lagrangian approach, as stated above, is only applicable to *holonomic* systems. This requires that the coordinates q_r are independent, which is assumed by the definition of generalized coordinates. If constraints are present, they must be expressible in a simple algebraic form $\dot{x}_i = x_i(q_1, q_2, \ldots, q_n)$, so that superfluous coordinates can be eliminated directly (see also degrees of constraint[A]). Holonomic constraints reduce the degrees of freedom and consequently the number and complexity of the equations of motion as well. For completeness, although there is no relevance to this application, it is mentioned that the problem of *nonholonomic* constraints (as long as they are of the type $\sum_{r=1}^{n} a_r \cdot \delta q_r = 0$, where a_r are functions of the coordinates q_r), which can not be integrated, may be solved by the method of Lagrange multipliers [Meirovitch 70]. The forces of constraint are now explicitly specified in equation (4.3) by adding the term $\lambda_c \cdot a_{cr}$ to the right side for each constraint c.

It should be understood that the Lagrange equations or any of the other approaches, including the direct application of Newton's laws, are merely a means to construct an eventually equivalent set of equations of motion for a given system. The problem of solving the equations is addressed in the following section.

4.3. Integrating the Equations of Motion

The equations of motion can be solved in two ways. The method of Inverse dynamics calculates the forces given that the motion of a system is specified. If q_r, $\dot{q}_r$ and $\ddot{q}_r$ are known over the duration of the motion, this amounts to the algebraic evaluation of a discrete set of equations for each time step, where the solution at time t has no effect on the equations at time $t + \Delta t$.

Forward dynamic analysis is the determination of the motion of a system given the forces; they involve integrating a coupled system of second order, nonlinear differential equations. The solution process is continuous, in that the output from one time is the input to the equations at the next time step. In this thesis, forward dynamics were applied to calculate the joint angles over time. Interpolation techniques were employed, as discussed in a subsequent section, to approximate the force and torque profiles, that produce a desired locomotion cycle, but are unknown *a priori*. This approach should not be confused with inverse dynamics. It resembles more a boundary value problem since only the initial and final values for certain variables, as well as some timing constraints, are provided rather than predefining the complete motion (see section 4.5).

Because there are no analytical solutions, the equations of motion are integrated numerically. For this purpose, the equations are written in matrix form

$$A\,\ddot{q} = B(q, \dot{q}) \ ,$$

where A = inertia matrix ($n \times n$ matrix for n degrees of freedom)
q = solution vector
B = vector of transient terms (including the generalized forces) .

For example, the three equations of the stance phase as formulated in the previous section have the following matrix form:

$$A\,\ddot{q} = \begin{bmatrix} m_2 & 0 & -m_2 r_2 \sin(\theta_2 - \theta_1) \\ 0 & I_1 + m_1 r_1^2 + m_2 \omega^2 & m_2 r_2 \omega \cos(\theta_2 - \theta_1) \\ a'_{1,3} & a'_{2,3} & I_2 + m_2 r_2^2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{\omega} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

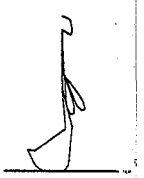
$$B(q,\dot{q}) = \begin{bmatrix} F_\omega + m_2\,\omega\,\dot{\theta}_1^2 + m_2\,r_2\,\dot{\theta}_2^2\cos(\theta_2-\theta_1) - m_2\,g\cos\theta_1 \\ F_{\theta_1} - 2\,m_2\,\omega\,\dot{\omega}\,\theta_1 + (m_1\,r_1 + m_2\,\omega)\,g\sin\theta_1 + m_2\,r_2\,\omega\,\dot{\theta}_2^2\sin(\theta_2-\theta_1) \\ F_{\theta_2} + m_2\,g\,r_2\sin\theta_2 - 2\,m_2\,r_2\,\dot{\omega}\,\dot{\theta}_1\cos(\theta_2-\theta_1) - m_2\,r_2\,\omega\,\dot{\theta}_1^2\sin(\theta_2-\theta_1) \end{bmatrix}$$

Note that due to the structure of the system, A is symmetric. Initially, the following simple combination of an explicit and implicit Euler method was used to integrate the equations:

$$\dot{q}(t+\Delta t) = \dot{q}(t) + \Delta t\,\ddot{q}(t)$$

$$q(t+\Delta t) = q(t) + \Delta t\,\dot{q}(t+\Delta t)\ .$$

After careful analysis of the resulting motion, it became obvious that the method was only reasonably stable if the initial conditions ($q, \dot{q}, \ddot{q}$) were kept close to zero and no external forces were applied, i.e. the system was conservative. This was not acceptable since a major aspect of the locomotion algorithm requires the presence of controlling forces to achieve a variety of movements. The instability of Euler's method was somewhat expected because of the several springs and dampers in the system which cause the problem to be *stiff* [Burden 85]. No formal analysis of the eigenvalues was done to check whether they lie within the stability region of the numerical method, which is to verify that the errors were introduced by the integration process and not due to an inaccurate dynamic model. Since the system is nonlinear, the eigenvalues are not constant and would have to be numerically calculated.



For simulating the motion of a multilink system, an *A-stable* method [Burden 85] such as LIMEX [Zugck 84] or LSODI [Hindmarsh 80] is preferred, that has a large stability region covering the entire left half of the complex plane. The LSODI integration package was eventually chosen, which produced reliable results for this thesis. LSODI solves the initial value problem for an ordinary system of first order differential equations. It was therefore necessary to transform the second order equations $A\,\ddot{q} = B(q,\dot{q})$ into a first order representation $A'\,\dot{u} = B'(u)$, where the solution vector u is of the form [Burden 85]

$$u_{2i-1} = q_i$$
$$u_{2i} = \dot{u}_{2i-1} = \dot{q}_i \quad \text{for} \quad i = 1,2,\ldots,n\ .$$

This transformation essentially doubles the number of equations (see appendix C for stance and swing phase). Frequently, standard integration techniques require the specification of the *Jacobian* matrix J (with $j_{ik} = \frac{\partial b'_i}{\partial u_k}$; $i,k = 1,2,\ldots,2n$) of the transient terms. LSODI has to be supplied with the full Jacobian matrix defined by

$$j_{ik} = \frac{\partial R_i}{\partial u_k}, \quad \text{where} \quad R_i = b'_i - \sum_{l=1}^{2n} a'_{il} \cdot s[l]$$

and the vector s is an internally generated approximation to $\dot{u}$.

4.4. External Constraints

The motion of an articulated body (in general, any physical body) is a result of its interaction with the environment and therefore subject to external constraints. For legged locomotion, the principal constraint is imposed by the ground. It enables the body to transform its rotational movements around the joints of the lower extremities into an overall translation in space. There have been different approaches in human motion simulation as to how the constraints should be mathematically formulated and included into the model.

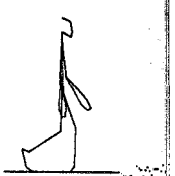
Wilhelms [Wilhelms 85] makes use of *springs and dampers* to confine the motion of body segments. Consequently, an artificially stiff system is induced that might lead to integration errors. This is especially the case for simulating ground contact, where large spring and damping constants are required to keep the foot, which carries the whole body weight, on the ground. Although this method is not employed here to express external constraints, springs and dampers are introduced at two places to maintain internally desired states: they control the torque values F_{θ_2} to balance the upper body, and F_{θ_3} at the end of the swing phase to keep the desired hip angle fixed until heel strike(see chapter 6).

A more appealing approach to the application of external constraints is to *analytically constrain* the motion of the model by reducing it to a subspace of smaller dimension. The ground constraint during the stance phase was represented this way. Since the foot of the dynamic stance model does not move (the added kinematic foot does as explained in section 5.4), only three degrees of freedom ($\omega, \theta_1, \theta_2$) are necessary to describe the system. If, on the other hand, the constraint was relaxed, as during the flight phase in running, the changing position of the foot would have to be accounted for by two additional degrees of freedom (in total, $x, y, \omega, \theta_1, \theta_2$). In general, this approach involves the derivation of two sets of equations for the system, one for which the constraint is active and one for which it is inactive. Unlike the spring and damper method, constraints here do not complicate the actual integration process because the extra amount of work is done beforehand when deriving the equations.

Legged locomotion can be characterized as a cyclic activity where each leg follows a strict alternation of stance and swing phases (see section 5.1). In terms of ground constraints, this means that they

have to be "turned" on or off with every phase change; using the above technique, this is identical to switching between equations of motion. In order to produce smooth transitions, i.e. a continuity in movements across phases, the equations of a new phase have to be initialized properly. In particular at heel strike, which occurs between the swing and the stance phase, some extra measures have to be taken to correctly capture the abrupt change in the motion of the leg. The problem can be expressed as a mechanical collision between the leg and the ground. For this purpose, the simplifying assumption is made that the support is transferred instantaneously at impact from the hind to the front leg, even though in bipedal walking the body weight is really shifted between the legs over the duration of the double support phase (in running, this assumption would not be necessary since the body is airborne before heel strike). It is also assumed that the spring in the leg of the stance model (see figure C-1) is massless, so that for an instant after ground contact, there is no reaction force (or velocity) along the leg axis and therefore the linear momentum P_ω is conserved during the collision. Since the angular momenta of the foot $L_{f_{st}}$ and the hip $L_{h_{st}}$ of the stance leg are conserved as well, we can write

$$L_{f_{st}}^- = L_{f_{st}}^+$$
$$L_{h_{st}}^- = L_{h_{st}}^+$$
$$P_{\omega_{st}}^- = P_{\omega_{st}}^+ ,$$



where $^-$ means before, $^+$ after impact. From these equations (for details see appendix C.3) the three velocities after impact $\dot{\omega}^+, \dot{\theta}_1^+, \dot{\theta}_2^+$ can be calculated and input as initial conditions (together with $\omega, \theta_1, \theta_2$) into the subsequent stance phase.

The formulations of the linear and angular momenta above imply that the physical models before and after heel strike are described by the same coordinates; but this is not the case, since the swing leg was defined earlier as a double pendulum in terms of θ_3 and θ_4, and not by ω and θ_1 (see appendix C). However, as will be discussed in detail in chapter 6, the knee of the swing leg is extended and "locked" shortly before heel strike. Therefore, the model of the swing leg is transformed into a single pendulum of fixed length l and hip angle θ_{34} (which can be substituted into the above linear and angular momenta equations for ω and θ_1, respectively). The extension of the knee between the double and single pendulum subphases is a collision that, analogous to above, determines the value of $\dot{\theta}_{34}$ (only one equation is needed, see appendix C.3) based on the conservation of angular momentum at the hip of the swing leg

$$L_{h_{sw}}^- = L_{h_{sw}}^+ .$$

The simulation of constrained motion and the application of the laws of collision, as approached in this section, are good examples of areas in computer animation where dynamics are clearly better than pure kinematic techniques.

4.5. Interpolation of the Generalized Forces

So far, the equations of motion have been presented as an initial value problem of the form $A'\dot{u} = B'(u)$ for every phase, given the initial condition $u(t_0) = \alpha$. However, this method can not be used to produce a realistic locomotion cycle, not to mention a variety of locomotion sequences. Although it is possible to apply different initial conditions as well as force and torque values, no mechanism is provided to enforce a desired motion. The problem is therefore redefined for each stance and swing phase:

$$A'\dot{u} = B'(u, F_{q_r}), \quad \text{subject to} \quad u(t_0) = \alpha, \; u(t_e) = \beta \quad \text{and} \quad t_0 \le t \le t_e \; . \tag{4.4}$$

Two new constraints are introduced specifying the values of the generalized coordinates at the end of a phase (final conditions) and the duration of a phase. These constraints are determined for each instance of a phase by the higher level control principles as explained in subsequent sections. The generalized forces F_{q_r} are now expressed as independent variables of equation (4.4). If the general solution to the problem is written as

$$u = f(t, F_{q_r}, C),$$

then F_{q_r} and the constant C are calculated by

$$u(t_0) - f(t_0, F_{q_r}, C) = 0$$
$$u(t_e) - f(t_e, F_{q_r}, C) = 0,$$

which is a classical *root-finding problem* where the roots F_{q_r} are approximated by interpolation techniques.

In practical terms, the objective is to find the proper forces or torques such that, given the initial conditions $u(t_0)$, the system reaches the final conditions $u(t_e)$ in exactly time t_e. As an example, the stance phase is considered here: the initial conditions come from the preceding swing phase and the collision laws (as already discussed). The final conditions are the hip angle θ_{1_des} and the length of the leg ω_{des} at time t_{stance}. The equations of motion are now integrated over the duration t_{stance} in a loop by modifying the hip torque F_{θ_1} and the leg axis force F_w until the final conditions are met.

The interpolation of F_{q_r} 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 (see [Burden 85]). The upper and lower bounds for F_{q_r} to start the Bisection method are determined by the following simple algorithm, which ensures that there is a root between the bounds and that they are no more than `inc` apart; it is shown for the stance phase, where F represents the hip torque:

```
F = F_init;
upper = lower = false;
while( !upper || !lower )
   integrate equations from t = t0 to t = tstance;
   if ( θ1(tstance) < θ1_des(tstance) )
      lower = true;
      F_lower = F;
      F = F + inc;
   else
      upper = true;
      F_upper = F;
      F = F - inc;
   endif
endwhile                                                    (4.5)
```

A solution to F_{q_r} using the above interpolation technique 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_r} profiles from one step (stance + swing) are carried over to initialize the next (`F_init` above). It should be noted, though, that some of the F_{q_r} are actually represented by functions of time and other parameters, that is, they are not applied uniformly over the duration of a phase. This aspect is dealt with in chapter 6. (Aside: from a neurophysiological viewpoint, the process of forwarding the leg torques F_{q_r} from one step to the next could be considered to represent the muscle memory of the lower level nervous motor program for walking as described in section 3.2.)

Chapter 5
High Level Control Concepts

In the last few sections, the study of legged locomotion focused on a simple dynamic model, whose internal structure was specifically designed to naturally support bipedal walking. The equations of motion, therefore, can be regarded as the functional synergies of the motion control hierarchy in figure 3-2, that are capable of executing a definite range of locomotion patterns under the control of specialized motor programs. Each motor program "knows" about a particular action. To this extent, the discussion of the locomotion cycle in the subsequent sections and the conclusions drawn essentially symbolize the motor program for bipedal locomotion.

Another analogy might be opportune: if one were to view walking as a programming language, virtually compiled by the control algorithm devised in this thesis, the dynamics in the preceding sections could represent the *syntax* of the motion, whereas the walking *semantics* would be inherent to the conceptual understanding or knowledge of the locomotion cycle. In the following, then, a "meaning" is given to the dynamics by bringing the two aspects together. In particular, it is shown how the conditions for each phase can be derived to solve the problem as formulated in equation (4.4), i.e. how to guide the dynamic model. Also, knowledge of the leg kinematics is utilized to visually upgrade the basic locomotion pattern which is generated by the dynamics.

5.1. Locomotion Cycle Characteristics

Generally, a sequence of animal locomotion can be subdivided into three stages: a short *developing* period characterized by an increase in speed, followed by a dominant *rhythmic* period where the forward velocity is kept fairly constant, and a brief *decay* period to decelerate the body. Almost all research on locomotion concentrates on the rhythmic phase of a gait on level surface; very little is known about how the basic pattern of each individual subject changes under the influence of acceleration, deceleration, or by going up and down inclines. Since rhythmic locomotion is just a series of recurring movements with the natural period of one stride (locomotion cycle) for every gait and any number of legs, it is sufficient to just direct investigations, as well as present kinetic and kinematic results, relative to one cycle.

The human walking cycle has been thoroughly studied. A good reference is a book by Inman [Inman 81], which includes a lot of experimental data from various authors. For bipedal walking or running, a locomotion cycle consists of two steps. As long as a symmetric gait (see gait[A]) is assumed, where the left and right leg perform the same movements, just out of phase, the principal unit of locomotion can be reduced to one step. There are two independent parameters (expressed in terms of the step unit) to define a specific instance of a gait: step length (sl) and step frequency (sf). Together with their product $sl \cdot sf$ which is the speed of the locomotion (v), they form the three locomotion parameters as introduced in section 3.3. Walking is possible at a wide variety of combinations of sl and sf. However, a person, when asked to walk at a particular velocity, is most likely to naturally "choose" the parameters which minimize energy expenditure. This observation is expressed in the experimentally derived equation [Inman 81], called *normalizing formula*, which shows a linear relationship between sl and sf, where sl and *body_height* are measured in m, sf in $steps/min$:

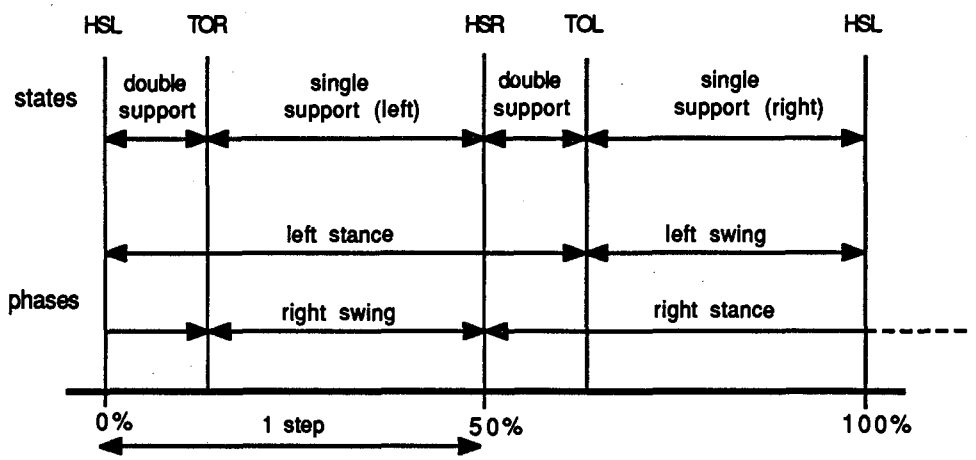
$$\frac{sl}{sf \cdot body\_height} = 0.004 \quad \Leftrightarrow \quad sf^2 = \frac{v}{0.004 \cdot body\_height}, \quad \text{because of} \quad sl = \frac{v}{sf} . \qquad (5.1)$$

The *body_height* normalizes the equation. It indirectly represents the length of the legs, which, of course, have an effect on the preferable step length. As stated in [Inman 81], equation (5.1) is based on data almost exclusively obtained from male subjects; women tend to walk with a shorter step length and greater step frequency at a given speed. Also, the normalizing formula applies only up to a certain step frequency sf_{norm} of about $132\,steps/min$ (which corresponds to $sl_{norm} \approx 0.528 \cdot body\_height$), if the speed is increased after that, the step length is kept constant. Another obvious condition to maintain a walking cycle is that neither of the locomotion parameters is allowed to exceed a maximum value ($sl_{max} = 0.6 \cdot body\_height$, $sf_{max} = 182\,steps/min$, $v_{max} = sl_{norm} \cdot sf_{max}$), otherwise running would result. Although the limits on the locomotion parameters as specified are quite constant for different subjects, they can be changed by the user for experimental purposes as part of the locomotion attributes[E].

Given equation (5.1) and the above constraints, an algorithm (for details see appendix D) is now designed to supplement the remaining locomotion parameters, 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 step length and step frequency are specified (assuming these as well as the resulting velocity are within the anatomical limits), a more angular motion might result.

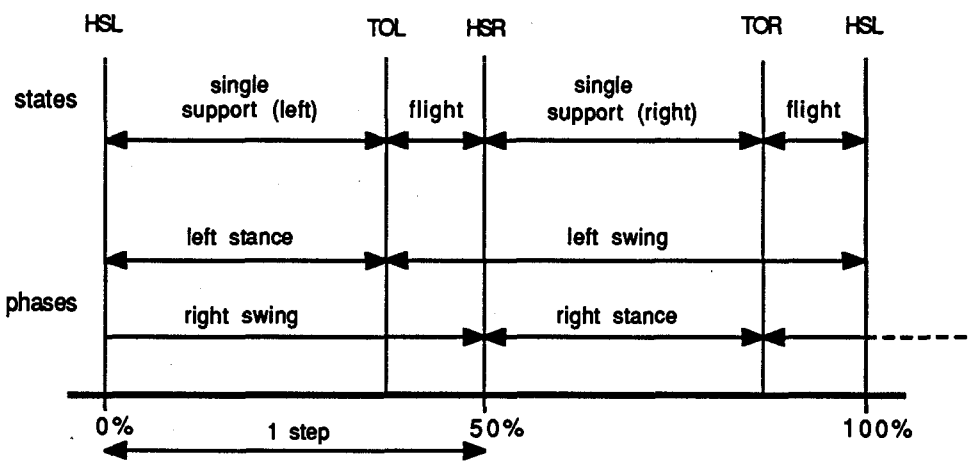
Once the locomotion parameters are complete, the step frequency is used, as shown below, to derive the *state-phase-timings*, whereas the step length is applied to the *symmetry of steps* concept (section 5.2) in

order to compute the final conditions for each phase of the current step. Hence, the two, up to now still undetermined ingredients to the central problem stated in equation (4.4) are deduced.



a) walking cycle

HSL = heel strike left
TOR = toe off right
HSR = heel strike right
TOL = toe off left



b) running cycle

Figure 5-1: Locomotion cycles for bipedal walking and running.

As already mentioned, a walking cycle consists of two steps. From figure 5-1 (a), it can be seen that each step contains one double support state (ds) where both feet are on the ground, and one single support state (ss) where one foot is off the ground. Formally, assuming t to denote a duration, this can be written as

$$t_{step} = t_{ds} + t_{ss} .$$

In terms of the individual legs, which go through the same stance-swing-stance cycle, just shifted in time, the following holds:

$$t_{step} = t_{stance} - t_{ds} \qquad (5.2)$$
$$t_{step} = t_{swing} + t_{ds} .$$

Experimental data gathered by different authors (see [Inman 81]) suggest a linear relationship in walking between the step frequency (sf) and the double support state as a percentage of a cycle. After interpolating the results from the different experiments, the time for the double support (t_{ds}) amounts to

$$t_{ds} = (-0.16 \cdot sf + 29.08) \cdot t_{cycle} / 100 . \qquad (5.3)$$

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

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

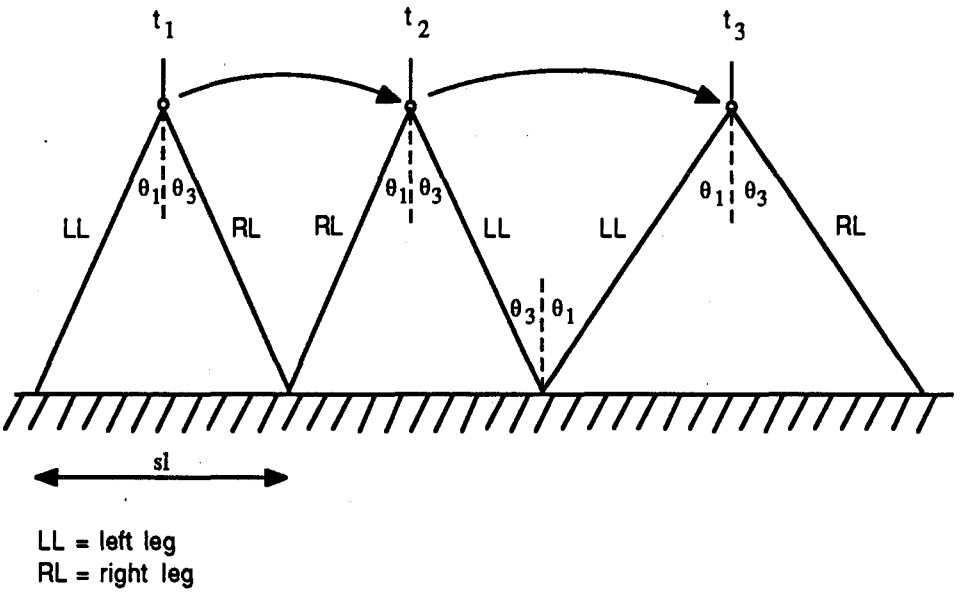
t_{ds} can be determined, and consequently the values for t_{stance} and t_{swing} are obtained from equation (5.2).

The above calculations form the basis of the state-phase-timing concept. Although not directly relevant, a few remarks on walking vs. running (see figure 5-1) pertain to reveal similarities between the two gaits: equation (5.3) expresses the fact that with increasing step frequency, or speed in general, the time for double support decreases in walking. At the phase level, this means that with bigger sf the amount of overlap between the stance phases of the legs is diminishing. When sf reaches about $182 steps/min$ (sf_max), the double support state disappears. At this point, walking is equal to running. If the step frequency increases further, t_{ds} in equation (5.3) becomes negative, which signifies the flight state, where both legs are off the ground; in other words, the swing phases start to overlap. Abstractly speaking, the two diagrams of figure 5-1 are quite similar. In going from walking to running, the right vertical bar indicating the end of double support is just "slid" to the left, and as soon as TOL comes before HSR (i.e. the left leg enters the swing phase before the right leg terminates its swing phase), the gait changes to running. Thus, if there exists a relationship between the step frequency and t_{flight}, running could be implemented by the same technique. Of course, running is not just possible when the sf exceeds sf_max, but generally, can also take place for the

same locomotion parameters as walking. In figure 5-1, both diagrams are based on the same time for a step, that is the same step frequency, and reasonably so, equal step length. To cover all of these possibilities for "artificial" running, additional information would be required such as the relationship between the maximum vertical distance between the foot and the ground during flight and t_{flight}.

5.2. Symmetry of Steps

If one were to have stiff legs (no knee joints) and no feet, the only way to move forward would be by means of a compass gait as shown in figure 5-2. For planar motion, the leg would drop below the ground during its swing phase. An interesting fact about this simplest form of bipedal locomotion is step symmetry. 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, and only depend on the step length sl. Most importantly, this still holds true when the body is accelerating or decelerating, which is indicated in figure 5-2 by the increased step length at time t_3 (i.e. the body accelerated from t_2 to t_3).



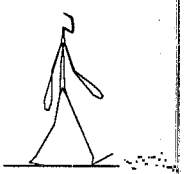


Figure 5-2: Symmetry of compass gait for different step lengths.

It should be noted that the concept of step symmetry introduced here is different from elsewhere in the literature [Raibert 86a], where symmetry is defined for the stance phase relative to the center of mass (which is located at the hip in the model). In this case, symmetry is only maintained if the step length does

not change, which means there is constant speed. Thus, as shown in figure 5-2, the angle θ_3 of the left leg at the beginning of the stance phase at time t_2 is not equal to the angle θ_1 at the end of the stance phase at time t_3; whereas θ_3 at time t_1 would be the same as θ_1 at time t_2 for the stance phase of the right leg.

The symmetry of step principle is now adapted to the model in figure 5-3 to determine ω, θ_1 and θ_3 at the end of a step (heel strike). Although the actual step configuration at heel strike is is not symmetric any more 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 really that the absolute value of θ_3 is smaller than it would be in the absence of a foot (it also holds that $\theta_3 < \theta_1$). In addition, the foot raises the position of the hip at impact, which has to be compensated by lengthening the telescope stance leg beyond its initial length, i.e. $\omega > l_1$. Details about controlling the changing length of the stance leg are given in chapter 4. The origin for the simulation of the stance leg stays fixed at $ankle_1$, which is at a distance l_9 above the ground. It is important to realize that the kinematic foot of the stance leg can be ignored at this point. Since it is superimposed on the motion of the dynamic leg after the simulation of the stance phase, it does not affect the calculations here, unlike the foot of the swing leg. The stance foot is explained as part of the separate concept of *virtual leg* in a subsequent section.

