

**CONTROLLING BIPEDAL LOCOMOTION  
FOR  
COMPUTER ANIMATION**

by

Joseph Laszlo

A thesis submitted in conformity with the requirements  
for the degree of Master of Applied Science in the  
Graduate Department of Electrical and Computer Engineering  
University of Toronto

© Copyright Joseph Laszlo, 1996

# **Controlling Bipedal Locomotion for Computer Animation**

Master of Applied Science, 1996

Joseph Laszlo

Department of Electrical and Computer Engineering

University of Toronto

## **ABSTRACT**

Some seemingly simple behaviours such as human walking are difficult to model because of their inherent instability. This thesis proposes an approach to generating balanced 3D walking motions for physically-based computer animations by viewing the motions as a sequence of discrete cycles in state space. First, a mechanism to stabilize open loop walking motions is presented. Once this basic “balance” mechanism is in place, the underlying open loop motion can then be modified to generate variations on the basic walking gait. In addition to other interesting variations, the speed, stride rate and direction of a walk can each be controlled. These variations can be parameterized and potentially used to provide the animated character with the ability to perform autonomous motions such as following a path specified by the animator. While this work is somewhat specific to physically-based animation, some of the underlying ideas may prove useful in other disciplines such as robotics and biomechanics.

## **ACKNOWLEDGEMENTS**

I would like to express my thanks to all the people who have helped me and supported me in my work. My supervisors, Dr. Michiel van de Panne and Dr. Eugene Fiume, have always been wonderfully approachable and ever helpful in their efforts to guide me. In particular, Michiel's boundless enthusiasm never failed to motivate me in tough times (or in good times) and Eugene's ability to step back and view the greater context of a problem has always impressed and inspired me – I hope that one day I'll develop similar skills. To my better half, Anastasia Cheetham, who spent many a late night in the lab, keeping me company and helping me to maintain my sanity when pressured by deadlines, thank you a thousand fold for everything! You'll never know how much your support and patience has meant. Finally, I'd like to thank the members of my defense committee, Dr. Bruce Francis, Dr. James Stewart and Dr. Jonathan Rose for their careful reading of my thesis on such short notice and for their constructive questions and comments.

I would also like to thank the ITRC and NSERC. Without their funding, this work (and the DGP lab in which it was done) might not exist.

# TABLE OF CONTENTS

<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 Kinematic Approaches.....	1
1.2 Dynamics.....	2
1.3 Goals .....	3
1.4 Thesis Organization.....	4
<b>2. BACKGROUND</b> .....	<b>5</b>
2.1 Definitions .....	5
2.2 Previous Work .....	5
2.2.1 Kinematic Animation.....	5
2.2.2 Dynamics-based Animation .....	7
2.2.3 Bipedal Locomotion.....	9
2.2.4 Limit Cycle Control .....	12
2.3 Pose Control.....	12
2.3.1 Linear Parametric PCG Perturbations.....	16
2.4 The Animation System .....	17
2.4.1 Ground Model .....	19
2.5 Biped Models.....	20
<b>3. DISCRETE LIMIT CYCLE CONTROL</b> .....	<b>23</b>
3.1 Limit Cycles .....	23
3.2 Control Formulation .....	25
3.3 Application to Bipedal Locomotion .....	31
3.4 Nominal Open-loop Control, $U^{nom}$ .....	32
3.5 Choice of Regulation Variables, $Q$ .....	37
3.6 Choice of Perturbation Parameters, $\Delta U$ .....	40

3.7 Linear, Sampled "Balance" Control.....	46
3.7.1 Desired RV Values .....	47
3.7.2 Constructing and Applying the Discrete System Model .....	48
3.8 Torso Servoing.....	53
3.9 Conclusions.....	56
<b>4. BALANCED WALKING RESULTS.....</b>	<b>57</b>
4.1 Up Vector Regulation Variables.....	58
4.1.1 Other Observations.....	63
4.2 Swing-COM Regulation Variables.....	64
4.3 Stance-COM Regulation Variables.....	70
4.4 Torso Servoing.....	72
4.5 Robo-bird Running .....	73
4.6 Conclusions.....	74
<b>5. WALKING VARIATIONS.....</b>	<b>76</b>
5.1 Speed Control .....	76
5.2 Base PCG Parameterization.....	81
5.3 Turning Perturbations .....	82
5.3.1 Point and Path Following .....	87
5.4 Stride Rate Perturbation .....	89
5.5 Other Interesting Variations.....	91
5.5.1 Bent-Knee Walking.....	91
5.5.2 Bent-Over Walking.....	92
5.5.3 Ducking.....	93
5.6 Conclusions.....	95
<b>6. CONCLUSIONS AND FUTURE WORK.....</b>	<b>96</b>
6.1 Future Work .....	97
6.1.1 Better Discrete System Models .....	97

6.1.2 Additional Forms of Locomotion.....	98
6.1.3 Natural Motion.....	98
6.1.4 Extension to Aperiodic Motions and Further Generalization.....	98
6.1.5 Automatic Synthesis .....	99
<b>REFERENCES.....</b>	<b>100</b>
<b>APPENDIX A – TERMS AND DEFINITIONS.....</b>	<b>108</b>
<b>APPENDIX B – MODEL PARAMETER SCRIPTS .....</b>	<b>113</b>
<b>APPENDIX C – SAMPLE ANIMATION SCRIPT .....</b>	<b>120</b>

## LIST OF FIGURES

Figure 2.1 - Animator control vs autonomy.....	9
Figure 2.2 - A periodic PCG for a simple, planar 4 DOF biped model.....	13
Figure 2.3 - Typical articulated creature model used with pose control graphs.....	14
Figure 2.4 - Rotational PD controller for pose control.....	14
Figure 2.5 - Pose control graph structures .....	15
Figure 2.6 - Overview of the simulation process.....	17
Figure 2.7 - Spring and damper ground force model (2D example) .....	19
Figure 2.8 - Friction cone ground slip model.....	20
Figure 2.9 - Complex human model .....	21
Figure 2.10 - Robo-bird creature.....	22
Figure 3.1 - Passive Limit Cycle Stability.....	24
Figure 3.2 - An Active Limit Cycle .....	25
Figure 3.3 - Discrete System.....	27
Figure 3.4 - Linear parameters of a 1D discrete system in state space .....	29
Figure 3.5 - Three cycles of a typical 1-dimensional system.....	30
Figure 3.6 - Overall discrete limit cycle control structure for a walking biped.....	32
Figure 3.7 - Forward walking base PCG .....	34
Figure 3.8 - Unbalanced motion of human model.....	35
Figure 3.9 - Possible walking and running steps.....	36
Figure 3.10 - Initial configuration for simple human model .....	37
Figure 3.11 - End-of-step sampling times for $Q_i$ .....	38
Figure 3.12 - Balance RV vectors.....	39
Figure 3.13 - Decomposition of up vector projection into RVs.....	40

Figure 3.14 - Right hip pitch and roll perturbations.....	42
Figure 3.15 - Typical effect of stance hip perturbations .....	43
Figure 3.16 - Balance RV components vs linearly scaled hip pitch perturbation .....	44
Figure 3.17 - Balance RV components vs linearly scaled hip roll perturbation.....	45
Figure 3.18 - Direction of change of forward RV components with hip pitch.....	46
Figure 3.19 - Estimate of desired lateral RV components .....	48
Figure 3.20- Balancing process for each step.....	48
Figure 3.21 - Unobserved state problem .....	49
Figure 3.22 - Model construction and extrapolation .....	49
Figure 3.23 - 2D sampling strategies .....	51
Figure 3.24 - Sampling point pitfalls in discrete model construction .....	53
Figure 3.25 - Torso servo parameters .....	54
Figure 3.26 - Pelvis-based up vector balance indicator.....	55
Figure 3.27 - Falling with torso servoing enabled .....	55
Figure 4.1 - A sequence of steps from a walking limit cycle.....	57
Figure 4.2 - Continuous-time up-vector RV component phase diagrams .....	58
Figure 4.3 - Nominal up-vector RV target range and plan-views .....	60
Figure 4.4 - Discrete RV values for L-F sampling, $Q^d = [.2,0]$ .....	62
Figure 4.5 - Discrete RV values for SP sampling, $Q^d = [.3,0]$ .....	62
Figure 4.6 - Discrete RV values for L-F sampling, $Q^d = [.35,0]$ .....	62
Figure 4.7 - Step length vs step number.....	64
Figure 4.8 - Continuous-time swing-COM RV phase diagrams.....	65
Figure 4.9 - Swing-COM RV target range and hip plots .....	67
Figure 4.10 - Discrete RV values for L-F sampling, $Q^d = [0.05,0]$ .....	68
Figure 4.11 - Discrete RV values for SP sampling, $Q^d = [0.0,0]$ .....	68
Figure 4.12 - Step length vs step number, SP sampling .....	69
Figure 4.13 - Discrete RV values for SP sampling, $Q^d = [0.0,\pm 0.03]$ .....	69

Figure 4.14 - Variation of stance-COM RV with stance hip pitch.....	70
Figure 4.15 - Up vector RV range and hip plots with torso servoing enabled .....	72
Figure 4.16 - Continuous-time up vector component phase diagram with torso servoing .....	73
Figure 4.17 - Robo-bird base PCG for running.....	74
Figure 4.18 - A running robo-bird.....	74
Figure 5.1 - Speed control using composite RV.....	79
Figure 5.2 - Forward velocity (m/s) versus time for velocity feedback.....	80
Figure 5.3 - Steady-state velocity vs $Q_{bias}^d$ .....	80
Figure 5.4 - Acceleration and deceleration .....	80
Figure 5.5 - Turning perturbations for human model with ball-and-socket hips .....	83
Figure 5.6 - Typical operation of a right turn perturbation.....	84
Figure 5.7 - Hip plots for the most successful turning perturbation trials.....	85
Figure 5.8 - Hip plots for less successful turning perturbation trials .....	86
Figure 5.9 - Turning perturbation with torso servoing applied .....	86
Figure 5.10 - Point-following .....	88
Figure 5.11 - Path following.....	88
Figure 5.12 - Stride rate perturbation for human model with ball-and-socket hips.....	89
Figure 5.13 - Results of applying different stride rate perturbation scalings.....	90
Figure 5.14 - Average speed for varying stride rates .....	91
Figure 5.15 - Bent-knee perturbation for a human model .....	92
Figure 5.16 - Bent-knee walking with a parameter scaling of 1.0.....	92
Figure 5.17 - Bent-over walking perturbation .....	92
Figure 5.18 - Bent-over walking .....	93
Figure 5.19 - Composite bent-knee, bent-over and bent-neck perturbation.....	94
Figure 5.20 - Ducking .....	94
Figure A-1 - States of a 1 degree-of-freedom swinging pendulum plotted vs time .....	109
Figure A-2 - State-space trajectory of a simple swinging pendulum.....	109

Figure A-3 - Reference planes of the human body.....	110
Figure A-4 - Phases of bipedal walking and running.....	111

## **LIST OF TABLES**

Table 4.1 - Results of first controlled step using stance-COM RVs.....	71
Table 5.2 - Speed control parameters .....	78

# 1. INTRODUCTION

Computer animation has long been an integral part of the simulation, motion picture, television and consumer entertainment industries and promises to play a much greater role in the future. As it becomes more pervasive, improved techniques will be needed to simplify and speed up the process of creating convincing, high quality animations. One key area of interest is the generation of motion for various types of creatures and characters to be used in an animation. These creatures are the actors of the computer graphics world. The way they move and interact with their environment has a great effect on a viewer's perception of the animation, whether it appears intentionally cartoon-like, or as an integral part of a realistic scene. In the quest for tools to generate realistic motion, one of the key directions of research has been the use of physically-based animation. This thesis presents an approach to the generation of bipedal locomotion for computer animations using physics-based simulations.

There are essentially two basic models used in the generation of motion for the purpose of animating articulated figures: kinematic models and dynamic models. The following brief overview of these approaches is useful for placing the research topic of this thesis in an appropriate context.

## 1.1 Kinematic Approaches

The most straightforward method for character animation is *kinematic* in nature. Kinematic animation is concerned only with the specification of joint angles and angular velocities over time. It does not deal with the forces and torques acting on or within a creature or their effect on the creature's motion.

Motion capture is a special case of the kinematic approach in which the joint angles and/or velocity data are measured from a real motion and then re-used on an animated character. The most common way of capturing a motion at present is to attach a series of markers to various points on the subject's body and to use multiple video cameras or other sensory devices to record the motion of the markers. The subject's motions are mapped directly onto the animated character, thereby ensuring that the animated motion will be realistic. The ability to modify, blend and transition between pre-recorded motions is important to provide the animator with sufficient control over the final motion. However, results based on modifications of captured motions are not guaranteed to remain realistic.

## **1.2 Dynamics**

An alternate approach toward providing realism is the use of physically-based animation. In this scheme, motions are the result of physical simulations, which include detailed modeling of internal and external forces and torques, the creature's mass and moments of inertia, and its interaction with the environment. All these parameters affect the final result, as they would in the real world. The essence of this approach is to ensure realism by constraining the motion of the system to abide by the laws of physics. Dynamics-based animation has the advantage that the task of ensuring that motion is physically realistic has been automated. The animator is, in principle, free to apply his or her abilities to the more artistic aspects of the animation process. Note that "realism" in this context refers to behaviour consistent with a simulated model of the real world. Similarity to the real world depends completely on the fidelity of this model.

This approach introduces new and challenging problems to be solved in order to be of practical use. First, incorporating dynamics effects involves the integration of the equations of motion over time, and has historically been computationally expensive for all but the most simplistic problems. While it seems that no amount of computing power is truly enough, efficient simulation algorithms and faster hardware are beginning to bring the simulation of complex systems of

interest to animators into the realm of real-time performance. A second problem, which remains largely unsolved, is often called the *control problem* of dynamic animation. Briefly stated, in the context of computer animation, the control problem is:

Given a creature, an environment and a desired motion specified by the animator, what are the control forces and torques required to achieve the desired motion or a close, physically-realistic approximation?

While the incorporation of dynamics in the generation of computer animations has been a topic of significant research interest for approximately a decade [AG85] [WB85] [Wil86], it is only now beginning to play a more serious role in commercial computer animation systems. A dynamics simulator is now a part of one of the most popular computer animation packages [Alias]. The delay is due to both the performance issues and a lack of suitable solutions to the control problem.

This thesis provides an approach to solving the control problem for bipedal locomotion.

### **1.3 Goals**

The primary goal of this thesis is to provide a technique for the animation of physically-based bipedal locomotion. More specifically, we present a control solution for articulated figures performing cyclic motions such as walking. Aperiodic motions such as sitting down and standing up are not addressed.

Within this context, this thesis has a number of more specific objectives:

- The technique should work for statically unstable articulated figures. This means that it must provide some form of ongoing corrective control actions.
- The basic approach should be general in nature, allowing for a wide variety of periodic motions without changes to the basic control structure.

- The approach should work for creatures of reasonable complexity without making any fundamental assumptions about the creature's structure. In particular, it should at least be suitable for animating a human model with tens of degrees of freedom (DOFs).
- The control representation should be relatively compact, and flexible enough to allow straightforward specification of walking variations.
- If desired, the motion should exhibit autonomy. For example, the animator might be allowed to specify the start and end points of a walk rather than being required to specify the placement of the foot for each step.

Two desirable objectives which we do not directly address are "naturalness" of the resulting motion (as opposed to physical realism) and interactivity. While both of these features are important to have in an animation system, the problem of generating bipedal locomotion subject to the above goals is a sufficiently challenging intermediate goal. Nevertheless, the proposed technique affords the animator the freedom to potentially obtain natural looking motions with reasonable additional effort compared to generating basic course motions. As well, expected increases in computer performance over the next year or two promise to make interactive use of the system a realizable goal.

## **1.4 Thesis Organization**

This thesis is divided into 6 chapters. Chapter 2 summarizes the previous related work and presents the background material necessary to understand the chapters which follow. It also provides an overview of our animation system. Chapter 3 discusses the underlying principle of our control approach. It further describes the general control structure and its application to the generation of balanced, cyclic locomotion. Chapter 4 presents the basic results of applying the control formulation to bipedal walking. Chapter 5 describes further results for variations on walking gaits. The ability to have the walking biped follow a desired path is also demonstrated. Finally, Chapter 6 concludes the thesis and discusses a number of possible directions for future work.

## 2. BACKGROUND

This chapter presents the background information required to understand later chapters in the thesis. First, important definitions are introduced. Previous related work is then reviewed, followed by a description of the underlying control representation on which our balance control approach is based. Finally we describe our animation system and our biped models.

### 2.1 Definitions

A number of important terms and acronyms are used throughout the thesis. Their definitions and descriptions can be found in Appendix A. The majority of the terms are commonly used terms in the robotics and biomechanics literature. More in-depth information can be found in [HR86], [SV89], [Fr86] and [IRT81].

### 2.2 Previous Work

Bipedal locomotion is a topic of interest to a number of disciplines. This section describes a representative subset of the work in these fields. First, we provide an overview of the various approaches to motion generation in computer animation. This is followed by discussion of work specific to bipedal locomotion in computer animation, biomechanics and robotics. Finally, some relevant work in the control literature is addressed.

#### 2.2.1 Kinematic Animation

Research in computer animation has evolved significantly in its relatively short life span. The earliest approaches to computer animation use *keyframing*, a technique based on classical animation. In keyframing, the configuration of the animated objects at various points in time is specified by the animator and the computer generates the in-between frames using linear or other forms of interpolation. In early systems, specification of keyframes required the animator to

directly manipulate the DOFs of an object [Mez68, BW71, Csu71, KB84, Stu84, MTT85b, SB85, Las87]. Later systems allowed the animator to specify the position of specific points on the objects being animated (such as a hand or foot ) and used inverse kinematics to determine the appropriate values for the creature's internal DOFs [KB82] [GM85] [BMW87]. Procedural descriptions of motion, often based on real-world data and observations, can be used to model very specific classes of movement, effectively "programming" the animated movement [Zel82] [GM85]. In all cases, the quality of the resulting motion is heavily dependent on the ability of the animator who is responsible for ensuring that the perceived dynamics of the motion are appropriate. This is a task which requires significant skill. It potentially distracts the animator from the primary task at hand, but it also allows him or her complete artistic control.

Rotoscoping and motion capture are techniques commonly used to obtain kinematic data from real-world sources. Directly recording a phenomenon to be animated guarantees realistic and natural-looking motion. Specialized hardware is generally required, but such equipment is becoming more accessible. A number of problems with this approach make the investigation of other motion generation techniques desirable. First, captured motions are limited to real-world motions that can easily be recorded. Essentially, motion capture has many of the same restrictions as live actors. Also, approaches to parameterizing captured motions often produce results that are no longer fully realistic. Physical constraint violations, such as ground interpenetration and sliding are common examples of failure. While solutions to enforcing such constraints for particular classes of motion have been demonstrated [BMTT90] [KB93], no general solution currently exists. More recent parameterization approaches seem oriented toward more broadly modifying captured motions and are likely to have similar problems [BW95] [WP95] [UAT95]. Finally, captured motions cannot easily be modified to respond realistically to environments different from the one in which they are obtained. Varying terrain and collisions are two examples of such potentially desirable changes. As the demand for fully interactive environments increases, this issue becomes more important. In recent years the interactive home-entertainment industry

has begun to exceed the motion picture industry in consumer revenues. Today's interactive video games are being made with the same effort as low-budget movies, incorporating complex scripts and storylines, well-known actors, and mixing real and computer-generated imagery [Sni95].

### 2.2.2 Dynamics-based Animation

The techniques used to integrate physics into the generation of computer animations can be divided into two basic approaches, *trajectory-based* and *controller-based*.

Trajectory-based techniques such as [WK88], [BN88], [Coh92] and [LGC94] attempt to find a physically realistic or near-realistic trajectory from one point in the state space of a creature to another. Since the systems are typically highly underconstrained, the trajectory is usually optimized in some way, for example for smoothness, minimum control energy or minimum time. A disadvantage of the technique is that a new trajectory must be generated for each new desired instance of motion. Also, interactions with the environment such as collisions and friction are often difficult to properly incorporate into the dynamics specification. One decided advantage of trajectory-based techniques is that they relate well to keyframing. The animator can control details of the end result through the specification of keyframes and other trajectory-based constraints. Trajectory-based techniques are also able to find the most physically plausible solution, even if no completely physical solution is possible.

In the controller-based approach, a dedicated *controller* or *control algorithm* is used to activate the simulated muscles of a creature, causing it to perform some motion or action within a simulated environment. The use of such controllers has a number of advantages over the trajectory-based approach. In many cases, controllers can be designed to be *reusable* and *composable* [van89] [Hod91] [SC92]. Reusability implies that a controller can be used to achieve a given motion with a variety of initial states. Composability implies that a sequence of motions can be generated by

switching between several controllers over time, possibly subject to some form of transition requirements.

Early versions of controllers for animation used force and torque functions, specified either by the user [Wil86] [AGL87] [FW88] or based on measured or observed data [MZ90] [Mil88]. Other controllers are based on state machines, dividing the motion into a number of phases, each of which is represented by a single state. Controllers are designed by hand in many cases [RH91] [SC92] [HSL92] [H+95]. Hand-designed controllers require the use of carefully chosen parameters to simplify the control program and are typically specific to a particular type of motion (e.g. hopping).

Controllers can also be automatically synthesized. Automatic synthesis uses various stochastic search strategies to explore the space of possible controllers [VF93] [vKF94] [vKF94b] [vL95] [NM93] [A+95] [Sim94] [GT95]. Each controller is assigned a fitness value which characterizes its "goodness" and a mechanism is provided for keeping and refining good controllers and eliminating poor ones. In [Sim94], the structure of the creature itself is allowed to evolve, as well as the controller. Current automatic synthesis techniques are best at finding controllers for relatively stable creatures and motions such as a crawling ant or motion in a single plane. This is because they rely on the fact that a good first guess can be stochastically determined with a reasonable amount of computation. A relatively smooth fitness function is also typically required to allow incremental progress toward an acceptable solution. Unstable motions such as human walking do not meet these requirements since the solution space is exceedingly small compared to that of a more stable creature and motion, especially when motion in 3 dimensions is desired.

The use of motion controllers increases the autonomy of the creature being animated, thereby requiring less direct animator intervention as compared to kinematic and trajectory-based approaches. The cost of this increased autonomy is in the degree of control the animator has over

the final motion. Allowing the animator to specify that the biped "follow that taxi" often removes the animator's choice of which precise path to take. The tradeoff between autonomy and degree of animator control is illustrated in Figure 2.1.