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[E]. In the actual implementation, a value for θ_4 (knee angle of the swing leg) at heel strike may also be selected to prevent the knee from fully extending. This aspect is disregarded below for simplification (i.e. $\theta_4 = 0$ and $l_{34} = l_1$). Given l_{34}, l_8 and l_9, $l_{11} = \sqrt{l_8^2 + l_9^2}$ and $\cos(\omega_8) = \frac{l_8}{l_{11}}$ are determined. The application of the cosine law yields

$$r^2 = l_{34}^2 + l_{11}^2 - 2\, l_{34}\, l_{11} \cos(\theta_5 + \omega_8) \quad \text{and} \quad \omega_3 = \operatorname{acos}\left(\frac{l_{34}^2 + r^2 - l_{11}^2}{2\, l_{34}\, r}\right).$$

Since

$$ankle_1 = (x_a, y_a) = (x_{nh'} + l_8, l_9), \quad \text{where } x_{nh'} \text{ is } x_{nh} \text{ from previous step, and}$$

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

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

it follows that

$$\omega = \sqrt{(y_h - y_a)^2 + (\frac{sl}{2})^2}$$

$$\theta_1 = \text{asin}\left(\frac{sl}{2\,\omega}\right)$$

$$\theta_3 = -\omega_3 - \text{asin}\left(\frac{x_{ah} - x_h}{r}\right). \qquad (5.4)$$

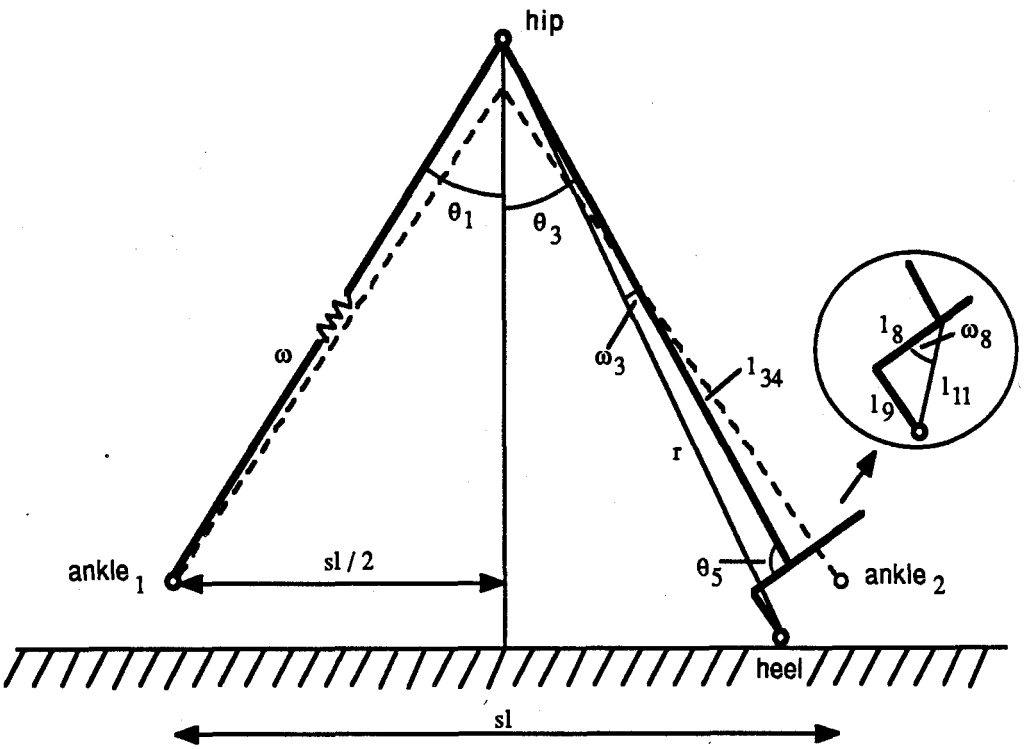


Figure 5-3: Dynamic model at heel strike: the swing leg is extended, a foot has been added kinematically; θ_3 is assumed to be negative, all other angles are positive.

One great advantage of step symmetry, as well as of the state-phase-timing concept from the previous section, is that all computations are done relative to one step which is the impending step in a locomotion sequence. Changes in the locomotion parameters *sl* and *sf* can therefore be accounted for with the "granularity" of one step. This allows for acceleration and deceleration of the motion, and even the extreme cases of starting from rest and coming to a full stop are possible (with minor adaptations of the underlying dynamics as shown in chapter 6). On the last step, for instance, *sl* is set zero (no "ground is

gained" since the hind leg only pulls even with the front leg), which, in the above calculations, results in the desired final conditions $\theta_1 = \theta_3 = 0$ and $\omega = l_1$.

However, one danger exists in treating each step separately: extreme changes in the locomotion parameters could cause large variations in the times and final conditions for successive steps, which, in turn, would produce torque and force profiles in the dynamic model that exceed the natural, anatomical limits of a human body. To avoid the problem, three locomotion attributes have been defined, v_{acc}, v_{dec} and sf_{accdec}, to restrict the amount of (constant) acceleration, deceleration or change in step frequency from one step to the next (the latter became necessary, since even for fairly constant velocity, an increase in *sf* can result in skyrocketing torque values). Further experimentation with the system is required to determine boundary values for these attributes that are "reasonable" for bipedal walking.

So far, all the necessary information to control the dynamic simulation of the leg phases has been derived. The next step is to concentrate on the details of the motion by adding kinematic, human-like features to it. Although the dynamics of the walking model are self-sufficient, including lower body kinematics can, in certain cases, have a direct effect on the dynamics. This was true above, where the addition of a foot to the dynamic swing leg changed the calculations of the final conditions for a step. Kinematics can also affect the dynamics during the actual simulation. A good example is the animation of the pelvis, which basically "splits" the dynamic hip into two, with the original hip remaining "on" the stance leg, while the motion of the hip for the swing leg is expressed as a function of the former. Due to *pelvic list* (to be discussed in the following section), the knee during the dynamic simulation of the swing phase now has to be flexed more to avoid ground contact. Therefore, the dynamics and lower body kinematics must be executed concurrently.

A last note is directed to running. Extending the step symmetry concept to this gait would involve the distance traveled during flight being subtracted from the step length *sl* before calculating the final conditions for a step.

5.3. Determinants of Gait

The net result of a locomotion sequence may be considered as a translational displacement in space. However, the motion, when analyzed in more detail, exhibits subtle deviations from the line of progression due to the rotational movements of the pelvis and the legs. During a locomotion cycle, the body rises and falls slightly, weaves from side to side, as well as speeding up and slowing down a bit, even if the overall forward velocity is constant. The displacement of the center of mass of the body, which represents the motion

of the body as a whole, is illustrated in figure 5-4 for one stride. The vertical displacement (A) prescribes a sinusoidal curve with a maximum at mid stance and a minimum at heel strike. This component of the motion is generated directly by the simulation of the telescope stance leg (see chapter 7 and appendix F). The curve (B) caused by the lateral motion of the center of mass is also sinusoidal at half the frequency of the vertical displacement. The lateral movements arise because the upper body is shifted over the respective weight-bearing stance leg.

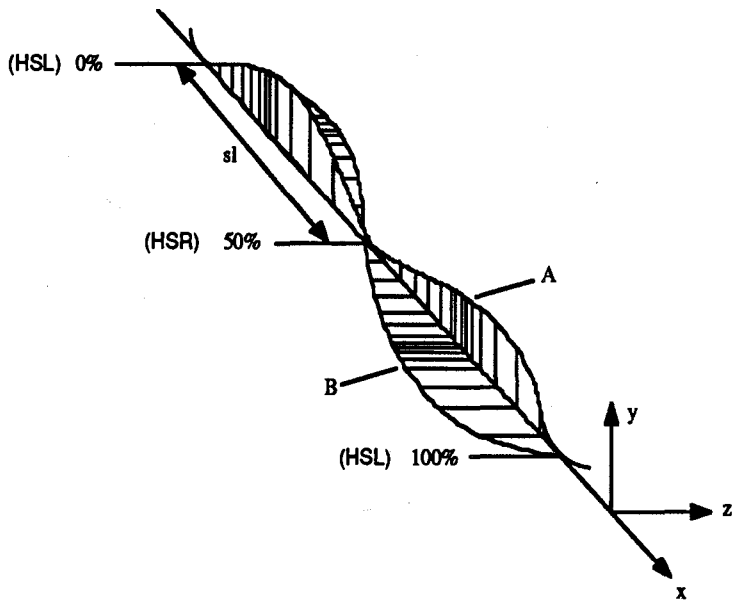


Figure 5-4: Exaggerated movements of the center of mass during one stride; different line spacing indicates a change in velocity (lines closer together mean deceleration, larger line spacing acceleration).

There is a direct correlation between the motion of the center of mass and energy expenditure in walking. Figure 5-5 shows the kinetic (translational and rotational), potential and total energy for half a stride. Employing the concept that the weight of the upper body is a passive element, which is carried on the active lower limbs [Inman 81], the energy levels of the HAT (head, arms and trunk) are displayed, rather than the energy profiles for the whole body. The potential energy curve is essentially identical with the vertical displacement of the center of mass (see figure 5-4), whereas the kinetic energy changes with the forward velocity of the center of mass. A maximum in potential energy, which coincides with a minimum in kinetic energy, occurs at about the middle of the single support period, while the potential energy reaches a minimum at the middle of the double support period, i.e. just after heel strike. At the same time, the kinetic

energy has a maximum value. The shapes of the two energy curves resemble quite well those of a conservative physical pendulum (see figure 5-6). However, unlike in the case of a swinging pendulum, the energy exchange is not complete. Even if the difference in magnitude between the kinetic and potential energies is neglected, the total energy shows substantial fluctuations. Walking therefore is a nonconservative activity, where the losses in energy have to be "topped off" with the work produced by the muscles.

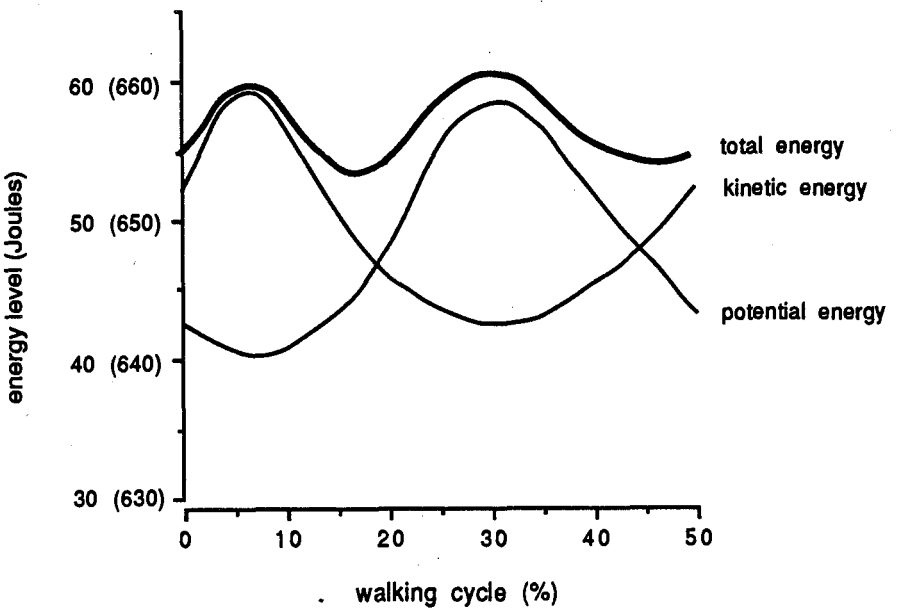


Figure 5-5: Approximations of kinetic, potential and total energy of the HAT for one walking step at average walking speed (adapted from [Inman 81]). The total energy is computed with the minimum potential energy level assumed at zero. The ordinate values in parentheses are for the potential energy.

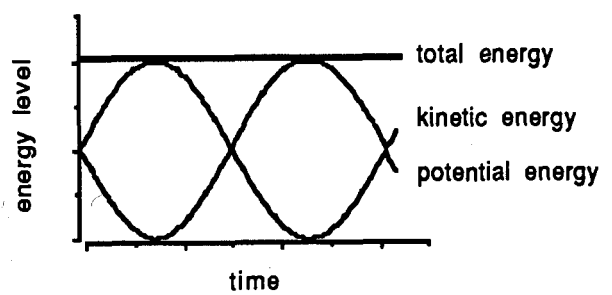


Figure 5-6: Exchange of kinetic and potential energy of a pendulum (neglecting friction). The total energy is constant, thus the system is conservative.

From the above, it is concluded that smaller deviations of the center of mass from the line of progression generate smaller fluctuations in the energy levels and consequently, the less energy is expended. During human walking, the absolute location of the center of mass stays within the region of the pelvis [Inman 81]. This is mainly achieved through the combined efforts of the *determinants of gait.* Inman [Inman 81] distinguishes between six determinants:

- pelvic rotation (transverse plane)
- pelvic list (coronal plane)
- pelvic tilt (sagittal plane)
- lateral displacement of body
- knee flexion of stance leg
- plantar flexion of stance leg ankle

Compared with the compass gait (already introduced in figure 5-2 as the simplest form of bipedal locomotion), where the pathway of the center of mass is through a series of arcs, one for each step, each of the above determinants smooths this path in one way or another to produce the sinusoidal curve of figure 5-4. Whereas the movements of the pelvis just flatten the arcs, the combination of knee and ankle motion alleviates the sudden changes in vertical displacement at heel strike, i.e. they remove the "tips" of the arcs by transforming them into a smooth curve. From this point of view, the determinants of gait contribute directly to minimize energy expenditure. For the purpose of animating human walking, the determinants greatly improve the realistic looks of the motion.

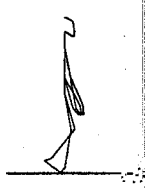
In the following, the determinants of gait are described in more detail. All determinants have been incorporated into the walking algorithm except pelvic tilt, which was neglected because it did not appear to have a perceptible visual effect in walking. Furthermore, the forward and backward tipping of the pelvis mainly affects the upper body, which is balanced independently anyway (see section 5.5). For the implementation, a kinematic pelvis and foot had to be added to the dynamic stance model. It is submitted that all the determinants can be expressed as linear functions, although experimental results [Inman 81] suggest a better approximation by higher order polynomials. This step is justified since the displacements generated by each determinant are relatively small (a few centimeters for the maximum values), so that the simplicity of a linear approximation outweighs the little qualitative improvement of a more complicated function. Various effects of the determinants on the motion are illustrated in appendix F.

Pelvic rotation about the vertical axis is a minimum at mid-stance and a maximum at heel strike. The rotation causes a larger step length and consequently, a greater radius for the vertical arcs of the center of mass. Since the step length, as defined by the locomotion parameter sl, includes the maximal displacement (from pelvic rotation) at heel strike, the actual step length used in the calculations of the step symmetry, denoted sl_act for now, needs to be adjusted for each step:

$$sl\_act = sl - \sin(pelvic_{rot}) \cdot l_0, \quad \text{where} \quad pelvic_{rot} = pelvic_{rot\_max} \cdot \frac{sl}{sl\_max}$$

and l_0 denotes the width of the pelvis (see appendix B.1); $pelvic_{rot\_max}$ is specified as one of the locomotion attributes[E]. For each step, the rotation of the pelvis is now linearly incremented until $pelvic_{rot}$ is reached at heel strike. This causes the hip of the hind leg (h_1) which is at a maximum behind the hip of the front leg (h_2) at heel strike, to move forward faster than h_2 (by the increments) during the double support and swing periods to eventually reach a maximum position in front of h_2 at the end of a step (see figure F-5).

Pelvic list is the rotation about the x-axis in figure 5-4 which makes the hip of the swing leg drop lower than the hip of the stance leg. As an effect, the center of mass does not need to be elevated so much and therefore the change in potential energy is reduced. A maximum angle occurs, i.e. the swing hip is lowest, just after toe-off, and thereafter the hip of the swing leg rises until it arrives at the same level as the hip of the stance leg at heel strike. This last characteristic of pelvic list, that is to equal the height of the hips at impact, is essential for the validity of step symmetry (which is based on a one hip model as shown in figure 5-3). A natural pelvic list is implemented as follows: after the simulation of the stance phase, the minimum height of the hip is computed, which comes shortly after heel strike (i.e. at the very beginning of the stance phase), and taken as the value for the swing hip at the end of the double support period (toe-off). The difference between this minimum hip value and the vertical position of the stance hip at toe-off yields the maximum angle for pelvic list for that step. As above, linear interpolation is employed to fill in all the remaining angles. A pelvic list factor[E] may be selected by the user to exaggerate or reduce pelvic list (see figure F-7).



The *lateral displacement* along the z-axis in figure 5-4 represents a shift of the weight of the body from stance leg to stance leg during a locomotion period (see also figure F-6). Moving the center of mass closer to the weight-bearing leg results in less effort being required to balance the upper body. The lateral displacement from the line of progression is a maximum at mid-stance and zero at the time of heel-strike (again, a necessary condition for step symmetry). There are two variables which influence the magnitude of lateral displacement, a stride width (sw) measuring the lateral distance between the feet (in m), and the step

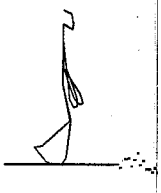
frequency (sf). A bigger stride width causes the body to rock more from side to side, and with increasing step frequency (usually synonymous with a greater speed), the lateral displacement decreases. The following relationship has been established for the maximum lateral displacement (lt_max) for a step from various experimental results [Inman 81]:

$$lt\_max = -0.00017241 \cdot sf + \frac{1}{6} \cdot sw + 0.0013448 \ .$$

A stride width factor is specified as a percentage of the pelvic width (l_0) in the list of locomotion attributes[E], from which sw is then determined.

The *knee* flexion of the stance leg together with the *plantar ankle* flexion of the stance foot, fulfill several functions. They serve as a major shock absorber at impact and smooth the transition of the stance leg from the double support period to the swing phase. In particular, the foot also retains the center of mass in a more elevated position at heel strike compared with a basic compass gait as was shown in figure 5-3. The implementation of these determinants is more ambitious and treated separately in the next section, since they involve the concept of a *virtual leg*.

5.4. Virtual Leg



The simulation of the stance phase based on the dynamic model of figure C-1 (a) generates the motion of an inverted, length-changing pendulum for one "step" under the given constraints for time and final conditions (it should be noted that the upper body is ignored in this discussion). What is desired, though, is the realistic motion of a human leg during the stance phase, which inherits the leg determinants described in the previous section. For this reason, a human leg may be superimposed onto the dynamic pendulum at each time increment as shown in figure 5-7. 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 and the task is to find the angles of the kinematic chain spanned between these endpoints. At least two of the four angles ($\theta_3, \ldots, \theta_6$) must be known to fully specify a particular configuration. In order to predetermine some of the angles so that the remaining ones can be determined unequivocally, additional information is required. In the present case, the information is supplied according to some general rules about the stance phase, which is therefore subdivided into a *normal*, *heel-off* and *meta-off* period. These periods are strictly successive in time and all three occur for any stance phase.

The *normal* period is illustrated in figure 5-8 (a). It characterizes the early state during stance,

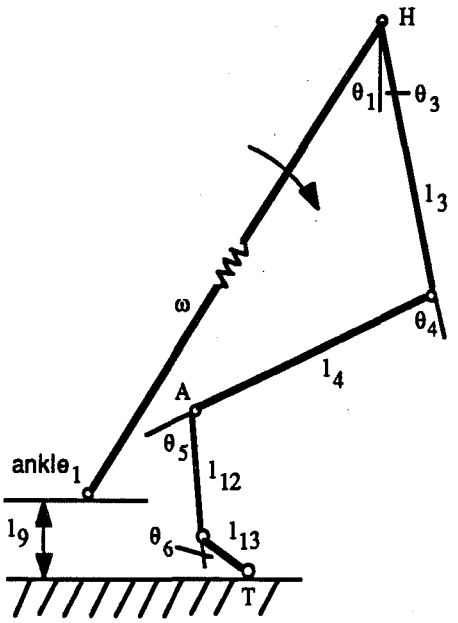


Figure 5-7: Superposition of a leg over the dynamic stance leg model. The size of the mid-foot (l_{12}) and toe (l_{13}) are exaggerated; the location of the hip (H) comes from the dynamic simulation, for which $ankle_1$ is assumed to be fixed at (x_a, y_a). The tip of the toe (T) maintains ground contact during the whole stance phase at position $(x_a + l_5 + l_6, 0)$; l_{12} and l_{13} are simply defined as $\sqrt{l_5^2 + l_7^2}$ and $\sqrt{l_6^2 + (l_9 - l_7)^2}$, respectively (see appendix B.1).

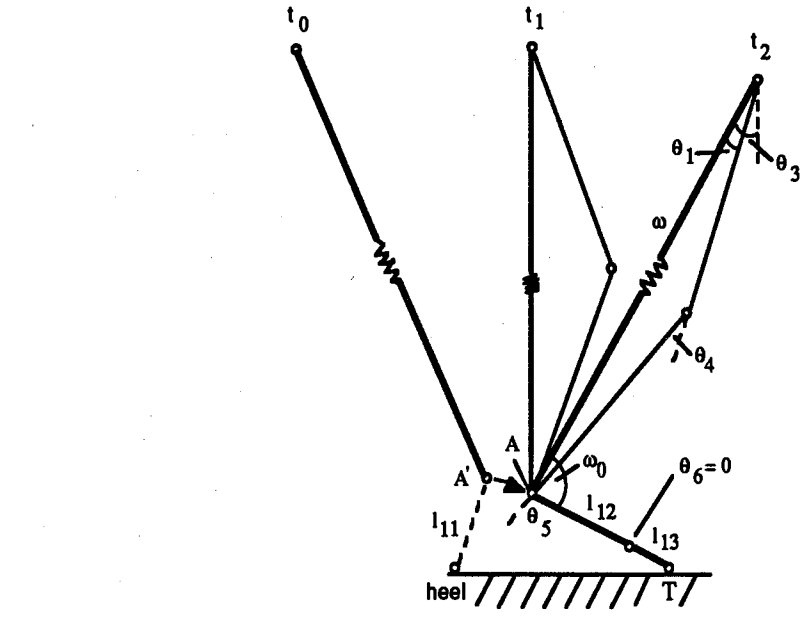
where, after heel strike at time t_0 the foot is rotated about the heel to come down onto the ground, and subsequently stays flat on the ground (e.g. at t_1) until some time after the leg passes through the vertical (that is, the center of mass moves in front of the body). To determine the leg angles, the algorithm makes use of the fact that the position of the ankle is always known, since the foot is assumed to rotate with constant increments on the circular arc $A' \rightarrow A$ with radius l_{11} (see also figure 5-3), and the ankle remains fixed at A once the foot is firm on the ground. Furthermore, this kinematic ankle is always located at the bottom end of the dynamic leg model (i.e. $ankle_1 = A$; this is not self-evident, as it is not the case in the meta-off period below). The knowledge of the location of the ankle splits the kinematic chain into two halves. In the upper half, θ_3 and θ_4, as shown at time t_2, are readily obtained by trigonometric calculations, given that θ_1 is known from the dynamic simulation. Because θ_6 stays zero during the whole period, θ_5 can then be determined. The ending condition for the normal period, which marks the beginning of the heel coming off the ground, is twofold: most frequently, the period is terminated as soon as condition ($c1$): $\omega_0 \leq \frac{\pi}{2}$ becomes

true ($c1$ gets obvious below in the discussion of the heel-off period). This is the case at t_2 and is identical to saying that the ankle slightly dorsiflexes, if $\theta_5 = \frac{\pi}{2}$ is assumed neutral, due to the superposition of the shank. There is an anatomical limit for the ankle dorsiflexion in walking (around 15°), individual deviations may be accounted for by adjusting the stiffness of the dynamic leg spring (which is a quite delicate task).

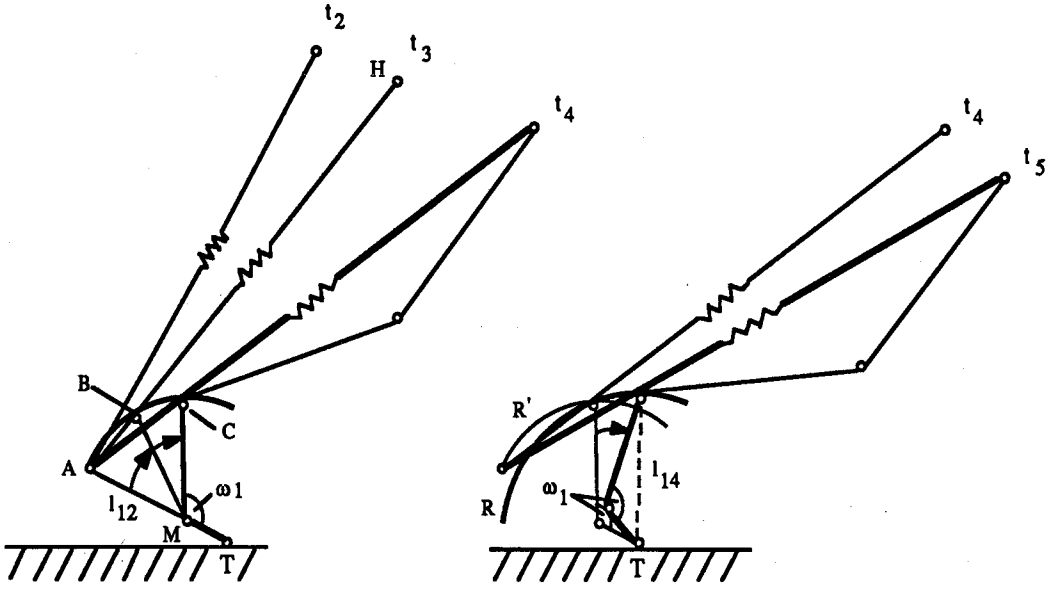
The normal period is also brought to an end if ($c2$): $\omega \geq (l_3 + l_4)$ is satisfied, in which case it becomes impossible to superimpose a thigh and shank with the foot flat on the ground. Condition $c2$ can come true in the presence of a stiff spring in the dynamic leg or for short step lengths. The latter is explained by the fact that the smaller the step length, the higher the hip at heel strike and consequently the more the dynamic leg must be lengthened (see also figure 5-3). In human walking, the leg actually never fully extends, i.e. θ_4 is not quite zero at time t_2. A locomotion attribute[E] has therefore been introduced to specify a minimum desired value for θ_4 at the end of the normal period; of course, the right side of $c2$ above is modified correspondingly.

During the *heel-off* period, the kinematic ankle travels on an arc with radius l_{12} around the metatarsophangeal joints (M, from here on denoted as the meta joint) while the toe remains on the ground, as shown in figure 5-8 (b). The position of the ankle on the arc is computed by intersecting the arc with the current dynamic leg, illustrated at time t_3 and t_4 where the intersections are B and C, respectively. Since θ_1 increases steadily with time, i.e. the dynamic leg is turning clockwise around A, the angle ω_1 is strictly decreasing (indicated by the arrows) and thus the distance between the heel and the ground is getting bigger and bigger, as desired. Unlike the situation in the normal period, the kinematic ankle is not fixed any more, but "travels up" the dynamic leg; with the position of the ankle known, the leg angles can be determined in the same way (of course, the meta angle θ_6 is now not zero any more).

Usually, the heel-off period is activated in the configuration shown at t_2, where condition $c1$ holds. This ensures that the dynamic leg intersects with the arc at subsequent time increments. However, if the normal period is terminated by $c2$, there might not exist such intersections. Moreover, during the heel-off period, the distance between the intersection point on the arc and the hip (for example $B \rightarrow H$ at time t_3) can exceed $(l_3 + l_4)$ due to a rapidly lengthening dynamic leg (problem $p1$). In both cases, the portion of the dynamic leg assigned to superimpose a thigh and a shank must be shortened, which means that the heel must come further off the ground. To find the new position of the ankle, the arc is intersected with a virtual circle centered at the hip H with radius $(l_3 + l_4)$ (actually a slightly shorter length to satisfy the locomotion attribute above for a minimum value of θ_4). The effect is that the angle does not lie on the dynamic leg any



a) normal period; foot is flat on ground



b) heel-off period; mid-foot rotates around M with radius l_{12}

c) meta-off period; foot rotates around T with radius l_{14}

Figure 5-8: Virtual leg concept for stance leg; the proportions of the foot (l_{12} and l_{13}) are exaggerated.

longer. Another problem ($p2$) might arise, if one of the just described exceptions applies, but eventually the regular algorithm can be employed again, that is the intersection of the arc with the dynamic leg yields a valid position for the ankle: there is a possibility for a sudden increase in ω_1 (decrease in the meta angle θ_6), which should constantly be decreasing. For this purpose, another locomotion attribute^E was defined to guarantee a continuous minimum decrease in ω_1 and therefore a smooth motion.

The end of the heel-off period is reached at time t_4 when ω_1 approaches a certain limit (around 65°). This signals the beginning of the *meta-off* period, during which ω_1 stays constant until the end of the stance phase (toe-off). Consequently, the foot rotates around the tip of the toe (T) with radius l_{14} (where $l_{14} = l_{12}^2 + l_{13}^2 - 2 l_{12} l_{13} \cos \omega_1$, with ω_1 taken at time t_4) as shown in figure 5-8 (c). The new arc R traced by the ankle is bigger and extends higher off the ground than the previous arc R'. The leg angles are calculated as in the previous heel-off phase (note θ_6 is fixed), by intersecting the dynamic leg with R; unfortunately, the same problems as in the heel-off period ($p1$ and $p2$ above) might occur. In addition, after the implementation it was realized that the meta-off period was only activated for fairly large step lengths ($\geq 80\,cm$ for $1.8\,m$ body); for walks at slow speed with a shorter step length the meta joint (M) never came off the ground even though in real human walking it does. The algorithm was therefore abandoned in favor of a different approach. From studies of human walking [Inman 81], two helpful observations were made. First, it was discovered that the meta-off period closely correlates with the double support period. This makes sense since during double support the weight of the body is shifted from the hind over to the front leg, which allows the relieved hind leg to rotate about the tip of the toe to "prepare" itself for the subsequent swing phase. Therefore, the event of the double support phase (whose time is known as discussed in section 5.1) is taken rather than an explicit bound on ω_1 to initiate the meta-off period; in fact, ω_1 now changes during meta-off and reaches a limit shortly before toe-off.

The second observation was that from the beginning of the double support phase, the knee of the hind leg flexes increasingly up to about 10 % of the cycle time into the swing phase. This flexion is quite independent of the walking speed and amounts to about 63° [Inman 81]. The increase in the knee and meta joint angles coincides with a rapid plantar flexion of the ankle. Using this information, the new meta-off algorithm linearly increments the knee and ankle angles to their values at the end of the double support phase; the former is known from above (by assuming a linear increase in the knee angle, the amount that the knee flexes during the first 10 % of the swing phase is subtracted from 63°) and the latter is specified as a locomotion attribute^E (in the actual implementation, the above percentage as well as the maximal knee flexion may also be modified by locomotion attributes^E). Hence, the leg determinants of the knee and ankle

are directly enforced. With these two angles known, the hip and meta angles are determined by simple trigonometric calculations.

It is important to notice that the meta-off period only applies to the hind leg during double support. At the same time, the front leg is in its normal period. The heel-off period of the stance leg takes place while the other leg is in an advanced stage of the swing phase. Another significant aspect concerns the implementation of the meta-off period: the motion of the inverted, springy pendulum is simulated for one walking step at a time, which does not comprise a whole stance phase but only the duration of one double support and one swing state (see figure 5-1). As soon as the swing leg signals heel strike, a new step is simulated for that leg. As a consequence, only the stance phase of the front leg is really modeled by the dynamics during double support, whereas the motion of the hind leg is totally expressed by the kinematic computations of the meta-off algorithm. This implies, that the position of the hip (H) for the hind leg during meta-off actually comes from the front leg, which takes on the "leading" role. In a sense, therefore, figure 5-8 (c) is drawn inaccurately; the dynamic leg at time t_4 and t_5 should rather be a virtual line to the hip of the other leg, whose current orientation would be somewhere between t_0 and t_1 as shown in part (a) of the figure.

The virtual leg concept introduced above animates knee flexion and plantar ankle flexion for the stance leg to complete the list of determinants of gait (section 5.3) that occur during a human walking cycle. Coincidentally, the term "virtual leg" is used elsewhere with a different meaning. Sutherland [Sutherland 84] studied the locomotion of machines with more than two legs. In order to simplify the control and coordination of the legs, he defined the idea of virtual legs as follows: when two (or more) legs act in unison, they can be functionally expressed as one virtual leg which is located halfway in between them. The virtual leg exerts the same forces and torques on the body as the original pair of legs. For instance, in a 4-legged pacing gait, the two left legs act in unison as do the two right legs. It follows that the pace is functionally identical to walking on two virtual legs, each of which located halfway between the hind and the front leg on either side; 2-legged hopping, where the left and right leg perform the same motion in parallel, reduces to a one-legged hop. Sutherland's virtual leg principle somewhat supports the claim made in an earlier section (3.2), that the algorithm for bipedal walking devised in this thesis is extensible to control other gaits as well as an increased number of legs. This is to say that once the motion of one leg is totally known in terms of stance and swing, additional legs can be readily added and coordinated to generate different gaits.

5.5. Upper Body Angles

The major focus of this research is directed towards the motion of the lower limbs during walking. The upper body can be thought of as a passive part, which is carried along by the lower body. However, certain movements of the upper body, like the swinging of the arms or the more subtle forward and backward tipping of the trunk, play an important part in contributing to an overall human appeal of the locomotion.

In the dynamic model, the upper body is represented by a single segment (see figure C-1), which has to be *actively* balanced -as discussed in section 2.2- in order to maintain an erect posture over the duration of a walking sequence. Similar to Raibert's approach [Raibert 86a], this is done by application of a hip torque of the form

$$F_{\theta_2} = -k_2(\theta_2 - \theta_{2\_des}) - v_2\dot{\theta}_2 \ ,$$

where k_2 and v_2 are spring and damping constants, respectively (their default values are listed in appendix C.4). The torque is computed at each time increment with the current values for θ_2 and $\dot{\theta}_2$. The desired upper body angle with respect to the vertical, θ_{2_des}, is usually zero, but can be modified as one of the locomotion attributes[E]; for instance, to produce the effect of a person leaning more forward during walking. Of course, k_2 and v_2 may also be varied if larger or smaller deviations from the upright position are desired.

As was the case for the legs, the dynamics generate a natural pattern over the duration of the locomotion (as far as the timing goes), but it is only generic (as far as the quality of the motion goes); namely the forward and backward tipping of the upper body in the sagittal plane. Kinematic calculations are therefore added based on the experimental data from Inman's book [Inman 81]. Since the upper body "follows" the motion of the lower body, each kinematic component can be directly formulated as a function of some determinant of gait. For this purpose, two types of upper body movements are distinguished: compensatory actions of the head and trunk to account for the pelvic motion, and the arm swing which is synchronized with the movements of the legs.

The rotational motion of the pelvis in the transverse plane is compensated by a shoulder rotation. These rotations are out of phase with each other, e.g. while the pelvis rotates forward on the side of the swinging leg, the shoulder rotates back. The magnitude of the shoulder rotation amounts to about 60 % of that of the pelvis, but can be altered by a locomotion attribute[E]. The total shoulder rotation is achieved by a stepwise rotation along the vertebral column, which is distributed over the five lumbar and twelve thoracic vertebrae in such a way that the seventh vertebra from the top (T_7) exhibits a minimum rotation. Below T_7

the joints rotate towards the orientation of the pelvis, whereas those above T_7 rotate towards the shoulder. The head is largely unaffected by these rotations and usually faces in the walking direction. The seven cervical joints are adjusted to account for this (i.e. each cervical vertebra is rotated with 1/7 th of the current shoulder rotation). Pelvic list is compensated by the vertebrae of the lumbar region to maintain an upright position of the upper body in the coronal plane.

The rotations of the arms are derived from the motion of the legs. Each arm swings forward with the opposite leg and moves back during the stance phase for that leg. Hence, there is a direct relationship between the angle for the arm swing at each time step and the hip angle of the opposite leg. The magnitude for the arm movements might diverge from the maximum hip angle by quite a perceivable amount, since it is not strictly dependent on the step length only, but to a certain degree, rather a matter of individual style. A locomotion attribute^E is provided to express the arm swing as a percentage of the hip angle. In practice, a smoother motion of the arms was obtained by making the arm angle dependent on the dynamic hip angle θ_1 on the back swing and not the kinematic hip angle θ_3, because the former is continuously decreasing, while on the contrary, θ_3 increases during stance due to the flexion of the knee (i.e. the thigh might not move back continually). It has also been observed that the elbow flexes a small amount near the end of the forward swinging period of the arm. More exactly, the flexion of the elbow sets in as soon as the arm angle, and likewise the hip angle of the opposite leg, has reached a maximum (which actually happens some time before heel strike, at the end of the SWING1 phase, as shown in the next chapter) and continues until heel strike. An upper and lower bound for the elbow flexion are specified by locomotion attributes^E.

A final remark on the upper body kinematics is that they do not complicate the control algorithm for the legs and, unlike the kinematics of the lower body, they can be applied to each step after the dynamic simulation of the legs.

Chapter 6
Low Level Control Details

The simulations of the stance and swing leg for a step are subject to constraints for timing and final orientation as expressed in equation (5.2) and (5.4); also the models themselves are constrained by the way the equations of motions are formulated (e.g. the stance leg turns around a fixed point on the ground). More than that, however, the dynamics 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 value at heel strike if some kind of a sinusoidal motion pattern is to be maintained (illustrated in figure 5-4); 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. Both aspects are explained in section 6.1 and 6.2 for the stance and the swing phases, respectively.

Bearing in mind that the walking algorithm exhibits a step oriented structure (i.e. the basic units of simulation are a double support state and a single support state), the following order of execution now becomes transparent (the figures in brackets denote section numbers):

```
for step i do                                                          (6.0)
  initialize step;          /* normalization formula [5.1]; state-phase timings [5.1];
                               symmetry of step [5.2]                          */
  stance phase dynamics;    /* integration [4.3]; interpolation [4.5]; STANCE1,2,3,4 [6.1] */

  lower body kinematics;    /* determinants of gait [5.3]; virtual leg [5.4]   */

  swing phase dynamics;     /* [4.3]; [4.5]; SWING1,2,3 [6.2]                  */

  upper body kinematics;    /* arm swing, pelvic movement compensation [5.5]   */

  increment i;
endfor
```

The separation of the leg dynamics for the stance and swing phases in the above sequence 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. As shown in appendix C.2, the degrees of freedom for the swing leg are therefore reduced to just θ_3 and θ_4. On the other hand, the swing leg has 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 % [Winter 79]). 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.

It should be noted that because the stance phase of a leg is only simulated for one step at a time (although the duration of a complete stance phase is $t_{step} + t_{ds}$, as seen in figure 5-1), the dominating role of the stance leg is taken up by the other leg at heel strike, i.e. as soon as the stance phase for that leg commences. One last important observation about algorithm (6.0) concerns the leg kinematics: the determinants of gait, which are applied after the simulation of the stance phase, only express the movements of the pelvis and the stance leg. For the dynamic swing leg, which is modeled as a double pendulum, the motion of the foot is added kinematically (for the dynamics, the foot is represented as a point mass at the end of the shank). This must take place during the dynamic simulation of the swing since the thigh and shank have to be moved in such a way as to avoid ground contact by the foot.

In order to maintain a locomotion cycle, the muscles of the lower limbs must develop flexion and extension moments at the hip, knee and ankle (neglecting minor moments around the metatarsophangeal joints). In terms of the energy level in walking (recall figure 5-5) these torques make up for the dissipation in energy, particularly at heel strike and around mid-stance, where a decrease in velocity causes a drop in kinetic energy that is not fully compensated for by an increase in potential energy. The two peaks in energy losses over the period of one step indicate that muscle activity is not distributed uniformly with time; in fact, some joint moments produced by the leg muscles become infinitely small for certain phases during a locomotion cycle. This has, for instance, led to the theory of ballistic walking [McMahon 84] (refer to section 2.1), which is based on the assumption of zero torques acting during the swing. Whereas a completely conservative swing phase seems an appropriate simplification for the rhythmic period of a locomotion sequence, it was not considered for the walking algorithm here because it would have been difficult to meet the step symmetry constraints by only adjusting the initial conditions (i.e. the angular velocities). Besides, in the case of a non-optimal walk, e.g. a walk with an unnaturally small or large step length at a given velocity, as well as for acceleration or deceleration in locomotion, the muscles of the swing leg are quite active.

Therefore, controlled moments at the hip and the knee are applied for the entire swing phase, as described in section 6.2. Nevertheless, the idea of partial application of joint moments has been employed during the stance phase; this is discussed in more detail in the following.

6.1. Stance Phase

Experiments on human subjects utilizing electromyography and force plates [Inman 81, Winter 79] have shown that a significant torque at the hip of the stance leg occurs only just after heel strike and lasts for about 20 % of the cycle time. Also, the torque during this time interval is such that it rapidly reaches a maximum value and decays quickly towards the end. The two observations have been adapted to the dynamic stance model (for the equations of motion, see appendix C.1) to determine the duration of the leg torque F_{θ_1}, as well as its shape, which is assumed to be constant over the whole duration. The magnitude of F_{θ_1} to move the stance leg from its initial position to the final leg angle θ_1 at the end of a step (θ_1 is supplied by the symmetry of step concept [5.2]) is calculated by the interpolation procedure explained in section 4.5. Since F_{θ_1} is turned off at $0.4 \cdot t_{step}$ (i.e. after 20 % of a cycle) the motion appears to be conservative thereafter until the end of a step; of course, the system is not truly conservative because a torque F_{θ_2} to balance the upper body of the model (see section 5.5) and an active force along the leg axis F_ω are applied throughout the simulation of the stance phase.

Whereas F_{θ_1} represents the hip torque, F_ω basically models the moments produced at the knee and ankle during the stance phase of a real human locomotion cycle; for the early period of the stance, F_ω simulates the flexion of the knee while in the latter part it essentially symbolizes the push-off effect generated by the plantar flexion of the ankle. Therefore, the values for F_ω directly influence the leg determinants (virtual leg). As a matter of fact, the leg axis force F_ω plays a key role in the proper and realistic functioning of the stance phase by controlling the length ω of the of the dynamic leg. Generally, F_ω must be chosen such that the hip of the stance model, which roughly corresponds to the center of mass, prescribes a vertical sinusoidal curve typical in walking (see figure 5-4). In practical terms, this means that the hip may never drop too low nor raise too high, i.e. the length of the leg can only vary within certain boundary values. Furthermore, the leg length ω has to reach the desired value at the end of a step to satisfy the constraint imposed by step symmetry.

In order to implement all of these demands, the stance phase is divided into four subphases (STANCE1, STANCE2, STANCE3, STANCE4) according to the calculations of F_ω and modifications of the dynamic equations. STANCE1, STANCE3 and STANCE4 are compulsory, in that they have to occur in

every stance phase, while STANCE2 may or may not be executed. All subphases are activated by certain events signaling violations in the above constraints. Once a constraint violation is detected at a time step, the exact time for a phase change is computed by interpolating back between the current time and the previous time step; for instance, a phase change between STANCE1 and STANCE2 (as explained below) takes place when $\omega \geq l_1$ is detected. The exact time to start STANCE2 is the time between the current and previous time step, where $\omega = l_1$.

The STANCE1 phase is initiated at heel strike and represents the normal mode of the stance phase simulation. The magnitude of the leg axis force F_ω at each time step is calculated from a linear spring and damper model:

$$F_\omega = k_\omega (l_1 + pa_1 - \omega) - v_\omega \dot{\omega} \ , \tag{6.1}$$

where k_ω and v_ω are spring and damping constants, respectively (their default values, as well as the values for all the subsequent constants, are listed n appendix C.4); l_1 is the original, unloaded length of the leg, ω is the current leg length and $\dot{\omega}$ the velocity along the leg axis. The purpose of the position actuator pa_1 (initially zero) is to ensure that the stance hip maintains a minimum height, which is the desired height (y_h) at the end of the step (y_h is defined in the calculations for step symmetry, see page 44). If the hip falls below y_h during the stance phase, possibly because the spring constant k_ω is chosen too small, pa_1 is incremented and the simulation is restarted for that walking step. This procedure might have to be repeated several times until y_h is satisfied. The increase in pa_1 corresponds to an artificial, short-term stiffening of the leg spring.

A growing pa_1 might, however, lead to another problem, which may also be caused by a large value for k_ω or a small damping constant: the actual length of the dynamic leg ω could exceed l_1 before the leg passes through the vertical; eventually, of course, ω has to extend to the desired length at the end of the step (i.e. $\omega > l_1$), but if this happens too quickly, in particular before the leg angle θ_1 becomes positive, a quite unnatural motion results. The reason is that according to the kinematic virtual leg concept (see section 5.4) the heel-off period begins as soon as $\omega > l_1$; in real human walking, however, the heel never comes off the ground before the leg has passed through the vertical, i.e. $\theta_1 \geq 0$ (this last statement is based on our own observations, but supported by experimental data [Inman 81]). Therefore, the event $\omega > l_1$ while $\theta_1 < 0$ terminates the SWING1 phase and marks the beginning of STANCE2, during which the leg spring is locked to freeze the length of the dynamic leg at l_1 until $\theta_1 \geq 0$. Consequently, the leg just moves like an inverted pendulum, which requires a modification in the equations of motion. For the matrix representation in appendix C.1,

$$b'_1 = b'_2 = 0, \ a'_{2,2} = 1 \ \text{ and } \ a'_{2,6} = a'_{6,2} = 0.$$

Once the leg has passed through the vertical ($\theta_1 \geq 0$), the STANCE1 phase is resumed. A smooth transition from STANCE2 to STANCE1 is achieved by initializing the STANCE1 phase with the value $\dot{\omega}$ from the time before the spring was locked. In practice, the SWING2 phase occurs mostly for short step lengths (note it is not compulsory), because due to a smaller impulse at heel strike the springy leg does not contract much, but rather extends quite quickly.

The leg axis force F_ω in STANCE1 might not be strong enough to lengthen the dynamic leg to the desired value at the end of the step (subsequently denoted by ω_{des}). For this purpose, another subphase (STANCE3) is introduced in which F_ω is continuously increased. The magnitude of F_ω at each time step amounts to

$$F_\omega = k_\omega (\omega_{des} + pa_3 - \omega) - v_\omega \dot{\omega} \ , \tag{6.2}$$

where k_ω, v_ω, ω and $\dot{\omega}$ have the same meaning as in equation (6.1). To maintain a continuous motion when changing from STANCE1 to STANCE3, the new position actuator pa_3 is initialized as follows:

$$pa_3 = l_1 + pa_1 - \omega_{des} \ .$$

pa_3 is then incremented by pa_{3_inc} at each time step throughout the STANCE3 phase. The event to activate STANCE3 has been chosen to coincide with the end of the SWING1 phase (see below); this is a somewhat arbitrary choice, but it represents a good compromise between extending the leg too early, and increasing ω too late and in a rapid fashion. Furthermore, the lengthening of the dynamic leg beyond l_1 basically simulates the heel coming off the ground in walking, which begins usually at about the end of SWING1 for the other leg.

Once the dynamic leg arrives at its desired length, i.e. $\omega \geq \omega_{des}$, the leg spring is locked again and in a new phase, STANCE4, the motion of the leg is simulated by a simple inverted pendulum until the end of the current step. In a sense, therefore, STANCE3 and STANCE4 are similar to STANCE1 and STANCE2, respectively. Unlike in the case of STANCE2, however, the STANCE4 phase is compulsory for a successful completion of the stance, that is, the stance leg model always has to be in the STANCE4 phase at the end of a step to satisfy the symmetry of step concept (i.e. ω_{des}). If STANCE4 is not reached, the simulation of the stance phase is repeated by increasing pa_{3_inc} to raise the magnitude of F_ω during STANCE3.

The dynamics of the stance phase combined with the leg kinematics produce a natural looking animation of a human stance leg during walking. This justifies the simplicity of the dynamic model, where a linear spring and damping construct simulate the shock-absorbing and push-off characteristics of the knee and ankle.

6.2. Swing Phase

As with the stance phase, the original swing phase is broken up into subphases in order to obtain a natural movement of the leg. Three subphases are distinguished (SWING1, SWING2, SWING3) -they are illustrated in figure 6-1. It should be noted that a similar method of subdividing the swing phase of a human locomotion cycle is described in a paper by Beckett and Chang [Beckett 68] (see also section 2.2). Whereas Beckett and Chung studied an "ideal" swing phase isolated from a walking cycle, the objective in this thesis is to assemble a complete locomotion sequence. This made a significantly different approach necessary for the subdivision as well as the calculation of the joint moment profiles. However, the principal idea of the SWING1 subphase as discussed below could be adapted from Beckett and Chang's second phase (phase II).

During SWING1 (from t_0 to t_1) the ankle is restrained to move along the polynomial curve P until the toe is exactly under the knee. At the same time, the hip angle reaches a maximum (subsequently denoted as θ_{3_des}), 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. 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 [Inman 81, Winter 79], 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} for the current step is known from the state-phase timing concept, the durations for the subphases can be readily determined. The parameters for the end of SWING1 and SWING2 are both specified as locomotion attributes^E to provide a means to slightly adjust the timings for different walking styles. In the following, the implementation of the subphases is covered in more detail. It is important to realize that in distinction from the stance phase, for which the whole phase is formulated as a root finding problem (section 4.5), each of the swing subphases is solved separately.

6.2.1. SWING1

In the SWING1 phase, the hip torque F_{θ_3} is interpolated such that the hip angle θ_3 reaches the desired value θ_{3_des} at time t_1. The magnitude of F_{θ_3} at each time step t is defined by an exponential function which closely models the rapid decrease in the hip moment during the beginning of the swing phase (this is based on experimental measurements by Inman [Inman 81]):

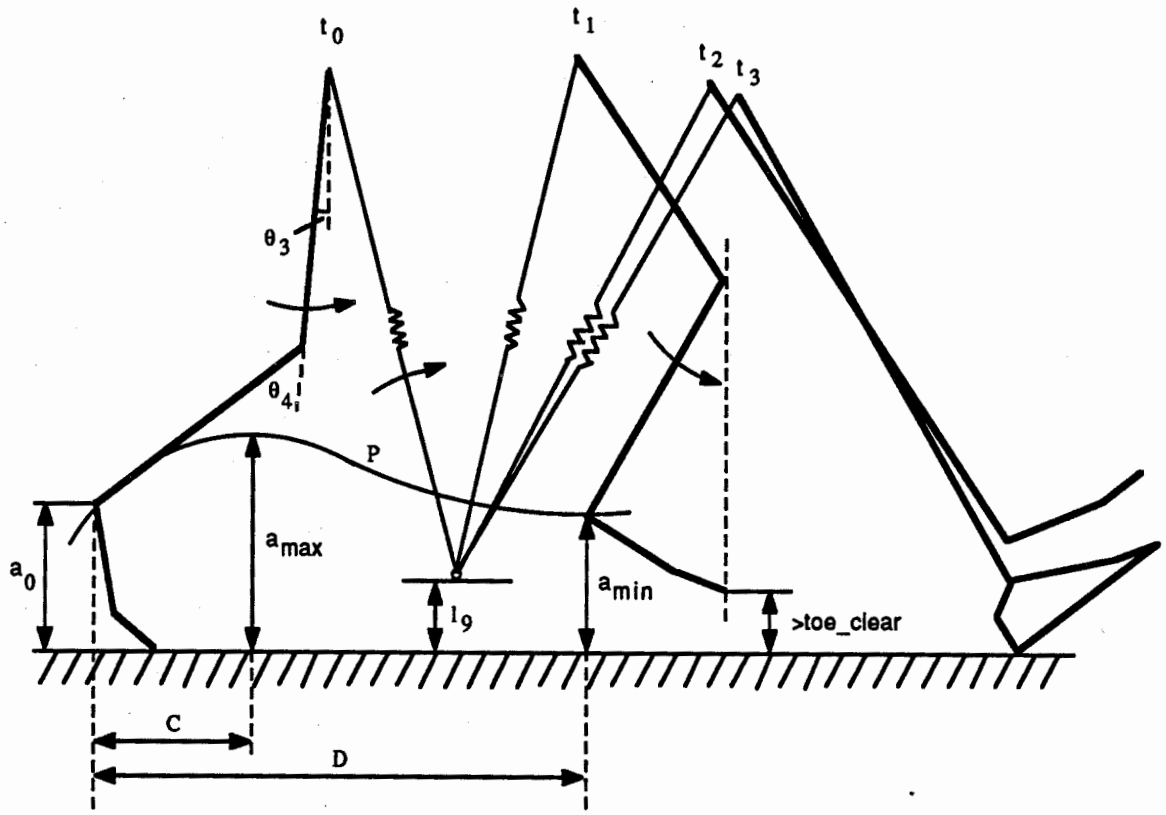


Figure 6-1: Illustration of swing phase (upper body is ignored); the proportions of the kinematic foot are exaggerated. The same legend applies to the leg angles and segments as in figure 5-7. The SWING1 phase goes from t_0 (time of toe-off) to t_1, SWING2 from t_1 to t_2 and SWING3 from t_2 to t_3 (time of heel strike).

$$F_{\theta_3} = B \cdot e^{-(At)^2}, \quad \text{where } 0 \le t \le t_1 \ . \tag{6.3}$$

B represents the initial maximal value of F_{θ_3} and is determined by the interpolation process; A is set to be a function of the step length (sl) and the duration of SWING1 (t_1):

$$A = sl_{max} / (\sqrt{2} \cdot sl \cdot 0.5 \cdot t_1), \quad \text{where } sl_{max} \text{ is the maximal possible step length.}$$

The calculation of A is justified as follows: the above exponential function has an inflexion point p at $(\frac{1}{A\sqrt{2}}, \frac{B}{\sqrt{e}})$; the more p moves to the left (i.e. the smaller $\frac{1}{A\sqrt{2}}$), the faster is the decrease in the resulting hip torque. As one might expect, F_{θ_3} should decay sooner for shorter step lengths, that is, if $sl_1 < sl_2$ then

$\frac{1}{A_1\sqrt{2}} < \frac{1}{A_2\sqrt{2}}$. On the other hand, the inflexion point should never occur too late to maintain a rapid decrease in F_{θ_3} relative to the time t_1 for SWING1. After experimentation with the system, it was found that the most natural motion resulted if the limit put on p was half the time for the SWING1 phase (i.e. $\frac{1}{A\sqrt{2}} \leq 0.5 \cdot t_1$).