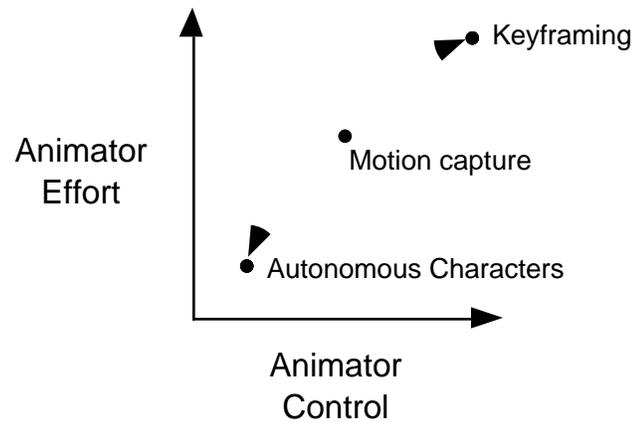


Figure 2.1 - Animator control vs autonomy

### 2.2.3 Bipedal Locomotion

The animation of bipedal locomotion has long been a topic of fascination to many. Zeltzer [Zel82] presents a hierarchical task-oriented animation system in which the low-level walking motions are implemented kinematically, based on measured human data. Girard and Maciejewski [GM85] use rules associated with dynamics (rather than dynamics simulations) for torso motion and inverse kinematics for leg motion to generate one of the first non-rotoscoped, natural looking walks. Bruderlin and Calvert [BC89] break each step into a number of kinematically-defined subphases based on known human gait mechanics and use simplified dynamics simulation to generate the motion in between each subphase. By allowing the user to vary a number of *gait determinants*, a wide variety of natural-looking walks can be generated. Since in this approach, the dynamics are highly constrained, replacing the dynamic interpolation with kinematic interpolation [BC93] is found to give results of similar quality while increasing performance significantly, allowing gait parameters to be adjusted interactively. This work currently represents the state-of-the-art in real-time, parameterized kinematic models of natural looking human walking motion. A similar

technique has also been applied to human running [BC96]. Badler [BPW93] uses primarily kinematic techniques as well as rotoscoped data with dynamic enhancements to achieve many human motions and behaviours. Boulic, Magnenat-Thalmann and Thalmann [BMTT90] and Ko and Badler [KB93] present techniques to generalize rotoscoped or motion captured walking data to other subjects and step lengths while reducing or eliminating the resulting ground constraint violations.

Raibert and Hodgins [RH91] use full dynamical simulation with robust hand-crafted hopping control to attain various bounding gaits for biped and quadruped robot models. As well, a similarly controlled planar kangaroo model is shown to compare well to its real-world counterpart. Stewart and Cremer [SC92] use their flexible constraint-based approach to generate fully dynamic 3D bipedal walking on level terrain and up a flight of stairs. One of the required constraints, however, is a 0 DOF "magnetic boot" on the stance foot. Van de Panne, Fiume and Vranesic [VFFV92] use optimal state-space control tables to control walking on level terrain and up and down ramps, smooth curved surfaces and stairs for a planar biped model. This approach requires a suitable control decomposition to make the generation of the state-space controllers tractable.

Auslander et al. demonstrate automatic synthesis of interesting 2D bipedal walking and tumbling motions but meet with difficulty in their initial attempts to extend this approach directly to 3D [Aus+95]. Van de Panne and Lamouret propose the use of guiding external forces to initially attain reasonable controllers using similar automatic synthesis [vL95]. The forces are then reduced in a number of steps and can sometimes be entirely eliminated to yield a fully-balanced, automatically synthesized motion. Examples of human walking, skipping and running and walking over varying terrain for a simple 3D biped are given. One difficulty with this approach is that the removal of guiding forces must be performed incrementally over the entire motion sequence (for example, each step of a walk). This process that can become prohibitively expensive for more complex creatures. Hodgins et al. [H+95] show how Raibert's hopping

principles along with various forms of additional control can be applied to running for a fully-dynamic, complex human model. The ground model for the runner uses a 1 DOF constraint which allows motion only in the body's pitch DOF.

Legged locomotion has also received significant attention in the robotics and biomechanics literature. Research in the biomechanics literature is directed primarily toward gait analysis. Of particular interest is the efficiency of natural motion in humans and animals [McM84] [Ale84] and the identification of various *determinants of gait* and their role in normal and pathological gaits [TT76] [SCD80] [MM80] [IRT81] [SSH82] [PB89]. To this end, a number of dynamic locomotion models have been proposed [VJ69] [MM80] [McM84] [Tow85] [PB89]. These approaches generally make assumptions which limit their usefulness for animation. Typical assumptions include a simplified biped model and/or motion only in a plane [VJ69] [TT76] [Tow85] [McM84] [PB89]. Some only consider the open-loop motion over one or two steps [TT76] [McM84] [PB89].

The robotics literature has more in common with the goals of physically-based computer animation than biomechanics does. It has as an objective the synthesis of legged locomotion, rather than analysis. While some works present only simulation results and others implement real robots, all systems, of necessity, incorporate some form of forward dynamics. Many approaches propose various reduced-order models for the equations of motion and rely on reference trajectories, typically based either on the motion of an inverted pendulum [FM84] [MS84] [KKI90] [KT91] or on measured human data [VS72] [HF77]. As with biomechanics, the complexity of the locomotion problem is often reduced through the use of simplified biped models. Constraining motion to the sagittal plane, is perhaps the most common simplification [HF77] [FM87] [KKI90] [KT91] [CHP92]. A number of approaches deal only with statically stable walking motions in which the biped is balanced at all points in the walk cycle [HF77] [ZS90] [SZ92]. While such motions may be quite useful for a robot due to their inherent stability,

they are of less interest to animators. Miura and Shimoyama present a stilt-leg biped that walks dynamically in 3D on point feet [MS84]. Takanishi et al. [TIYK85] achieve a dynamic but very rough, lurching 3D walk for a robot with anthropomorphic legs. Furusho & Sano [FS90] demonstrate the use of sensor-based feedback to produce smoother motions from a similar gait. Raibert et al. [Rai+84] present an elegant three-way decomposition of control to accomplish robust one-legged hopping in three dimensions which is later extended to bipedal and quadrupedal models using the notion of a *virtual leg* [Rai86] [Rai86b].

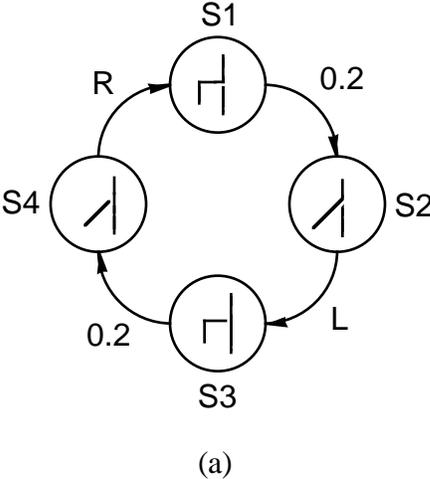
#### **2.2.4 Limit Cycle Control**

A number of papers view bipedal walking and running motions as limit cycles in state space. These are most closely related to the work in this thesis. McGeer [McG89] [McG90] [McG90b] demonstrates that various forms of passive legged locomotion, such as walking with and without knees and running can exist as natural modes of a mechanical device. By using Newton's method to search for motions which have identical initial and final system states, stable gaits could be found for a system which uses only a small downhill slope as a source of energy. Katoh and Mori [KM84] use high-gain PD control to drive a biped's motion toward a prescribed cyclic state space trajectory. Hmam and Lawrence [HL91] use nonlinear feedback control to drive a running biped onto a prescribed trajectory which is based on the passive motion of the system. The feedback is used to improve the robustness of the system to perturbation. These latter two works use very simple biped models and all three assume strictly planar dynamics.

### **2.3 Pose Control**

The fundamental control representation used throughout this thesis is the pose control graph, or PCG [vKF94]. Figure 2.2 shows a typical PCG, which is essentially a specialized type of finite state machine. Pose control provides a compact way to specify the torques to be applied to an articulated figure in order to attain a desired motion. Each state in the PCG specifies a set of desired joint angles for the creature with respect to some fixed reference position, called a *desired*

pose, and transition information. A *pose table* is a sequential list of each desired pose and its transition information. Figure 2.3 illustrates a typical creature which might be controlled using pose control graphs.



(a)

DOF		right hip	right knee	left hip	left knee	transition info
state	S1	0	0	-90	90	0.2
	S2	0	0	-45	0	0 L
	S3	-90	90	0	0	0.2
	S4	-45	0	0	0	0 R

(b)

Figure 2.2 - A periodic PCG for a simple planar, 4 degree of freedom biped model  
 (a) State diagram form.  
 (b) Pose table form. All DOF values are in degrees relative to a reference position with straight, vertical legs and upright torso. Time-based transitions are in seconds. For sensor-based transitions, L - left foot sensor, R - right foot sensor.

While pose control appears similar to keyframing, two distinctions should be emphasized. First, the PCG determines the *desired* joint angles, and not the *actual* joint angles. The joints must be driven toward the desired angles through the use of a low-level control mechanism. Second, the poses do not specify the creature's position and orientation with respect to the world frame of reference. Instead, these are determined by the creature's interaction with its environment.

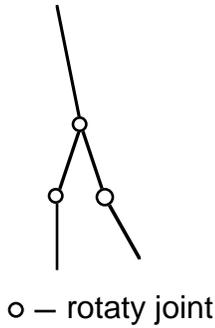


Figure 2.3 - Typical articulated creature model used with pose control graphs

In our implementation, joint angles are driven toward their desired values by joint actuator torques generated according to the following proportional-derivative (PD) control law:

$$\tau = k_p \cdot (\theta_{desired} - \theta) - k_d \cdot \dot{\theta}$$

where  $\tau$  is the control torque applied at the joint,  $\theta$  is the current joint angle,  $\theta_{desired}$  is the desired joint angle,  $\dot{\theta}$  is the relative angular velocity of the rigid links connected by the joint and  $k_p$  and  $k_d$  are proportional and derivative control constants.

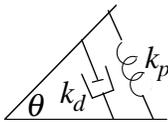


Figure 2.4 - Rotational PD controller for pose control

$k_p$  and  $k_d$  specify the strength of a rotational spring and damper pair which acts as the "muscle" controlling the joint, as indicated in Figure 2.4. While this is a simple model, it is sufficient for our purposes in that it ensures that only internal control forces are used to generate the creature's motion. The actuator PD gain constants are held fixed for each joint and are considered to be part of the model specification.

The low-level mechanism for driving individual DOFs to their desired angles incorporates feedback, and is thus an example of closed loop control. The basic pose control mechanism does

not make use of any other feedback to correct or adjust the overall motion of the creature. Therefore, we will refer to pose control using a fixed PCG as being a form of *open loop* control. Our definition is one of convenience and is appropriate since we choose to use pose control as our fundamental representation. In essence, the local closed loop control at the joints is treated as part of the system being controlled rather than part of the controller itself. In this context, *closed loop* refers to feedback used to modify the desired joint angle parameters of the PCG.

The transitions between the states of a PCG are based on a fixed *hold time* and/or are *sensor-based*. A hold-time transition occurs after a specified time has elapsed in the associated state. A sensor-based transition occurs when a particular binary sensor turns "on", or takes place immediately if the sensor is already on upon entering the state. The only form of sensors used in this work are ground contact sensors on the creature's feet, which are considered to be on when the foot is in contact with the ground.

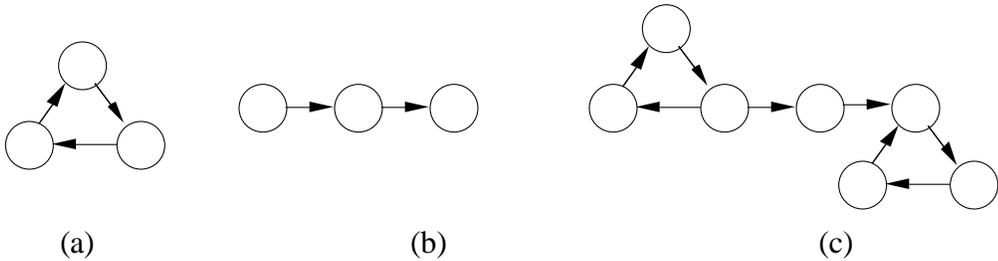


Figure 2.5 - Pose control graph structures  
 (a) cyclic  
 (b) aperiodic  
 (c) composite

PCGs may be *periodic*, *aperiodic*, or *composite* (see Figure 2.5). A periodic PCG is one in which the state machine is cyclic. In this form, the desired poses are perfectly periodic. The actual motion and the applied torques need not be so; they do not typically repeat from one cycle to the next. As discussed in [vKF94], creatures controlled using periodic PCGs are analogous to simple windup toys. An aperiodic PCG is a chain of successive poses, useful for performing a single motion instance, such as recovering after a fall. A composite PCG is a more general form of state machine composed of periodic and aperiodic PCGs.

While pose controllers can be automatically synthesized with reasonable efficiency for relatively simple creatures and motions [vKF94], they work best for naturally stable motions. The base pose controllers presented in this thesis are reasonably complex and have all been designed by hand. The possibility of using automatic synthesis with the control techniques described is discussed later as future work.

To achieve the types of motions we are interested in, it is necessary to adjust our PCG to perform appropriate corrective actions on each cycle. An approach to the selection of these control actions is one of the key contributions of this thesis. Adjustments to the PCG are accomplished by applying linearly scaled perturbations to the PCG joint parameters during each cycle of motion.

### 2.3.1 Linear Parametric PCG Perturbations

To distinguish between the PCG providing our basic motion and the perturbations we apply to it, we will introduce the notion of a *base PCG*. A base PCG is a pose control graph which provides the fundamental cyclic motion of the creature we are trying to animate. For example, in the case of walking, the base PCG might consist of a left step followed by a right step. Ideally, the execution of a base PCG from a suitable initial state would result in the desired motion (e.g. a walk). However, the creatures we are interested in animating are unstable. The periodic open-loop actions of the base PCG invariably results in the creature falling. In order to generate balanced locomotion, additional control must be provided. Chapter 3 introduces one approach to providing such additional control, which uses the notion of *linear parametric perturbations* (LPPs), which we define below.

We begin by defining a *relative* PCG, which describes a change in the pose control to be applied to a creature, typically used to effect a desired change a motion. Consider a base PCG,  $B$ , to which we add a relative PCG,  $\Delta P$ , scaled by an arbitrary scalar constant,  $k$ :

$$C = B + k \Delta P.$$

$k \cdot \Delta P$  is a linearly parameterized control perturbation where the parameter  $k$  is a *scaling factor* of  $\Delta P$ . The PCGs  $B$ ,  $\Delta P$  and the overall control,  $C$ , are each similar in form to Figure 2.2 (b). The addition operation denotes the addition of the corresponding desired joint angle and hold time parameters of two PCG pose tables of equal dimensions. Scalar multiplication is applied to each desired joint angle and hold-time parameter of a PCG pose table. Sensor-based transitions in a perturbed PCG remain unaltered. As an example, an LPP could be used to turn a creature's head to look in a particular direction or to say "no". To accomplish this,  $\Delta P$  might be chosen to vary only the desired angle of the neck and  $k$  would be chosen to turn the head to the right or left by the desired amount.

A variety of LPPs will be used in Chapters 3 and 5 as the mechanism to provide our biped with basic balance control and additional gait variations.

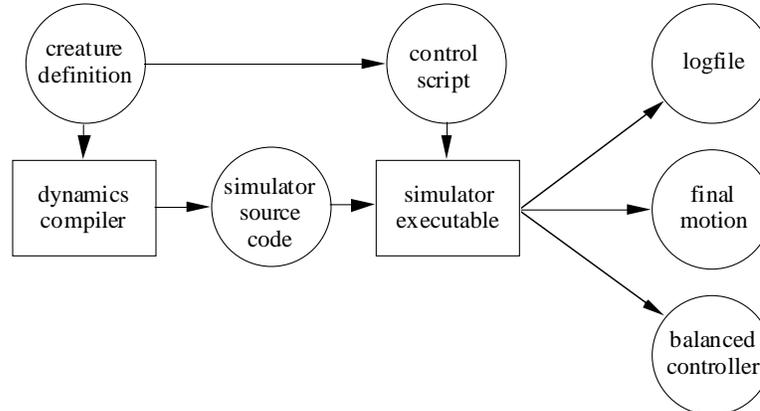


Figure 2.6 - Overview of the simulation process

## 2.4 The Animation System

Figure 2.6 provides an overview of the process of generating a physically-based animation with our system, beginning with a creature definition and a control script. The creature definition is used by a dynamics compiler to generate code which solves and integrates the equations of

motion. The dynamics code, together with code implementing the ground model, basic pose control and the balance control formulation of Chapter 3 is then compiled into the simulator executable. The control script provides the particular control parameters for the desired simulation. Sample control scripts can be found in Appendices B and C. The primary outputs of each simulation are the final balanced motion of the creature and the aperiodic PCG which was ultimately responsible for generating it. Note that the resulting aperiodic PCG output provides open loop control. It is therefore only reusable given an identical initial state. In essence it is a record of the applied control actions for the motion, already complete with feedback actions.

The dynamics compiler used is a commercially available software package [SDFAST]. The animation environment currently supports the simulation and control of a single articulated creature consisting of rigid links in a tree structure with rotary joints of up to 3 DOF each and no joint limits. Each DOF has individual PD constants which remain fixed for the entire simulation. Collision forces due to interpenetration of the links of the articulated figure are not simulated.

The equations of motion are integrated using a fixed time step, fourth order Runge-Kutta integrator which is part of the dynamics compiler software. Performance of the simulator varies with model complexity with the most complex human model (described in Section 2.5) requiring approximately 1 minute of wall clock time to compute 1 second of simulated motion on a Sun Sparkstation 10. The use of a fixed integration time step has a significant impact on performance since the worst-case (i.e. smallest) time step for the complete simulation must be used. It is estimated that the use of a variable integration time step could improve performance by a factor of 5-10. Recorded simulation results can be played back in real-time on a Silicon Graphics Indigo<sup>2</sup> Workstation with GR3-XZ graphics hardware. Display functions are implemented using the using the SGI-GL graphics library.

### 2.4.1 Ground Model

The ground reaction forces are modeled for all simulations using a spring and damper model. Figure 2.7 illustrates the model for a 2D system. Ground forces are applied to specific, pre-defined *monitor points* on the model. The ground forces exerted on a monitor point,  $M$ , which has penetrated the ground surface are:

$$F = k_p \cdot (P - M) - k_d \cdot \dot{M}$$

where  $F$  is the ground reaction force,  $M$  is the position of the monitor point,  $\dot{M}$  is its velocity,  $P$  is the initial point of contact with the ground, and  $k_p$  and  $k_d$  are proportional and derivative constants defining the stiffness and damping properties of the ground (see Figure 2.7).

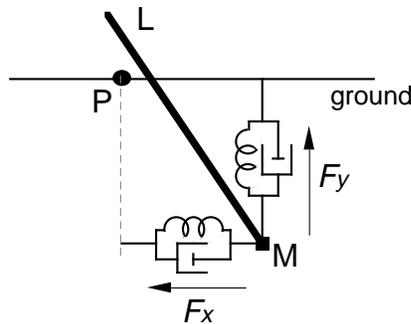


Figure 2.7 - Spring and damper ground force model (2D example)

- $L$  - a link of the creature
- $M$  - a monitor point attached to link  $L$
- $P$  - point of initial contact of  $M$  with ground plane
- $F_x$  - simulated ground forces in x direction
- $F_y$  - simulated ground forces in y direction

The ground reaction forces are bounded such that the vertical component is always positive, thus never allowing the damping term to impede the lifting of the foot. Monitor point slippage is implemented using a Coulomb friction model, which is used to limit the tangent of the ground reaction force. Slippage is allowed when the ratio of the vertical and tangent forces exceeds a specified limit as illustrated in Figure 2.8 for the 2D case. Slippage is simulated by moving the point of initial contact,  $P$ , to be directly above monitor point.

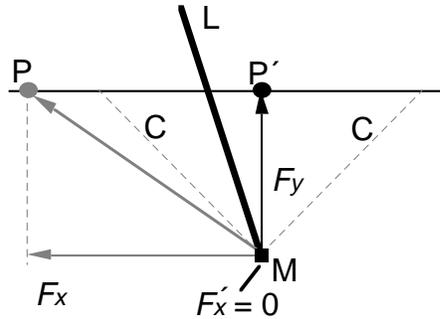


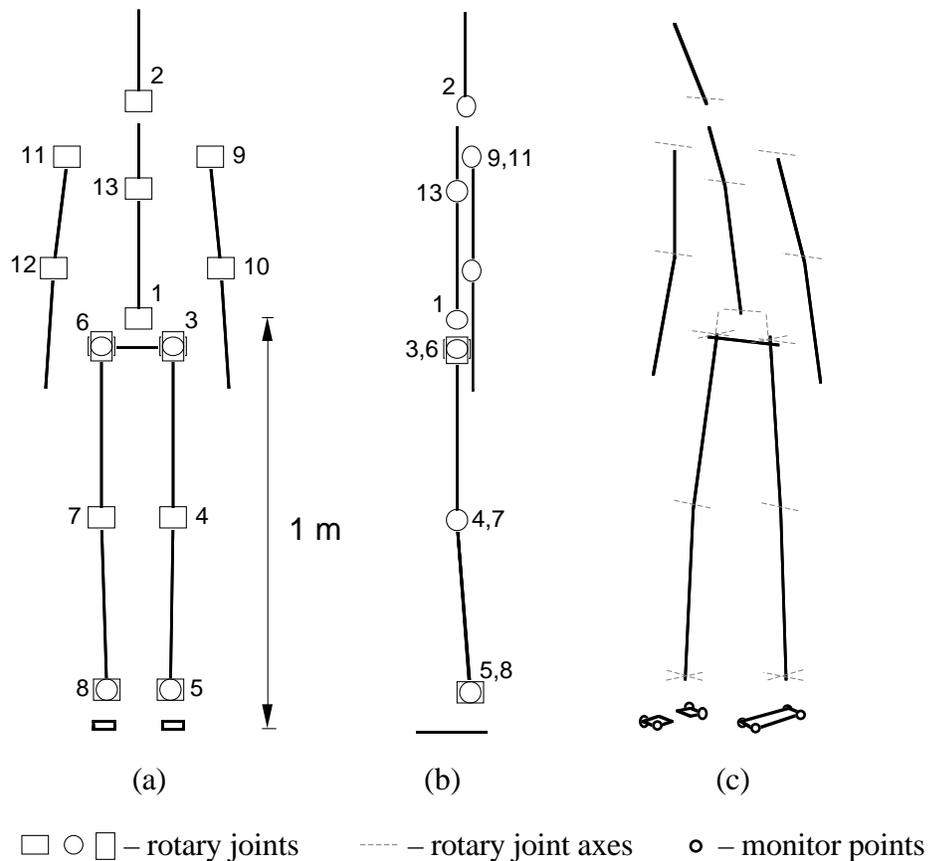
Figure 2.8 - Friction cone ground slip model (2D example)

- L* - a link of the creature
- M* - a monitor point on link *L*
- C* - boundaries of a friction cone for a ratio of 1.0
- P* - point of initial contact of *M* with ground plane
- P'* - new "point of initial contact" of *M* with ground plane after slipping is applied

No other ground forces are applied to the creature to constrain movement in any direction. The foot may be in full or partial contact with the ground, depending on which monitor points are in ground contact. It is free to pitch, roll and yaw and to slip within the described constraints provided by the friction cones. This contrasts with many approaches in animation and in robotics, which use planar dynamics or place less realistic constraints on the motion of the foot while in contact with the floor [HF77, FM87, KKI90, KT91, CHP92, VFV92, SC92, Hod+9].

## 2.5 Biped Models

The most complex human model used, shown in Figure 2.9, has 19 degrees of freedom including ball-and-socket hips, 2 DOF ankles and a jointed torso. All other joints are modeled using 1 DOF. Mass and inertia parameters are realistic for a human model and are from taken from [WH95]. Several other simpler human models with fewer DOFs are also used throughout our experiments in order to reduce simulation time requirements. The simplest of these has 12 DOFs, with 2 DOF hips (pitch and roll but no yaw), no arms, and a rigid torso which incorporates the mass and inertia parameters for fixed arms in the reference position of Figure 2.9.



#### Degrees of Freedom

- |  |  |
|--|--|
| 1 - waist pitch (sagittal plane)         |  |
| 2 - neck pitch (sagittal plane)          |  |
| 3:0 - left hip roll (coronal plane)      | 6:0 - right hip roll (coronal plane)       |
| 3:1 - left hip pitch (sagittal plane)    | 6:1 - right hip pitch (sagittal plane)     |
| 3:2 - left hip yaw (transverse plane)    | 6:2 - right hip yaw (transverse plane)     |
| 4 - left knee pitch (sagittal plane)     | 7 - right knee pitch (sagittal plane)      |
| 5:0 - left ankle pitch (sagittal plane)  | 8:0 - right ankle pitch (sagittal plane)   |
| 5:1 - left ankle roll (coronal plane)    | 8:1 - right ankle roll (coronal plane)     |
| 9 - left shoulder pitch (sagittal plane) | 11 - right shoulder pitch (sagittal plane) |
| 10 - left elbow pitch (sagittal plane)   | 12 - right elbow pitch (sagittal plane)    |
|  | 13 - mid-back pitch (sagittal plane)       |

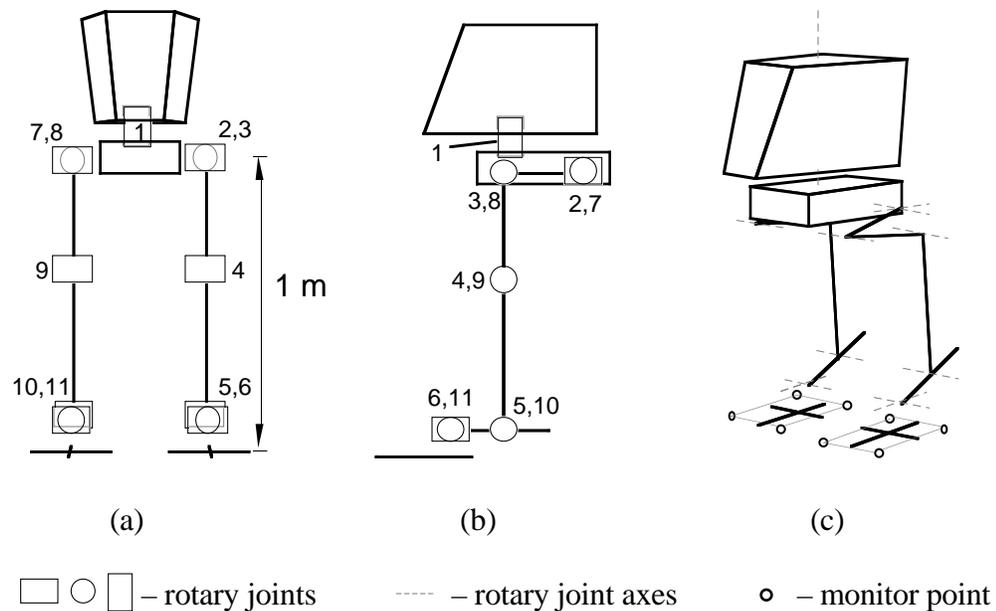
Figure 2.9 - Complex human model

- (a) front view (reference position)  
 (b) left side view (reference position)  
 (c) typical pose (with monitor points shown)

A second model has a bird-like structure similar to the biped All-terrain Scout Vehicle (ASV) robot in the motion picture "The Empire Strikes Back". The robo-bird creature, shown in Figure 2.10,

has 15 degrees of freedom, including 2 DOF hips and 2 DOF ankles. The models do not incorporate any physical joint limits. Dimensions, mass and inertia parameters for both models can be found in Appendix B.

These models are quite complex compared to much of the previous work on bipedal systems which have often assumed very simple bipedal models. In the next chapter, we will present an approach to generating animations of balanced motion for these creatures, which represents one of the major contributions of this thesis.



#### Degrees of Freedom

- |  |  |
|--|--|
| 1 - neck yaw (transverse plane)          |  |
| 2:0 - right hip roll (coronal plane)     | 7:0 - left hip roll (coronal plane)      |
| 2:1 - right hip pitch (sagittal plane)   | 7:1 - left hip pitch (sagittal plane)    |
| 3 - right knee1 pitch (sagittal plane)   | 8 - left knee1 pitch (sagittal plane)    |
| 4 - right knee2 pitch (sagittal plane)   | 9 - left knee2 pitch (sagittal plane)    |
| 5 - right knee3 pitch (sagittal plane)   | 10 - left knee3 pitch (sagittal plane)   |
| 6:0 - right ankle roll (coronal plane)   | 11:0 - left ankle roll (coronal plane)   |
| 6:1 - right ankle pitch (sagittal plane) | 11:1 - left ankle pitch (sagittal plane) |

Figure 2.10 - Robo-bird creature

- (a) front view (reference position)  
 (b) left side view (reference position)  
 (c) typical pose (with monitor points shown)

### 3. DISCRETE LIMIT CYCLE CONTROL

This chapter describes our basic control approach and its application to the generation of bipedal walking motions. Section 3.1 begins by describing the notion of limit cycles, on which our control formulation is based. Next, Section 3.2 presents the overall control strategy and develops a discrete system model to be used with a number of user-specified control elements to stabilize periodic open-loop motions. Section 3.3 discusses the application of this control system to bipedal walking. The underlying open-loop control, which serves as the basis for a desired gait, is discussed in Section 3.4. Sections 3.5 and 3.6 then go on to describe various possible observed and controlled variables for walking. Section 3.7 provides details on the application of the control elements introduced in earlier sections to the generation of balanced walks. Finally, minor variations on the basic control which are of particular use in improving the aesthetics of the human model's motion are described in Section 3.8. While the basic formulation is applied to bipedal walking, it is not inherently tied to any particular model or motion and could potentially be applied to the generation of limit cycles in other types of animated figures. Some of the terms we define in this and subsequent chapters have different meanings in the context of control system theory. While we make efforts to avoid such conflicts, we have chosen to sometimes give preference to the colloquial usage of terms. Our relaxed definition of a limit cycle is one example of such usage.

#### 3.1 Limit Cycles

One common characteristic of many non-linear dynamical systems is the existence of system-wide *limit cycles*. A limit cycle is a periodic, cyclic trajectory through the state-space of a system. Strictly speaking, a limit cycle involves the full state of the system. However, within the context of this thesis we will use a relaxed definition in which only part of the full system state must cycle

periodically. This definition is one of convenience since it allows periodic locomotion such as walking or running to be discussed in terms limit cycles.

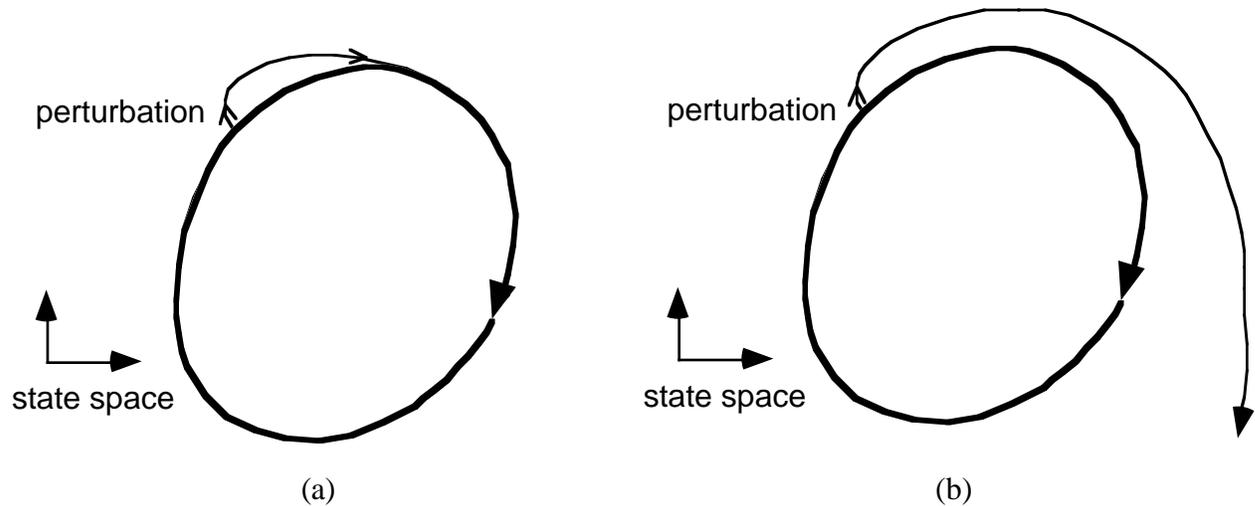


Figure 3.1 - Passive Limit Cycle Stability  
 (a) Passively stable  
 (b) Passively unstable

Limit cycles may be *stable* or *unstable*. A stable limit cycle is one in which slight perturbations to the state-space trajectory are driven back into the limit cycle as indicated in Figure 3.1 (a). An unstable limit cycle is one in which slight perturbations to the trajectory result in the system deviating further from the limit cycle as shown in Figure 3.1 (b). We will call limit cycles that do not require explicit control forces to maintain them *passive* limit cycles. Note that this definition does not preclude a system with active components (motors etc.) from exhibiting passive limit cycles. A motorized or windup toy is an example of such a system.

We wish to attain similar stable limit cycles with passively unstable bipedal systems by applying suitable control forces to periodically drive the system back into an *active* limit cycle. We define an *active* limit cycle as one that requires corrective control forces to be applied to the system for the explicit purpose of maintaining the cyclic trajectory. Figure 3.2 illustrates this idea.

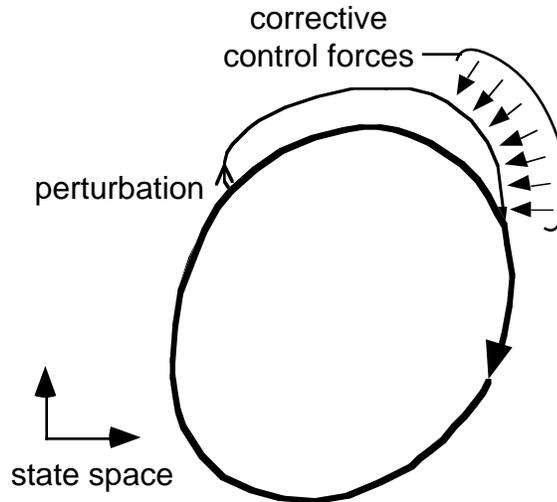


Figure 3.2 - An Active Limit Cycle

### 3.2 Control Formulation

The problem of choosing appropriate control perturbations to drive the *entire state* of a system to a desired value is a difficult one. Assuming a solution does exist, the number of parameters to be determined is large for all but very simple systems (a few DOFs or less). Non-linearities in a system mean that over the course of a full cycle even small perturbations of certain state variables can cause large changes in final state and/or result in almost no cycle at all. For example, a small change in the roll angle of the ankle in a walk might cause the next foot to miss the ground completely.

The essence of our control approach is to begin with a passively unstable system, discretize it into individual cycles and stabilize each cycle in turn. Each cycle is stabilized by applying control perturbations which drive its final state to a suitable state from which to begin the next cycle. The motivation for using a discrete version of the system is that the discrete dynamics are much simpler to model than the continuous system and therefore, simpler to control, as we shall see shortly.

The continuous dynamic system which we eventually wish to control can be modelled by the following system state equation:

$$x(t + \Delta t) = V(x(t), U). \quad (3.1)$$

where  $x$  is the system state and  $U$  is a set of applied control forces defined over  $[t, t + \Delta t]$ .

$V$  is a highly complex function which involves the integration of the forward dynamics of the animated model over time and includes the effect of both internal and external applied forces such as gravity, ground collisions, and muscular control forces. Instead of working directly with this complex continuous system, we assume that a strictly cyclic motion is desired and discretize Eq. 3.1 into individual motion cycles to obtain

$$x_{i+1} = g(x_i, U_i). \quad (3.2)$$

Here, the subscript  $i$  denotes the cycle number.  $U_i$  is the set of time-varying control forces applied over the  $i$ th cycle. The function  $g$  is a special case of  $V$  in which the sample times are not necessarily regular<sup>1</sup>, depending on the definition of a motion cycle. For example, the end of a motion cycle could be defined as the time of a particular transition in a state machine. We further assume that a user-supplied open loop controller,  $U^{nom}$ , produces a near-cyclic motion when applied to the system being controlled. To drive the final motion into a cycle, additional control forces are required. We denote these forces as  $\Delta U_i^*$ , which are the control perturbations required to drive each cycle of the nominal motion, toward a limit cycle. The discrete system then becomes

$$x_{i+1} = g(x_i, U^{nom} + \Delta U_i^*). \quad (3.3)$$

where  $\Delta U_i^*$  are still to be determined. Figure 3.3 illustrates this discrete dynamical system.

---

<sup>1</sup> Note that strictly speaking,  $g$  is a different function for each cycle since the size of the interval over which it is defined may vary from one cycle to the next.

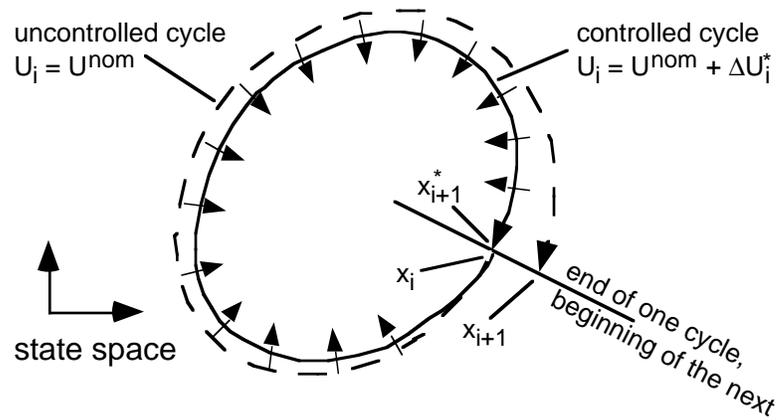


Figure 3.3 - Discrete System

- $x_i$  initial state of the  $i$ th cycle/final state of the  $(i-1)$ th cycle
- $x_{i+1}$  final state of the uncontrolled  $i$ th cycle
- $x_{i+1}^*$  final state of the controlled  $i$ th cycle/initial state of the  $(i+1)$ th cycle
- $U^{nom}$  basic cyclic control
- $\Delta U_i^*$  computed control perturbation for  $i$ th cycle

We will furthermore choose to express the corrective control forces,  $\Delta U_i^*$ , as the linear sum of several “basis” corrective actions:

$$U_i = U^{nom} + \sum_{j=1}^N k_{ij} \Delta U_j = U^{nom} + K_i \bullet \Delta U \quad (3.4)$$

where  $\Delta U_j$  are fixed control perturbations which are defined over a cycle and  $k_{ij}$  are linear scaling factors applied to  $\Delta U_j$ .  $K_i$  is the vector of perturbation scaling factors and  $\Delta U$  is a vector whose elements are the fixed control perturbations,  $\Delta U_j$ , which remain the same from one cycle to the next.  $N$  is the dimensionality of our control system and is equal to the number of state variables which we wish to observe (and control).

Rather than using the complete system state, we choose to work with a small number of *regulation variables* (RVs). Regulation variables are a projection of the system state and are the observed

part of state in our control system. Using a reduced set of state variables greatly reduces the computational effort required to construct a model of the discrete system. We will use  $Q$  to denote the vector of RVs and define  $\gamma(x)$  to be the projection function such that

$$Q = \gamma(x). \quad (3.5)$$

Note that choosing to work strictly in the reduced state space carries the implicit assumption that controlling the reduced state is sufficient to control the complete system state in a desirable way. For this to hold,  $\gamma(x)$  and  $\Delta U$  must be chosen appropriately. However, there is no guarantee that such an appropriate choice exists.

Replacing the full state in Eq. 3.3 with the reduced state,  $Q$ , and substituting the control formulation of Eq. 3.4 yields the reduced-order system which we will control directly:

$$Q_{i+1} = h(Q_i, U^{nom} + K_i \bullet \Delta U) \quad (3.6)$$

For a given cycle,  $Q_i$ ,  $U^{nom}$ , and  $\Delta U$  are treated as a priori information. We will use  $Q_{i+1} = h(K)$  as a convenient short form of Eq. 3.6 when we are interested in discussing only the effect of  $K$  on the system.

Eq. 3.6 can be restated as

$$Q_{i+1} = Q^{nom} + \Delta Q_{i+1} = h(Q_i, U^{nom} + K_i \bullet \Delta U) \quad (3.7)$$

where

$$Q^{nom} = h(Q_i, U^{nom}) \quad (3.8)$$

We choose to approximate the response of this system about the nominal operating point  $Q^{nom}$  (where  $K=0$ ) using the following linear predictive model:

$$\Delta Q_{i+1} = J K_i \quad (3.9)$$

where  $J$  is the  $N \times N$  discrete system Jacobian, defined as  $\left[ \frac{\partial q_i}{\partial k_j} \right]_{i,j \in [1,N]}$ :

$$J = \begin{bmatrix} \frac{\partial q_1}{\partial k_1} & \dots & \frac{\partial q_1}{\partial k_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_i}{\partial k_1} & \dots & \frac{\partial q_i}{\partial k_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_N}{\partial k_1} & \dots & \frac{\partial q_N}{\partial k_N} \end{bmatrix}. \tag{3.10}$$

$J$  relates the change in  $Q_{i+1}$  over a single cycle to the applied perturbation scaling,  $K_i$ .

Figure 3.4 illustrates the relationship between the parameters of the linear model and the perturbed system's cyclic motion in state space for a 1D system. In this example, the linear predictive model is constructed using two sample points on  $Q_{i+1} = h(k)$ , corresponding to applied perturbation scalings of  $k_1$  and  $k_2$ . The figure illustrates that we can predict (and hence control) the value of  $Q_{i+1}$ .

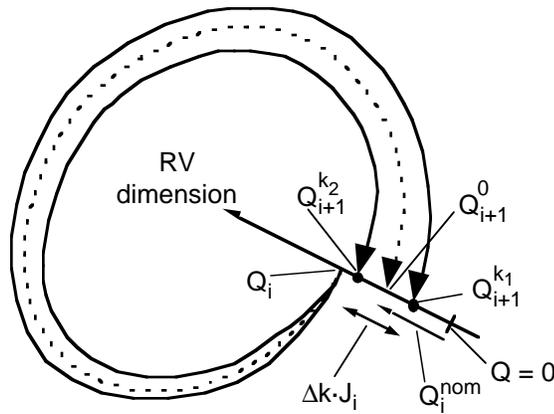


Figure 3.4 - Linear parameters of a 1D discrete system in state space.  
Here,  $\Delta k = k_2 - k_1$

Evidence supporting the use of a linear discrete system model for bipedal control and the details of model construction are presented later in the thesis (Section 3.6), after the particular choices of RVs and LPPs have been described.

The key to using the linear predictive model for determining the correct control action is the following. Once the linear discrete system model has been constructed, the control perturbation scalings required to drive  $Q_{i+1}$  to a desired value, can be computed using the inverse model:

$$K_i = J^{-1} \Delta Q_{i+1}^d \quad (3.11)$$

where  $\Delta Q_{i+1}^d = Q^{nom} - Q_{i+1}^d$ , the desired change in the RVs with respect to  $Q^{nom}$  for the current cycle.

Figure 3.5 shows the limit cycle viewed as state trajectory with respect to time. Three consecutive controlled cycles of a 1D discrete system are shown in this diagram,  $T$  is the cycle period,  $k_i$  is the perturbation scaling for cycle  $i$ , and  $V_i$  is the resulting state trajectory for cycle  $i$ . In this example,  $Q_{i+1}^d$ , the desired RV value for each cycle, is held constant for all three cycles.

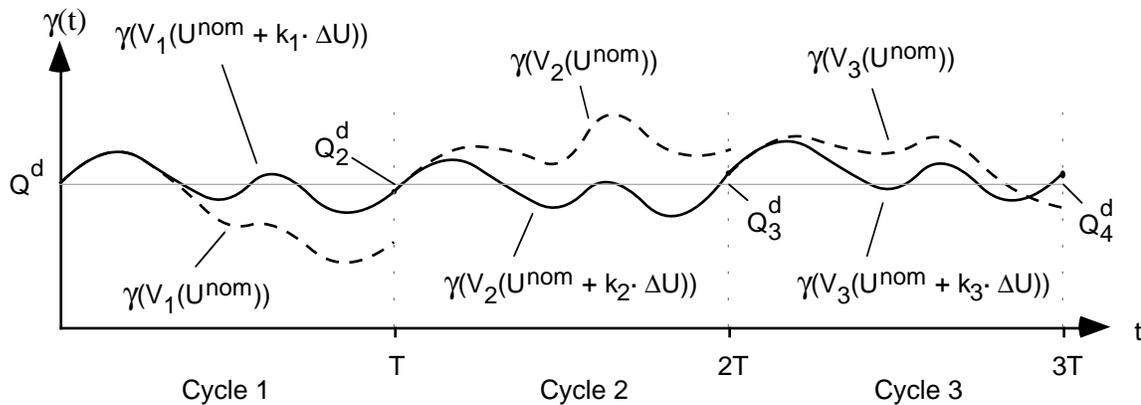


Figure 3.5 - Three cycles of a typical 1-dimensional system.  
In this case, each cycle has the same period,  $T$ .