During a human locomotion cycle, the extension moment at the hip of the swing leg starts in the double support phase and produces a continuous motion at toe-off. Equation (6.3) implies, however, that the hip torque sets in suddenly with a maximum value at the beginning of the SWING1 phase; there is no hip torque during the double support for the impending swing leg since its motion is completely expressed by the kinematics of the virtual leg concept. To assume a smooth transition at toe-off, the equations of motion for the SWING1 phase have to be initialized properly with the leg angles and angular velocities at the hip from the end of the meta-off period. The velocities are calculated simply from

$$\dot{\theta}_3 = \frac{\theta_3(t_0) - \theta_3(t_0 - \Delta t)}{\Delta t}, \quad \text{where } \Delta t \text{ is the time increment.} \tag{6.4}$$

While the hip torque drives the thigh to its desired orientation at the end of the SWING1 phase, the ankle moves along the 4th order polynomial curve

$$P: \; y(x) = \sum_{i=0}^{4} A_i x^{4-i}, \tag{6.5}$$

where x and y are the coordinates of the ankle; the evaluation of the coefficients A_i is delayed for a moment, in order to finish the discussion of the joint torques for SWING1. One method to constrain the ankle to the curve P during SWING1 involves the application of a knee torque F_{θ_4}. Since a moment produced at the knee affects the hip angle as well, this approach could be difficult to implement. Instead, a simpler method is employed, which totally eliminates the need for controlling a knee torque: the knee joint is locked for the dynamics. At each time step in the simulation, the knee angle is updated kinematically such that the constraint on the ankle motion is satisfied (i.e. the ankle lies on P). This updated position of the ankle, that is the new value for θ_4 is used for the dynamic simulation at the next time step. To account for the locking of the knee, the equations of motion (see appendix C.2) are modified as follows, assuming the matrix representation:

$$b'_3 = b'_4 = 0, \; a'_{4,4} = 1 \text{ and } a'_{2,4} = a'_{4,2} = 0.$$

The knee angle θ_4 at each time step is calculated from the three equations shown below; the hip angle θ_3 comes from the dynamics of the swing leg and the position of the hip (x_h, y_h) is known from the stance phase simulation. The three unknowns are θ_4, x and y; the latter two define the ankle position:

$$f_1(x, y, \theta_4) = x_h - l_3 \sin\theta_3 - l_4 \sin(\theta_3 + \theta_4) - x = 0$$

$$f_2(x, y, \theta_4) = y_h - l_3 \cos\theta_3 - l_4 \cos(\theta_3 + \theta_4) - y = 0$$

$$f_3(x, y, \theta_4) = A_0 x^4 + A_1 x^3 + A_2 x^2 + A_3 x + A_4 - y = 0 \ .$$

The values of x and y from f_1 and f_2, respectively, are now substituted into f_3, which is solved for θ_4. Unfortunately, there is no analytical solution. Therefore, numerical interpolation techniques are applied. The Bisection method is used to compute a good first approximation, which is subsequently refined by the Newton Raphson method [Burden 85]. Initial lower and upper bounds on θ_4 (which correspond to an ankle position (x,y) above and below the polynomial P) are obtained by an algorithm similar to (4.5). The Newton-Raphson method converges fast, usually within 3 or 4 iterations; it can be used in this case because the first derivative of f_3 with respect to θ_4 exists:

$$f'_3 = (4A_0 x^3 + 3A_1 x^2 + A_2 x + A_3) \cdot (-l_4) \cos(\theta_3 + \theta_4) - l_4 \sin(\theta_3 + \theta_4) \ .$$

So far, the calculation of the hip and knee angle has been discussed. The two remaining leg angles of the ankle and meta joint are determined by linear interpolation between the values at toe-off (time t_0) and the end of SWING1 (t_1). Whereas the former are known from the end of the kinematic meta-off phase, the latter are specified as locomotion attributes[E] (`swing1_ankle, swing1_meta`). However, the toe might stub the ground in the event of a flat shape of the polynomial P (if a_{max} is small, see figure 6-1). This is prevented by a constant checking for ground contact during SWING1 and if need be, the ankle is dorsiflexed temporarily just enough for toe-clearance.

The derivation of the P as defined in equation (6.5) is now described. In order to compute the coefficients $A_0 - A_4$ -which is done by a Gaussian elimination algorithm- five conditions for the motion of the ankle are required. The following conditions, which are illustrated in figure 6-1, establish a desirable shape for P :

1. $y(0) = a_0$
2. $y(C) = a_{max}$
3. $y'(C) = 0$
4. $y(D) = a_{min}$
5. $y'(D) = 0$

The horizontal position of the ankle at toe-off is assumed to be zero; a_0 is computed at the end of the previous meta-off period. At a distance C, the ankle reaches a maximum height a_{max} during swing. Based on observations by Beckett and Chang [Beckett 68] and Inman [Inman 81], C amounts to about 30 % of the value for D (see below). A locomotion attribute[E] was defined to allow for adjustments. An upper bound on a_{max} is set to be $l_5 + l_6$, which is the length of the toe plus the mid-foot. The actual maximum height for the current step in a locomotion sequence is expressed as a function of the minimum length ω_{min} of the dynamic stance leg for that step (this corresponds to a maximum knee flexion). This is justified by the fact that the faster a person walks, the bigger the knee flexion during stance, which causes a lower ankle path of the swing leg. Therefore,

$$a_{max} = \frac{\omega_{min}}{l_1} \cdot (l_5 + l_6), \quad \text{where } l_1 \text{ is the length of the extended leg.}$$

At the end of the SWING1 phase (at t_1) the ankle arrives at a distance D and assumes the minimal vertical value a_{min}. By this time, the hip angle has reached θ_{3_des} for heel strike and the toe is exactly under the knee. Since the position of the hip is determined by the stance leg and the ankle and meta angles are known as well (`swing1_ankle, swing1_meta`), the values for a_{min} and D can be readily obtained by trigonometric calculations (note that both are functions of the step length, because they depend on θ_{3_des}). Nevertheless, particularly at short step lengths, there is a possibility that the computed value for a_{min} turns out to be too small so that the toe would go through the ground at t_1 or at least the `toe_clear` constraint (specified by a locomotion attribute) might be violated. In either case, the hip angle is increased just enough (θ_{3_clear}) to satisfy the `toe_clear` constraint. This is the reason why the extension of the hip can get bigger at t_1 than at heel strike, which can also be observed in real human walking at short step lengths. Of course, for the calculation of D, θ_{3_clear} rather than θ_{3_des} is used. Conditions (3) and (5) above ensure the

maximum and minimum criteria of P by setting the first derivatives to zero at C and D. The resulting curve will take on different shapes depending on the characteristics of the current walking step.

There is one more significant aspect about the SWING1 phase. If the initial estimate of the hip torque B in equation (6.3) is very large, it might happen that the hip extends so much during SWING1 that the ankle can not be "placed" on the polynomial P, no matter what knee angle θ_4 is chosen; in this case, a recovery mechanism aborts the simulation of the current SWING1 phase, lowers the value of B and restarts the interpolation process.

6.2.2. SWING2

During the SWING2 phase from t_1 to t_2 (see figure 6-1) the thigh remains at a fairly constant position, whereas the shank swings rapidly forward. The dynamic simulation is based on the equations of motion as formulated in appendix C.2 (which implies that the knee is not locked any longer). To move the shank forward, a torque at the knee, F_{θ_4}, is calculated by interpolation (as introduced in section 4.5) in such a way that the knee extends exactly at time t_2 (a final knee angle bigger than zero may actually be selected by a locomotion attribute, which was discussed in section 5.2); F_{θ_4} thereby assumes the same exponential profile as F_{θ_3} during SWING1 defined in equation (6.3). This produces a quick decay for the knee torque, and results in an acceleration of the shank at the beginning of SWING2 and a gentle extension of the knee at the end. In order to sustain a continuity in the motion of the shank across the phase change at t_1, the SWING2 phase is initialized with the value of the angular velocity $F_{\dot{\theta}_4}$ at the end of SWING1, which is calculated from an equation similar to equation (6.4) (note that it is not determined by the dynamics since the knee was locked during SWING1).

Simultaneous with the knee moment, a hip torque F_{θ_3} is applied throughout SWING2 to hold the thigh in its desired position. The magnitude of this torque at each time step is defined as

$$F_{\theta_3} = k_3 (\theta_{3\_clear} - \theta_3) - v_3 \dot{\theta}_3 \ ,$$

where k_3 represents a spring, v_3 a damping constant; θ_3 and $\dot{\theta}_3$ are the current hip angle and angular velocity at the hip of the swing leg. θ_{3_clear} is initially set to θ_{3_des}. However, the foot might intersect with the ground while the knee is extending. This could result from k_3 and v_3 being chosen too small, or by an accentuated pelvic list (see section 5.3) and a short step length. To recover from this situation, a new increased hip angle is defined such that the foot is lifted just above the ground at the time step where the undesired impact occurred and the simulation of the SWING2 phase is then repeated with θ_{3_clear} set to the

new hip angle. Even with the new value for θ_{3_clear} which produces a larger F_{θ_3} and therefore a greater extension of the hip, it is possible for the foot to strike the ground again at a later time in the same SWING2 phase. Thus, the recovery measure might be applied several times. As the shank swings forward the foot reaches its lowest point in the rotation around the knee when the ankle passes through the vertical. Consequently, if $\theta_{3\_clear} > \theta_{3\_des}$, the algorithm attempts to reduce the hip angle to its desired value. The recovery and reduction measures are necessary to successfully generate SWING2 phases for different locomotion parameters and attributes. They also guarantee a low position of the foot during the latter part of the swing, which naturally occurs in a real human walk. Because of the feedback mechanism (i.e. the check for impact), this approach might very well be modified to account for clearance of small obstacles along the locomotion path.

As in the SWING1 phase, the angles for the ankle and meta joint during the SWING2 and subsequently the SWING3 phase are determined by linear interpolation between the initial angles (`swing1_ankle, swing1_meta`) at time t_1 and the final angles at heel strike at t_3. Whereas the meta angle at t_3 is assumed to be zero, the ankle angle is specified by a locomotion attribute as described in section 5.3.

6.2.3. SWING3

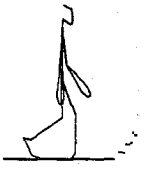
The SWING3 phase takes up the short period (by default 15 % of t_{swing}) at the end of the swing which is characterized by very little rotational movement at the joints of the swing leg (see figure 6-1, t_2 to t_3). Since the leg is extended and the knee is locked, the equations of motion are modified as in the SWING1 phase; the dynamic model of the swing leg is now defined as a single pendulum. To account for the impulse as a result of the knee extension at the end of SWING2, the equations are initialized according to the calculations in appendix C.3.2.

A constant torque is applied to the hip during SWING3 to bring about heel strike at exactly the time for the end of the swing. This is achieved by the interpolation process described in section 4.5. Usually, the hip torque takes on values to solely hold the leg in its position (and therefore its magnitude is similar to F_{θ_3} near the end of the SWING2, neglecting the rather minor compensation for the knee extension impulse), while forward motion of the hip directed by the stance leg is mainly responsible for bringing the heel down onto the ground. At heel strike, the orientation of the swing leg plus the impact conditions (see appendix C 3.1) are used to initialize the stance leg model for the subsequent stance phase.

6.2.4. Starting and Stopping

The subphases for the stance and swing described above apply not only during the rhythmic period of a locomotion sequence, but also during accelerated and decelerated motion. If the locomotion parameters change at any time during a walk, the different timing constraints and final conditions for the subphases become active as soon as the current step is completed. However, for the first and last step the swing subphases need to be adjusted to account for a different motion pattern of the leg.

When starting a walk from rest with both feet parallel on the ground, the SWING1 phase can not be executed as discussed in section 6.2.1. Instead, the following algorithm is employed, which is based on the same principle as the original SWING1 phase (i.e. the knee is locked for the dynamic simulation and the knee angle is updated kinematically while a hip torque produces a forward swing of the thigh to its desired orientation θ_{3_des}): rather than prescribing the motion of the ankle by a polynomial, the tip of the toe is lifted up vertically. If the hip angle has not reached θ_{3_des} by the time when the knee is directly over the toe (this happens automatically by increasing the hip angle and moving the toe straight up), the toe holds its position directly under the knee until the end of SWING1. The SWING2 phase can then proceed as usual.



On the last step, the SWING1 phase is applied normally except that the desired angle for the hip θ_{3_des} is not zero as one might expect, but calculated such that the knee of the swing leg at the end of SWING1 is exactly at the same horizontal position as the toe of the stance leg (of course, the toe of the swing leg is directly under the knee at this moment). In place of the SWING2 and SWING3 phases, the foot is now just lowered straight down until it is flat on the ground parallel to the stance foot at the end of the swing. This is implemented as sort of a SWING1 phase by locking the knee for the dynamics and applying a hip torque to arrive at a zero hip angle at t_{swing}. Because the lowering of the foot takes place much more quickly than the usual STANCE2 and STANCE3 phases, the duration is reduced to half the normal time; at the same time the duration for the SWING1 phase is increased to satisfy t_{swing}.

It is noted that the algorithms for the first and last step of a locomotion sequence have been chosen somewhat arbitrarily. Since no experimental kinematic data could be found in the literature on how human beings start and stop a walk, the steps taken above are solely justified by our own observations. Also, a compromise was made between generating a realistic appearance and controlling the dynamics.

Chapter 7
Results

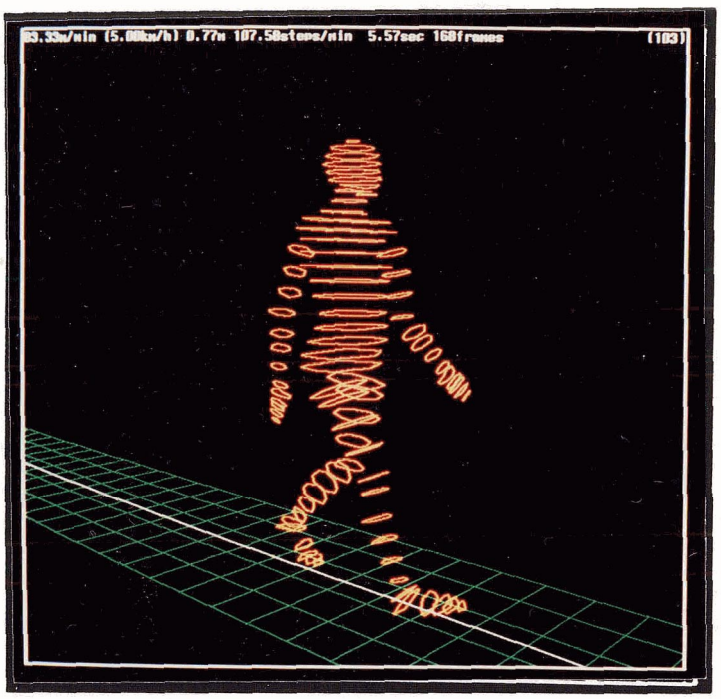


Figure 7-1: Display of a walking figure.

The KLAW (Keyframe-Less Animation of Walking) system has been implemented to animate bipedal walking. It consists of two C-program modules, *walk* and *walkdsp*. The former does all the calculations discussed in the previous sections, whereas the latter was written for displaying the results (i.e. a walking figure such as that shown in figure 7-1) on an IRIS 2400 workstation. Appendix B.2 shows the interface between the two programs. The computation of all the body angles is time-consuming: for instance, it took 31 *sec* of CPU time on a SUN3-50 computer with floating point processor (Motorola MC68881) to compute the animated walk illustrated in figure 7-1 (5.57 *sec* = 168 frames). In practice, the SWING2 phase has been most expensive because of its recovery and reduction measures (see section 6.2.2); 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.

Some of the results of the walking algorithm are now discussed in more detail. The convention is used that positive angles mean a flexion and negative angles denote an extension of a joint; for the ankle and meta joint, a positive angle is a plantarflexion, whereas a negative value is a dorsiflexion. Except for figure 7-10, the ordinate values for all diagrams start at heel-strike (i.e. the motion of the left leg in figure 5-1 is presented). All the joint rotations are assumed to be in the sagittal plane, that is, along the line of progression.

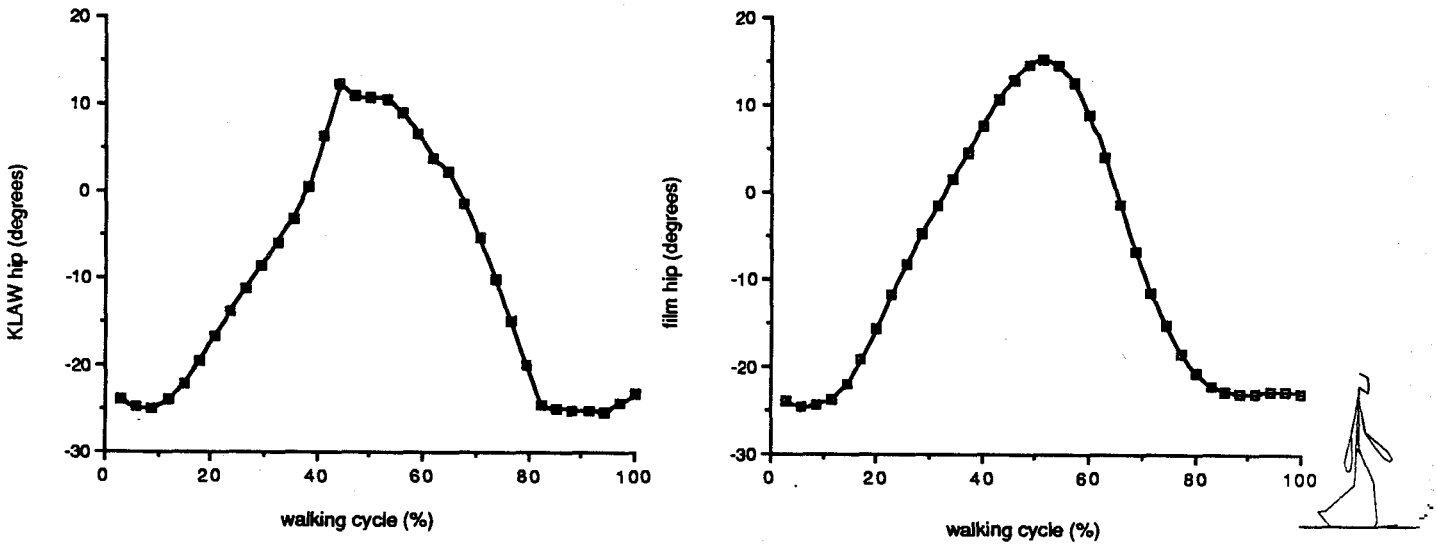


Figure 7-2: Comparison of hip angles during a locomotion cycle. Left: angles as calculated by KLAW; right: angles obtained from a walking human subject [Winter 79].

Figures 7-2, 7-3 and 7-4 show the hip, knee and ankle angles as calculated by the walking algorithm compared to real walking data collected by Winter [Winter 79] for one rhythmic walking cycle (denoted as *W* for further reference). The locomotion parameters for the "film" walk are approximately $v = 5\ km/h$, $sl = 0.79\ m$ and $sf = 107\ steps/min$ based on a body height of $1.8\ m$. Specifying a desired velocity of $v = 5\ km/h$, the walking algorithm calculated (see appendix D) a "natural" step length $sl = 0.77\ m$ and a step frequency $sf = 107.5\ steps/min$, which are very close to the real walk. Illustrations of both walks are given in appendix F (figure F-1 and F-2).

Since filtering and smoothing techniques were applied to the real walking data [Winter 79], these angles produce more continuous curves. A peak occurs in the calculated KLAW hip angle (maximum flexion) at around 40 % into the cycle. At the same time the knee angle is a minimum, which means that the

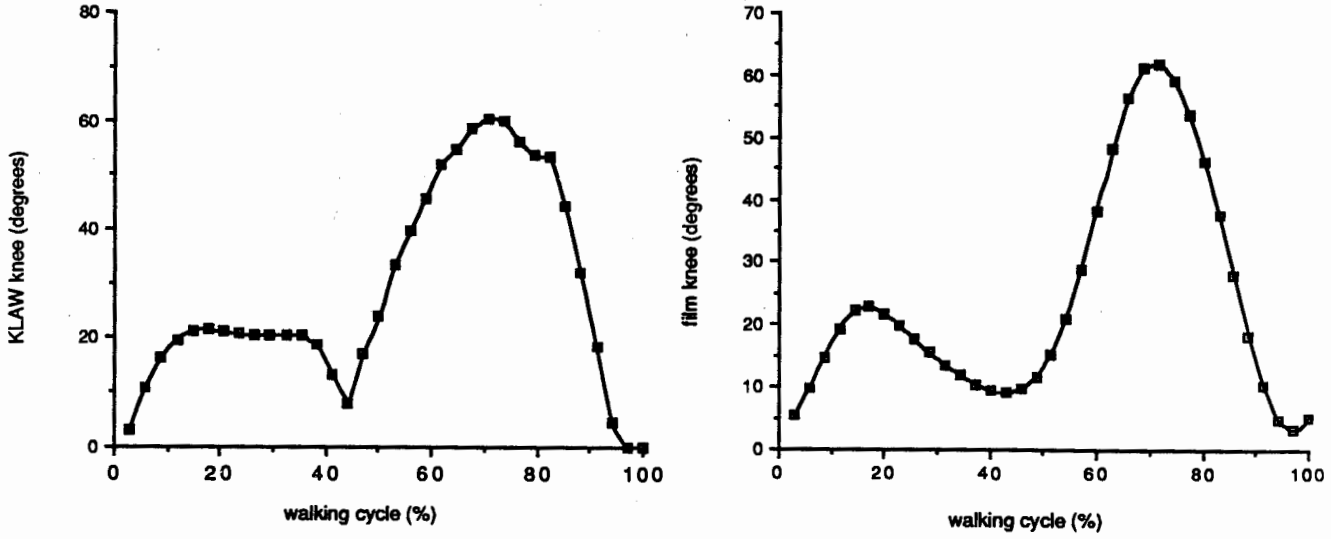


Figure 7-3: Comparison of knee angles during a locomotion cycle. Left: angles as calculated by KLAW; right: angles obtained from a walking human subject [Winter 79].

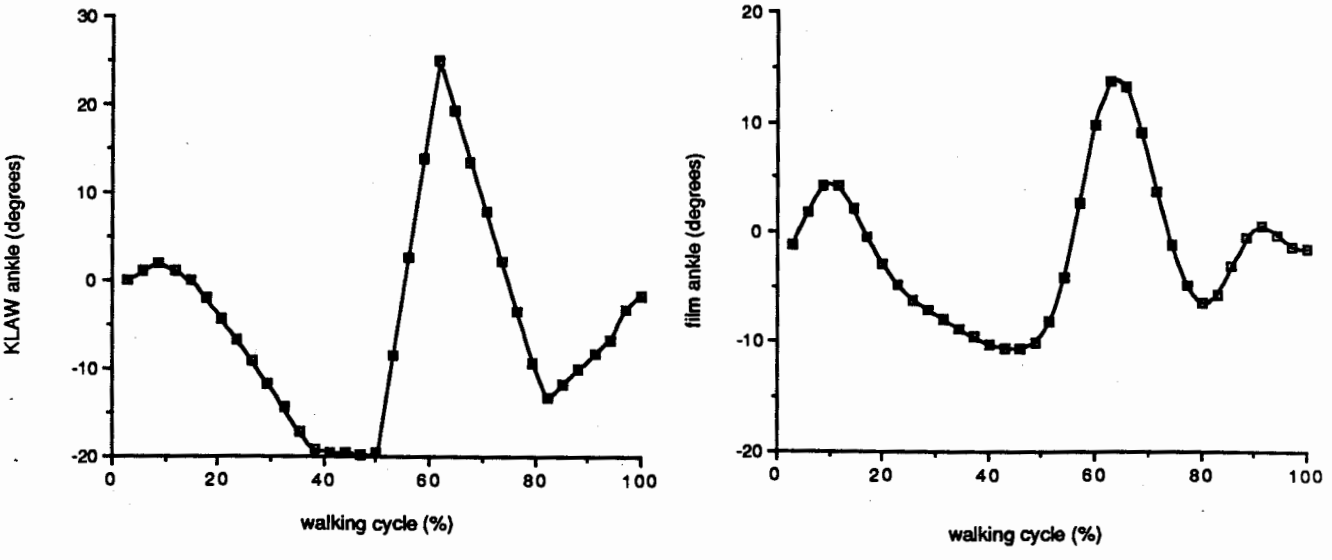


Figure 7-4: Comparison of ankle angles during a locomotion cycle. Left: angles as calculated by KLAW; right: angles obtained from a walking human subject [Winter 79].

leg is fully lengthened. The peaks signal that the leg extends too fast, which is caused by a quickly increasing leg axis force F_{ω}, that reaches a maximum at 40 % of the cycle time (see figure 7-7). Possibly an earlier activation of the STANCE3 phase (set to start at the end of the SWING1 phase as defined in section 6.1, in this case about 35 % of the cycle), or a smaller value for pa_{3_inc} could alleviate the problem. The relatively flat shape of the KLAW knee angle curve during stance from about 10-30 % of the cycle is produced by a stiff dynamic leg spring. A lower spring constant k_{ω} can help to more closely match the knee angle of the real walk. The magnitude of the KLAW ankle plantarflexion at the end of the double support phase, i.e. at toe_off (60 % of the cycle), is significantly larger than for the "film" ankle. This is not unnatural, as experimental data [Inman 81] show similarly large values (between 20 and 30 °) at comparable walking speeds. It should be noted that the deviations of the KLAW angles from the "film" angles are quite minor and hardly visible in an animation of a human figure. Furthermore, the measurements of articulated motion are not free of noise, and it is difficult to obtain simultaneous data of motion in all three reference planes (sagittal, coronal and transverse). The walking algorithm, on the other hand, includes motions such as pelvis rotation, pelvic list and lateral displacement of the body, which are only indirectly represented in figure 7-2, 7-3 and 7-4 (see also figure 7-12).

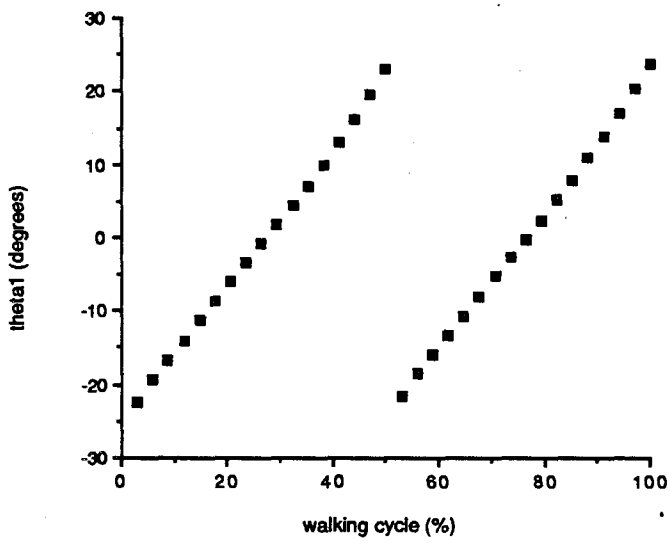


Figure 7-5: Dynamic hip angle θ_1 for walking cycle W.

Figure 7-5 shows the very regular behavior of the dynamic hip angle θ_1 over the duration of two rhythmic walking steps: the angle increases from -25 ° at heel strike to 25 ° at the heel strike of the other leg.

Towards the end of a step (40-50 % of the cycle), the curve bends upwards slightly, which represents an increase in the angular velocity. This corresponds to an acceleration of the center of mass as discussed in section 5.3.

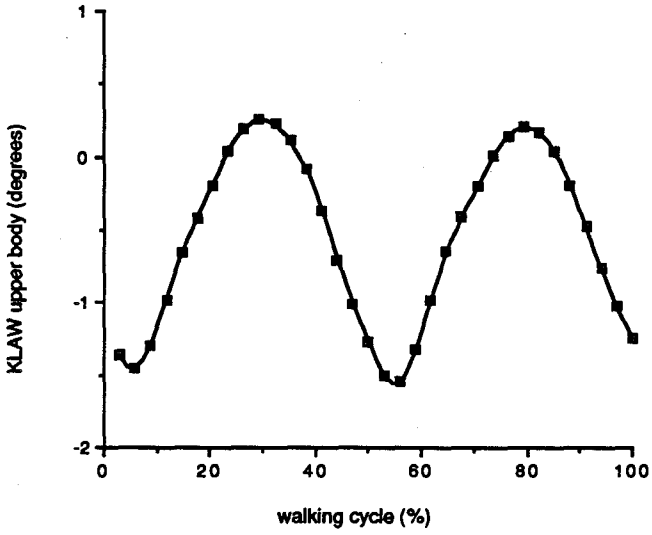


Figure 7-6: Upper body angle θ_2 (sagittal plane) for walking cycle *W*; a positive value corresponds to a forward tipping of the body.

In figure 7-6, the upper body angle θ_2 is displayed as calculated by the dynamics. Shortly after heel strike (at 0 % and 50 %) of the walking cycle, the upper body tilts forward as a result of the impulse. However, the net change in the upper body angle is very small.

Figure 7-7 shows the leg axis force F_ω and the leg torque F_{θ_1} during stance. The dynamic stance phase for a leg ends with each step, since the other leg takes over at heel strike. Therefore the force and torque pattern repeat with each step (this holds also for the swing phase torques illustrated in figure 7-8). F_{θ_1} follows a regular pattern; as described in section 6.1, it acts as a step input torque for about 20 % of the cycle time during a walking step. The value for F_ω never drops down to zero, which means that the SWING2 phase was not activated (i.e. the leg was never locked) and STANCE4 had to occur just at heel strike of the other leg. Shortly after heel strike, F_ω increases to account for the impulse, which is achieved by an internal increment in the position actuator pa_1. The end of the SWING1 phase is reached at about 35 % of the cycle, which causes another temporary increase in F_ω when STANCE3 sets in to lengthen the dynamic leg to its desired value ω_{des} at heel strike.