Our control formulation assumes the following a priori information, supplied by the user:

1. the open-loop control,  $U^{nom}$ .

2. the RV projection function,  $\gamma$ .
3. the fixed control perturbations,  $\Delta U$ .
4. the desired (target) values for the RVs,  $Q_{i+1}^d$ , for each cycle,  $i$ .

The parameters of the discrete system model ( $Q^{nom}$  and  $J$ ) and the required perturbation scalings,  $K_i^*$ , are calculated automatically. These computations will be discussed in Section 3.7 after the various elements of the bipedal control system have been described.

### 3.3 Application to Bipedal Locomotion

In attempting to generate balanced locomotion for a biped, we must first select the number of control dimensions to be used. For successful balance, the base of support must, on average, remain under the centre of mass and the torso should remain generally upright. Only two control dimensions are required to achieve this, one in each dimension of the horizontal plane. Thus to balance each step, our bipedal control system will use two RV dimensions and require two LPPs which span the RV space. Since we use pose control exclusively with our biped we will use  $B$  and  $\Delta P$  to represent the pose control equivalents of  $U^{nom}$  and  $\Delta U$  respectively.

The base PCG,  $B$ , describes one complete cycle of motion, or in the case of bipedal walking, two steps. For bipedal walking we choose to split the cycle into two symmetric halves and apply our control formulation to each step. The PCG control perturbations,  $\Delta P$ , affect the motion throughout the cycle, rather than at a single point in the cycle.

Figure 3.6 illustrates the overall pose control structure used for our bipedal systems. The structure is an expansion of Eq. 3.4, for  $N=2$ , with the left step and right step perturbations specified explicitly. In summary, each step is balanced by the choice of two scalar parameters, for example,  $k_{l0}$  and  $k_{r0}$  for step 0. These parameters are calculated automatically using the discrete system model.

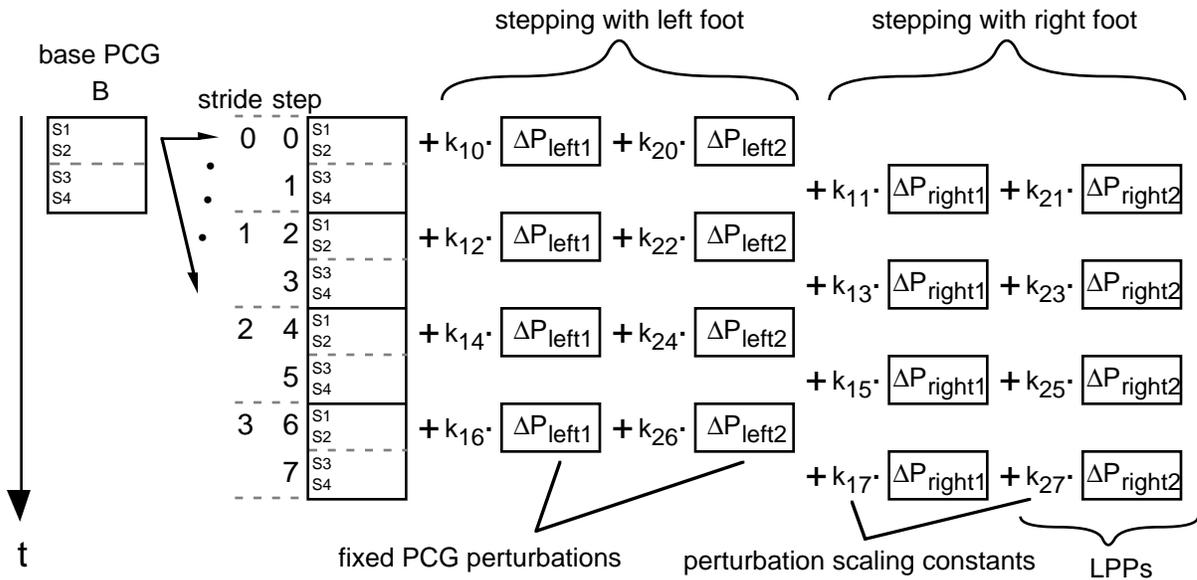


Figure 3.6 - Overall limit cycle control structure for a walking biped. The walk begins with the left leg as the stance leg. B is the open-loop base PCG. C is the overall applied control for each step, consisting of the base PCG B and additional (left or right) stance leg perturbations to balance the step. The  $S_i$  are the states of B (four in this example).

With suitable parameter choices, the control algorithm is able to generate stable walking limit cycles whose open loop motion is described by the base PCG.

Appropriate parameter choices are crucial for successful control. The next few sections describe each component of the control system for a walking biped. The process of choosing various parameters will be described as they are introduced.

**3.4 Nominal Open-loop Control,  $U^{nom}$**

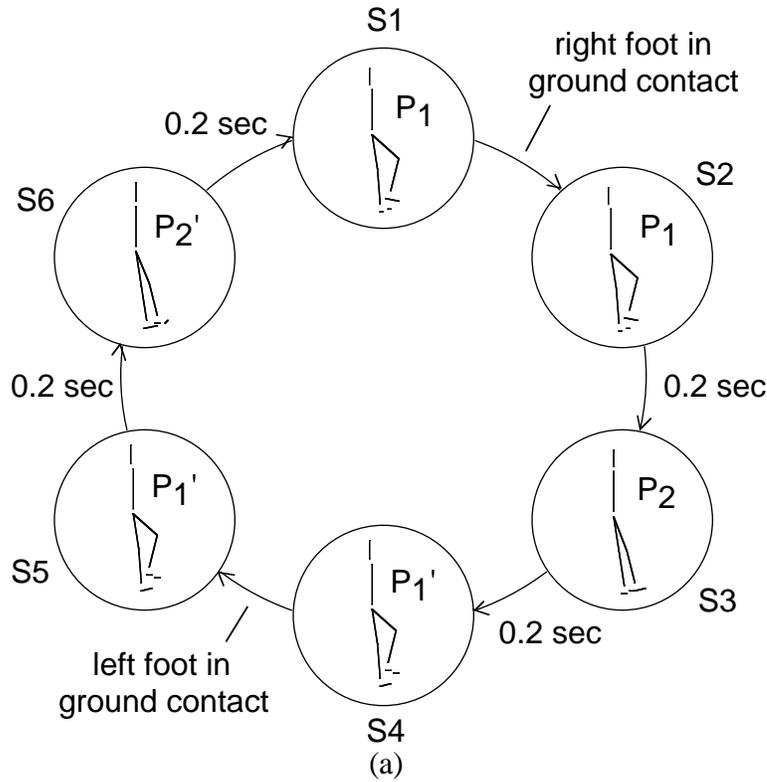
A base PCG provides the nominal open-loop control for the periodic motion of the biped. In the case of a forward walk, the base PCG consists of a sequence of poses to generate a forward step with one leg followed by a forward step with the other leg. The base PCG for walking must generate motion that is initially close to attaining a balanced walk when used without additional

compensating perturbations. A base PCG producing an open-loop walk of a few steps is more likely to be successfully stabilized than one that falls in the first step.

Figure 3.7 shows a base PCG used to generate a forward walking motion for the simplest human model along with its equivalent graphical representation. The base PCG has right-left symmetry, although PCG asymmetry can also be quite useful as will be demonstrated by the turning perturbations presented in Section 6.2.

The base PCG consists of six states. The poses in states S1 and S2 are identical. The poses in states S4, S5, and S6 are identical to those of states S1, S2 and S3 respectively, with the left and right sides exchanging roles. Having a total of only four unique poses, the controller describes a simpler motion than a typical human walk. This relatively small number of poses simplifies the specification of a base PCG as well as the creation of suitable parametric perturbations.

Sensor-based transitions are necessary to synchronize the biped's motion to the external world. The base PCG of Figure 3.7 uses such transitions to move from S1 to S2 and from S4 to S5. A transition occurs when the swing foot for the current step (i.e. the next stance foot) contacts the ground. If this has already happened before the state is entered, the transition occurs immediately. Actively ensuring ground contact at fixed points in the walk cycle is important in certain situations such as when starting from an initial resting state. While the sensors are typically no longer necessary once a walking limit cycle is reached, changing from one limit cycle to another can require sensor-based synchronization again.



(a)

DOF:	1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1	transition info
S1	0	0	0	-50	60	-5	0	0	-10	0	0	0	0 R
S2	0	0	0	-50	60	-5	0	0	-10	0	0	0	.2
S3	0	0	0	-20	0	5	0	0	-10	0	0	0	.2
S4	0	0	0	-10	0	0	0	0	-50	60	-5	0	0 L
S5	0	0	0	-10	0	0	0	0	-50	60	-5	0	.2
S6	0	0	0	-10	0	0	0	0	-20	0	5	0	.2

(b)

Figure 3.7 - Forward walking base PCG for a simple human model  
 (a) State diagram (right foot dashed). Poses  $P_i$  and  $P_i'$  are left-right symmetric.  
 (b) Pose table. All DOFs are in degrees relative to reference position. Transition information is given as time (seconds) followed by an optional foot sensor type (left or right) for sensor-based transitions.

- | State | Description   |
|-------|---|
| S1    | Right foot placed on ground<br>Left leg begins forward swing (knee bent for ground clearance)       |
| S2    | Right leg propels body forward<br>Left leg continues forward swing (knee bent for ground clearance) |
| S3    | Left leg extends in anticipation of ground contact  |
| S4    | Left foot placed on ground<br>Right leg begins forward swing (knee bent for ground clearance)       |
| S5    | Left leg propels body forward<br>Right leg continues forward swing (knee bent for ground clearance) |
| S6    | Right leg extends in anticipation of ground contact   |

Figure 3.8 shows the open loop motion resulting when the base PCG alone is used for control. As expected, the biped falls over after several steps.

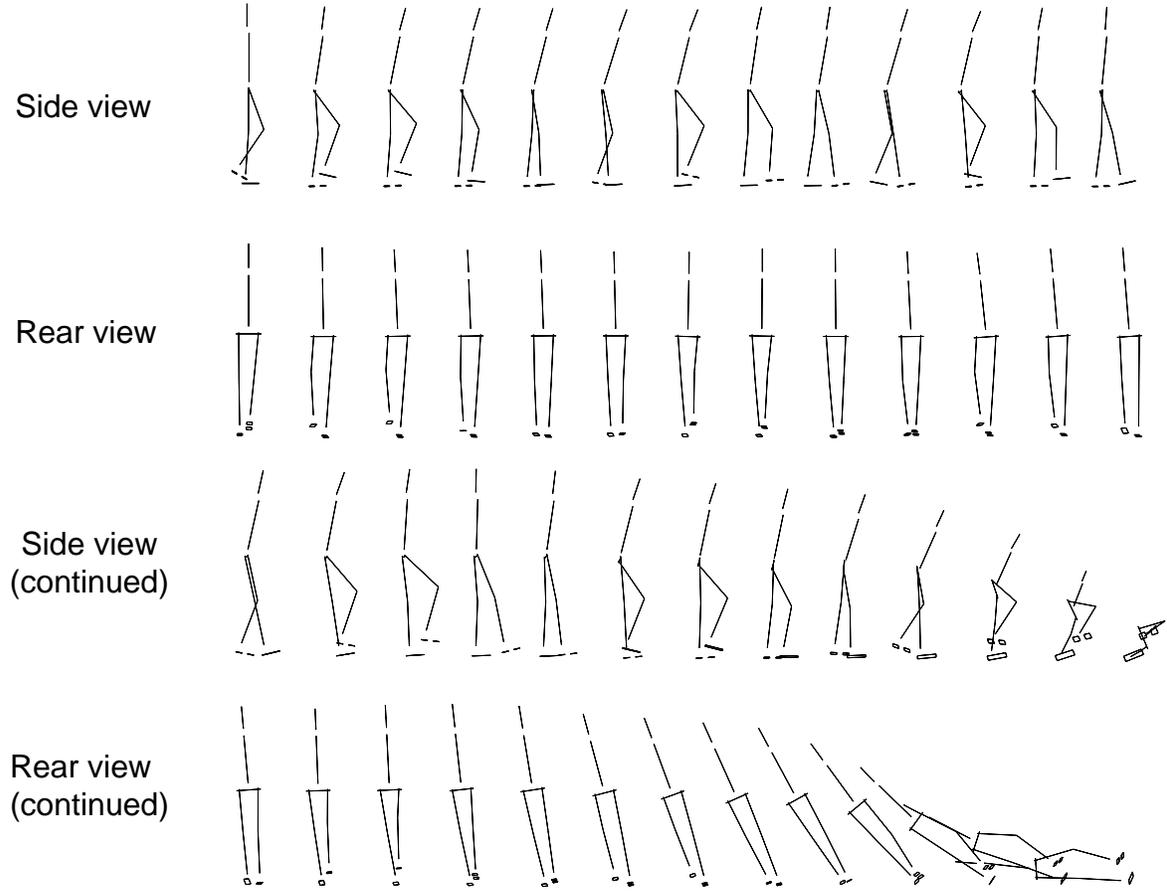


Figure 3.8 - Unbalanced motion of human model using base PCG shown in Figure 3.7.

Surprisingly, the small number of poses is sufficient not only for walking, but also for a limited form of running. Running can be achieved simply by replacing the 0.2 second time-based transitions of the walking base PCG with 0.1 second transitions. The flexibility of the pose controller to accommodate the different motions is due to the ground sensors which are used in state transitions. The sensor-based transitions exist to ensure that the next stance foot is in contact with the ground before proceeding into its stance phase. However, they do not constrain the initial time of ground contact. In the case of running, ground contact typically occurs shortly after entering state S1 or S4 as the new stance leg moves to propel the biped forward. In the case of

walking, ground contact typically occurs in the previous state (S6 or S3) toward the end of the swing leg extension. These two scenarios are illustrated in Figure 3.9.

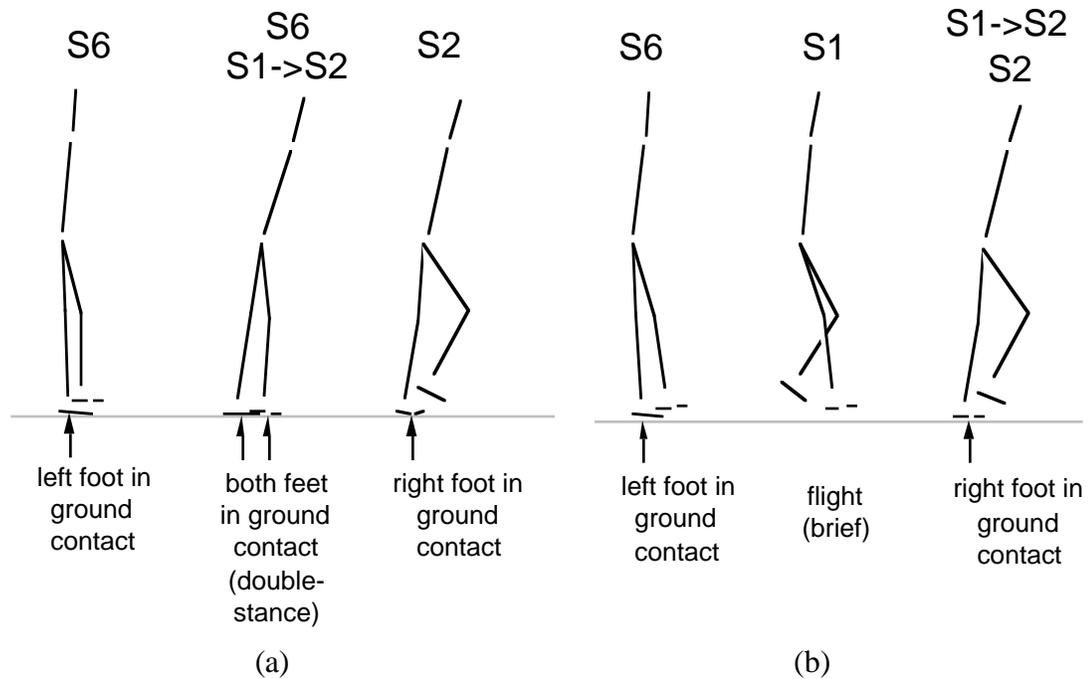


Figure 3.9 - Possible walking and running steps described by base PCG of Figure 3.7. Related states and transitions are indicated.  
 (a) walking  
 (b) running

An initial state must also be chosen for the walk. The initial state provides a starting position from which a successful walking motion can begin. While the simplest initial state to use would have the creature at rest in the reference position, it is easier to achieve a limit cycle when the initial state more closely resembles a state on a balanced walk cycle. Finding an appropriate initial state currently requires trial-and-error on the part of the animator. Figure 3.10 shows the initial state which was generated manually for the simple human model controlled by the base PCG of Figure 3.7.

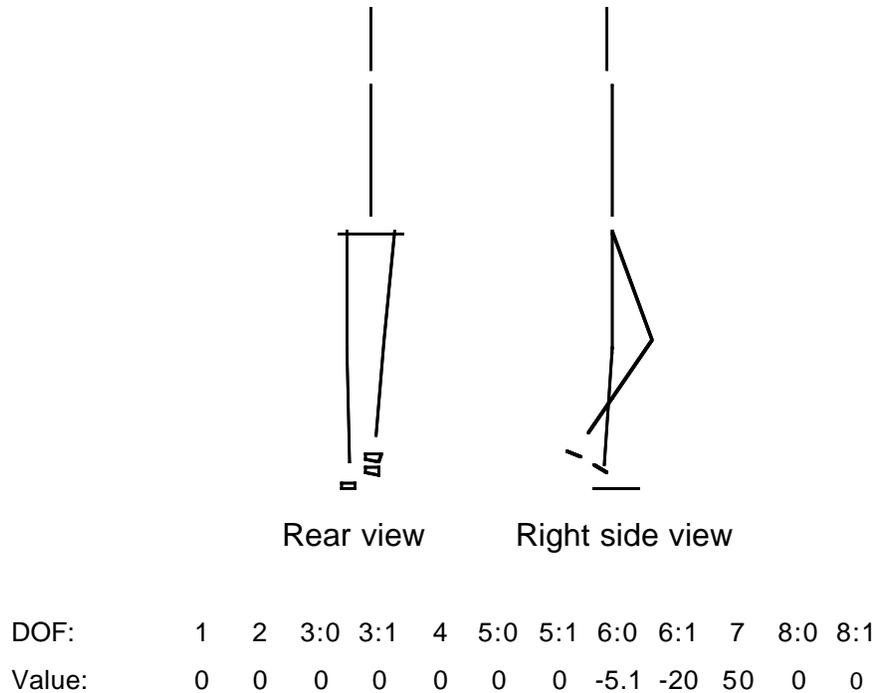


Figure 3.10 - Initial configuration for simple human model.  
Right foot shown as two segments for clarity. All velocities are 0.

### 3.5 Choice of Regulation Variables, $Q$

As discussed earlier, the discrete balance control of Eq. 3.6 is applied only once per step. This implies that each regulation variable must represent the behaviour of some part of the system state over an entire step as a single scalar value. It is important to choose suitable functions of state and sampling times which give a reasonably smooth response to perturbations as we assume in Section 3.2 by using the discrete system Jacobian.

All of the RVs presented in this thesis are projections of the system state at a specific point in the cycle as shown in Figure 3.11. The sample times correspond to the approximate time of foot placement for each step. Variations such as sampling an average or peak value of some function of state over the whole step are also promising but unexplored possibilities.

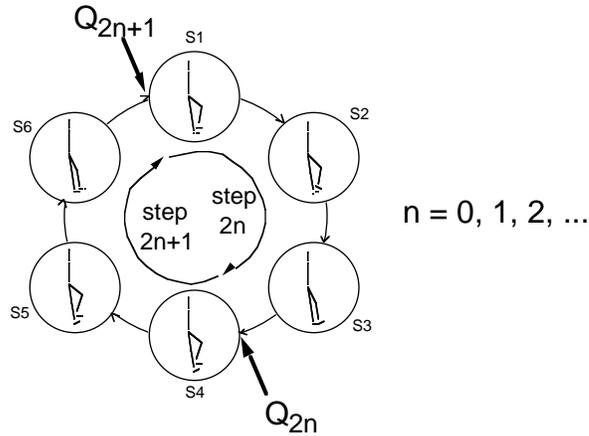


Figure 3.11 - End-of-step sampling times for  $Q_i$ .

RVs must be chosen to be meaningful functions of the system state in order to improve the chances of attaining a limit cycle. A number of general criteria should be met:

1. RVs should incorporate some key element(s) of the desired cycle and should reflect movement or position relative to the world frame. An example that meets both criteria is “uprightness” with respect to the world vertical axis for the case of walking.
2. The RVs should be easily controllable. It should be possible to reach a wide range of RV values from various initial states by choosing an appropriate parametric control perturbation. For example, the system must have DOFs which can affect the chosen RVs.
3. RVs should vary smoothly around a reasonable range of interest. For example, in the case of a walking biped, the range of interest might correspond to configurations with a relatively upright torso.
4. We should be able to estimate a target value for each RV which will keep the motion close to a desirable limit cycle. For example, to attain a walking limit cycle, we must be able to determine target RV values that will keep the biped balanced.

5. When multiple RVs are used, independence or near-independence of the RVs can be useful. RV interdependencies can be complex (e.g. non-smooth), making it more difficult to find parametric perturbations which cause only smooth variations over a wide range of RV values. For our bipedal control, we assume near-independent RVs.