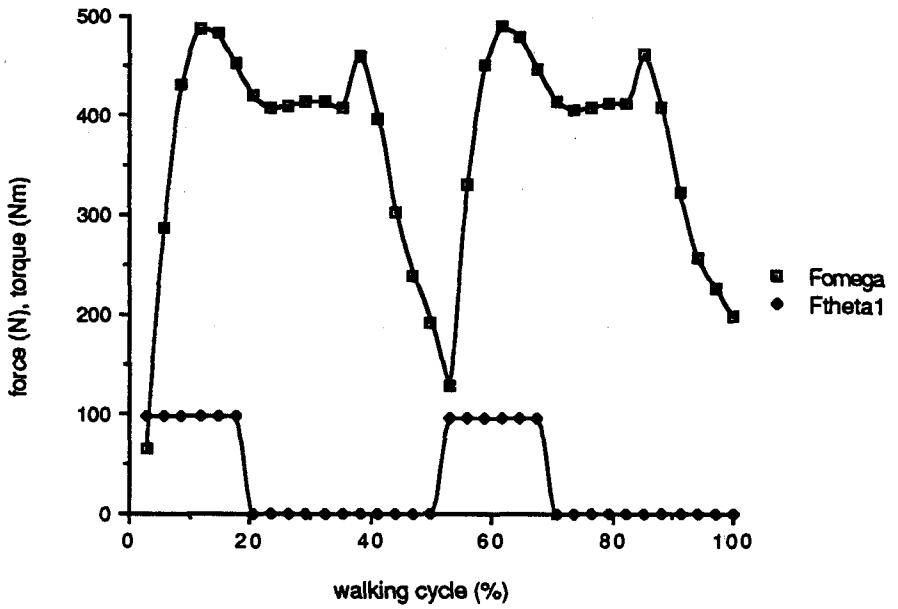


Figure 7-7: Leg force F_{ω} and torque F_{θ_1} during stance for walking cycle W.

Figure 7-8 illustrates the torques at the swing leg. A knee torque F_{θ_4} develops only during SWING2 from 35 % until about 48 % of the cycle for the first step; it sets in with a maximum (negative extension) magnitude and decreases according to the exponential function in equation (6.3). F_{θ_3} follows the same pattern during SWING1, from about 12 % (after double support with the beginning of the swing phase) until 35 % of the cycle time. In the SWING2 phase, the magnitude of F_{θ_3} is determined by a spring and damping construct, which causes irregularities in the profile particularly at the phase change from SWING1 to SWING2 (see at about 80 % of the cycle time, where F_{θ_3} actually becomes positive, which indicates a flexion moment). The irregular behavior of F_{θ_3} is caused by the sudden extension torque at the knee, which has to be compensated in order to keep the hip angle constant.

Figure 7-9 displays the motion of one leg for a complete walk (including starting and stopping). The sinusoidal path of the hip (i.e. the vertical displacement of the center of mass) is more pronounced for the faster walk during the rhythmic period, which is in accord with observations by Inman [Inman 81] on real human walks. The diagrams of figure 7-10 show the leg angles at each frame for the same walking sequence (for the same leg). It can be seen that the maximum flexions of the hip, ankle and meta joints (ankle and meta joint plantarflex) at toe-off are bigger during the second cycle, where the full velocity of 5 km/h is reached,

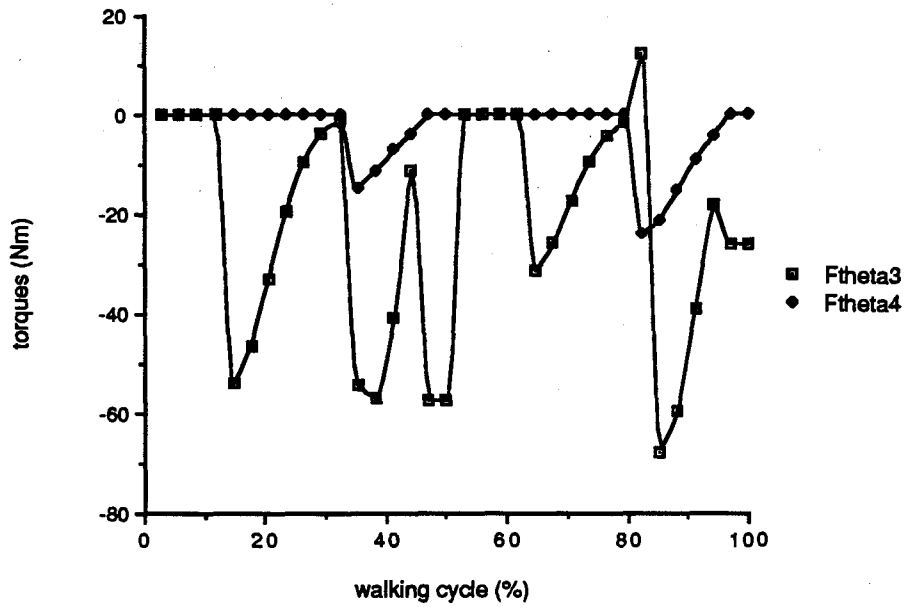


Figure 7-8: Hip torque F_{θ_3} and knee torque F_{θ_4} during swing for walking cycle W.

than in the first and third cycle, which are characterized by acceleration and deceleration, respectively. Only the maximum knee angle stays fairly constant in all three cycles. This is in conformity with studies on real human walking [Inman 81] (see also the discussion of the meta-off phase, section 5.4). (It is noted, that the walking sequence of figure 7-9, bottom, and 7-10 is also printed on the right margin of this write-up in reverse order.)

A comparison of hip and knee angles for 3 walks with different step length and step frequencies, but the same speed, is given in figure 7-11 over the duration of one rhythmic walking cycle. Photographic Images of the same walks are presented in appendix F (see figure F-2, F-3 and F-4). An analysis is left to the reader.

Figure 7-12 illustrates the effect of pelvic list in the coronal plane on the leg angles in the sagittal plane. Between 40 % and 70 % of the walking cycle, the hip joint of the walk with an accentuated pelvic list (3-3) does not flex as much as the hip joint during a normal walk (3-1). In other words, the bigger the pelvic list, the more the thigh of the hind leg extends forward towards the end of the double support phase. Of course, at heel strike, both the hip and the knee angles must be the same because the walks are based on the same locomotion parameters. The knee flexion of (3-3) is increased from about 55 % to 75 % of the cycle compared to (3-1) to account for the lower hip position.

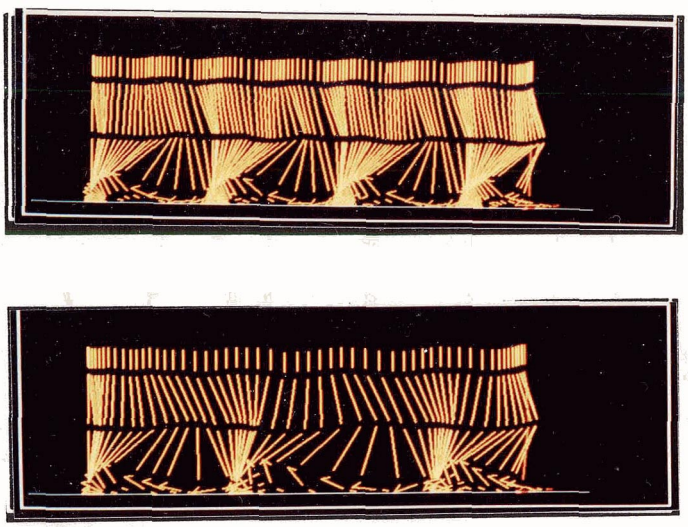
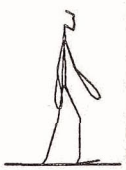


Figure 7-9: Motion of leg for two complete walking sequences at different speeds (only one leg is displayed). Both walks cover about the same distance from the start to the end.
Top: 2 *km/h* , 204 frames, 4 walking cycles (2nd and 3rd cycles are rhythmic).
Bottom: 5 *km/h* , 108 frames, 3 walking cycles (2nd cycle is rhythmic).



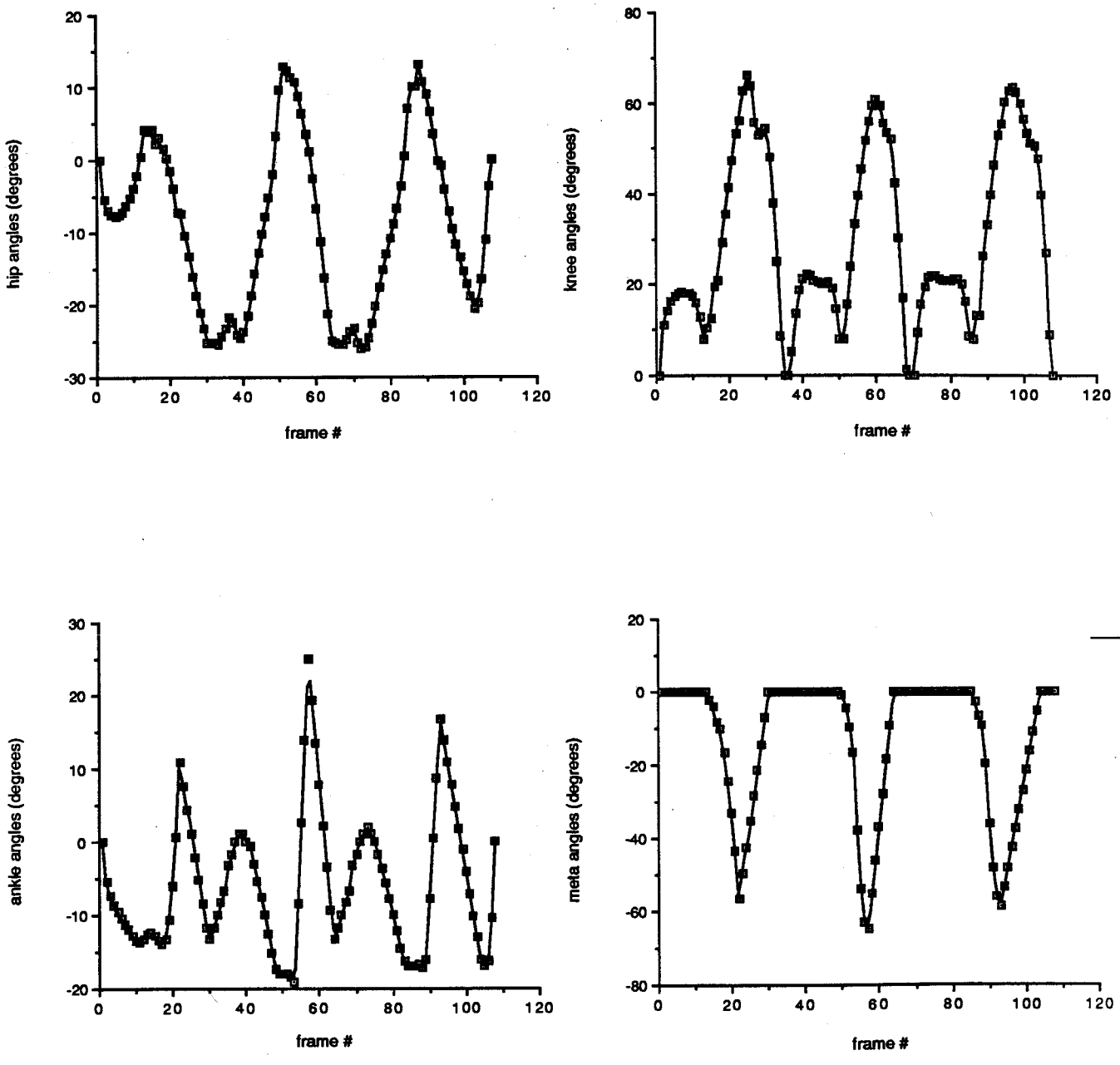


Figure 7-10: Hip, knee, ankle and meta angles for a complete walking sequence (including starting and stopping) of 108 frames, which corresponds to the walk of figure 7-9, bottom.

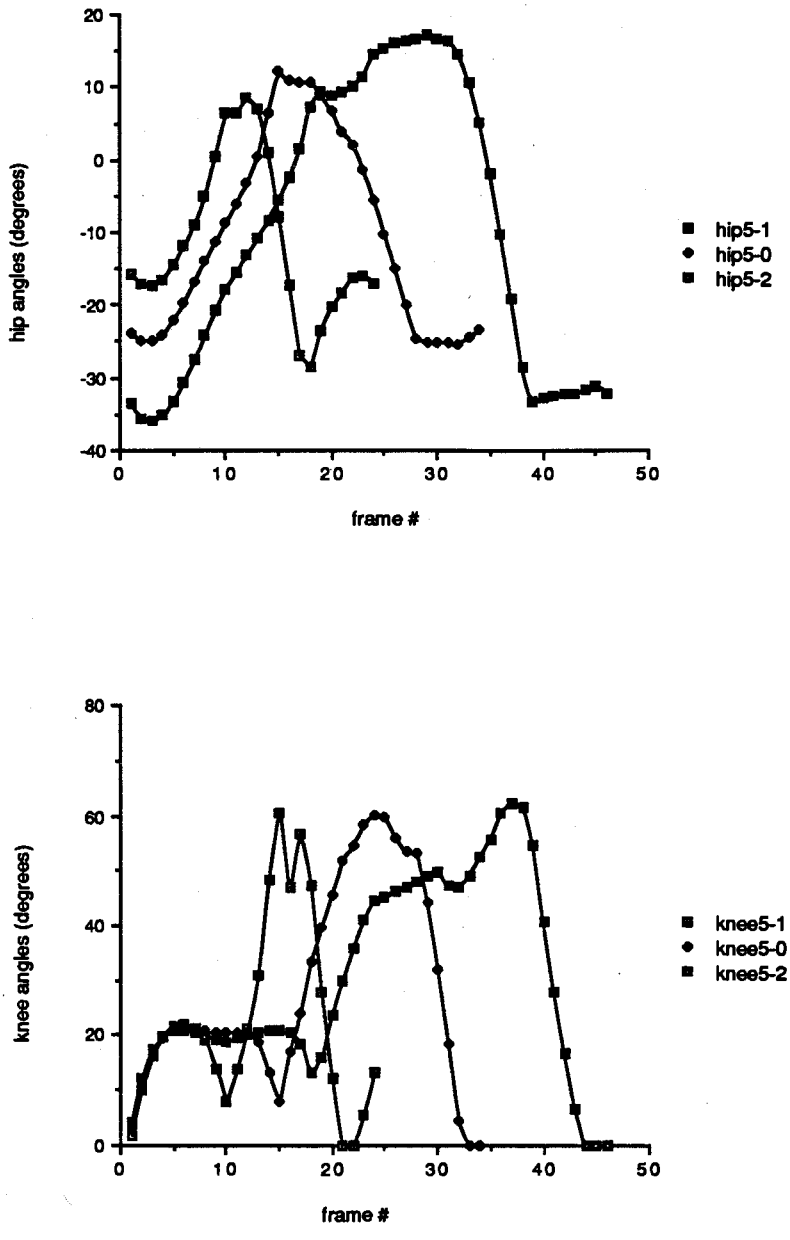


Figure 7-11: Hip and knee angles for 3 walks with the same speed, $v = 5\ km/h$, but different step lengths ($sl_{5-1} = 0.5\ m$, $sl_{5-0} = 0.77\ m$ and $sl_{5-2} = 1.05\ m$) and different step frequencies ($sf_{5-1} = 166.7\ steps/min$, $sf_{5-0} = 107.5\ steps/min$ and $sf_{5-2} = 79.4\ steps/min$). The number of frames for each walking cycle w are: $w_{5-1} = 23$, $w_{5-0} = 34$ and $w_{5-2} = 46$. Walk 5–0 is a "natural" walk.

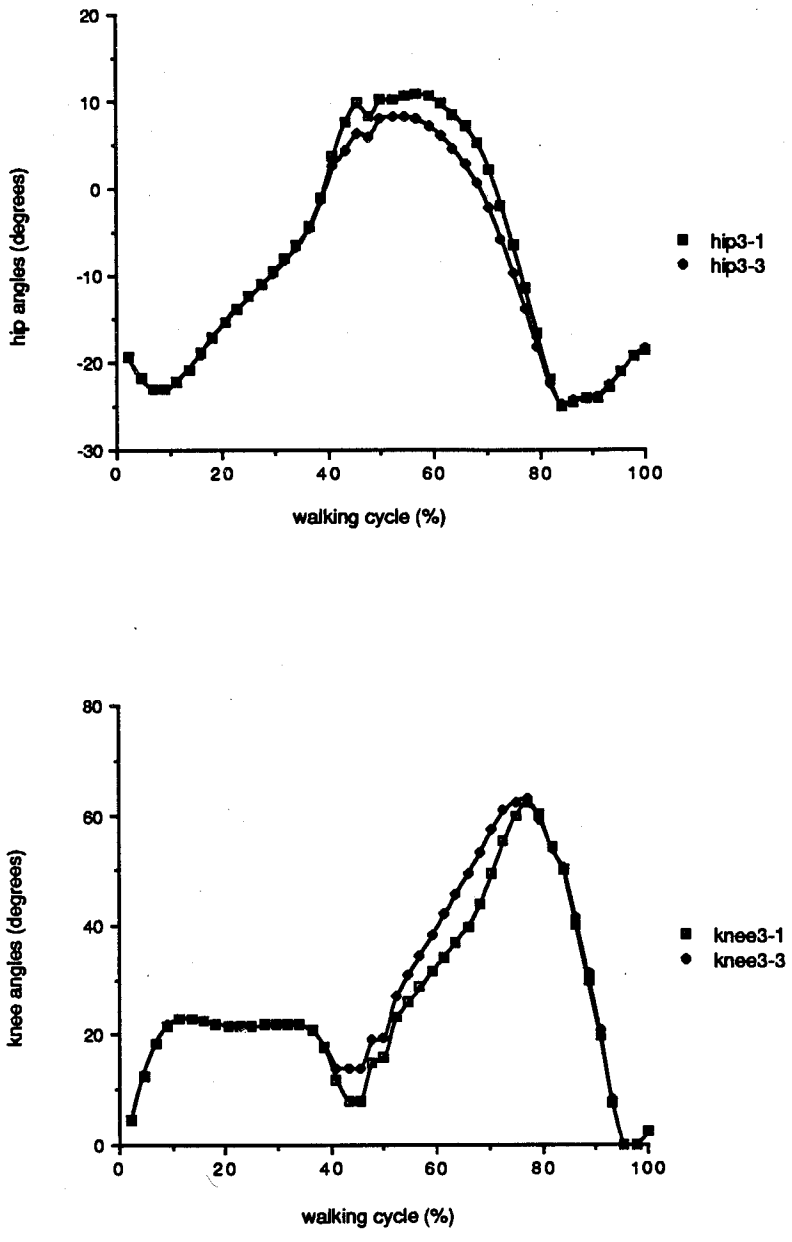


Figure 7-12: Hip and knee angles of 2 walks with the same locomotion parameters ($v = 3\ km/h$, $sl = 0.6\ m$ and $sf = 83.3\ steps/min$), but a different value for the `pelvic_list_factor` locomotion attribute: $\texttt{pelvic\_list\_factor}_{3\text{–}1} = 1$, $\texttt{pelvic\_list\_factor}_{3\text{–}3} = 3$. A rhythmic walking cycle is assumed.

Chapter 8
Conclusions

The principal objective of this thesis was to design a control algorithm for the animation of bipedal locomotion. The KLAW system which has been implemented can generate a large variety of walking sequences and the two sub-goals of developing a high-level control scheme and producing realistic motions have been successfully translated into action. A natural animation of a walking human figure is obtained by a hierarchical decomposition of a task specified by a few walking parameters. The decomposition is achieved by combining several motion control techniques in an elegant way, eliminating a lot of the disadvantages that occur in the separate application of any one of these techniques (as described in chapter 2).

It has been shown that a simple dynamic model is sufficient to simulate a generic locomotion pattern. The effects of gravity, ground collisions and active muscle moments during a human walking cycle are adequately accounted for by the system. The simplicity of the dynamic model makes it possible to determine the governing forces and torques based on internal knowledge and rules about locomotion. This eliminates a major problem attached to applying traditional forward dynamics to computer animation: rather than having to specify the cause of a motion in terms of forces and torques, the user can select a number of more obvious motion parameters such as velocity and step length. From a hierarchical viewpoint, the equations of motion provide the low-level, natural execution of the leg movements. They symbolize the functional synergies that are controlled by the motor program for walking as discussed in section 3.2. The motor programs embody the rules about a locomotion cycle, which are incorporated at several levels in the motion control hierarchy. High-level concepts such as step-symmetry and state-phase timings define the boundary conditions for a specific walking configuration. Rules at the middle level govern the coordination of the legs; this involves the proper sequencing of the leg-phases. At the bottom level, knowledge about walking leads to a subdivision of the original phases and the calculation of the force and torque profiles. Kinematic rules about the motion of the foot, leg, pelvis and upper body are integrated into the dynamic simulation.

Goal-directed and dynamic approaches to motion control have been tied together in this thesis. Although the implementation was done specifically for bipedal walking, we believe that the basic ideas can

be extended and applied to other forms of articulated motion as well. 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). Therefore, the ultimate goal is to built 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 (see also figure 3-2 (b)). The usefulness of such a system would greatly depend on the choice of parameters assigned to each task.

The KLAW system developed in this thesis can be integrated into current animation systems to produce sequences of walking. For instance, in a keyframing system such as the *V-2000* system of Vertigo, an initial and final key position of a figure could be specified and the KLAW system would then generate a walk in-between. Some of the extensions mentioned below (e.g. the possibility of changing directions) would be necessary to make this symbiosis more practical.

A direct extension of the algorithm would be the inclusion of other bipedal gaits and locomotion with more than two legs. Suggestions are given in sections 3.2, 5.1 and 5.2. Whereas running, for example, would require a modification of the dynamic model 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. The algorithm was designed for legged locomotion on level ground. However, we think that walking up and down inclines and stairs could be implemented in a straightforward manner based on the same principles. For instance, walking up stairs would involve the definition of a new local motor program (see figure 3-2) for the swing phase, whereby the feedback mechanism for foot clearance (discussed in section 6.2.2) could be adapted to move the leg from step to step; for walking downhill, a greater knee flexion during stance would have to be accounted for in the local motor program for the stance leg by reducing the stiffness of the dynamic leg spring.

The usability of the algorithm could be improved by a better user interface, where the locomotion parameters (i.e. velocity, step length and step frequency) and possibly the locomotion attributes[E] would be specified and modified interactively as functions of time. In the current version of KLAW, it is only possible to generate walks with an acceleration period at the beginning and a deceleration at the end of the locomotion sequence. By adding functions for the direction of locomotion, KLAW could be interfaced with path-planning algorithms. A change in direction could be implemented by adding "kinematic" degrees of freedom (as in the case of the leg-determinants) to enable the legs to swing slightly sideways.

Is is also desirable to make the algorithm more efficient in order to come close to real-time performance. A significant speed-up could be expected if a faster integration method was used (the LSODI [Hindmarsh 80] FORTRAN routine does a lot of extra, time consuming checking). Also, it would be worthwhile to examine the numerous trigonometric expressions used in calculating the kinematic gait-determinants for a more efficient formulation.

Appendix A
Terminology

The following is an alphabetically ordered list of frequently used terms from Computer Animation, Physics and Artificial Intelligence. They are explained or defined in the way in which they are used in the context of this thesis.

Animation

This literally means 'to bring to life'; it is the process of representing changes of movements over time. Generally, the motion is generated through the rapid display of successive images, fast enough (24 frames/sec for film, 30 frames/sec for video), that the human eye interprets them as continuous motion. Originally, animation was restricted to 2-D (Walt Disney, Warner Brothers) or 2½-D (pictures are drawn on parallel planes, that can be moved against each other).

When computer technology was ready to be applied to animation, it first served mainly as an *assisting* tool. The computer supported color drawings, the calculation of in-betweens and film-recording.

Today, 3-D computer animation is common practice and is an entirely different medium. Often termed *modeled* animation, it supplies the animator with a whole new set of tools for the modeling-animating-rendering process. 3-D models, color, shading, movement of camera (viewpoint) and other techniques greatly enhance realism.

Articulated Body

The body is made up of segments or links (usually rigid) connected by joints. The motion of the segments relative to each other is somewhat restricted. The human body is expressed as such an articulated body. It can be represented as a tree structure (transformation tree) where the segments (joints) are taken as nodes connected by joints (segments). Each node has its own coordinate system (transformation matrix).

Nodes located more deeply in the tree will be transformed by matrices from ancestor nodes, when the tree is traversed, e.g. a wrist-node gets transformed by the elbow-node above it, if the lower arm is moved.

Each joint has one to three rotational DOF (up to two degrees of constraints). Joints with 1 DOF are called *hinges* (e.g. elbow), with 2 DOF they are called *universal joints* (e.g. finger) and with 3 DOF, joints are referred to as *ball joints* (e.g. shoulder). The total number of DOF of the body is the sum of the DOF at each joint plus the 6 DOF of an initial joint (position and orientation of the body as a whole) which "connects" the body to the world.

Center of Mass

Traditionally, the center of mass of a system is defined as the sum of the products of all its particle masses and the distance from the origin divided by the total mass of the system. With articulated bodies, the calculation of the center of mass is simplified, since the mass of each segment can be expressed as a function f_i of the total mass M (e.g. thigh/body= 0.1, total arm/body= 0.05). The center of mass x therefore is:

$$x = \frac{f_1 M x_1 + f_2 M x_2 + \cdots + f_n M x_n}{M} = f_1 x_1 + f_2 x_2 + \cdots + f_n x_n$$

In locomotion of articulated bodies the center of mass constantly changes, so it has to be recalculated for each time-step. It plays an important role in balancing the body.

Conservative System

In this context, a physical system (e.g. a double pendulum) is conservative, if no external forces and torques are acting on it except gravity. An exact definition is given below.

If the work done by a force in moving an object from one position to another only depends on these two positions but not on the path followed, the force is said to be conservative and the system is referred to as a conservative system. For example, if a block is raised from the ground to a height h, the work done on the block by the gravitational force is mgh. Since there is no work by gravity in the horizontal direction, the work done on the block to lift it to height h remains the same, whether it is moved up vertically in a straight line, zig-zag or along any arbitrary path. Hence, gravity is a conservative force and, if air resistance is

neglected, the system is conservative. But if the block is pushed horizontally on a table from A to B, the work on the block is done against a frictional force, which is always directed opposite to the motion. Now the amount of work depends on the path taken between A and B, in that the longer the path is, the more work has to be done. In this case, the system is non-conservative and friction represents a non-conservative force.

Degrees of Constraint

A single particle in space has 3 DOF. If its motion is restricted to a line ($y = ax + b$, $z = 0$), one coordinate is sufficient. The particle is said to have 2 degrees of constraint and therefore only 1 DOF.

Degrees of Freedom (DOF)

DOF are the number of independent parameters (not including time) required to completely specify the position of every part of the system. For example, a single particle has 3 DOF in space, a segment or rigid body has 6 DOF (3 translational coordinates for position and 3 rotational ones for orientation), a double pendulum with moving reference point free to move in space has 9 DOF (3 translational coordinates for position and 6 rotational ones for orientation).

Differential Equations of Motion

Dynamic analysis of a system involves expressing its behavior over time. For example, if we have a particle of mass m fastened to a spring with force constant k, and the displacement of the particle is x, then the behavior of this system is formulated as

$$m\ddot{x} + kx = 0,$$

which is derived from Newton's 2nd law of motion, $F = ma$ by letting $F = -kx$ (Hooke's law) and $a = \ddot{x}$, the 2nd derivative of the displacement over time.

The above equation is called the differential equation of motion for the system, which can be set up in various ways (Lagrange, Newton, D'Alembert, Hamilton) [Wells 67]. Except in the Hamiltonian form, they are all of 2nd order.

In this simple case, only one coordinate (displacement in x-direction) is considered. Usually, in

particular when dealing with articulated bodies, the number of DOF is large, thus resulting in many equations of motion, since there is one equation for each DOF. Because the equations are non-linear, complex and coupled most of the time, a solution (to the direct dynamics problem[A]) can only be found by numerical methods.