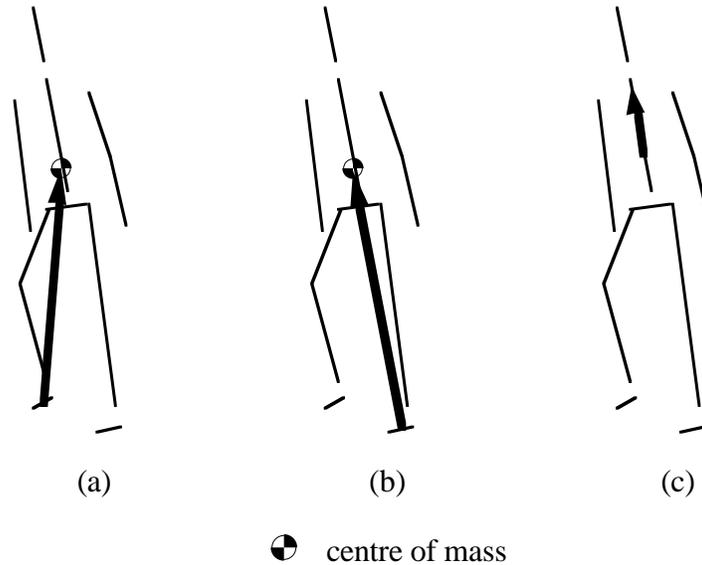


Figure 3.12 - Balance RV vectors  
 (a) swing-COM vector  
 (b) stance-COM vector  
 (c) up vector

In this thesis, we experiment with three choices of RVs, based on the vectors shown in Figure 3.12. The first, the *up-vector*, is based on the notion of torso "uprightness". The up-vector is fixed to and runs along the length of the torso in the human model and the head in the robo-bird model. The *swing-centre of mass (swing-COM) vector* describes the position of the COM of the biped with respect to the current swing foot. The *stance-centre of mass (stance-COM) vector* indicates the position of the COM with respect to the stance foot. The sampling time for all three types of RV are at the end of states S3 and S6. For the purpose of computing RVs, the swing and stance legs do not exchange until after the last base PCG state of the step. The leg which is the swing leg for most of the current step is used to compute the swing-COM RV. The stance-COM RV is treated in a similar fashion.

The RVs are two scalar components extracted from the given vector. The vector is sampled at the end of the current step, normalized, and projected onto the horizontal plane. The projection is then decomposed into the two components as illustrated in Figure 3.13 for the up-vector case. The components provide an indication of the forward and lateral lean of the chosen vector. The forward direction is defined as being orthogonal to the ground-plane projection of a line joining the biped's hips.

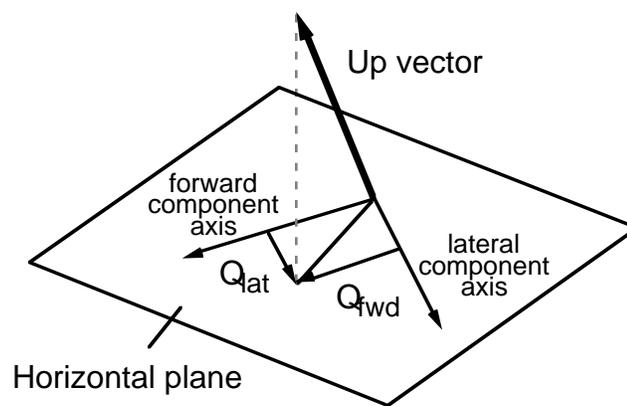


Figure 3.13 - Decomposition of up vector projection into RVs

### 3.6 Choice of Perturbation Parameters, $\Delta U$

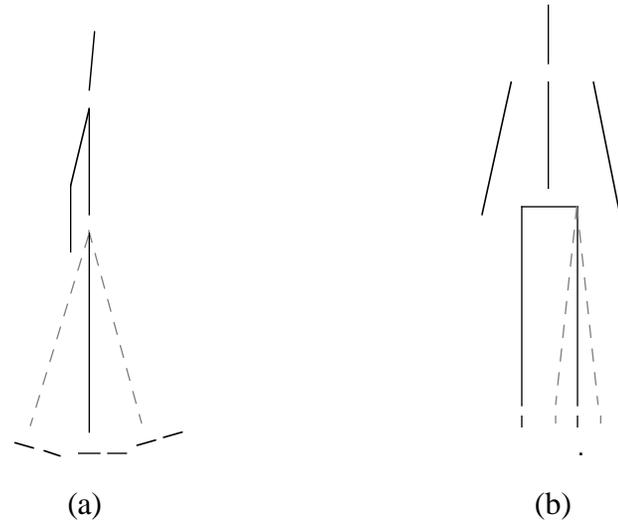
The balance control formulation of Figure 3.6 uses two linear parametric perturbations (LPPs) to control each step in the base PCG cycle. Recall from Section 2.3 that each LPP consists of two basic components, a *fixed PCG perturbation*, and a scalar multiplier. New values for each scalar multiplier are automatically computed for each step to balance that step. The fixed part of the PCG perturbations remain the same throughout a given walk. This section discusses the choice of fixed PCG perturbations which must be supplied by the animator. Throughout this section, the term "perturbation" will be used to refer strictly to the fixed part of an LPP rather than an arbitrarily scaled perturbation.

As in the case of RVs, the choice of perturbations should meet a minimal set of requirements to be effective. The criteria for good PCG perturbations are related to the criteria for suitable RVs, discussed in Section 3.5. In general, the following should hold:

1. The scaled perturbation should be able to reach a wide range of desired RV values from a wide variety of initial conditions.
2. Perturbations should provide smooth control over the RVs so that a suitable linear model of the discrete system can be constructed. The inverse of this model is used to determine the appropriate scaling factors for each perturbation for each step. For bipedal walking, the chosen perturbations provide near-linear control over each of the three balance control RVs presented in Section 3.5.
3. Perturbations should be designed to be as independent as possible in order to reduce or eliminate non-linear interactions between RVs.

The control perturbations chosen in this thesis affect the stance hip pitch and stance hip roll DOFs. They are shown in Figure 3.14. Items (c) and (d) show the pose table form of the perturbations which are scaled and added to a base PCG pose table such as that of Figure 3.7 (b). In our case, each of the chosen PCG perturbations affects a single DOF. Unity-valued perturbations are used so that the scalar multiplier units are in degrees. Each perturbation is applied to all poses (i.e. all states) of the related step, left or right. This provides a smoother and more effective control perturbation.

Figure 3.14 illustrates the hip pitch and roll perturbations for a body in free space. When applied to a body in contact with the ground, the perturbations primarily affect the biped as shown in Figure 3.15. Applying a stance hip pitch perturbation varies the torso angle in the sagittal plane. Applying a stance hip roll perturbation varies the torso angle in the coronal plane.



DOF:	1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1	transition info
state S1	0	0	0	0	0	0	0	0	1	0	0	0	0
S2	0	0	0	0	0	0	0	0	0	1	0	0	0
S3	0	0	0	0	0	0	0	0	0	1	0	0	0
S4	0	0	0	0	0	0	0	0	0	0	0	0	0
S5	0	0	0	0	0	0	0	0	0	0	0	0	0
S6	0	0	0	0	0	0	0	0	0	0	0	0	0

(c)

DOF:	1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1	transition info
state S1	0	0	0	0	0	0	0	1	0	0	0	0	0
S2	0	0	0	0	0	0	0	0	1	0	0	0	0
S3	0	0	0	0	0	0	0	1	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0	0	0	0	0
S5	0	0	0	0	0	0	0	0	0	0	0	0	0
S6	0	0	0	0	0	0	0	0	0	0	0	0	0

(d)

Figure 3.14 - Right hip pitch and roll perturbations for a simple human model. During states S1 to S3, the right leg is the stance leg. DOF 6:1 is right hip pitch. DOF 6:0 is right hip roll.  
 (a) hip pitch (viewed from right)  
 (b) hip roll (viewed from rear)  
 (c) hip pitch (pose table)  
 (d) hip roll (pose table)

Figure 3.16 shows the relationship between the stance hip pitch perturbation and the forward and lateral component of each of the three types of RVs introduced in Section 3.5. Figure 3.17 shows similar results for the stance hip roll perturbation. Each curve represents the value of the

particular RV at the end of a single step for a range of applied perturbation scalings. For each curve, a different initial state is used. The initial state for step  $n$  was generated by first simulating  $n-1$  steps with the open-loop base PCG. By the fourth step, the figure is falling noticeably.

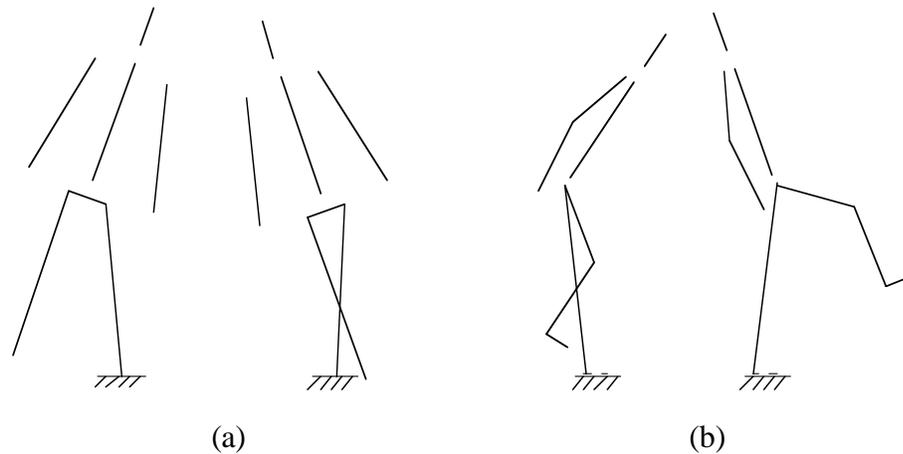


Figure 3.15 - Typical effect of stance hip perturbations when dynamics are considered  
 (a) right stance hip roll  
 (b) right stance hip pitch

The graphs illustrate that hip pitch varies nearly linearly with  $Q_{fwd}$  for all three choices of RV. Similarly, hip roll is nearly linear with respect to  $Q_{lat}$ . These relationships provide evidence which supports our assumption that the discrete system can be modelled using a linear model. Despite the fact that the perturbations themselves are mutually independent, their effect on the RVs is not cleanly decoupled. Thus, they do not provide truly independent control over each RV dimension as desired. In general, however, the magnitudes of the undesired variations are not excessively large relative to the accessible range of RV values in the desired control dimensions. Completely independent control of the RVs would imply a diagonal discrete system Jacobian. The relatively small effects of each perturbation on other's control dimension mean that the off-diagonal elements of the discrete system Jacobian will be small compared to the diagonal elements.

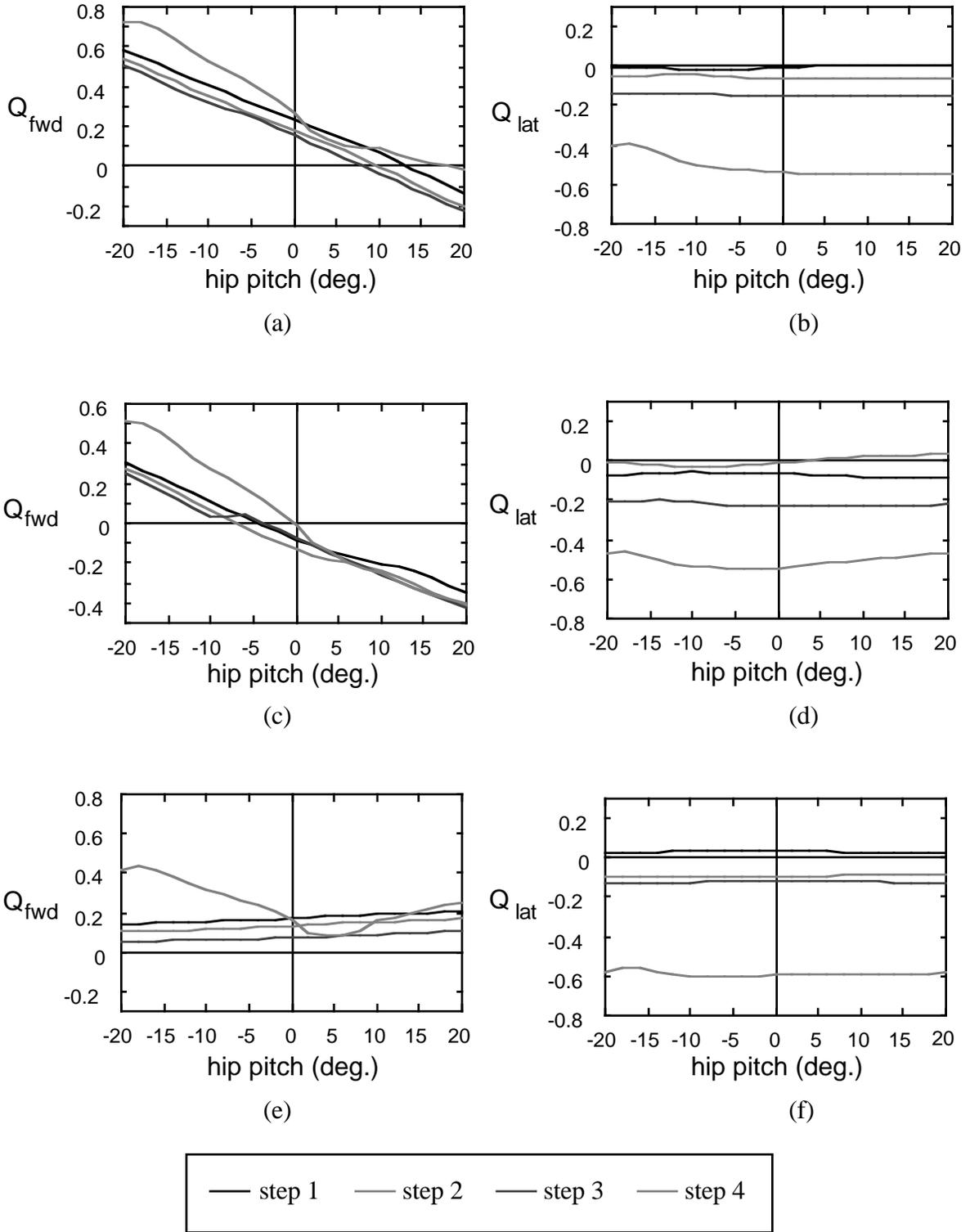


Figure 3.16 - Balance RV components vs linearly scaled hip pitch perturbation  
 (a) - (b) up vector  
 (c) - (d) swing-COM vector  
 (e) - (f) stance-COM vector

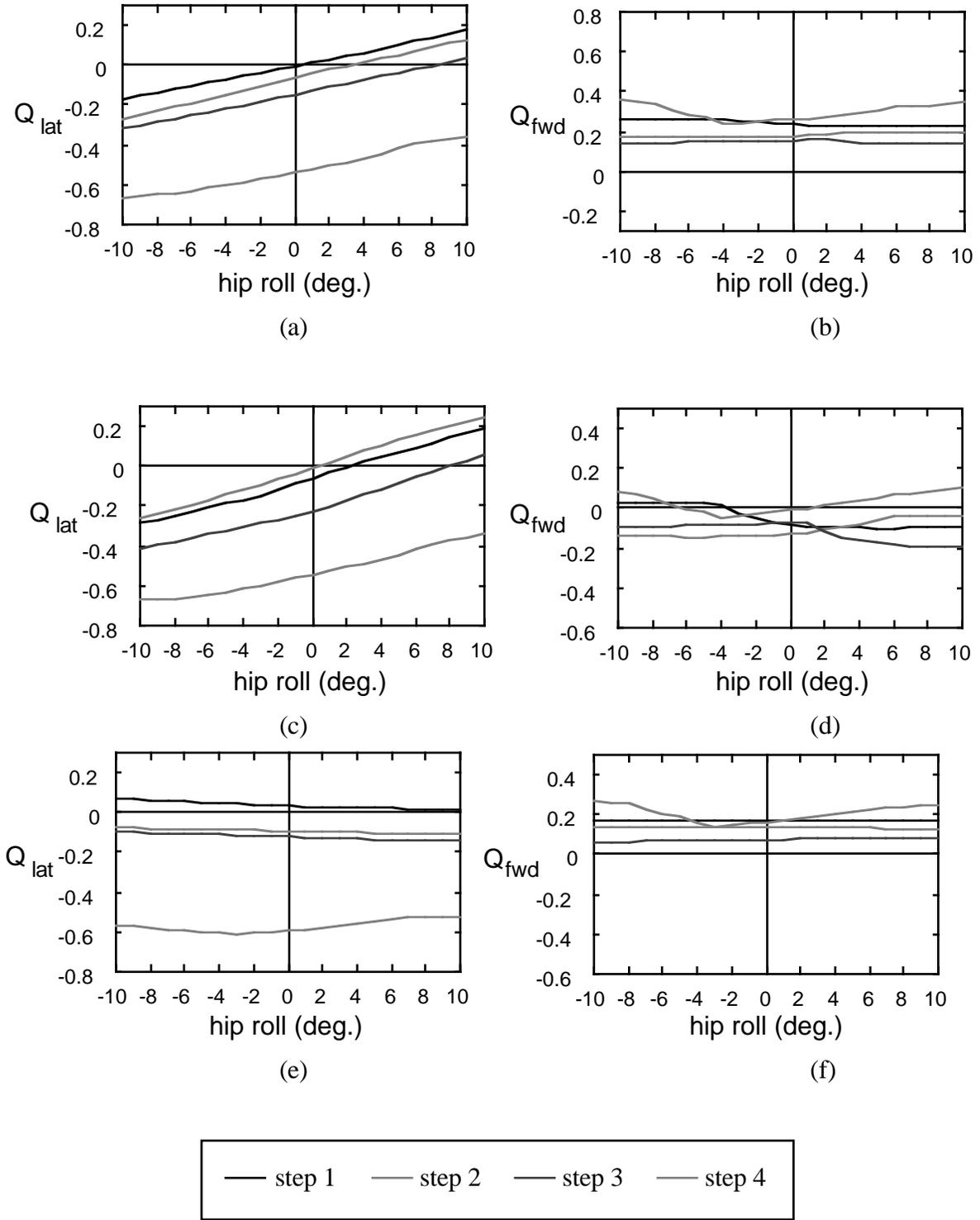


Figure 3.17 - Balance RV components vs linearly scaled hip roll perturbation  
 (a) - (b) up vector  
 (c) - (d) swing-COM vector  
 (e) - (f) stance-COM vector

These relationships allow us to attain final values for  $Q$  close to the desired values for each step using a simplified form of the Jacobian, discussed in the next section.

One interesting point to note is that the slope of the stance-COM curves in Figure 3.16 and Figure 3.17 are opposite in sign compared to the corresponding slopes for the up vector and swing-COM RVs. This is due to the mechanism through which the stance-COM angle changes. Figure 3.18 shows this effect exaggerated for clarity.

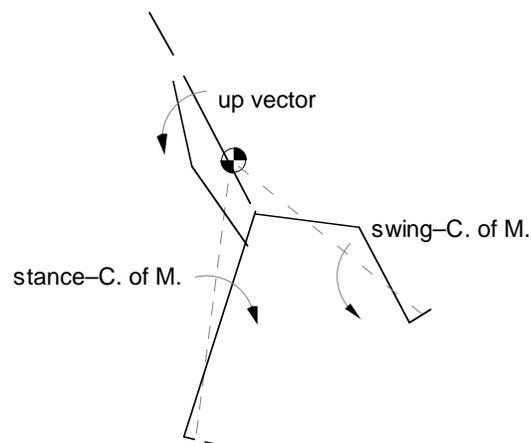


Figure 3.18 - Direction of change of forward RV components with hip pitch

### 3.7 Linear, Sampled "Balance" Control

This section describes how the base PCG, perturbations and various RVs discussed in earlier sections can be used to compute and apply the discrete system model parameters ( $J$  and  $Q^{nom}$ ) to generate a balanced walk. The "balancing" is done by choosing appropriate RV target values for each step and finding the scaling factors to apply to the PCG perturbations to reach them. The scaling factors are determined automatically using the inverse of the linear discrete system model (Eq. 3.11). The balancing process is repeated, one step at a time, for as many steps as desired. In some cases, the resulting walk is erratic and wanders. In others, a walking limit cycle is reached.

Two separate problems must be solved in order to achieve the motions we seek:

1. We must choose an RV target value,  $Q^d$ , on the cycle, which is likely to drive the system into a reasonable limit cycle.
2. We must construct the discrete system model of Eq. 3.9 and then use it to compute the final control perturbation for each step. This involves calculating the discrete system Jacobian,  $J$ , and the nominal operating point,  $Q^{nom}$  for each step.

### 3.7.1 Desired RV Values

The desired (target) values for the RVs should be chosen each step to keep the system near a limit cycle. We choose to use a constant target value for all steps. The idea is that by forcing the RVs to the same desired value at the same point in each cycle, a limit cycle will be generated. This approach is sufficient both to drive the system into a stable limit cycle initially and to maintain it. Allowing the RV targets to vary from one step to the next can be useful, but requires a way to select them. This possibility is further explored in Chapter 5 where variation in  $Q^d$  is used to provide control over the biped's walking speed.

Once the choice is made to use constant RV targets, the particular values must be chosen. This is essentially a trial-and-error process. However, common sense and the behaviour of the open-loop system can provide valuable clues. For example, if a straight forward walking motion is desired, the lateral RV target can be quickly estimated using the final pose in one PCG step as shown in Figure 3.19. Similarly, the first few steps of the open loop motion (such as that in Figure 3.8) can provide a reasonable estimate for the forward component targets. Finally, performing a number of trials with different forward  $Q^d$  values provides a solution, if one exists, as well as information on the useful range of target values. In this case, a "trial" is an attempt to balance the biped's motion by applying our proposed control technique with some particular trial value of  $Q^d$ .

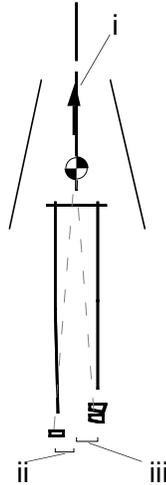


Figure 3.19 - Estimate of desired lateral RV components  
 (based on S6 from Figure 3.7 (a) - rear view)  
 i) up vector (vertical)  
 ii) stance-COM  
 iii) swing-COM

### 3.7.2 Constructing and Applying the Discrete System Model

Once an RV target has been chosen, the discrete system model is reconstructed and applied at each step as shown in Figure 3.20. In general, a single fixed model is not sufficient to represent all possible steps. While RVs provide a reduced, low dimensional representation of system state, they are also an incomplete and ambiguous representation of the system state. For example, the chosen RVs use position information and no velocity information. Such unobserved parts of the system state cause variations in both  $Q^{nom}$  and  $J$ . Figure 3.21 illustrates this idea. This result is evident in graphs of Figure 3.16 and Figure 3.17 where both the average slope and the offset of the curves can vary from one step to the next. These variations are especially apparent in the lateral control dimension.

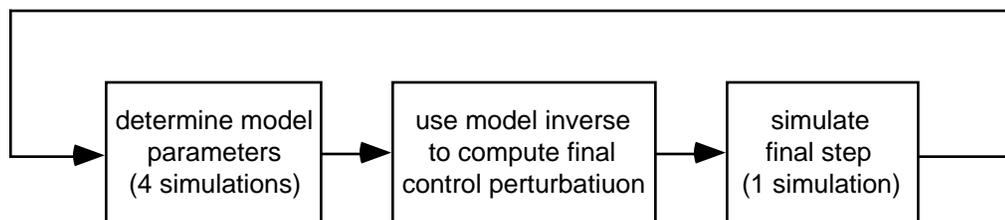


Figure 3.20 - Balancing process for each step.

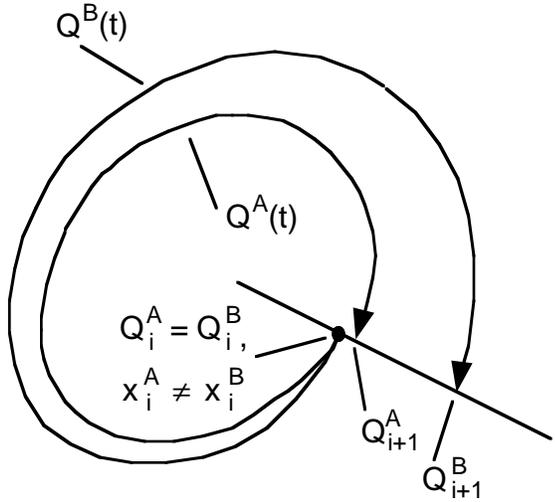


Figure 3.21 - Unobserved state problem. Two unperturbed RV trajectories beginning with identical values for  $Q_i$  but different initial states.

Because of the variation of  $Q^{nom}$  and  $J$  from one step to the next, it is (unfortunately) necessary to construct a new discrete system model for each step. This is done by sampling  $h(K_i)$  to determine the model parameters for the current step,  $Q_i^{nom}$  and  $J_i$ . The inverse of this linear model,  $J^{-1}$ , is then used to choose the perturbation scalings necessary to reach  $Q_{i+1}^d$ . Figure 3.22 illustrates this process for a one-dimensional system.

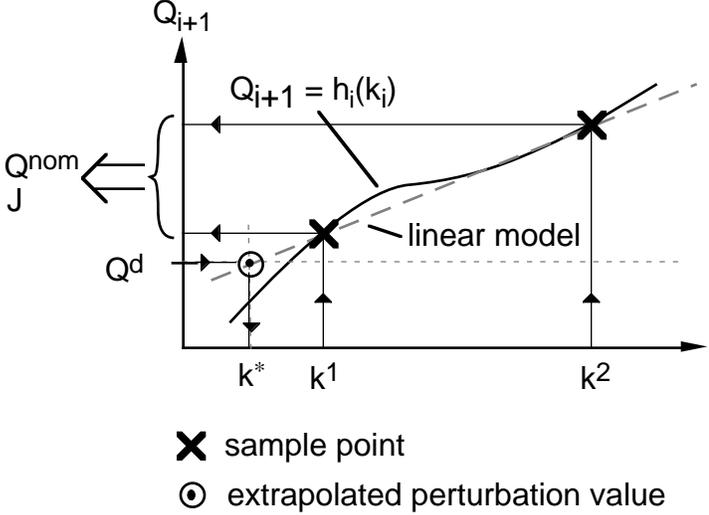


Figure 3.22 - Model construction and extrapolation.  $k^1$  and  $k^2$  are two sample points used to construct the forward model. The inverse model and  $Q^d$  yield the necessary applied perturbation scaling,  $k^*$

In general, there is no guarantee that the discrete system Jacobian will be invertible for each step. However, with well-chosen choices of RVs and LPPs, singularities generally occur only for trajectories far enough from useful limit cycles to be of little interest. Typical singularities for the human model, occur when it is lying on its side and when the current stance leg does not contact the ground at all during the current step.

The general form of the Jacobian for a 2D system is:

$$J = \begin{bmatrix} \frac{\partial q_1}{\partial k_1} & \frac{\partial q_1}{\partial k_2} \\ \frac{\partial q_2}{\partial k_1} & \frac{\partial q_2}{\partial k_2} \end{bmatrix} \quad (3.10)$$

Using finite differences, a minimum of  $N+1$  samples are required to construct the linear model of  $h(K)$  for a control system of dimension  $N$ : one for the nominal operating point and one for each control dimension, each of which yields a column in the Jacobian. In this case, each “sample” consists of a simulation of one step with a different value of  $K$ . In practice, a greater number of samples may be required.

For the two dimensions of our bipedal control system, we use four sample simulations, two for each dimension. An additional simulation computes the final motion for the step after the final PCG scalings have been chosen using the model. Rather than using the complete Jacobian, we choose to work with two simplified forms:

1. Assume independent control dimensions and observe only the primary RV. This corresponds to the assumption of a diagonal Jacobian (i.e.  $\partial q_i / \partial k_j = 0, i \neq j$ ). We will refer to this form as *superposition* (SP) sampling.
2. Assume near-independent control dimensions but allow the operating point to move as each final perturbation scaling is determined. This corresponds to a form of triangular

Jacobian which makes use the most up-to-date values of the model parameters in a manner similar to the Gauss-Seidel method for solving linear systems.

For the first, two independent 1D models are constructed then combined in the final simulation. For the second, two variations are considered. In the first, a 1D model is constructed in the forward control dimension and inverted to obtain the required  $K_{fwd}^*$ . A second 1D model is then constructed in the lateral control dimension making use of the known value of  $K_{fwd}^*$ . We refer to this as forward-then-lateral (F-L) sampling. Lateral-then-forward (L-F) sampling is similar but performs 1D control in the lateral dimension first, then uses this result to construct a 1D model in the forward dimension. We might expect that one or both of these approaches perform better than superposition sampling since they both incorporate additional knowledge of the perturbations to be applied. Somewhat surprisingly, this turns out not to be the case as we shall see in Chapter 4. All three approaches give comparable results, with superposition slightly outperforming the other two. The three sampling strategies investigated are illustrated in Figure 3.23.

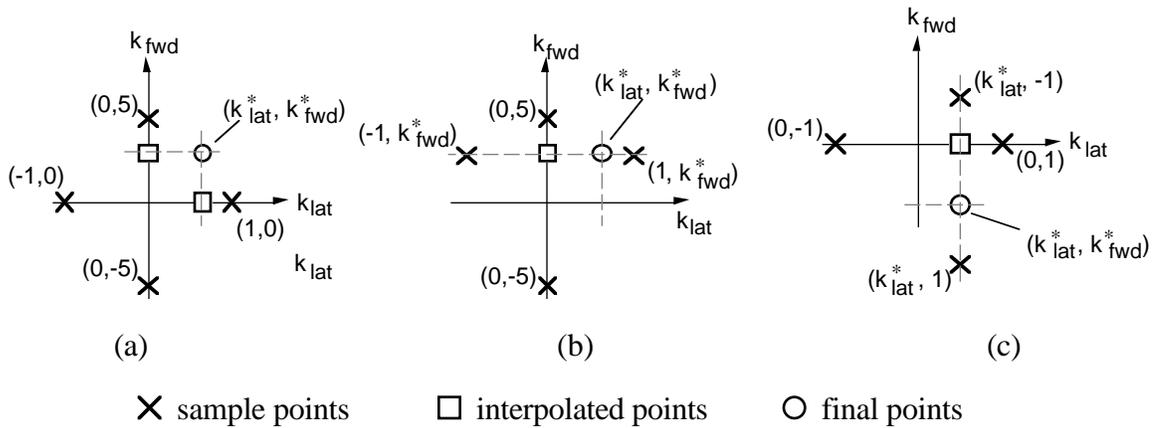


Figure 3.23 - 2D sampling strategies  
 (a) superposition (SP)  
 (b) forward then lateral (F-L)  
 (c) lateral then forward (L-F)

The final parameters to consider in constructing a good linear model of the discrete system are the perturbation scaling factors used to sample  $h(K)$  (e.g.  $k^1$  and  $k^2$  in Figure 3.22). Since the

discrete system is quite linear, the precise location of the sample points is not critical. There are, however, a few potential pitfalls to be avoided in choosing them:

1. Since we cannot necessarily predict the sign of the required perturbations from one step to the next, it is often best to sample  $h(K)$  symmetrically about the nominal operating point,  $K=0$ , rather than at the nominal point itself. Figure 3.24 (a) shows such a case.
2. Samples should span a sufficiently large range in order to avoid local distortions in  $h(K)$ . A typical failure due to this problem can be seen in Figure 3.24 (b).
3. Sample points should not consist of excessively large control perturbations. When the current state is far from the limit cycle, large control perturbations can result in the biped falling over. Including a fall as a sample can lead to a poor model. This situation is illustrated in Figure 3.24 (c).

Sample scaling factors of  $k_{fwd} = \pm 5$  degrees for hip pitch and  $k_{lat} = \pm 1$  degree for hip roll give acceptable linear models for our biped models over a wide variety of gaits.

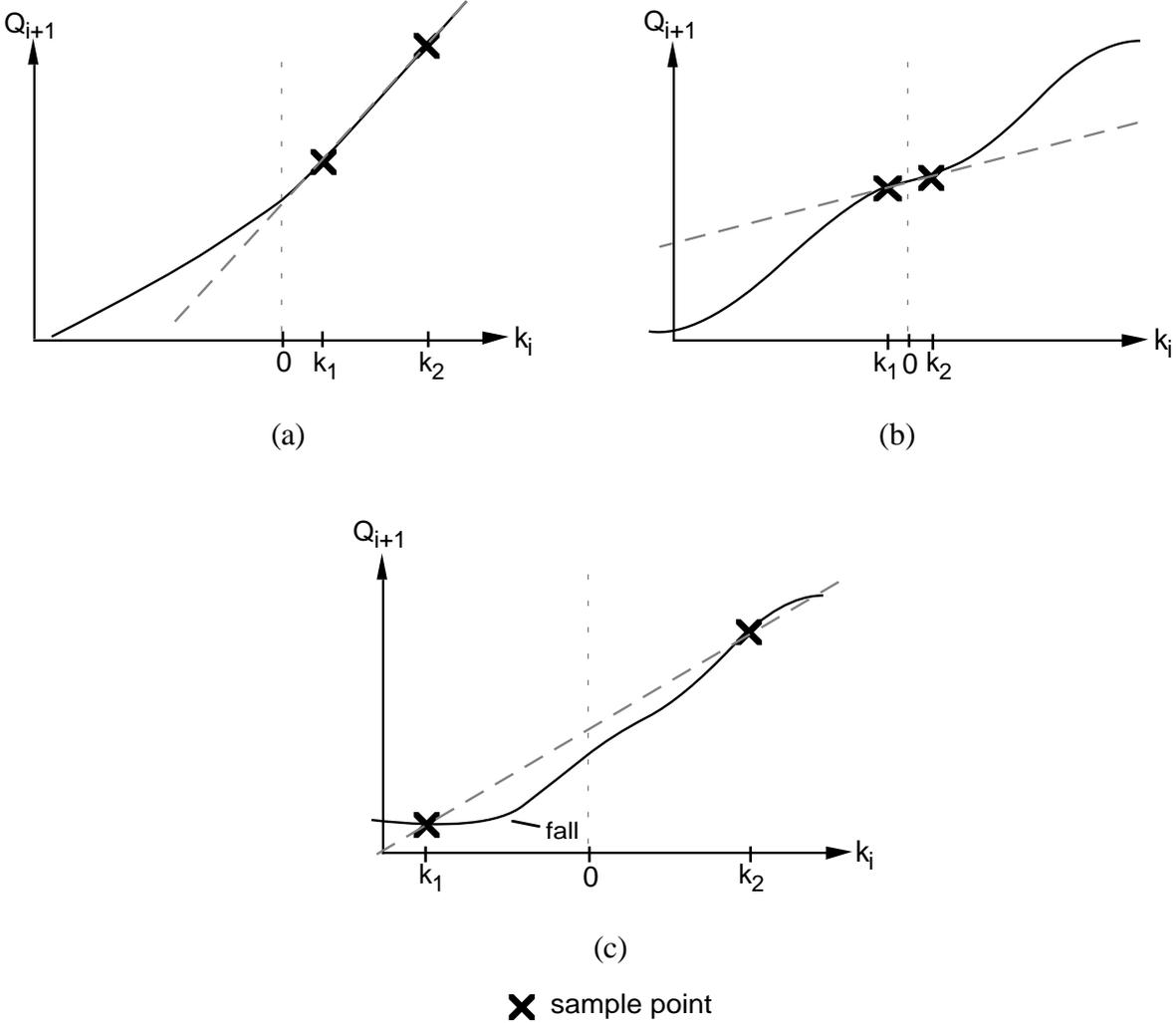


Figure 3.24 - Sampling point pitfalls in discrete model construction  
(a) sample scalings non-symmetric  
(b) sample scalings too small  
(c) sample scalings too large

### 3.8 Torso Servoing

While it is preferable for animation to have general control solutions, more specific control solutions are sometimes useful or necessary. The control described in previous sections can be used to generate successful walking gaits. However, the resulting walks exhibit a characteristic bobbing<sup>2</sup> of the torso in the sagittal plane. This is due, in part, to the choice of simple stance hip

---

<sup>2</sup> rocking back and forth.

perturbations and, in part, to the fact that the motion is only constrained at a single point in the cycle for each step.

One possible solution to this problem is to use the waist pitch degree-of-freedom of our human model to continuously servo the torso to a desired angle from the vertical in the sagittal plane. The torso servoing control is applied to the biped's waist pitch DOF. The applied torque for a planar example is calculated as

$$\tau = k_p \cdot (\phi - \phi_{desired}) - k_d \cdot \dot{\phi}$$

where

$\tau$  is the applied joint torque

$\phi_{desired}$  is the desired torso angle from vertical in the world frame (constant)

$\phi$  is the torso angle with respect to world frame vertical

$\dot{\phi}$  is the time derivative of  $\phi$

and

$k_p$  and  $k_d$  are the proportional and derivative gains

The parameters  $\tau$ ,  $\phi_{desired}$ , and  $\phi$  are shown in Figure 3.25

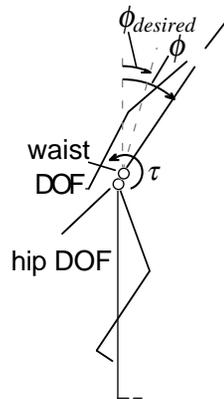


Figure 3.25 - Torso servo parameters

With continuously-applied torso servoing, the torso motion becomes considerably smoother. However, it also presents an additional problem when used with balance control. The application of a torso servo precludes the use of torso-based RVs since these will no longer vary with hip

perturbations. Redefining the up vector using a vector fixed to the pelvis as illustrated in Figure 3.26, rather than the torso, is sufficient to overcome this problem.

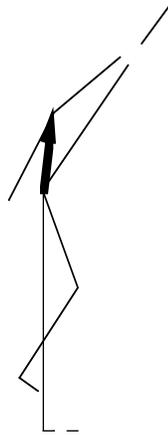


Figure 3.26 - Pelvis-based up vector balance indicator

Torso servoing ensures that the upper body of the biped remains upright, and provides increased stability by damping large upper body movements. However, it should be noted that it does nothing to prevent the biped from falling over since the legs must still support the upper body. An example of a fall with torso servoing applied can be seen in Figure 3.27.

There are a number of benefits in addition to a more aesthetic result. Reducing the highly dynamic oscillations of the torso (a significant portion of the body mass) results in a successful limit cycle for a larger range of  $Q^d$ . As well, longer stride times and very slow walks become possible.

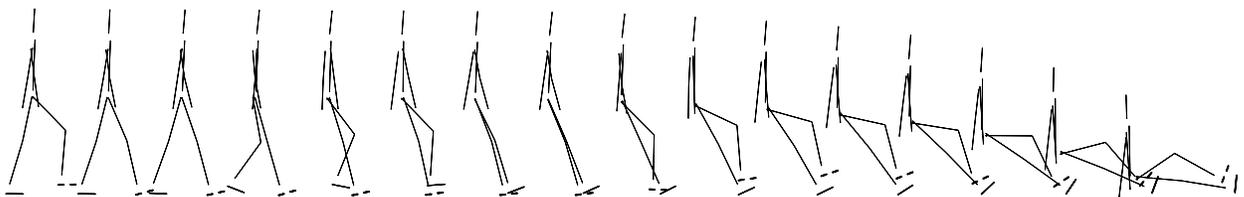


Figure 3.27 - Falling with torso servoing enabled.

### **3.9 Conclusions**

In this chapter, we have described a control system capable of generating and controlling stable 3D bipedal walking limit cycles from unstable open loop motions. A number of key simplifications were made, leading to a well-behaved discrete system which is remodelled for each step using a discrete linear model. The details of each of the system components and their roles in the overall system were described and examples were given for a walking human model.

In the following chapter, we will present some of the results obtained by applying this control strategy to our dynamic models.

## 4. BALANCED WALKING RESULTS

This chapter describes the results of a series of experiments performed using the balance control approach of Chapter 3. Various parameter choices are explored to determine their importance in generating successful limit cycles and their effect on the resulting motion. Control of the human model is the primary focus of these experiments, although trials using the robo-bird model are also included to demonstrate success with a significantly different bipedal model. Sections 4.1, 4.2, and 4.3 present results obtained using each of the three choices of RVs introduced in Section 3.5. Section 4.4 examines the effectiveness of applying a torso servo to the human model.

Figure 4.1 shows a few steps from a typical balanced walk. The motion reaches a limit cycle and continues to walk in a straight line for the full 60 steps of the trial. Using the nominal open-loop control alone results in only 5 steps before the human figure falls over. The results clearly demonstrate that with suitable parameter choices, active stabilization of bipedal locomotion cycles is both possible and useful.

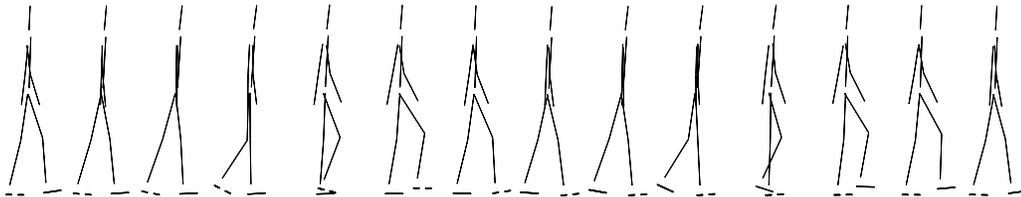


Figure 4.1 - A sequence of steps from a walking limit cycle.  
Up vector-based,  $Q^d = [ .25, 0 ]$ , F-L sampling.

The motion resulting from the application of the described balance control varies significantly, depending on the choice of RVs and other parameter values. Many choices of parameter values

result in a number of successful steps but eventually fall. Other values move the system onto a successful walking limit cycle. Some walks don't fall despite seemingly chaotic motion.

### 4.1 Up Vector Regulation Variables

The first results we show use the base PCG of Figure 3.7 and the stance hip perturbations described in Section 3.6. The RVs consist of the components of the up vector, as illustrated in Figure 3.12 (c). Torso servoing is not applied. Figure 4.2 shows a representative set of RV trajectories corresponding to successful walking trials of 60 steps. These curves are the experimentally-based equivalents of the idealized cyclic trajectories shown earlier, such as those of Figure 3.3. In three of the plots, a clear limit cycle emerges. In the fourth, the trajectory is chaotic.

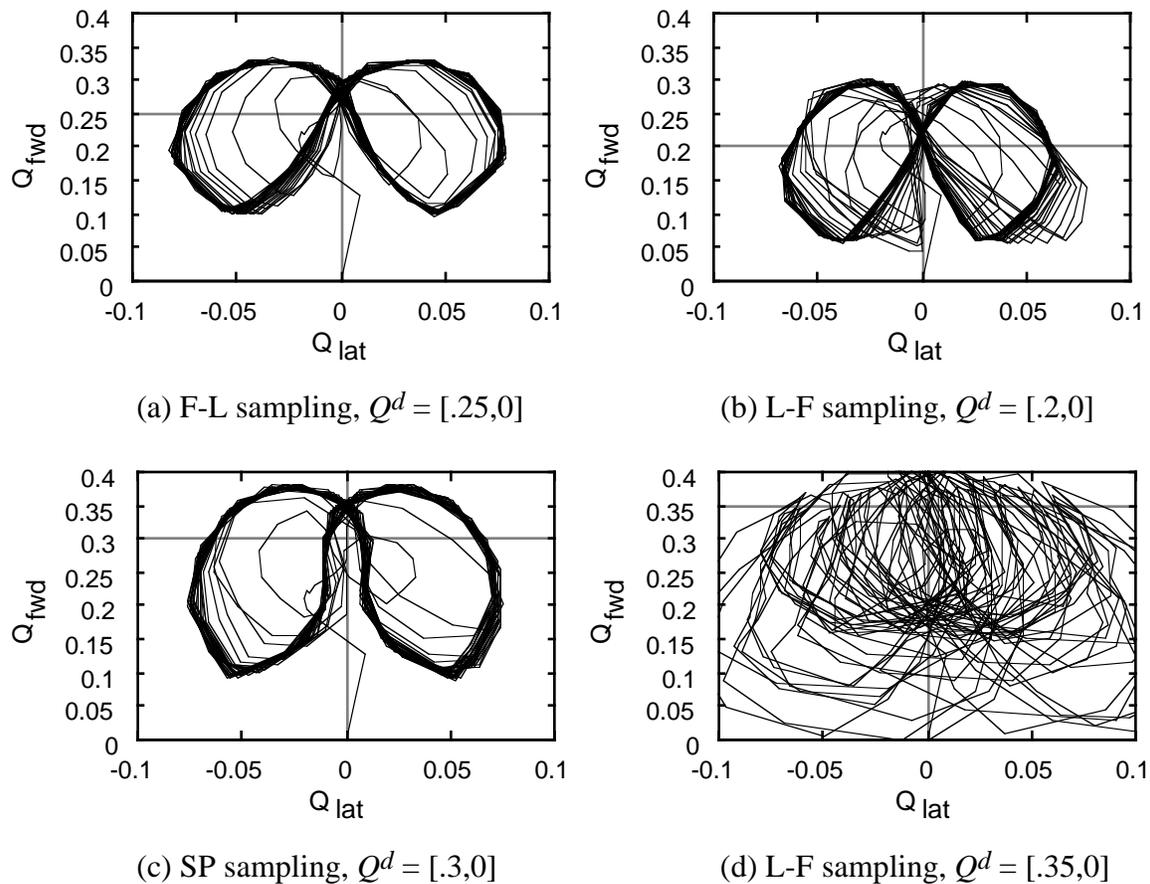
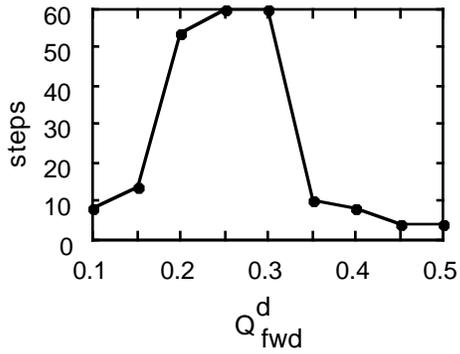


Figure 4.2 - Continuous-time up-vector RV component phase diagrams

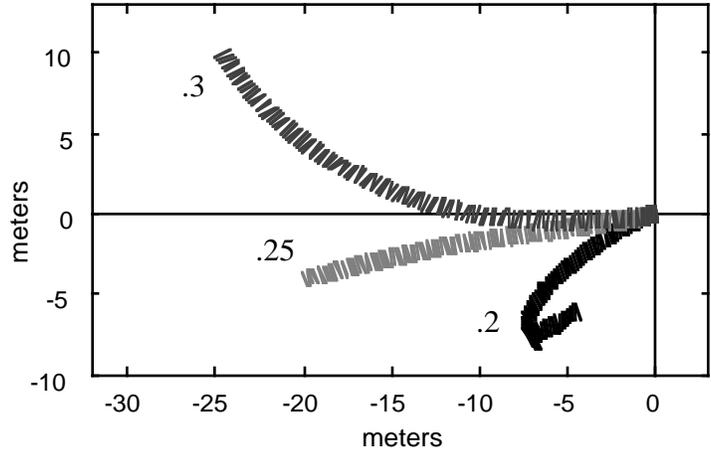
While such phase plots are useful tools for observing limit cycle behaviour in a system, they tell us relatively little about the difference between sampling strategies and/or the visual appearance of the walk.