Duty Factor

The duty factor of a foot is the fraction of the duration of the stride for which this foot is on the ground.

Dynamics

This is the study of changes in motion due to forces and torques. In Biomechanics [Winter 79], this discipline is often called *kinetics*. Considering the human body, internal forces (caused by muscular activity, ligaments, frictions in muscles and joints) as well as external forces (ground reaction forces, collisions with active and passive bodies) have an influence on movement. Classical dynamics is based on Newton's 3 laws of motion. Equations of motion can be solved in 2 directions:

direct (forward) dynamics problem
: given the forces and torques, solve for the accelerations, i.e. determine the resulting motion of the system. In Computer Animation, dynamics is usually applied in this form.

inverse dynamics problem
: given the motion of a system (i.e. the accelerations), solve for the forces and torques that produce these accelerations. In Biomechanics, one often has to resort to this approach, since the measurements of forces and torques (which have to be known for direct dynamics to be applied) are mostly inaccurate and difficult, if not impossible, to measure. On the other hand, positional data (i.e. accelerations) can be readily gathered. Thus, via inverse dynamics, the forces and torques can be calculated.

Frames

A frame-based representation scheme specifies a knowledge base[A], which is a collection of frames logically connected and organized in some (mostly) hierarchical manner. A frame [Minsky 75] is a data structure for representing stereotypical situations, such as being in a certain kind of living room or going to a child's birthday party. The salient idea behind this is to group pieces of knowledge into *microworlds*, which may be useful for understanding a particular concept. Each frame has terminals or *slots* for attaching pointers

to substructures. Different frames can share slots, e.g. the frames for a table and a chair might share the slot for a leg, if both have legs of the same geometrical nature. Terminals of a frame are usually stored with "weakly" bound default assignments, which can be updated or replaced for a particular instance.

Gait

Different styles of locomotion depend on speed and on the structure of the animal. Gaits can be classified as *walks* and *runs*. Walks have duty factors[A] ≥ 0.5 (for a certain time during a stride all feet are on the ground), whereas runs have duty factors < 0.5 (there is a time during a stride when all feet are off the ground). A gait is called symmetric, if the left and right feet of each pair have equal duty factors and relative phases[A] differing by 0.5 . Thus human walking is a symmetric gait. Also amble, trot and pace are symmetric; gallop, hopping or canter are asymmetric [Alexander 84].

Kinematics

This is the analysis or description of movements of bodies or parts of bodies in time and space independent of the forces, that cause these movements [Wilhelms 85, Dagg 77]. It involves the calculation of linear and angular displacements, velocities and accelerations. The kinematics of articulated bodies[A] results in the following two related problems:

direct (forward) kinematics problem
: find the position of a distal segment of the body given the joint angles of the proximal segments, e.g. find the position of the foot given the angles of the hip, knee and ankle.

inverse kinematics problem
: find all the joint-angles of the proximal segments required to produce a particular absolute position of a distal segment, e.g. given the position of the foot, find the joint-angles at the ankle, knee, hip, which will place the foot at that position. Inverse kinematics finds applications in robotics, where, for instance, the robot hand has to be placed at a certain known position in order to grasp something; or in locomotion to place and hold a foot at a position on the ground, while the body is passing over it. With articulated bodies the inverse kinematics problem can be very difficult to solve [Girard 85].

Knowledge Base

The central issue in knowledge representation is to find proper internal representations for expert knowledge along with associated knowledge handling facilities to create *intelligent* systems.

The different notations for representing knowledge are referred to as representation schemes. By making use of such a representation scheme, a knowledge base can be specified, which is basically a model of the world containing various facts. There are different classes of representation schemes: logical, network, procedural, frame-based, etc. [Mylopoulos 83]. A procedural representation scheme, for instance, can specify a production system. The knowledge base would consist of a number of production rules and a database. The production rules are of a "if *pattern* then *action*" type. If the pattern matches some fact in the database, the action is executed.

Moment of Inertia

See Radius of Gyration.

Particle

An imaginary bit of matter (visually a dot), so small, that it is totally determined in space by 3 coordinates.

Radius of Gyration (γ)

The distance from any given axis of rotation, at which the mass of a rigid body could be concentrated without changing the moment of inertia, I, about that axis of rotation. For the application here, the moment of inertia about the center of mass, I_0, is important. In this case, γ is such that 2 equal point masses (located at a distance γ at either side of the center of mass) have the same moment of inertia as the original distributed mass:

$$I_0 = m\gamma^2 = \frac{m}{2}\gamma^2 + \frac{m}{2}\gamma^2$$

It should be noted that the moment of inertia about any other axis of rotation (usually the proximal or distal end of a segment), which is parallel to the axis of I_0, can be determined by applying the Parallel Axis Theorem [Winter 79]:

$$I = I_0 + mx^2,$$

where x is the distance between the center of rotation and the center of mass.

Relative Phase

The setting down of a foot with respect to the setting down of an arbitrary chosen reference foot, expressed as a fraction of the duration of the stride. The reference foot is assigned relative phase 0 and the others have relative phases in the range 0-1. Alexander [Alexander 84] presents various examples.

Rigid Body

Defined by a number of points, that must move together as a whole; they may not move relative to each other.

Simulation

In this context, simulation is defined as the process of studying the behavior of a physical system under various conditions using a computer model. A model is an internal representation of a system where mathematical equations define its components and their interactions [Miller 75]. The graphics is completely decoupled from the model. In simulation as opposed to animation, a system is described by governing dynamical equations, whereas animation is frequently associated with kinematics. Here, simulation is treated as a particular animation technique, where motion control is determined by dynamic equations of motion.

Step

During each stride for bipedals, there are 2 steps, one left and one right. A step is the movement of a single leg from the time it is lifted off the ground until its next lift-off. A step is divided into stance and swing phases.

Stride

Complete cycle of leg movements, e.g. from the setting down of a particular foot to the next setting down of the same foot.

Appendix B
Body Data

B.1. Anthropometric Data

The anthropometric data shown below are used for dynamic analysis and lower body kinematics, respectively. It is claimed in this context that in determining the dynamics of the stance leg and the shank of the swing leg, the segment weights, centers of mass and radii of gyration[A] (moments of inertia) include the physical properties of the foot. Data for the lengths of segments were drawn from various anatomy books, all other measurements are borrowed from a book by Winter [Winter 79].

for segment i,

l_i = segment length / body height
m_i = segment weight / body mass
r_i = center of mass / segment length
γ_i = radius of gyration [A] / segment length

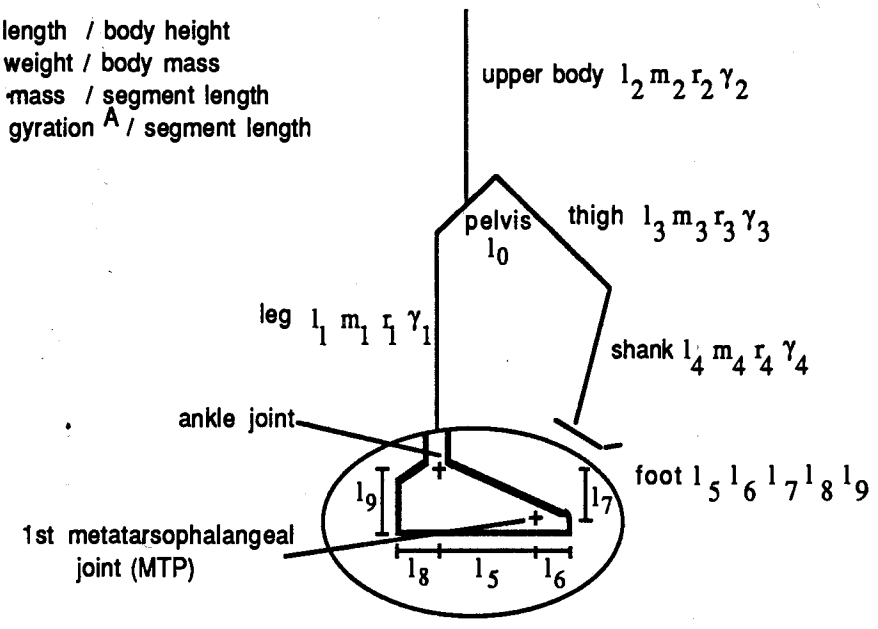


Figure B-1: Indices for anthropometric values of lower body segments.

Table B-1 lists the *relative* anthropometric data for each of the segments illustrated in figure B-1. 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	-	-	-
ankle-MTP*	7	0.02367	-	-	-
hind foot	8	0.02959	-	-	-
ankle-footbase*	9	0.03846	-	-	-

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

The *absolute* anthropometric data, including the moments of inertia I_i, are calculated 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 $75\,kg$, the following values result for the thigh:

$$
\begin{aligned}
l_{3_{abs}} &= 0.426\,m \\
m_{3_{abs}} &= 8\,kg \\
r_{3_{abs}} &= 0.184\,m \\
\gamma_{3_{abs}} &= 0.138\,m \\
I_3 &= m_{3_{abs}} \, (l_{3_{abs}} \cdot \gamma_{3_{abs}})^2 = 0.0276\,kg\,m^2
\end{aligned}
$$

B.2. Human Walking Figure Data

The following data structure is updated by the KLAW system for each frame ($\frac{1}{30} sec$) to animate a human walking figure using a left-handed coordinate system. Rotations about the x, y, z axes are denoted by the suffix *_rx, _ry, _rz*, respectively.

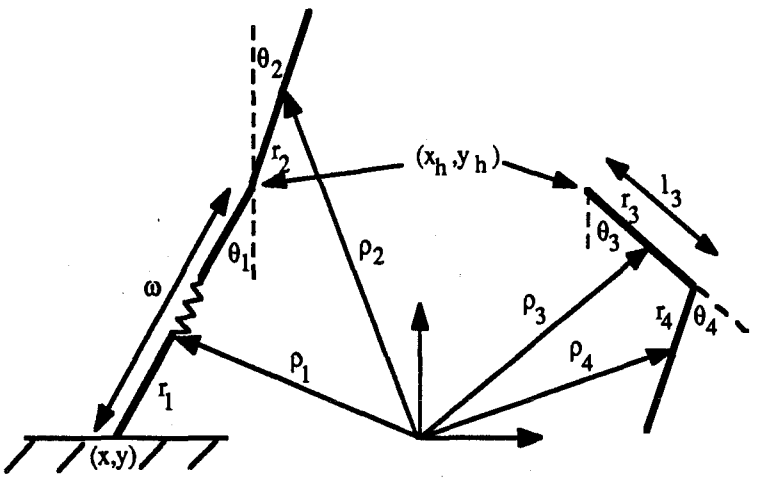
```
typedef struct  walk_data
{
    float       time;           /* current time                          */
    int         state;          /* state (single or double support)      */
    int         cycle;          /* cycle (develop, rhythmic, decay)      */
    int         right_phase;    /* leg phase (STANCE1,2,3,4; SWING1,2,3 */
    int         left_phase;
    float       body_tx;        /* reference position of the body        */
    float       body_ty;        /* (located halfway between hips)        */
    float       body_tz;
    float       pelvis_ry;      /* pelvic rotation                       */
    float       pelvis_rx;      /* pelvic list                           */
    float       right_hip_rz;   /* θ3                                    */
    float       right_knee_rz;  /* θ4                                    */
    float       right_ankle_rz; /* θ5                                    */
    float       right_meta_rz;  /* θ6                                    */
    float       left_hip_rz;
    float       left_knee_rz;
    float       left_ankle_rz;
    float       left_meta_rz;

                                /* distribution of upper body angle θ2   */
    float       lumbar1_rz;
    float       lumbar2_rz;
    float       lumbar3_rz;
    float       lumbar4_rz;
    float       lumbar5_rz;
                                /* distribution of compensation for      */
                                /* pelvis_rx                             */
    float       right_hip_rx;
    float       right_ankle_rx;
    float       left_hip_rx;
    float       left_ankle_rx;
    float       lumbar1_rx;
    float       lumbar2_rx;
    float       lumbar3_rx;
    float       lumbar4_rx;
    float       lumbar5_rx;
                                /* distribution of compensation for      */
                                /* pelvis_ry                             */
    float       right_hip_ry;
    float       left_hip_ry;
    float       lumbar1_ry;
    float       lumbar2_ry;
    float       lumbar3_ry;
    float       lumbar4_ry;
    float       lumbar5_ry;
    float       thorax1_ry;
    float       thorax2_ry;
    float       thorax3_ry;
    float       thorax4_ry;
    float       thorax5_ry;
```

```
    float       thorax6_ry;
    float       thorax7_ry;
    float       thorax8_ry;
    float       thorax9_ry;
    float       thoraxl0_ry;
    float       thoraxl1_ry;
    float       thoraxl2_ry;
    float       cervical1_ry;
    float       cervical2_ry;
    float       cervical3_ry;
    float       cervical4_ry;
    float       cervical5_ry;
    float       cervical6_ry;
    float       cervical7_ry;
    float       right_shoulder_ry;
    float       left_shoulderight_ry;

                                /* arm  swing                          */
    float       right_shoulder_rz;
    float       left_shoulderight_rz;
    float       right_elbow_rz;
    float       left_elbow_rz;
} WALKDATA;
```

Appendix C
Derivation of Equations of Motion



a) stance leg model with upper body (inverted double pendulum with springy leg)

b) swing leg model (double pendulum)

Figure C-1: Dynamic models for simulating different phases in legged locomotion; θ_3 is assumed to be negative in the given configuration, all other angles are positive.

The equations of motion for the stance and swing phase are derived below using the Lagrange method. The vectors ρ_i $(i = 1, \ldots, 4)$ extend from the inertial frame to the centers of mass of each segment i as described in section 3.4.1. The values of the anthropometric data are given in appendix B. The actual length of the springy stance leg is denoted by ω; initially $\omega_0 = l_1$. The I_i are the moments of inertia around the center of mass as calculated in appendix B.

C.1. Stance Phase

The Lagrange equation for the stance phase can be written (see section 3.4.2) as

$$\frac{d}{dt}\left(\frac{\partial T_{st}}{\partial \dot{q}_{st_i}}\right) - \frac{\partial T_{st}}{\partial q_{st_i}} + \frac{\partial U_{st}}{\partial q_{st_i}} = F_{q_{st_i}} \quad , \text{ where } q_{st} = [\,\omega, \theta_1, \theta_2\,]^T, \quad i = 1, 2, 3 \; . \tag{C.1}$$

The kinetic energy of the system (see figure C-1 (a)) is expressed by

$$T_{st} = \tfrac{1}{2} m_1 \dot{\rho}_1^2 + \tfrac{1}{2} m_2 \dot{\rho}_2^2 + \tfrac{1}{2} I_1 \dot{\theta}_1^2 + \tfrac{1}{2} I_2 \dot{\theta}_2^2 \; ,$$

where the squares of the translational velocities of the centers of mass $\dot{\rho}_1^2$ and $\dot{\rho}_2^2$ are calculated as follows:

$$\rho_1 = [\,x + r_1 \sin\theta_1 \,,\; y + r_1 \cos\theta_1\,]^T$$

$$\dot{\rho}_1 = [\,\dot{x} + r_1 \dot{\theta}_1 \cos\theta_1 \,,\; \dot{y} - r_1 \dot{\theta}_1 \sin\theta_1\,]^T$$

$$\dot{\rho}_1^2 = \dot{x}^2 + \dot{y}^2 + r_1^2 \dot{\theta}_1^2 + 2 r_1 \dot{\theta}_1 (\dot{x} \cos\theta_1 - \dot{y} \sin\theta_1)$$

Since the foot remains fixed on the ground during stance, i.e. $\ddot{x} = \ddot{y} = \dot{x} = \dot{y} = 0$, this reduces to

$$\dot{\rho}_1^2 = r_1^2 \dot{\theta}_1^2 \; .$$

Similarly,

$$\rho_2 = [\,x + \omega \sin\theta_1 + r_2 \sin\theta_2 \,,\; y + \omega \cos\theta_1 + r_2 \cos\theta_2\,]^T$$

$$\dot{\rho}_2^2 = \dot{\omega}^2 + \omega^2 \dot{\theta}_1^2 + r_2^2 \dot{\theta}_2^2 - 2\dot{\omega} r_2 \dot{\theta}_2 \sin(\theta_2 - \theta_1) + 2\omega r_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_2 - \theta_1)$$

Therefore, the kinetic energy becomes

$$T_{st} = \tfrac{1}{2} m_2 \dot{\omega} + \tfrac{1}{2}(I_1 + m_1 r_1^2 + m_2 \omega^2)\dot{\theta}_1^2 + \tfrac{1}{2}(I_2 + m_2 r_2^2)\dot{\theta}_2^2$$
$$+ m_2 r_2 \dot{\theta}_2 (\omega \dot{\theta}_1 \cos(\theta_2 - \theta_1) - \dot{\omega} \sin(\theta_2 - \theta_1)) \; .$$

The potential energy of the system amounts to

$$U_{st} = m_1 g (y + r_1 \cos\theta_1) + m_2 g (y + \omega \cos\theta_1 + r_2 \cos\theta_2) \; .$$

Substitution of T_{st} and U_{st} into equation (C.1) and differentiation with respect to ω, θ_1 and θ_2 leads to 3 equations of motion, which totally describe the system. Step by step,

$$\frac{\partial T_{st}}{\partial \dot{\omega}} = m_2 \dot{\omega} - m_2 r_2 \dot{\theta}_2 \sin(\theta_2 - \theta_1)$$

$$\frac{d}{dt}\left(\frac{\partial T_{st}}{\partial \dot{\omega}}\right) = m_2 \ddot{\omega} - m_2 r_2 \ddot{\theta}_2 \sin(\theta_2 - \theta_1) - m_2 r_2 \dot{\theta}_2 (\dot{\theta}_2 - \dot{\theta}_1) \cos(\theta_2 - \theta_1)$$

$$\frac{\partial T_{st}}{\partial \omega} = m_2 \omega \dot{\theta}_1^2 + m_2 r_2 \dot{\theta}_2 \dot{\theta}_1 \cos(\theta_2 - \theta_1)$$

$$\frac{\partial U_{st}}{\partial \omega} = m_2 g \cos\theta_1$$

$$\Rightarrow \quad F_{\omega} = m_2 \ddot{\omega} - m_2 \omega \dot{\theta}_1^2 - m_2 r_2 \ddot{\theta}_2 \sin(\theta_2 - \theta_1) - m_2 r_2 \dot{\theta}_2^2 \cos(\theta_2 - \theta_1) + m_2 g \cos\theta_1$$

$$\frac{\partial T_{st}}{\partial \dot{\theta}_1} = (I_1 + m_1 r_1^2 + m_2 \omega^2)\dot{\theta}_1 + m_2 r_2 \omega \dot{\theta}_2 \cos(\theta_2 - \theta_1)$$

$$\frac{d}{dt}\left(\frac{\partial T_{st}}{\partial \dot{\theta}_1}\right) = (I_1 + m_1 r_1^2 + m_2 \omega^2)\ddot{\theta}_1 + 2 m_2 \omega \dot{\omega} \theta_1 + m_2 r_2 \dot{\omega} \dot{\theta}_2 \cos(\theta_2 - \theta_1) + m_2 r_2 \omega \ddot{\theta}_2 \cos(\theta_2 - \theta_1) - m_2 r_2 \omega \dot{\theta}_2 (\dot{\theta}_2 - \dot{\theta}_1) \sin(\theta_2 - \theta_1)$$

$$\frac{\partial T_{st}}{\partial \theta_1} = m_2 r_2 \dot{\theta}_2 (\omega \dot{\theta}_1 \sin(\theta_2 - \theta_1) + \dot{\omega} \cos(\theta_2 - \theta_1))$$

$$\frac{\partial U_{st}}{\partial \theta_1} = -(m_1 r_1 + m_2 \omega) g \sin\theta_1$$

$$\Rightarrow \quad F_{\theta_1} = (I_1 + m_1 r_1^2 + m_2 \omega^2)\ddot{\theta}_1 + 2 m_2 \omega \dot{\omega} \theta_1 - (m_1 r_1 + m_2 \omega) g \sin\theta_1 + m_2 r_2 \omega \ddot{\theta}_2 \cos(\theta_2 - \theta_1) - m_2 r_2 \omega \dot{\theta}_2^2 \sin(\theta_2 - \theta_1)$$

$$\frac{\partial T_{st}}{\partial \dot{\theta}_2} = (I_2 + m_2 r_2^2)\dot{\theta}_2 + m_2 r_2 (\omega \dot{\theta}_1 \cos(\theta_2 - \theta_1) - \dot{\omega} \sin(\theta_2 - \theta_1))$$

$$\frac{d}{dt}\left(\frac{\partial T_{st}}{\partial \dot{\theta}_2}\right) = (I_2 + m_2 r_2^2)\ddot{\theta}_2 + m_2 r_2 \dot{\omega} \dot{\theta}_1 \cos(\theta_2 - \theta_1) + m_2 r_2 \omega \ddot{\theta}_1 \cos(\theta_2 - \theta_1) - m_2 r_2 \omega \dot{\theta}_1 (\dot{\theta}_2 - \dot{\theta}_1) \sin(\theta_2 - \theta_1) - m_2 r_2 \ddot{\omega} \sin(\theta_2 - \theta_1) - m_2 r_2 \dot{\omega} (\dot{\theta}_2 - \dot{\theta}_1) \cos(\theta_2 - \theta_1)$$

$$\frac{\partial T_{st}}{\partial \theta_2} = -m_2 r_2 \dot{\theta}_2 (\omega \dot{\theta}_1 \sin (\theta_2 - \theta_1) + \dot{\omega} \cos (\theta_2 - \theta_1))$$

$$\frac{\partial U_{st}}{\partial \theta_2} = -m_2 g r_2 \sin \theta_2$$

$$\Rightarrow \quad F_{\theta_2} = -m_2 r_2 \ddot{\omega} \sin (\theta_2 - \theta_1) + m_2 r_2 \omega \ddot{\theta}_1 \cos (\theta_2 - \theta_1) + (I_2 + m_2 r_2^2) \ddot{\theta}_2$$

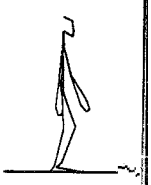
$$- m_2 g r_2 \sin \theta_2 + 2 m_2 r_2 \dot{\omega} \dot{\theta}_1 \cos (\theta_2 - \theta_1) + m_2 r_2 \omega \dot{\theta}_1^2 \sin (\theta_2 - \theta_1)$$

After rewriting the equations in matrix form $A \ddot{q}_{st} = B (q_{st} , \dot{q}_{st})$, they are transformed to a set of first order equations (see section 3.4.3) of the form

$A' \dot{u}_{st} = B' (u_{st})$, where

$$u_{st} = [\omega, \dot{\omega}, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T$$

$$B' (u_{st}) = \begin{bmatrix} \dot{\omega} \\ F_\omega + m_2 \omega \dot{\theta}_1^2 + m_2 r_2 \dot{\theta}_2^2 \cos (\theta_2 - \theta_1) - m_2 g \cos \theta_1 \\ \dot{\theta}_1 \\ F_{\theta_1} - 2 m_2 \omega \dot{\omega} \theta_1 + (m_1 r_1 + m_2 \omega) g \sin \theta_1 + m_2 r_2 \omega \dot{\theta}_2^2 \sin (\theta_2 - \theta_1) \\ \dot{\theta}_2 \\ F_{\theta_2} + m_2 g r_2 \sin \theta_2 - 2 m_2 r_2 \dot{\omega} \dot{\theta}_1 \cos (\theta_2 - \theta_1) - m_2 r_2 \omega \dot{\theta}_1^2 \sin (\theta_2 - \theta_1) \end{bmatrix}$$



$$A' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & -m_2 r_2 \sin (\theta_2 - \theta_1) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_1 + m_1 r_1^2 + m_2 \omega^2 & 0 & m_2 r_2 \omega \cos (\theta_2 - \theta_1) \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & a'_{2,6} & 0 & a'_{4,6} & 0 & I_2 + m_2 r_2^2 \end{bmatrix}$$

C.2. Swing Phase

The Lagrange equation for the swing phase can be written as

$$\frac{d}{dt}\left(\frac{\partial T_{sw}}{\partial \dot{q}_{sw_i}}\right) - \frac{\partial T_{sw}}{\partial q_{sw_i}} + \frac{\partial U_{sw}}{\partial q_{sw_i}} = F_{q_{sw_i}} \quad , \text{ where } q_{sw} = [\theta_3, \theta_4]^T \; , \quad i = 1, 2 \; . \tag{C.2}$$

From figure C-1 (b) we derive (note that θ_3 is defined to be negative for consistency)

$$\rho_3 = [x_h - r_3 \sin\theta_3 \, , \; y_h - r_3 \cos\theta_3]^T \quad \text{and}$$

$$\rho_4 = [x_h - l_3 \sin\theta_3 - r_4 \sin(\theta_3 + \theta_4) \, , \; y_h - l_3 \cos\theta_3 - r_4 \cos(\theta_3 + \theta_4)]^T \; .$$

Substituting

$$\dot{\rho}_3^2 = \dot{x}_h^2 + \dot{y}_h^2 + r_3^2 \dot{\theta}_3^2 - 2 r_3 \dot{\theta}_3 (\dot{x}_h \cos\theta_3 - \dot{y}_h \sin\theta_3) \quad \text{and}$$

$$\begin{aligned}\dot{\rho}_4^2 = {} & \dot{x}_h^2 + \dot{y}_h^2 + l_3^2 \dot{\theta}_3^2 + r_4^2 (\dot{\theta}_3 + \dot{\theta}_4)^2 - 2 l_3 \dot{\theta}_3 (\dot{x}_h \cos\theta_3 - \dot{y}_h \sin\theta_3) \\ & - 2 r_4 (\dot{\theta}_3 + \dot{\theta}_4)(\dot{x}_h \cos(\theta_3 + \theta_4) - \dot{y}_h \sin(\theta_3 + \theta_4)) \\ & + 2 l_3 r_4 \dot{\theta}_3 (\dot{\theta}_3 + \dot{\theta}_4) \cos\theta_4\end{aligned}$$

into

$$T_{sw} = \tfrac{1}{2} m_3 \dot{\rho}_3^2 + \tfrac{1}{2} m_4 \dot{\rho}_4^2 + \tfrac{1}{2} I_3 \dot{\theta}_3^2 + \tfrac{1}{2} I_4 (\dot{\theta}_3 + \dot{\theta}_4)^2$$

yields

$$\begin{aligned}T_{sw} = {} & \tfrac{1}{2}(m_3 + m_4)(\dot{x}_h^2 + \dot{y}_h^2) + \tfrac{1}{2}(I_3 + m_3 r_3^2 + m_4 l_3^2)\dot{\theta}_3^2 \\ & + \tfrac{1}{2}(I_4 + m_4 r_4^2)(\dot{\theta}_3 + \dot{\theta}_4)^2 - (m_3 r_3 + m_4 l_3)\dot{\theta}_3 (\dot{x}_h \cos\theta_3 - \dot{y}_h \sin\theta_3) \\ & - m_4 r_4 (\dot{\theta}_3 + \dot{\theta}_4)(\dot{x}_h \cos(\theta_3 + \theta_4) - \dot{y}_h \sin(\theta_3 + \theta_4)) \\ & + m_4 l_3 r_4 \dot{\theta}_3 (\dot{\theta}_3 + \dot{\theta}_4) \cos\theta_4\end{aligned}$$

for the kinetic energy. The potential energy for the swing phase is

$$U_{sw} = m_3 g (y_h - r_3 \cos\theta_3) + m_4 g (y_h - l_3 \cos\theta_3 - r_4 \cos(\theta_3 + \theta_4))$$