The graphs shown in Figure 4.3 provide additional information. Graphs (a), (c) and (e) show the effect of various choices for  $Q^d$ , in particular, for the forward component of  $Q^d$ . The trials explore the useful range of this parameter by uniformly sampling a broad set of values. For each trial, the forward component of the  $Q^d$  is held constant for all steps. The lateral component is fixed at 0.0 for all trials. Walks are up to a maximum of 60 steps long and fewer than 60 successful steps indicates a fall.

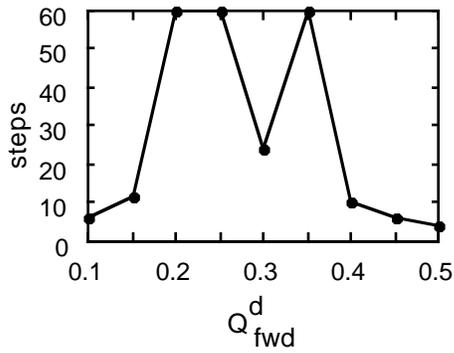
The "*hip plots*" corresponding to the most successful trials for each sampling strategy are shown in Figure 4.3 (b), (d) and (f). Hip plots indicate the position and orientation of the biped's pelvis in the horizontal plane as viewed from above. This orientation information indicates the direction the biped is facing as it walks, allowing a forward motion to be easily distinguished from a lateral (sidestepping) motion. The lines on the plots are approximately 1 meter in length, significantly wider than the biped's actual hips so that their orientation can be easily seen. This affects the perceived scale of the plot, making the walks appear shorter than they actually are. The dimensions of the terrain are given in meters. The elapsed time between adjacent samples on the hip plots is approximately 0.16 seconds.



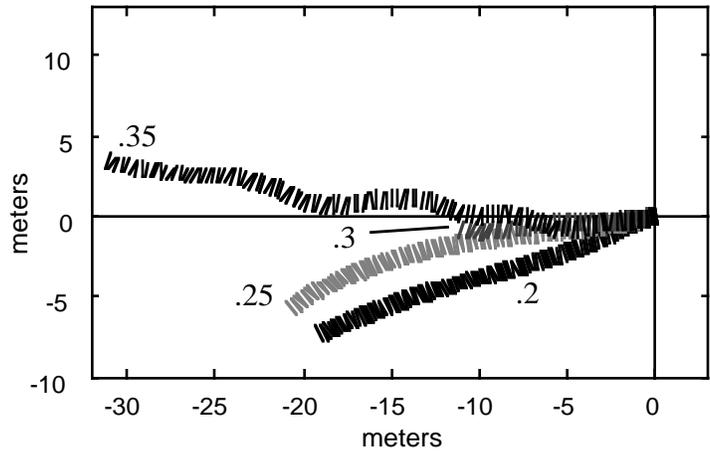
(a)



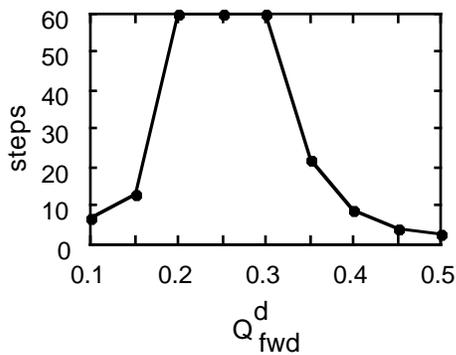
(b)



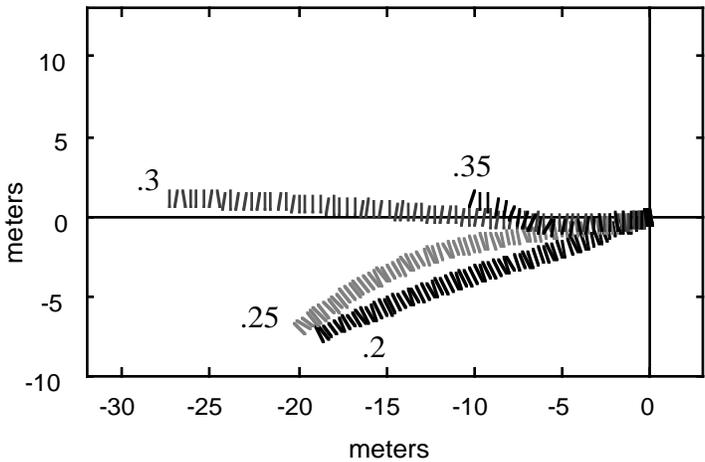
(c)



(d)



(e)



(f)

Figure 4.3 - Nominal up-vector  $Q^d$  range and corresponding hip plots for each sampling method. All trials are limited to a maximum of 60 steps. The biped initially faces left (i.e. in the  $-x$  direction).

- (a) - (b) F-L sampling
- (c) - (d) L-F sampling
- (e) - (f) SP sampling

A number of observations can be made from these graphs and the corresponding animations:

1. Not all choices of  $Q^d$  work. Successful trials fall within a "nominal range" of  $Q^d$  which is similar across sampling strategies.
2. Choice of sampling strategy noticeably affects the appearance of the walks. Somewhat surprisingly, no one strategy is demonstrably superior.
3. Walks are not of uniform quality. Some walks initially appear stable but then fall over. Other walks appear chaotic but don't fall. Choices of  $Q^d$  toward the centre of the nominal ranges give the most consistent walks in terms of stability and appearance.
4. One walk begins moving forward, slows to a stop and then moves backwards chaotically for about 15 steps before finally falling. The fact that such a walk emerges from a base PCG designed to produce forward walking motions illustrates the potential for a broad range of useful motions with little or no change in the underlying balance mechanism.

Figure 4.4 through Figure 4.6 show the end-of-cycle components of the reduced state vector,  $Q_i^*$ , over time, for the walks of Figure 4.3 (b), (d) and (f). The sampled points are joined by lines to better visualize oscillations. The lines do not represent the continuous motion of the RV during each stride.  $Q^d$  is indicated by a dotted line.

A number of qualitative observations can be made about the behaviour of  $Q_i^*$ :

1. A startup phase is clearly evident. The number of steps required to reach a steady-state response varies from 5 to over 20, with 10 being most typical. Trials using values of  $Q^d$  in the middle of nominal range stabilize most quickly.
2. The desired value,  $Q^d$ , is not always achieved, but a limit cycle is still attained in many such cases.
3. No particular sampling strategy appears superior with respect to steady state error or rate of stabilization.

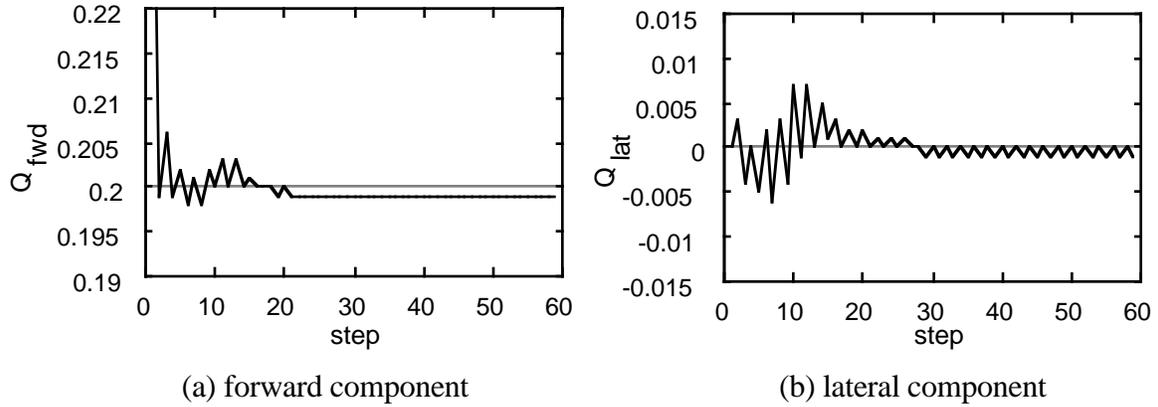


Figure 4.4 - Discrete RV values,  $Q_i^*$ , for L-F sampling,  $Q^d = [.2,0]$

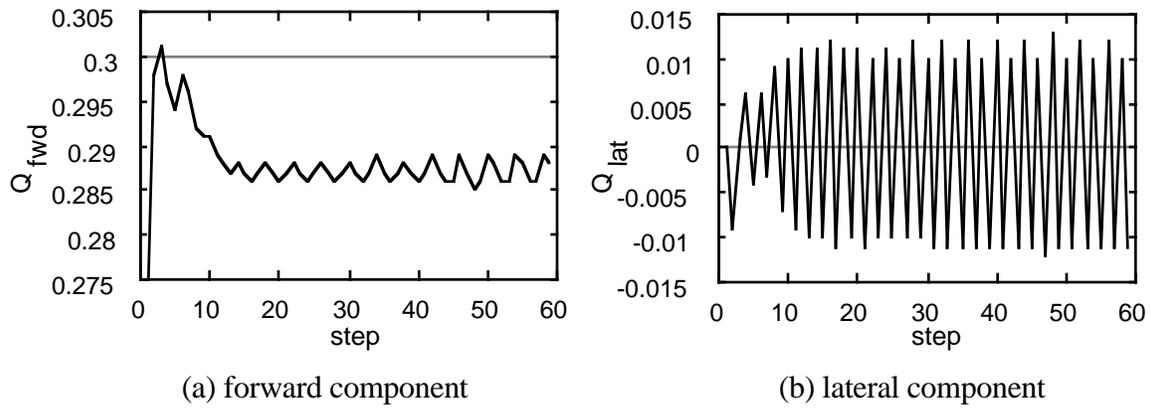


Figure 4.5 - Discrete RV values,  $Q_i^*$ , for SP sampling,  $Q^d = [.3,0]$

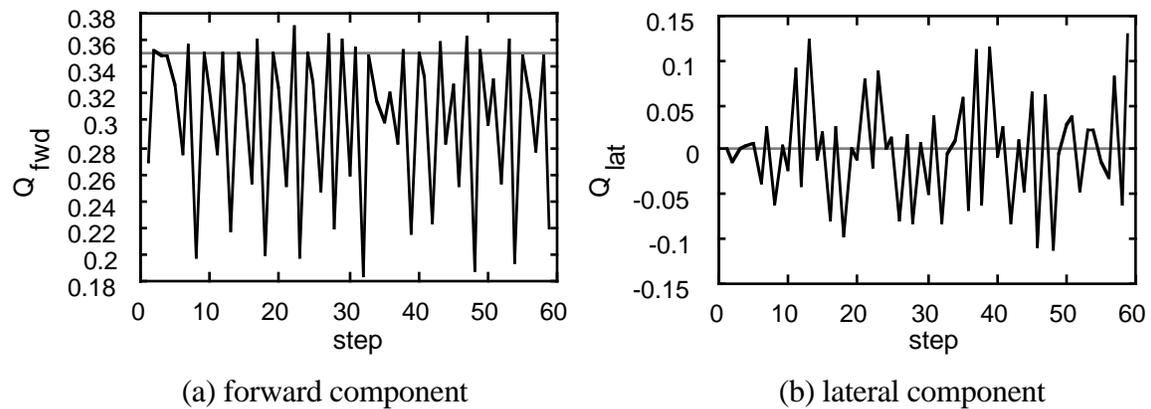


Figure 4.6 - Discrete RV values,  $Q_i^*$ , for L-F sampling,  $Q^d = [.35,0]$

4. The chaotic walk (see Figure 4.6) has significantly larger error with respect to the desired RV values,  $Q^d$ , as compared to stable limit cycles (note the scale of the graph). The visual appearance of this walk is noticeably more clumsy than the other walks.
5. Generally speaking, the forward component of  $Q$  stabilizes more quickly than the lateral component and exhibits smaller variation in steady-state (e.g. Figure 4.4).
6.  $Q_i^*$  can oscillate from step to step. This is noteworthy since the overall motion should ideally have nearly identical characteristics from one step to the next.

#### 4.1.1 Other Observations

The walks generated using the up-vector display a number of notable characteristics. First, while step length is not a regulated variable, virtually all of the walks travel a uniform distance from one step to the next. Figure 4.7 shows the step lengths over the course of a typical walk. This result is encouraging since an irregular walk would be much less appealing and a suitable remedy is not immediately obvious. This also serves as evidence that the unobservable state variables approach a limit cycle as desired.

Another characteristic of the resulting motions is that not all of the walks follow a straight line. This can be seen clearly in the hip plots of Figure 4.3. Without any form of directional control, most of the walks follow a curved path. The chaotic walk corresponding to the  $Q^d = [.35,0]$  trial of Figure 4.3 (d) follows a less regular path, weaving back and forth over the course of the trial. As we shall see in the next chapter, this can be solved by explicitly controlling the biped's direction with an additional feedback loop.

An adverse effect apparent in some of the walks is a tendency to place each stance foot directly in line with the previous in the lateral dimension as if walking a tightrope. In a few cases, this even results in the biped crossing legs slightly each step, with one leg passing through the other since such interpenetrations are not prohibited in our simulations. While the *tightrope walking* is

undesirable, it is also an artifact of using the up vector RV in the lateral dimension. The lateral hip perturbations which result in this effect are chosen to maintain lateral torso uprightness. Using a different choice of RV for the lateral dimension can correct this effect, as will be demonstrated in Chapter 5.

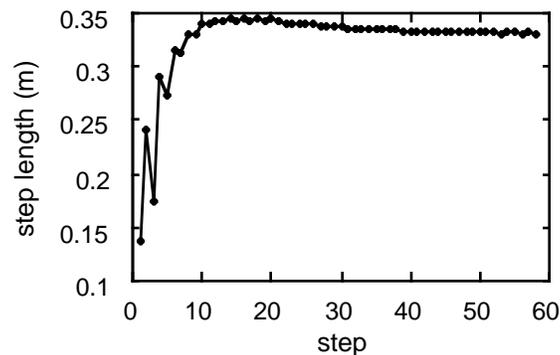


Figure 4.7 - Step length vs step number.  
 $Q^d = [0.25, 0]$  walk of Figure 4.3 (b)

One final characteristic of the walks is excessive front-to-back and side-to-side motion of the torso. This body motion is a result of the particular choice of control perturbations. This effect and a suitable solution will be discussed in Section 4.4.

## 4.2 Swing-COM Regulation Variables

Our balance control technique is also effective with the use of swing-COM RVs. The fact that significantly different choices of RV can be used successfully with otherwise identical control serves to illustrate that the proposed control approach is reasonably general.

Figure 4.8 shows a representative set of RV trajectory plots for the swing-COM vector trials. The plots indicate the trajectories of both feet through the whole cycle since the swing leg, and hence the swing-COM RV, changes legs each step. In contrast with the similar up-vector trajectories of Figure 4.2, the swing-COM-based RV trajectories do not exhibit "strong" limit cycle behaviour,

but rather tends to drift around in near-cycles and/or converge very slowly onto a limit cycle if at all.

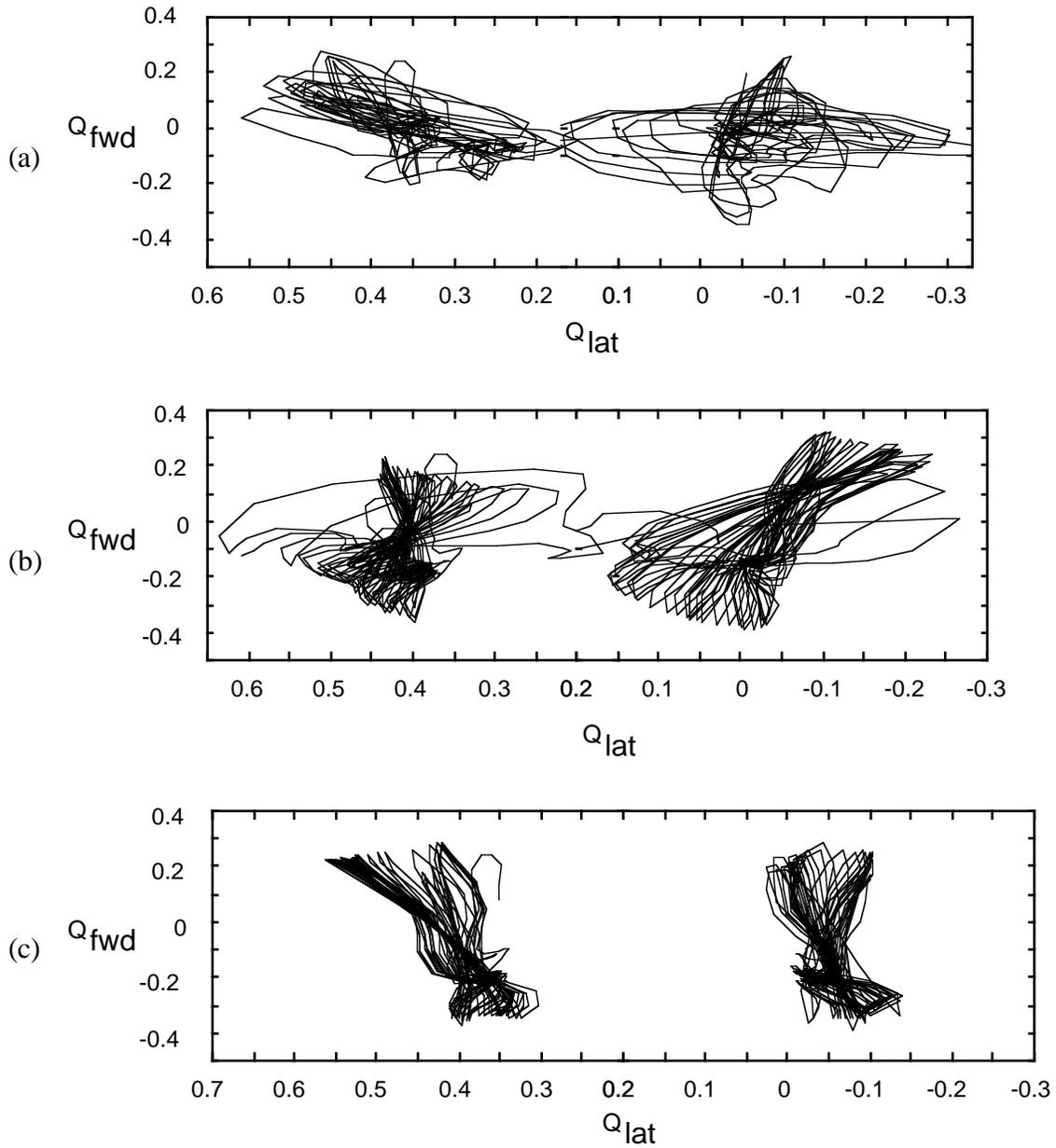


Figure 4.8 Continuous-time swing-COM RV phase diagrams. The two curves represent the left and right feet respectively (i.e. swing-COM and stance-COM at various times in the cycle).

(a) F-L-sampling,  $Q^d = [-0.05, 0]$

(b) L-F-sampling,  $Q^d = [0, 0]$

(c) SP-sampling,  $Q^d = [0, 0]$

Figure 4.9 shows the nominal range plots and hip plots for a series of trials using swing-COM RVs and stance hip control perturbations with each of the three sampling strategies (F-L, L-F and SP). The trials explore the useful range of the forward component of  $Q^d$ , with uniform sampling. The lateral component of  $Q^d$  is fixed at 0.0 for all trials. Torso servoing is not applied. All trials are a maximum of 60 steps long.

A few of observations can be made based on Figure 4.9 and on the corresponding animated motion:

1. The swing-COM RVs generate walking motions generally similar in nature to those generated using up-vector RVs. As with the up-vector results, there is a “nominal range” of  $Q^d$  values which produce the most successful walks.
2. In contrast to the up vector case, different sampling strategies give significantly different results. SP sampling generates the most stable motions and F-L sampling the least stable.
3. Unlike the up vector walks, which generally move straight forward, many swing-COM walks demonstrate a tendency of the biped to twist sideways about its own vertical axis. This tendency is most prevalent when using the F-L sampling strategy and is almost entirely absent for the SP sampling strategy.