The two equations of motion can now be derived by differentiating the Lagrangian analogous to the stance phase. This results in

$$F_{\theta_3} = (I_3 + m_3 r_3^2 + m_4 l_3^2 + I_4 + m_4 r_4^2 + 2 m_4 l_3 r_4 \cos\theta_4)\ddot{\theta}_3$$
$$+ (I_4 + m_4 r_4^2 + m_4 l_3 r_4 \cos\theta_4)\ddot{\theta}_4$$
$$- (m_3 r_3 + m_4 l_3)(\ddot{x}_h \cos\theta_3 - \ddot{y}_h \sin\theta_3)$$
$$- m_4 r_4 (\ddot{x}_h \cos(\theta_3 + \theta_4) - \ddot{y}_h \sin(\theta_3 + \theta_4)) - m_4 l_3 r_4 \dot{\theta}_4 (2\dot{\theta}_3 + \dot{\theta}_4) \sin\theta_4$$
$$+ m_3 g r_3 \sin\theta_3 + m_4 g (l_3 \sin\theta_3 + r_4 \sin(\theta_3 + \theta_4))$$

$$F_{\theta_4} = (I_4 + m_4 r_4^2 + m_4 l_3 r_4 \cos\theta_4)\ddot{\theta}_3 + (I_4 + m_4 r_4^2)\ddot{\theta}_4$$
$$- m_4 r_4 (\ddot{x}_h \cos(\theta_3 + \theta_4) - \ddot{y}_h \sin(\theta_3 + \theta_4))$$
$$+ m_4 l_3 r_4 \dot{\theta}_3^2 \sin\theta_4 + m_4 g r_4 \sin(\theta_3 + \theta_4) .$$

It should be noted that 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:

$$x_h = x + \omega \sin\theta_1 \qquad y_h = y + \omega \cos\theta_1$$
$$\dot{x}_h = \dot{\omega} \sin\theta_1 + \omega \dot{\theta}_1 \cos\theta_1 \qquad \dot{y}_h = \dot{\omega} \cos\theta_1 - \omega \dot{\theta}_1 \sin\theta_1$$
$$\ddot{x}_h = \ddot{\omega} \sin\theta_1 + 2\dot{\omega}\dot{\theta}_1 \cos\theta_1 \qquad \ddot{y}_h = \ddot{\omega} \cos\theta_1 - 2\dot{\omega}\dot{\theta}_1 \sin\theta_1$$
$$+ \omega(\ddot{\theta}_1 \cos\theta_1 - \dot{\theta}_1^2 \sin\theta_1) \qquad - \omega(\ddot{\theta}_1 \sin\theta_1 + \dot{\theta}_1^2 \cos\theta_1) ,$$

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

After transformation to a set of first order equations, the matrix formulation

$$A' \dot{u}_{sw} = B'(u_{sw})$$

is expressed by

$$u_{sw} = [\theta_3, \dot{\theta}_3, \theta_4, \dot{\theta}_4]^T$$

$$B'(u_{sw}) = \begin{bmatrix} \dot{\theta}_3 \\ F_{\theta_3} + (m_3 r_3 + m_4 l_3)(\ddot{x}_h \cos\theta_3 - \ddot{y}_h \sin\theta_3) + m_4 r_4 (\ddot{x}_h \cos(\theta_3 + \theta_4) \\ \quad - \ddot{y}_h \sin(\theta_3 + \theta_4)) + m_4 l_3 r_4 \dot{\theta}_4 (2\dot{\theta}_3 + \dot{\theta}_4) \sin\theta_4 \\ \quad - m_3 g r_3 \sin\theta_3 - m_4 g (l_3 \sin\theta_3 + r_4 \sin(\theta_3 + \theta_4)) \\ \dot{\theta}_4 \\ F_{\theta_4} + m_4 r_4 (\ddot{x}_h \cos(\theta_3 + \theta_4) - \ddot{y}_h \sin(\theta_3 + \theta_4)) \\ \quad - m_4 l_3 r_4 \dot{\theta}_3^2 \sin\theta_4 - m_4 g r_4 \sin(\theta_3 + \theta_4) \end{bmatrix}$$

$$A' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & I_3 + m_3 r_3^2 + m_4 l_3^2 & 0 & I_4 + m_4 r_4^2 + m_4 l_3 r_4 \cos\theta_4 \\ & + I_4 + m_4 r_4^2 + 2 m_4 l_3 r_4 \cos\theta_4 & & \\ 0 & 0 & 1 & 0 \\ 0 & a'_{2,4} & 0 & I_4 + m_4 r_4^2 \end{bmatrix}$$

C.3. Collisions

C.3.1. Heel Strike

At impact (heel strike), the following calculations are done based on the conservation of linear and angular momenta ($^-$ means before, $^+$ after impact). For the angular momentum at the hip of the stance leg

$$L_h^- = L_h^+ \ , \quad \text{where} \quad L_h = I_2 \dot{\theta}_2 - \vec{r}_2 \times m_2 \dot{\rho}_2 \ .$$

With

$$\vec{r}_2 = [\, r_2 \sin\theta_2 \,,\ r_2 \cos\theta_2 \,]^T \quad \text{and}$$

$$\dot{\rho}_2 = [\, \dot{x} + \dot{\omega} \sin\theta_1 + \omega \dot{\theta}_1 \cos\theta_1 + r_2 \dot{\theta}_2 \cos\theta_2 \,,\ \dot{y} + \dot{\omega} \cos\theta_1 - \omega \dot{\theta}_1 \sin\theta_1 - r_2 \dot{\theta}_2 \sin\theta_2 \,]^T$$

and using the fact, that $\dot{\omega} = 0$ before heel strike, i.e. there is no velocity along the leg axis in the air, and $\dot{x} = \dot{y} = 0$ after heel strike, since the (dynamic) foot does not move on the ground, we have

$$(I_2 + m_2 r_2^2)\, \dot{\theta}_2^- + m_2 r_2 (\dot{x}^- \cos\theta_2 - \dot{y}^- \sin\theta_2) + m_2 r_2 \omega \dot{\theta}_1^- \cos(\theta_2 - \theta_1)$$

$$= (I_2 + m_2 r_2^2)\, \dot{\theta}_2^+ - m_2 r_2 \dot{\omega}^+ \sin(\theta_2 - \theta_1) + m_2 r_2 \omega \dot{\theta}_1^+ \cos(\theta_2 - \theta_1) \qquad (C.3)$$

The angular momentum at the foot before and after impact is expressed by

$$L_f^- = L_f^+ \ , \quad \text{where} \quad L_f = I_1 \dot{\theta}_1 - \vec{r}_1 \times m_1 \dot{\rho}_1 + I_2 \dot{\theta}_2 - \vec{\rho} \times m_2 \dot{\rho}_2 \ .$$

For simplification, this can be written as

$$L_f^- - L_h^- = L_f^{+\cdot} - L_h^+ \ .$$

With $\dot{\rho}_2$ as defined above and

$$\vec{r}_1 = [\, r_1 \sin\theta_1 \,,\ r_1 \cos\theta_1 \,]^T \ ,$$

$$\dot{\rho}_1 = [\, \dot{x} + r_1 \dot{\theta}_1 \cos\theta_1 \,,\ \dot{y} - r_1 \dot{\theta}_1 \sin\theta_1 \,]^T \ ,$$

$$\vec{\rho} = [\, \omega \sin\theta_1 + r_2 \sin\theta_2 \,,\ \omega \cos\theta_1 + r_2 \cos\theta_2 \,]^T$$

and again, $\dot{\omega}^- = 0$, $\dot{x}^+ = \dot{y}^+ = 0$, this yields

$$(I_1 + m_1 r_1^2 + m_2 \omega^2)\,\dot{\theta}_1^- + (m_1 r_1 + m_2 \omega)(\dot{x}^- \cos\theta_1 - \dot{y}^- \sin\theta_1)$$

$$+ m_2 r_2 \dot{\theta}_2^- \cos(\theta_2 - \theta_1) = (I_1 + m_1 r_1^2 + m_2 \omega^2)\,\dot{\theta}_1^+ + m_2 r_2 \dot{\theta}_2^+ \cos(\theta_2 - \theta_1)\ . \qquad (C.4)$$

A third condition comes from the conservation of linear momentum along the leg axis,

$$P_\omega^- = P_\omega^+\ .$$

For this purpose, the reaction force R is considered

$$R\,\Delta t = m_2(\dot{\rho}_2^+ - \dot{\rho}_2^-)\ , \quad \text{with } \dot{\rho}_2 \text{ as defined above.}$$

Since for the instant at impact, there is no contribution of the force in the direction of the leg, it follows that

$$(R_x \Delta t)\sin\theta_1 + (R_y \Delta t)\cos\theta_1 = 0\ .$$

With $\dot{\omega}^- = 0$ and $\dot{x}^+ = \dot{y}^+ = 0$, this yields

$$m_2(\dot{x}^- \sin\theta_1 + \dot{y}^- \cos\theta_1) - m_2 r_2 \dot{\theta}_2^- \sin(\theta_2 - \theta_1)$$

$$= m_2 \dot{\omega}^+ - m_2 r_2 \dot{\theta}_2^+ \sin(\theta_2 - \theta_1)\ . \qquad (C.5)$$

It should be noted that $\dot{x}^-$ and $\dot{y}^-$ of the foot before impact as used above are determined as follows from the swing leg:

$$\dot{x}^- = \dot{x}_h - l_{34}\,\dot{\theta}_{34}\cos\theta_3$$

$$\dot{y}^- = \dot{y}_h + l_{34}\,\dot{\theta}_{34}\sin\theta_3\ ,$$

where l_{34} is the length of the swing leg after extension. The three equations (C.3), (C.4) and (C.5) can now be solved for $\dot{\omega}^+, \dot{\theta}_1^+, \dot{\theta}_2^+$.

C.3.2. Locking of the Knee

The value for the hip angle, $\dot{\theta}_{34}$, after the extension of the swing leg is calculated from the conservation of angular momentum at the hip:

$$L_h^- = L_h^+\ ,$$

which can be expressed as

$$I_3\,\dot{\theta}_3 + \vec{r}_3 \times m_3\,\dot{\rho}_3 + I_4(\dot{\theta}_3+\dot{\theta}_4) + \vec{\rho} \times m_4\,\dot{\rho}_4 = I_{34}\,\dot{\theta}_{34} + \vec{r}_{34} \times m_{34}\,\dot{\rho}_{34} \ , \qquad (C.6)$$

where

$$r_{34} = \frac{m_3 r_3 + (l_3 + r_4)\, m_4}{m_3 + m_4} \ ,$$

$$m_{34} = m_3 + m_4 = m_1 \quad \text{and} \quad I_{34} \approx I_1 \ .$$

Assuming $\dot{\theta}_3$ and $\dot{\theta}_4$ to be defined as in appendix C.2 and

$$\vec{r}_3 = [-r_3 \sin\theta_3 \, , \ -r_3 \cos\theta_3 \,]^T \ ,$$

$$\vec{\rho} = [-l_3 \sin\theta_3 - r_4 \sin(\theta_3+\theta_4) \, , \ -l_3 \cos\theta_3 - r_4 \cos(\theta_3+\theta_4)\,]^T \ ,$$

$$\dot{\rho}_{34} = [\,\dot{x}_h - r_{34}\,\dot{\theta}_{34} \cos\theta_3 \, , \ \dot{y}_h + r_{34}\,\dot{\theta}_{34} \sin\theta_3\,]^T \ ,$$

equation (C.6) becomes

$$\begin{aligned}
&(I_3 + m_3 r_3^2 + I_4 - m_4 r_4^2 - m_4 l_3^2)\,\dot{\theta}_3 + (I_4 - m_4 r_4^2)\,\dot{\theta}_4 \\
&\quad + m_3 r_4 (\dot{x}_h^- \cos\theta_3 - \dot{y}_h^- \sin\theta_3) + m_4 l_3 (\dot{x}_h^- \cos\theta_3 - \dot{y}_h^- \sin\theta_3) \\
&\quad + m_4 r_4 (\dot{x}_h^- \cos(\theta_3+\theta_4) - \dot{y}_h^- \sin(\theta_3+\theta_4)) - l_3 r_4 (1+\dot{\theta}_3+\dot{\theta}_4)\cos\theta_4 \\
&= (I_{34} + m_{34} r_{34}^2)\,\dot{\theta}_{34} + m_{34} r_{34} (\dot{x}_h^+ \cos\theta_3 - \dot{y}_h^+ \sin\theta_3) \ , \qquad (C.7)
\end{aligned}$$

from which $\dot{\theta}_{34}$ can readily be determined.

C.4. Spring and Damping Constants

The following are the default values assigned to the various spring and damping constants (k_i, v_i) and position actuators pa_i described in chapter 6 (a body weight of 75 kg and the anthropometric data in appendix B are assumed).

$k_\omega = 30000$
$v_\omega = 1200$
$k_2 = 1800$
$v_2 = 200$
$k_3 = 5000$
$v_3 = 400$
$pa_{1\_init} = 0$
$pa_{3\_init} = 0.01$
$pa_{3\_inc} = 0.003$

Appendix D
Locomotion Parameters

The algorithm below calculates the non-specified walking parameters and/or verifies the specified ones. The three walking parameters are velocity (`v`), step length (`sl`) and step frequency (`sf`). If a walking parameter is zero below, it means that it is not specified.

```
if ( v >= v_max !! sl >= sl_max !! sf >= sf_max )
    /* reject, since walking is not possible                    */
    return;

if ( v != 0 )   /* velocity is specified                        */
{
    if ( sl != 0 && sf != 0 )
    /* velocity, step length and step frequency are specified   */
    {
        if ( sf * sl  != v )
        {
            /* reject, since walking is not possible            */
            return;
    }   }
    else
    {
        if ( sl != 0 && sf == 0 )
        /* velocity and step length are specified               */
        {
            sf = v / sl;
        }
        else
        {
            if ( sl == 0 && sf != 0 )
            /* velocity and step frequency  are specified       */
            {
                sl = v / sf;
            }
            else
            {
                if ( sl == 0 && sf == 0 )
                /* only velocity is specified                   */
                {
                    sf = sqrt( v / ( body_height * 0.004 ) );
                    if ( sf > sf_norm )
                    {
                        sf = v / sl_norm;
                        sl = sl_norm;
                    }
                    else
                    {
                        sl = v / sf;
}   }   }   }   }   }
```

...continued on next page

```
else    /* velocity is not specified                                    */
{
    if ( sl != 0 && sf != 0 )
    /* step length and step frequency are both given                   */
    {
        if ( ( v = sl * sf ) > v_max )
        {
            /* reject, since walking is not possible                   */
            return;
    }   }
    else
    {
        if ( sf == 0 && sl != 0 )
        /* only step length is given                                   */
        {
            if ( sl <= sl_norm )
            {
                sf = sl / ( 0.004 * body_height );
            }
            else
            {
                if ( sl > sl_norm && sl < sl_max )
                {
                    sf =  ( v_max / sl_max )  - sf_norm;
                    sf *= ( sl - sl_norm ) / ( sl_max - sl_norm );
                    sf += sf_norm;
        }   }   }
        else
        {
            if ( sf != 0 && sl == 0 )
            /* only step frequency is given                            */
            {
                if ( sf <= sf_norm )
                {
                    sl = 0.004 * sf * body_height;
                }
                else
                {
                    if ( sf > sf_norm && sf < sf_max )
                    {
                        sl = sl_norm;
            }   }   }
            else
            {
               '/* nothing is specified, thus ...                      */
                return;
        }   }
        v = sl * sf;
}   }
```

Appendix E
Locomotion Attributes

The following is a brief explanation of the 28 locomotion attributes that can be modified to endow the motion with an individual style, i.e. to produce different instances of a walk for the same locomotion parameters (step length and step frequency). The numbers in parentheses denote the default value settings, the numbers in square brackets are page indices to where the attribute was first used in the text. Most of the attributes should be altered with care, since they directly influence the dynamic simulation. For instance, if `pelvis_list_factor` is increased from 1 (normal pelvic list) to 10 (10 times exaggerated pelvic list), the dynamics are likely to collapse since the hip on one side drops so low that toe clearance can not be achieved any more during the swing phase; however, a value of 3 would be fine, generating a quite accented pelvic list. On the other hand, some attributes, like `arm_rot_factor`, can be chosen more freely since their impact is purely kinematic. Some experimentation with the system is required to get a feel as to how a particular change influences the outcome of the locomotion. All attributes are specified as real numbers; angles are shown in degrees.

`arm_rot_factor` (0.8) [58]

scaling factor for the arm swing relative to the hip angle of the opposite leg.

`duration_kneeflex` (0.1) [55]

percentage of cycle time, for which the knee flexion of the swing leg still increases after double support. This variable is used together with `max_knee` below to calculate the value of the knee angle for the hind leg at the end of double support.

`elbow_rot_max` (35) [58]

maximum elbow flexion; this angle for the elbow is reached at heel strike of the opposite leg.

`elbow_rot_min` (0) [58]

minimum elbow flexion in degrees; this angle for the elbow is maintained throughout most of the forward arm swing until the end of the SWING1 subphase. Then the elbow flexes up to the final value `elbow_rot_max`. When the arm swings back, `elbow_rot_min` is reached at the same time that the arm back swing is completed.

`end_swing1` (0.5) [64]

percent of cycle time for SWING1 subphase.

`end_swing2` (0.85) [64]

percent of cycle time for SWING2 subphase.

`max_knee` (65) [55]

maximum knee flexion during swing, which occurs at `duration_kneeflex`$\cdot t_{cycle}$ after the beginning of the swing.

`meta_incSS` (2) [55]

minimum increment in meta angle for each time step during the stance heel-off period.

`pelvic_list_factor` (1) [50]

scaling factor for pelvic list in the coronal plane relative to the naturally chosen pelvic list; a value of 0 means no pelvic list, a value of 1 produces a natural motion, values greater than 1 exaggerate pelvic list.

`pelvic_rot_max` (13) [50]

maximum rotation of the pelvis in the transverse plane; reached at heel strike.

`percent_D` (0.3) [68]

percentage of the distance traveled by the ankle during SWING1, at which the y-value of the ankle reaches a maximum (0.3 means 30 %).

`sf_accdec` (40) [46]

maximum change in step frequency from one step to the next (*steps/min*).

`sf_max` (180) [40]

maximum step frequency for walking (*steps/min*).

`sf_norm` (132) [40]

maximum value of step frequency, for which normalization formula still holds (*steps/min*).

`shoulder_rot_factor` (0.6) [57]

scaling factor for shoulder rotation relative to pelvic rotation; of course, the rotation of the shoulder is exactly out of phase with the pelvic rotation.

`sl_max` (0.6) [40]

maximum step length for walking (relative to body height).

`sl_norm` (0.528) [40]

maximum value of step length (expressed relative to body height), for which the normalization formula still holds.

`stance1234_knee` (0) [53]

minimum knee angle during stance phase.

`stance6_ankle` (-65) [55]

desired ankle angle θ_5 at toe-off, which marks the end of the meta-off period.

`stride_width_factor` (0.9) [51]

scaling factor for stride width relative to the length of the pelvis l_0; a value of 1 means that the feet are positioned directly under the hips.

`swing1_ankle` (-105) [67]

desired angle of ankle θ_5 at the end of the SWING1 subphase.

`swing1_meta` (0) [67]

desired angle of metatarsophangeal joints θ_6 at the end of the SWING1 subphase.

`theta2_des` (0) [57]

desired upper body angle θ_2.

`theta4_des` (0) [44]

desired knee angle θ_4 at the end of the swing phase.

`theta5_des` (-90) [44]

desired ankle angle θ_5 at the end of the swing phase; a value of -90 means no plantar- or dorsiflexion.

`toe_clear` (0.009) [68]

mimimum vertical distance between the toe and the ground at the end of the SWING1 subphase (in m).

`v_acc` (2) [46]

maximum change in velocity-increase from one step to the next (km/h).

`v_dec` (4) [46]

maximum change in velocity-decrease from one step to the next (km/h).

It is submitted that in KLAW, global parameters like `body_height`, `body_mass`, simulation time, and dynamic parameters such as spring and damping constants can be set or modified as well, but are not strictly considered as locomotion attributes.

Appendix F

Photographic Images

In the following, various aspects of the walking algorithm are illustrated. The pictures were taken from the screen of an IRIS 2400 workstation. All the walks were produced assuming a body height of 1.8 m and a body mass of 75 kg.

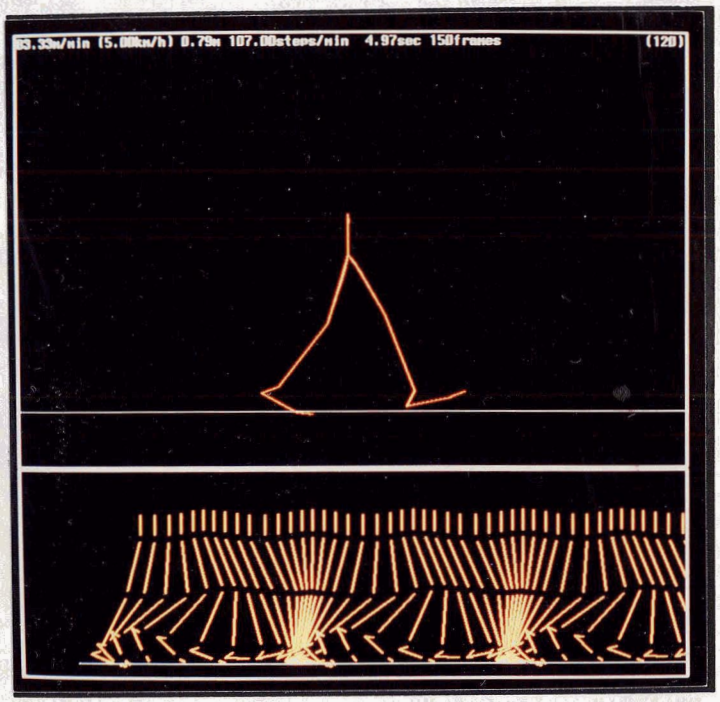


Figure F-1: Legs shown at heel-strike of a walking sequence (frame 120). The data for this walk are based on real human subjects and were collected by Winter [Winter 79] utilizing film recording techniques. The foot goes slightly through the ground due to the fact that the exact anatomical data of the subjects were not specified and had to be approximated; the walking speed is $v = 5\ km/h$, the step length $sl = 0.79\ m$ and the step frequency of $sf = 107\ steps/min$; no data was supplied for the upper body angles.

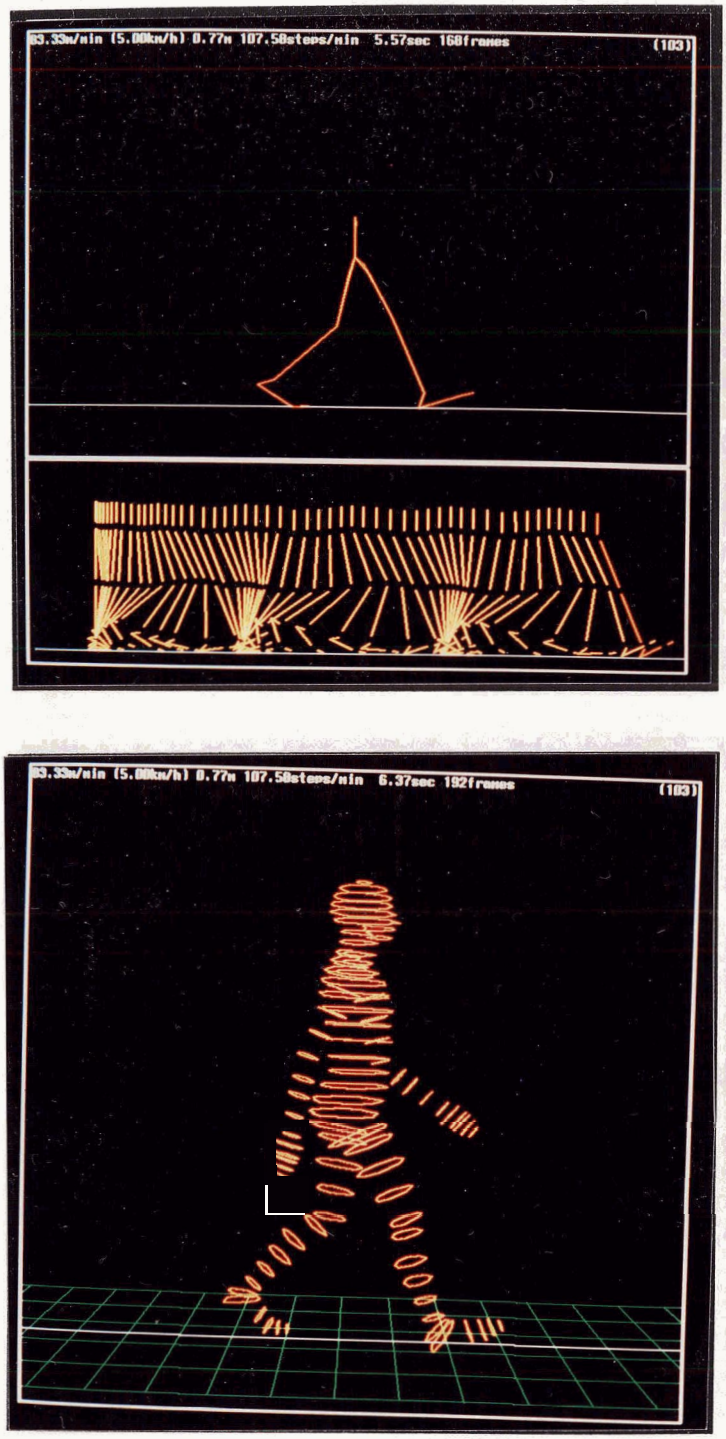


Figure F-2: Illustration of a walking sequence at heel-strike (frame 103), generated by the walking algorithm. Once the rhythmic phase is entered (after one step), the leg patterns come very close to a real walk (compare to figure F-1). The locomotion parameters are $v = 5\ km/h$, $sl = 0.77\ m$ and $sf = 107.5\ steps/min$; only v was specified, sl and sf were chosen by the system (see algorithm in appendix D).

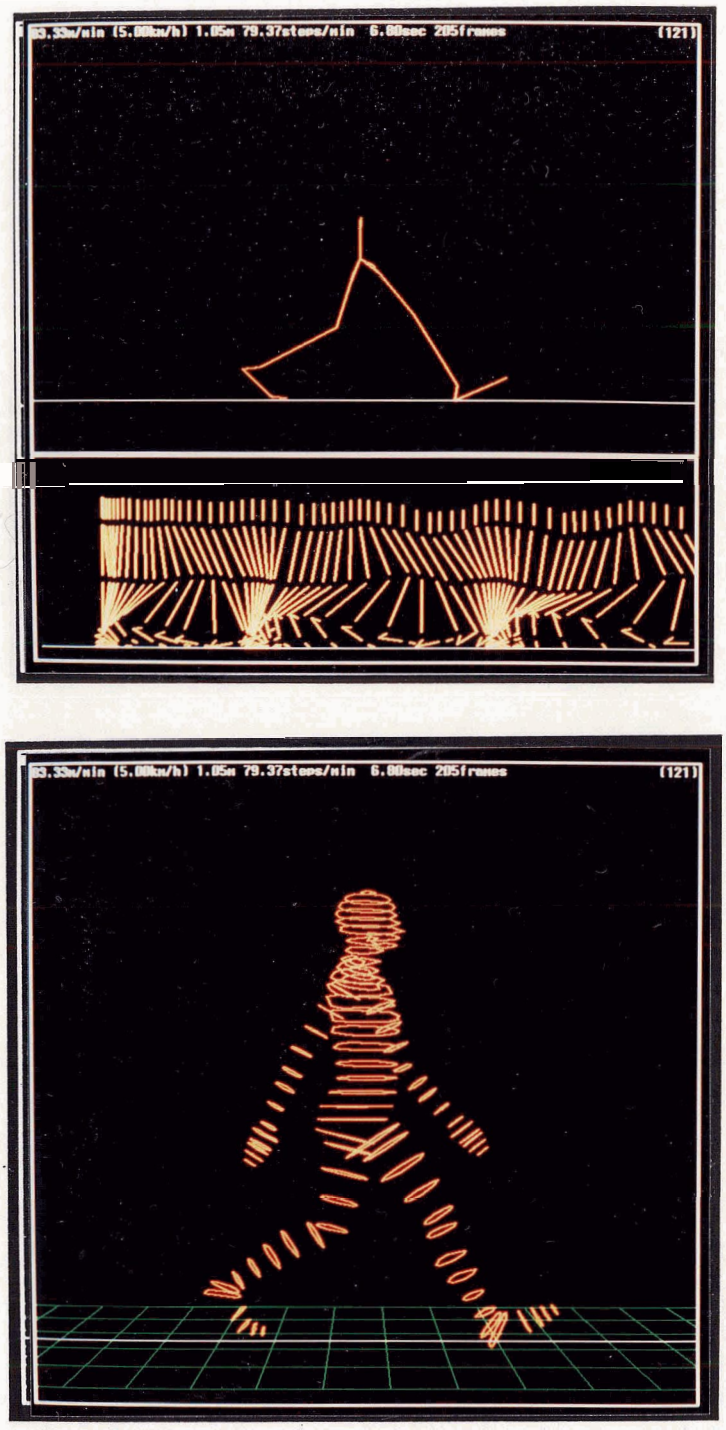


Figure F-3: Illustration of a walking sequence at heel-strike (frame 121), generated by the walking algorithm. The locomotion parameters are $v = 5\ km/h$, $sl = 1.05\ m$ and $sf = 79.4\ steps/min$; v and sl were specified, sf was chosen by the system. Although the walking speed is the same as for the walk in figure F-2, the leg patterns show significant differences; sl approaches a maximum value ($sl_{max} = 1.08\ m$).

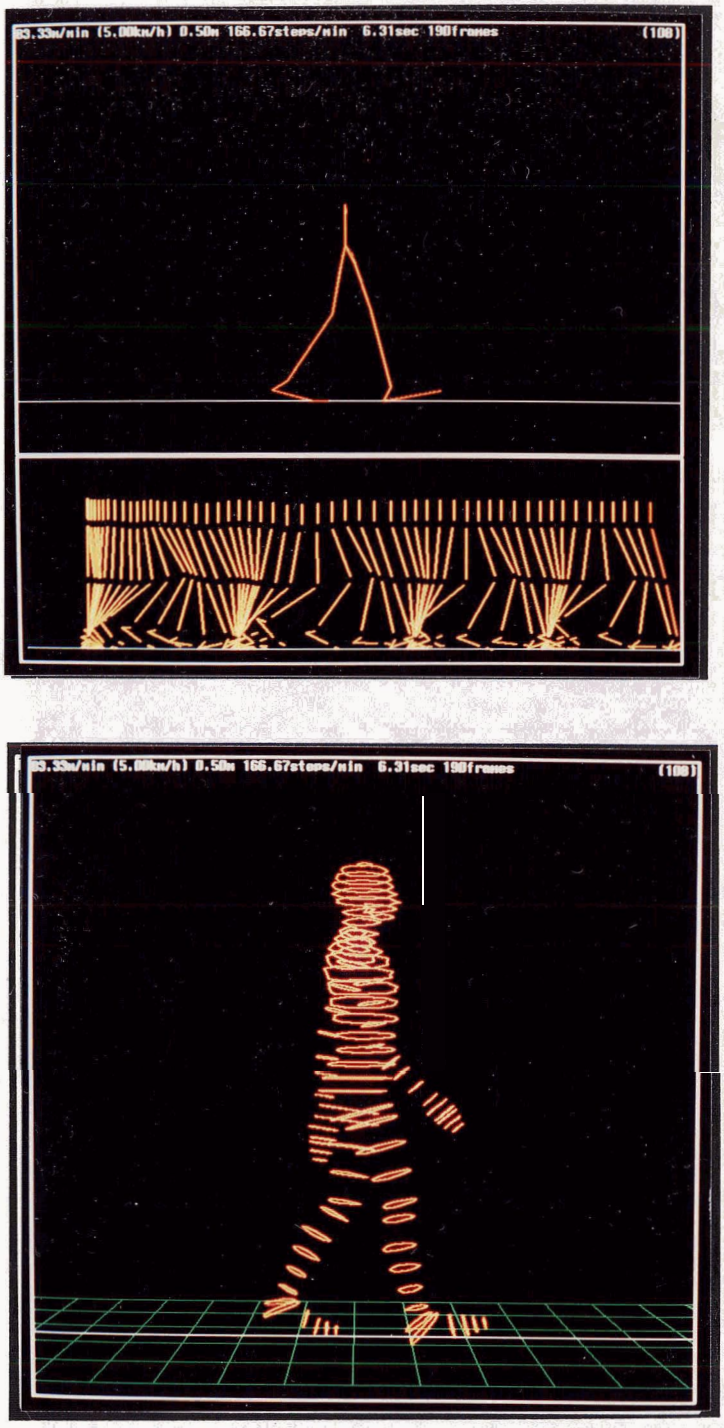


Figure F-4: Illustration of a walking sequence at heel-strike (frame 108), generated by the walking algorithm. The locomotion parameters are $v = 5\ km/h$, $sl = 0.50\ m$ and $sf = 166.7\ steps/min$; v and sl were specified, sf was chosen by the system. Although the walking speed is the same as for the walk in figure F-2 and F-3, the leg patterns show significant differences. sf approaches a maximum value ($sf_{max} = 182\ steps/min$).

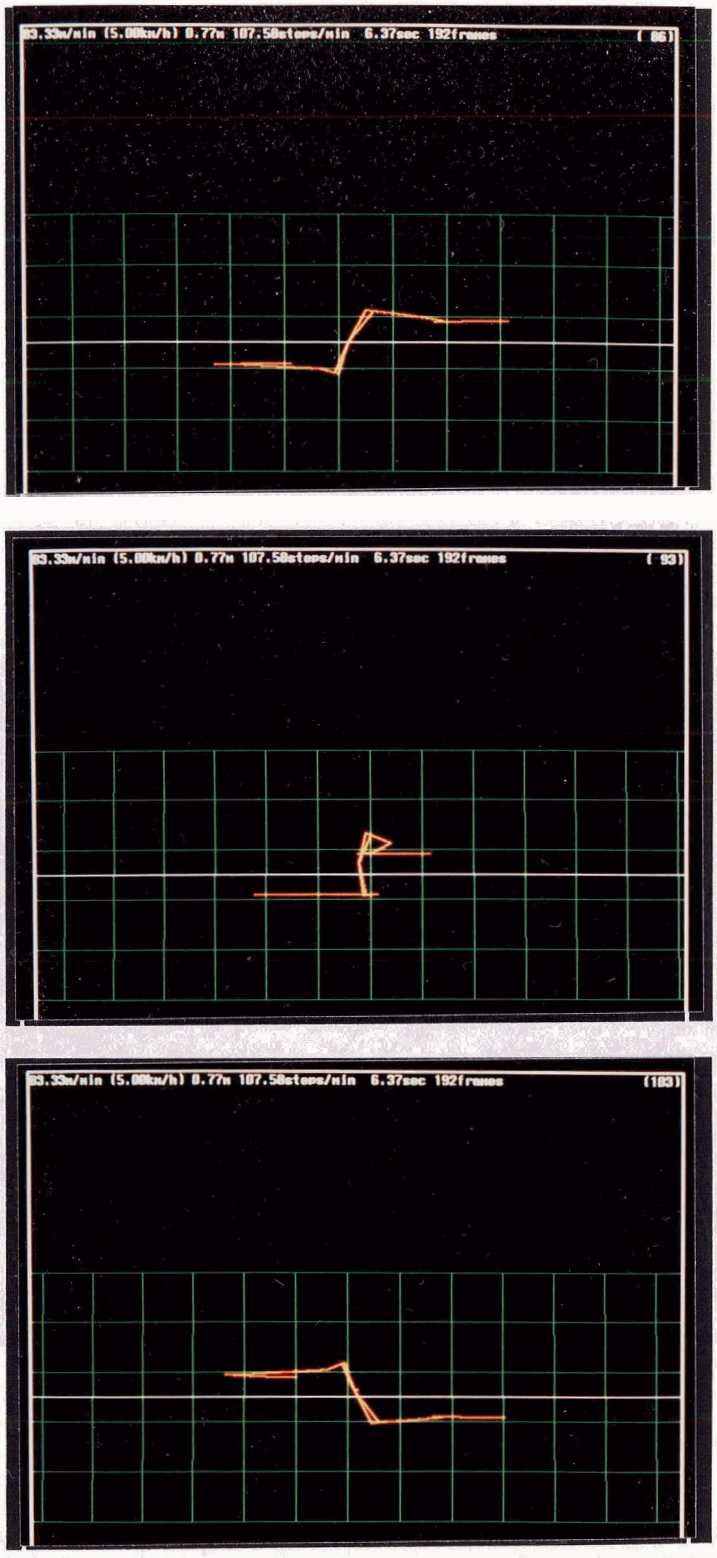


Figure F-5: Illustration of pelvic rotation in the transverse plane (a top view is assumed) for the walk shown in figure F-2. Top: heel-strike of left leg at frame 86. Middle: mid-stance of left leg at frame 93. Bottom: heel-strike of right leg at frame 103.

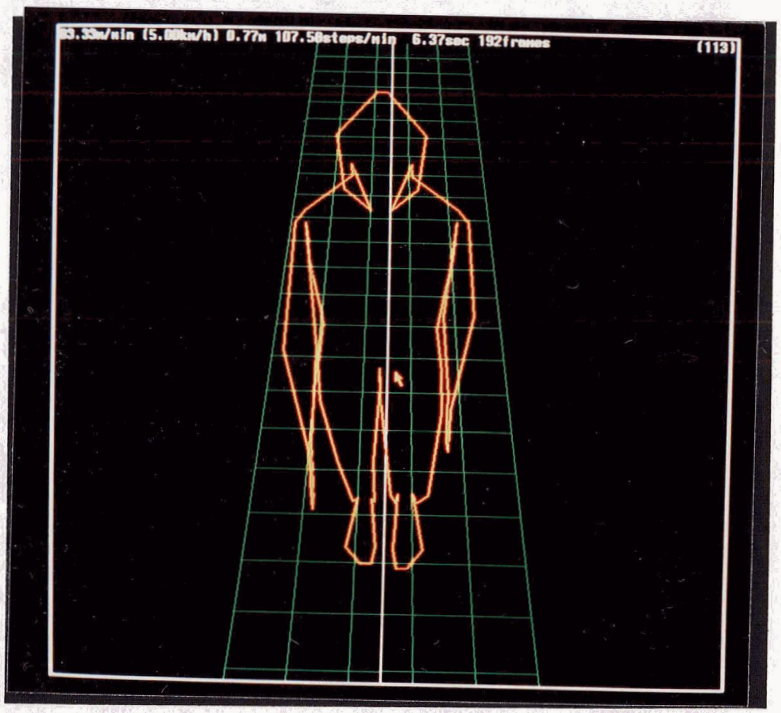
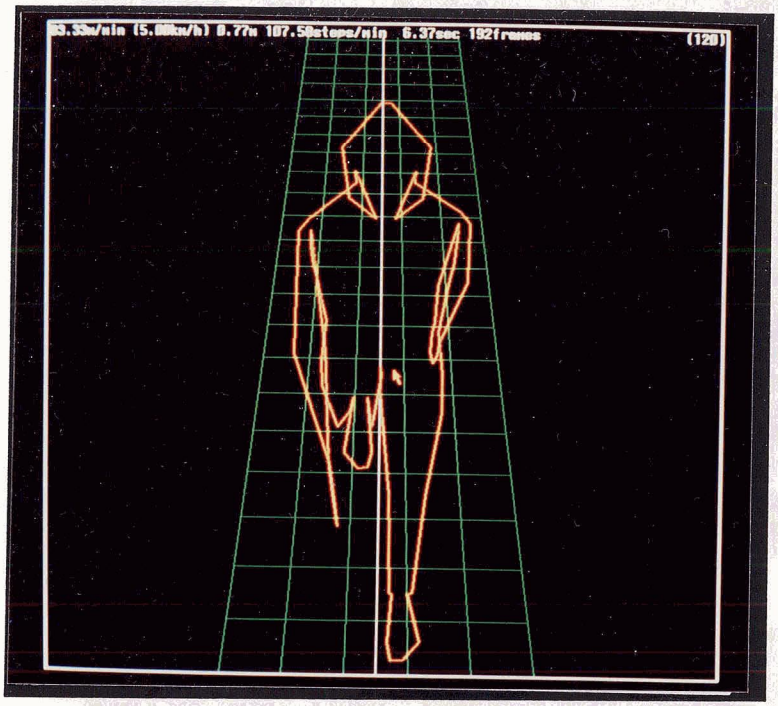
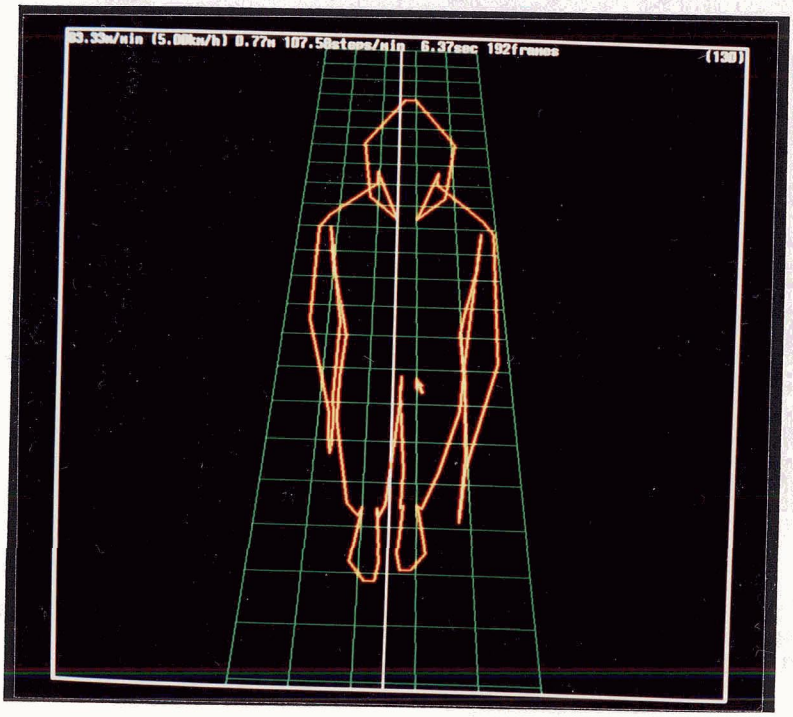


Figure F-6: Illustration of lateral displacement of the body for the walk shown in figure F-2. The body is shifted over the right leg (see arrow) at mid-stance for the right leg (frame 113).

- next page -

Top: at heel-strike (frame 120), the body is centered between the legs.
Bottom: at mid-stance for the left leg (frame 130), the body is shifted over the left leg.


(120)


(130)

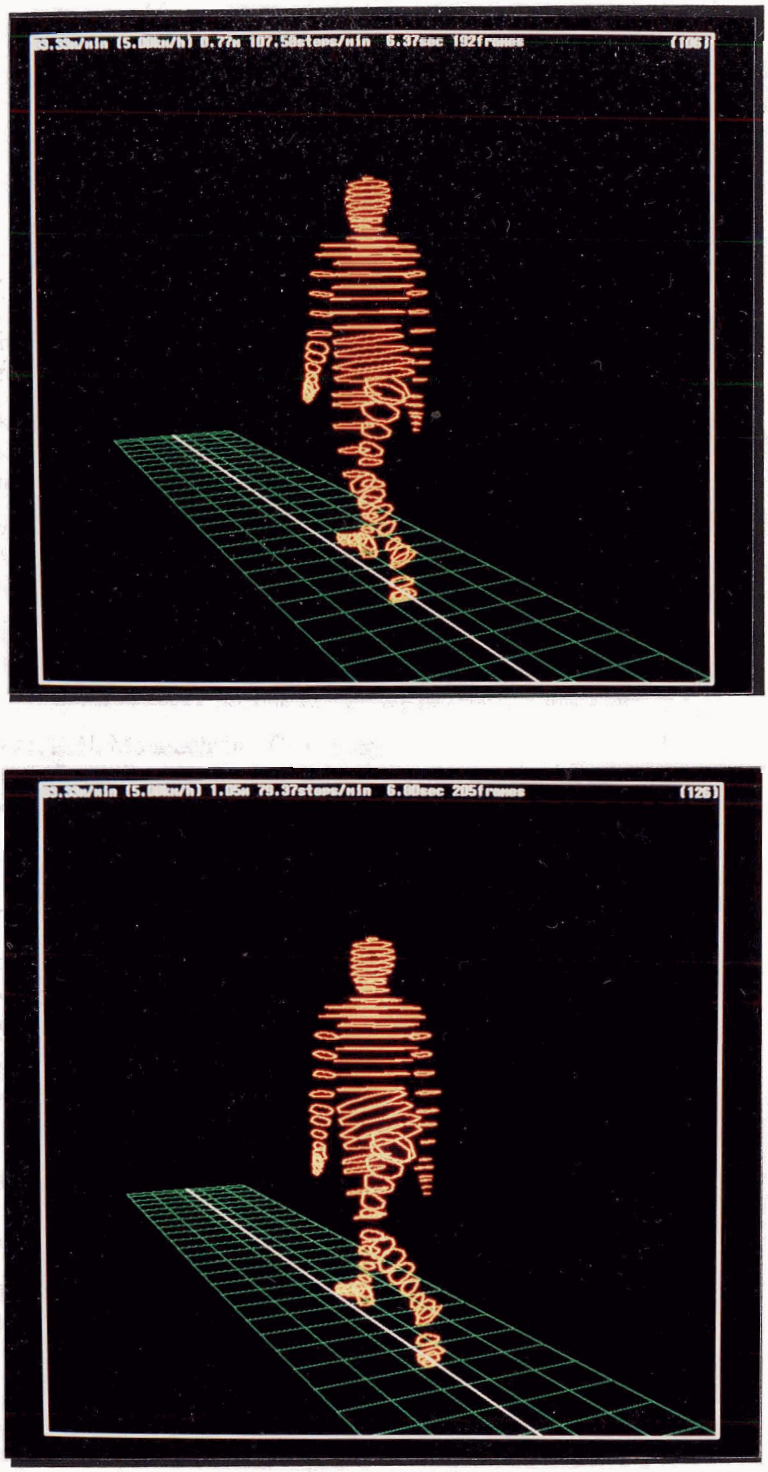


Figure F-7: Illustration of pelvic list in the coronal plane. Top: natural pelvic list at toe-off (frame 106) for the walk shown in figure F-2. Bottom: accentuated pelvic list at toe-off (frame 126) for the walk shown in figure F-3.

References

[Alexander 84] R.McN. Alexander.
The Gaits of Bipedal and Quadrupedal Animals.
The International Journal of Robotics Research 3(2):49-59, 1984.

[Armstrong 85] W.W. Armstrong, M. Green.
The Dynamics of Articulated Rigid Bodies for Purposes of Animation.
In *Graphics Interface '85, Proceedings*, pages 407-415. 1985.

[Badler 79] N.I. Badler, S.W. Smoliar.
Digital Representation of Human Movement.
Computing Surveys 11(1):19-38, March, 1979.

[Badler 80] N.I. Badler, J. O'Rourke, B. Kaufman.
Special Problems in Human Movement Simulation.
In *ACM SIGGRAPH '80, Proceedings*, pages 189-197. July, 1980.

[Badler 87] N.I. Badler, K.H. Manoochehri, G. Walters.
Articulated Figure Positioning by Multiple Constraints.
IEEE Computer Graphics and Applications 7(6):28-38, June, 1987.

[Badler 88] N.I. Badler.
Human Figure Modeling and Animation.
In *Computer Graphics Symposium '88*. Hewlett Packard, 1988.

[Beckett 68] R. Beckett, K. Chang.
An Evaluation of the Kinematics of Gait by Minimum Energy.
J. Biomechanics 1:147-159, 1968.

[Burden 85] R.L. Burden.
Numerical Analysis.
Prindle, Weber & Schmidt, 1985.

[Calvert 82] T.W. Calvert, J. Chapman, A. Palta.
Aspects of the Kinematic Simulation of Human Movement.
IEEE Computer Graphics and Applications 2(9):41-50, November, 1982.

[Calvert 88] T.W. Calvert.
The Challenge of Human Figure Animation.
In *Graphics Interface '88, Proceedings*, pages 203-210. 1988.

[Catmull 78] E. Catmull.
A Hidden-Surface Algorithm with Anti-Aliasing.
In *SIGGRAPH '78, Proceedings*, pages 6-11. July, 1978.

[Csuri 81] C. Csuri.
Goal-Directed Movement Simulation.
In *CMCCS '81 / ACCHO '81*, pages 271-280. 1981.

[Dagg 77] A.I. Dagg.
Running, Walking, Jumping.
Wykeham Publications, London, 1977.

[Drewery 86] K. Drewery, J. Tsotsos.
Goal-Directed Animation Using English Motion Commands.
In *Graphics Interface '86, Proceedings*, pages 131-135. 1986.

[Forest 86] L. Forest, N. Magnenat-Thalmann, D. Thalmann.
Keyframe based Subactors.
In *Graphics Interface '86, Proceedings*, pages 213-216. 1986.

[Girard 85] M. Girard, A.A. Maciejweski.
Computational Modeling for the Computer Animation of Legged Figures.
In *ACM SIGGRAPH '85, Proceedings*, pages 263-270. July, 1985.

[Gomez 85] J.E. Gomez.
TWIXT: A 3-D Animation System.
Computer & Graphics 9(3):291-298, 1985.

[Greene 72] P. H. Greene.
Problems of Organization of Motor Systems.
Progress in Theroretical Biology.
Academic Press, New York, 1972, pages 303-338.

[Hanrahan 85] P. Hanrahan, D. Sturman.
Interactive Animation of Parametric Models.
The Visual Computer 1:260-266, 1985.

[Hindmarsh 80] Alan C. Hindmarsh.
LSODE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers.
ACM-SIGNUM Newsletter 15(4):10-11, 1980.

[Inman 81] V.T. Inman, W.J. Ralston, F. Todd.
Human Walking.
Williams & Wilkins, Baltimore, 1981.

[Isaacs 87] P.M. Isaacs, M.F. Cohen.
Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics.
Computer Graphics 21(4):215-224, July, 1987.

[Korein 82] J.U. Korein, N.I. Badler.
Techniques for Generating Goal-Directed Motion for Articulated Structures.
IEEE Computer Graphics and Applications 2(9):91-81, November, 1982.

[Lozano 79] T. Lozano-Perez, M.A. Wesley.
An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles.
Communications of the ACM 22(10), October, 1979.

[Magnenat 83] N. Magnenat-Thalmann, D. Thalmann.
The Use of High-Level 3-D Graphical Types in the MIRA Animation System.
IEEE Computer Graphics and Applications 3(9):9-16, December, 1983.

[Magnenat 85] N. Magnenat-Thalmann, D. Thalmann.
An Indexed Bibliography on Computer Animation.
IEEE Computer Graphics and Applications 5(7):76-85, July, 1985.

[McGeer 88] B.T. McGeer.
Passive Dynamic Walking.
January, 1988.
submitted for publication.

[McMahon 84] T.A. McMahon.
Mechanics of Locomotion.
The International Journal of Robotics Research 3(2):4-28, 1984.

[Meirovitch 70] L. Meirovitch.
Methods of Analytical Mechanics.
McGraw-Hill, New York, 1970.

[Miller 75] D.I. Miller.
Computer Simulation of Human Motion.
Techniques For The Analysis of Human Movement.
Lepus Books, London, 1975, pages 69-105, Chapter 3.

[Minsky 75] M. Minsky.
A Framework for Representing Knowledge.
The Psychology of Computer Vision.
McGraw-Hill, New York, 1975.

[Miura 84] H. Miura, I. Shimoyama.
Dynamic Walk of a Biped.
The International Journal of Robotics Research 3(2):60-74, 1984.

[Muybridge 70] E. Muybridge.
Human & Animal Locomotion.
Dover Publications, New York, 1970.

[Mylopoulos 83] J. Mylopoulos, H. Levesque.
An Overview of Knowledge Representation.
On Computational Modelling: Perspectives from AI, Databases, and Programming Languages.
Springer-Verlag, New York, 1983.

[Patla 85] A.E. Patla, T. W. Calvert, R. B. Stein.
Model of a Pattern Generator for Locomotion in Mammals.
American Journal of Physiology 17(4):R484-R494, 1985.
Section R: Regulatory, Integrative and Comparative Physiology.

[Pearson 76] K. Pearson.
The Control of Walking.
Scientific American 235(6):72-86, 1976.

[Raibert 84a] M.H. Raibert.
Hopping in Legged Systems - Modeling and Simulation for the 2D Case.
IEEE Transactions Syst. Man Cybern. 14(3):451-463, 1984.

[Raibert 84b] M.H. Raibert, H.B. Brown, Jr., M.Chepponis.
Experiments in Balance with a 3D One-Legged Hopping Machine.
The International Journal of Robotics Research 3(2):75-92, 1984.

[Raibert 86a] M.H. Raibert.
Legged Robots that Balance.
MIT Press, Cambridge Massachusetts, 1986.

[Raibert 86b] **M.H. Raibert.**
Legged Robots.
Communications of the ACM 29(6):499-514, 1986.

[Reynolds 82] C.W. Reynolds.
Computer Animation with Scripts and Actors.
In *SIGGRAPH '82, Proceedings*, pages 289-296. July, 1982.

[Ridsdale 86] G. Ridsdale, S. Hewitt, T.W. Calvert.
The Interactive Specification of Human Animmation.
In *Graphics Interface '86, Proceedings*, pages 121-130. 1986.

[Ridsdale 87] G.J. Ridsdale.
The Director's Apprentice: Animating Figures in a Constraint Environment.
PhD thesis, Simon Fraser Unversity, 1987.

[Sturman 86a] D. Sturman.
Interactive Keyframe Animation of 3-D Articulated Models.
In *Graphics Interface '86, Tutorial on Computer Animation.* 1986.

[Sturman 86b] D. Sturman.
A Discussion on the Development of Motion Control Systems.
In *Graphics Interface '86, Tutorial on Computer Animation.* 1986.

[Sutherland 84] I.E. Sutherland, M.K. Ullner.
Footprints in the Asphalt.
The International Journal of Robotics Research 3(2):29-36, 1984.

[vanBaerle 86] S. van Baerle.
Character Animation: Combining Computer Graphics with Traditional Animation.
In *Graphics Interface '86, Tutorial on Computer Animation.* 1986.

[Vaughan 82a] C.L. Vaughan, J.G. Hay, J.G. Andrews.
Closed Loop Problems in Biomechanics. Part I - A Classification System.
J. Biomechanics 15(3):197-200, 1982.

[Vaughan 82b] C.L. Vaughan, J.G. Hay, J.G. Andrews.
Closed Loop Problems in Biomechanics. Part II - A Optimization Approach.
J. Biomechanics 15(3):201-210, 1982.

[Wells 67] D.A. Wells.
Theory and Problems of Lagrangian Dynamics.
McGraw-Hill, New York, 1967.

[Wilhelms 85] J. Wilhelms.
Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots.
In *Graphics Interface '85, Proceedings*, pages 97-104. 1985.

[Wilhelms 86] J. Wilhelms.
Virya- A Motion Control Editor for Kinematic and Dynamic Aniamtion.
In *Graphics Interface '86, Proceedings*, pages 141-146. 1986.

[Winter 79] D.A. Winter.
Biomechanics of Human Movement.
John Wiley & Sons, 1979.

[Zeltzer 82a] D. Zeltzer.
Representation of Complex Animated Figures.
In *Graphics Interface '82, Proceedings*, pages 205-211. 1982.

[Zeltzer 82b] D. Zeltzer.
Motor Control Techniques for Figure Animation.
IEEE Computer Graphics and Applications 2(9):53-59, 1982.

[Zeltzer 83] D. Zeltzer.
Knowledge-Based Animation.
In *ACM SIGGRAPH/SIGART, Workshop on Motion*, pages 187-192. 1983.

[Zeltzer 85] D. Zeltzer.
Towards an Integrated View of 3-D Computer Character Animation.
In *Graphics Interface '85, Proceedings*, pages 105-115. 1985.

[Zugck 84] J. Zugck.
Numerische Behandlung Linear-impliziter Differentialgleichungen mit Hilfe von Extrapolationsmethoden.
Master's thesis, University of Heidelberg, 1984.