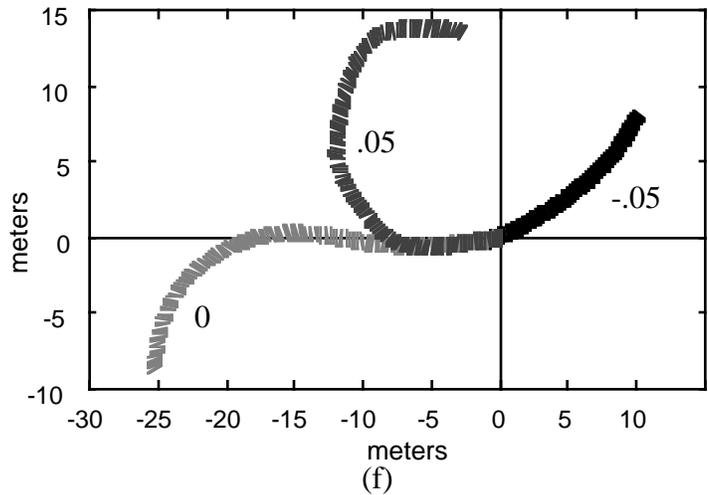
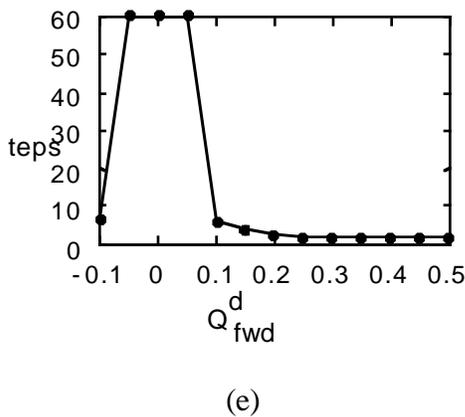
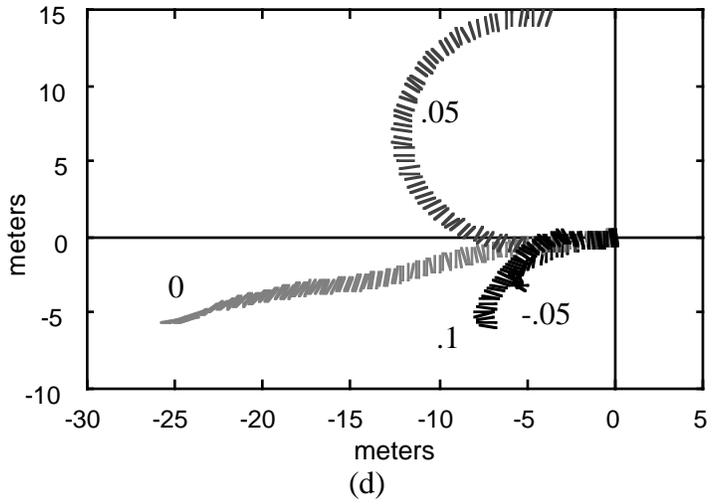
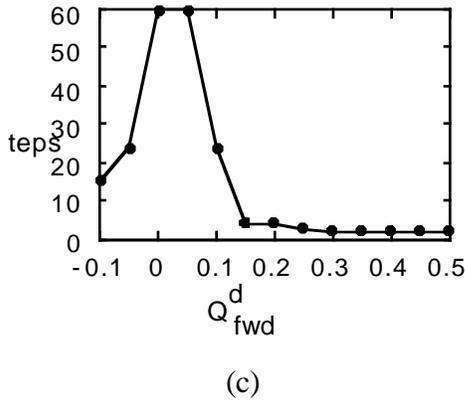
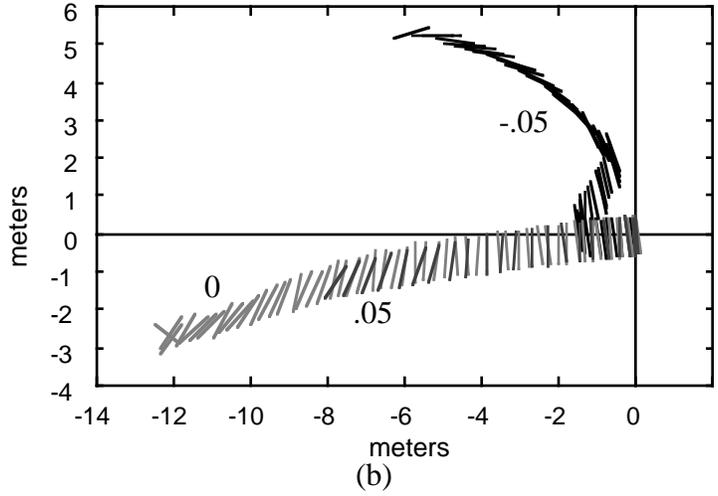
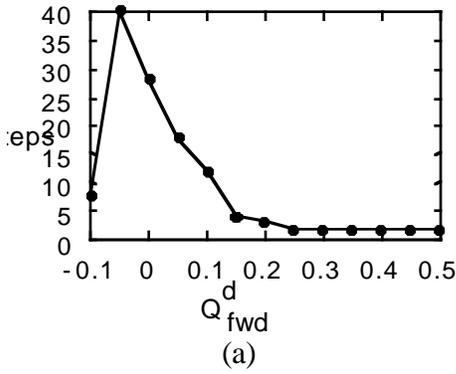
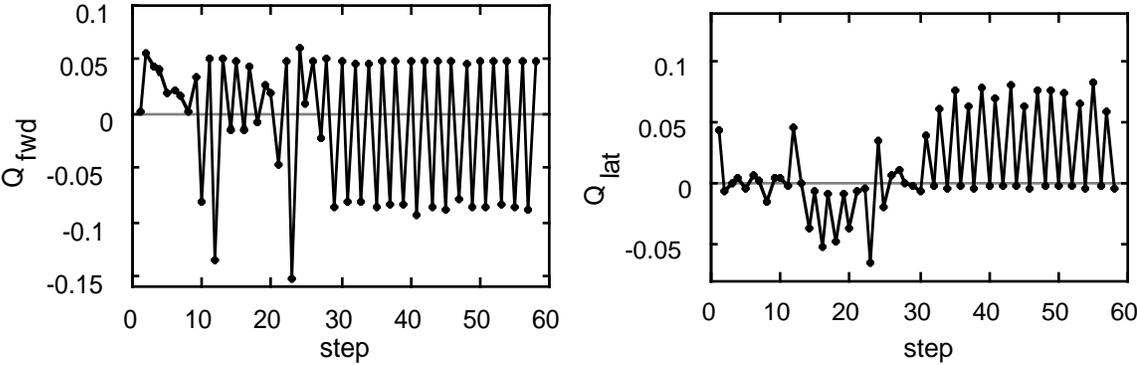


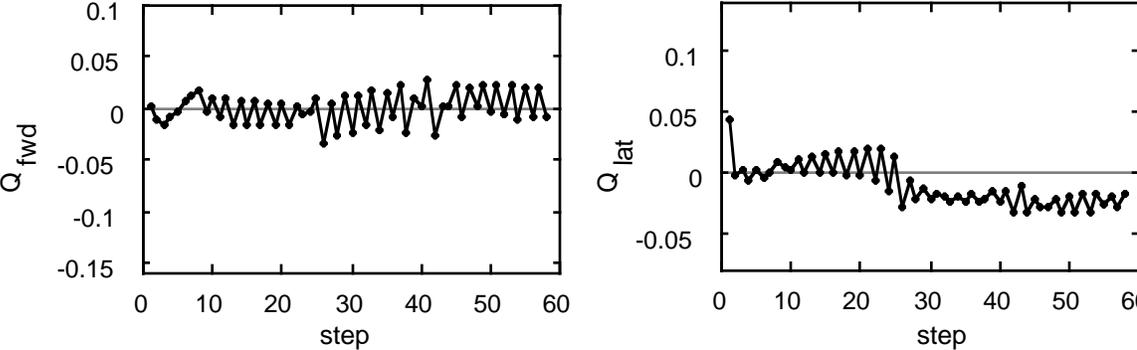
Figure 4.9 - Nominal swing-COM  $Q^d$  range and corresponding hip plots for each sampling method. Lateral  $Q^d = 0.0$  for all trials. The biped initially faces left (i.e. in the  $-x$  direction).  
 (a) - (b) F-L sampling  
 (c) - (d) L-F sampling  
 (e) - (f) SP sampling

Figure 4.10 and Figure 4.11 show discrete RV value plots typical of the results obtained using swing-COM RVs.  $Q^d$  is indicated by a dotted line.



(a) forward component (b) lateral component

Figure 4.10 - Discrete RV values,  $Q_i^*$ , for L-F sampling,  $Q^d = [0.05,0]$



(a) forward component (b) lateral component

Figure 4.11 - Discrete RV values,  $Q_i^*$ , for SP sampling,  $Q^d = [0.0,0]$

Overall, the results are not as good as those obtained with the up-vector RV. The swing-COM  $Q_i^*$  values are, in general, less stable, fall farther from their desired values with larger, more chaotic oscillations, and demonstrate a tendency to drift. Correspondingly, the continuous motion of the biped in most swing-COM trials is farther from a true limit cycle, qualitatively speaking.

Two particular problems are apparent when viewing the generated motion:

1. All three sampling strategies generate walks with step lengths which vary widely from one step to the next as shown in Figure 4.12. This contrasts sharply with the uniform step lengths generated when using up vector RVs.
2. "Tightrope" walking, similar to that present in some up vector-based walks, is also prevalent in the swing-COM trials. The use of swing-COM RVs, however, makes a reasonable solution possible by choosing  $Q^d$  with this problem in mind.

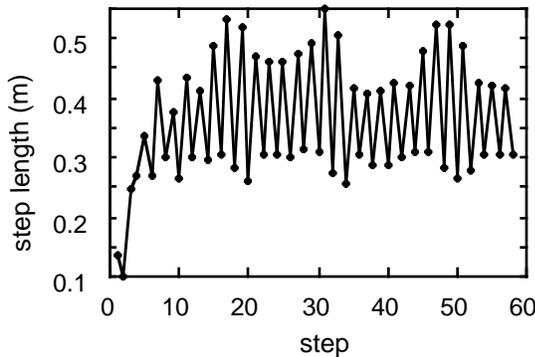


Figure 4.12 - Step length vs step number, SP sampling,  $Q^d = [0.05,0]$

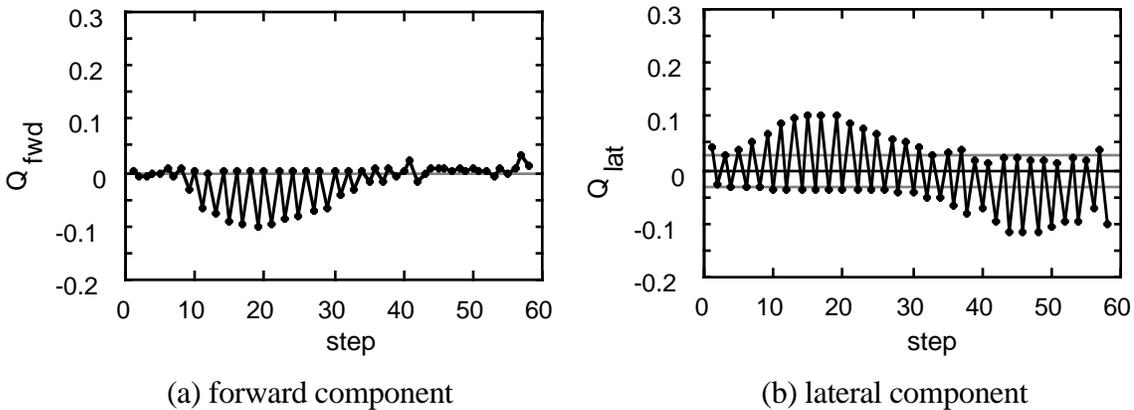


Figure 4.13 - Discrete RV values for SP sampling,  $Q^d = [0.0, \pm 0.03]$ . Compare with Figure 4.11. Larger lateral step-to-step variation means less tightrope walking.

The first problem will be addressed in the next chapter. To solve the second problem, the lateral component of  $Q^d$  is chosen to give the swing foot a displacement slightly away from the body and

the COM in the lateral dimension. Figure 4.13 shows the resulting  $Q_i^*$  for such an attempt based on the walk of Figure 4.11, which exhibits tightrope walking. A lateral  $Q^d$  component of  $\pm 0.03$  is used instead of 0.0. The sign of  $Q^d$  is such that foot placement target is away from the COM. The resulting walk does not fall and exhibits much more realistic lateral foot placement.

### 4.3 Stance-COM Regulation Variables

While using the up-vector and swing-COM RVs achieves reasonable success in generating walking motions, similar use of the stance-COM RVs does not. Most trials fall during the second controlled step. The failures are due to the fact that even modest changes in the stance-COM RVs require very large stance hip perturbations which cause very large changes in the biped's state. Figure 4.14 illustrates this idea. Because of this result, the chosen balance parameters fail to reach a suitable state from which to begin the next cycle, even though the desired RV value might be attained.

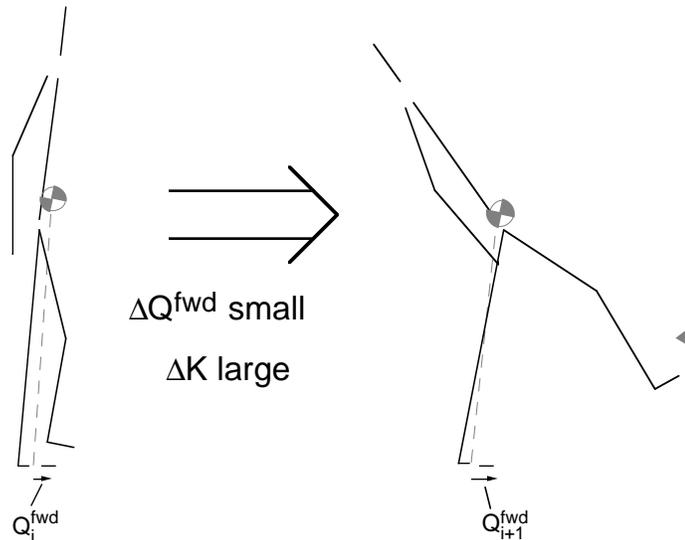


Figure 4.14 - Variation of stance-COM RV with stance hip pitch ( $\Delta K$ )

While the walks all fall quickly, two interesting observations can be made:

1. The first controlled step reaches  $Q^d$  with a high degree of accuracy, i.e. the discrete system model of the first step is accurate.

2. The above holds for final perturbation scalings,  $K^*$ , far from the sample point scalings used to construct the linear discrete system model. That is, the model applies over a large change in state. Table 4.1 shows this for a number of desired values of  $Q^d$ . At one extreme, the model is still accurate for  $K^*$  as high as 13 times the sample scalings used to construct the model.

Taken together, these suggest the possibility of controlling non-trivial aperiodic motions in a similar way to the periodic motions of this thesis. Such motions would typically require a much larger controllable range of  $Q$  than cyclic motions since they typically involve a larger change from initial state to final state. An example of such an aperiodic motion would be standing up from sitting in a chair.

$Q_{fwd}^d$	$Q_{lat}^d$	$K_{fwd}^*$	$K_{lat}^*$	$Q_{fwd}^*$	$Q_{lat}^*$
.1	0	-36.318	+5.544	0.072	+0.007
.15	0	-10.918	+7.823	0.154	+0.030
.2	0	+14.481	+14.603	0.208	-0.001
.25	0	+39.881	-45.124	0.254	-0.015
.3	0	+65.280	-9.943	0.335	-0.003

Table 4.1 - Results of first controlled step using stance-COM RVs.

$K_{fwd}^*$  and  $K_{lat}^*$  are final stance hip perturbation scalings in degrees. Sample scalings for construction of linear models are  $[K_{fwd}, K_{lat}] = [\pm 5, \pm 1]$  (compare to  $K_{fwd}^*$  and  $K_{lat}^*$  in table).

Finally, because the particular choice of perturbation is the primary cause of the poor stability results with the stance-COM, it is quite possible that a better choice of perturbations might give better results. Using a stance ankle pitch perturbation to vary the force with which the stance foot pushes off the ground is one possible example.

## 4.4 Torso Servoing

All of the walks presented thus far suffer from an unaesthetic "bobbing" of the torso. This can be readily observed in the continuous-time RV phase diagrams such as those in Figure 4.2. Excursions in torso pitch of around 12 degrees are typical and they are quite apparent when watching animations of the balanced walks. While these motions may in fact be desirable in some cases, they tend to adversely affect the perceived quality of the motion when a realistic, natural looking walk is the eventual goal.

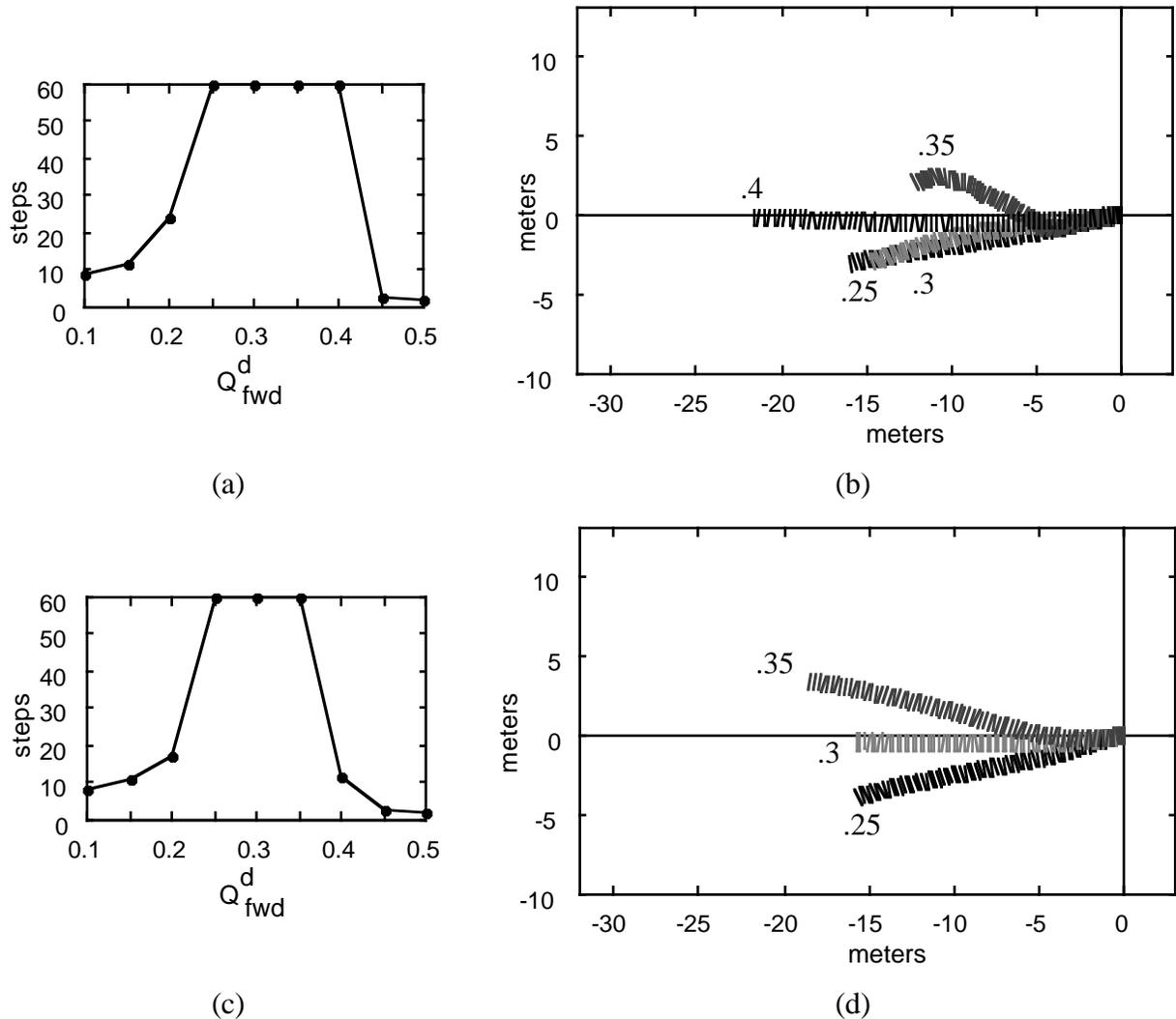


Figure 4.15 - Trial results based on the walks of Figure 4.3 with the addition of torso servoing to 5 degrees from world vertical. The biped initially faces left (i.e. in the -x direction).

(a) - (b) F-L sampling  
(c) - (d) L-F sampling

Torso servoing significantly reduces the magnitude of the excursions in the forward dimension and, as a side effect, can also reduce the excursions in the lateral dimension. Figure 4.15 shows the results for a set of trials which use the same balance control parameters as the walks of Figure 4.3. In this set of trials, torso servoing is applied using a desired angle of 5 degrees forward from vertical.

The resulting paths are straighter and generally more consistent in direction than the non-torso servoed results. The useful range of  $Q^d$  increases in general. The phase diagram in Figure 4.16 illustrates the effectiveness of torso servoing at reducing the bobbing effect caused by the particular stance hip perturbation used. In this trial, torso servoing reduces the range of torso pitch from approximately 12 degrees to 3 degrees.

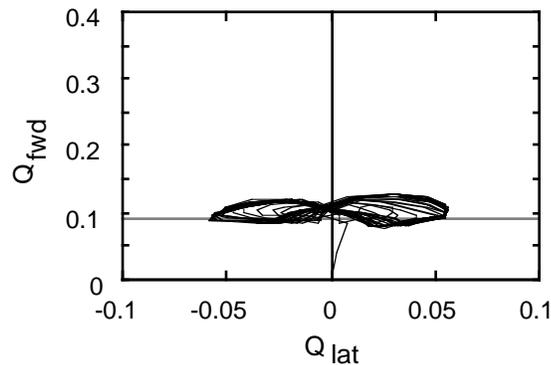


Figure 4.16 - Continuous-time up vector component phase diagram for human model with torso servoing to +5 degrees from world vertical ( $Q_{servo} \approx .09$ ). Compare to Figure 4.2.

## 4.5 Robo-bird Running

Figure 4.17 shows the base PCG used to generate a running motion for the robo-bird model, shown in Figure 4.18. While the base PCG differs from that used for the human model, it is balanced using up-vector RVs and stance hip pitch and roll perturbations as with the human model.

DOF:		1	2:0	2:1	3	4	5	6:0	6:1	7:0	7:1	8	9	10	11:0	11:1	transition info
state	S1	0	0	0	80	-95	45	-45	0	0	15	55	-65	25	-25	0	0.1
	S2	0	0	0	80	-95	45	-45	0	0	15	55	-65	25	-25	0	0.1
	S3	0	0	15	55	-65	25	-25	0	0	0	80	-95	45	-45	0	0.1
	S4	0	0	15	55	-65	25	-25	0	0	0	80	-95	45	-45	0	0.1

Figure 4.17 - Robo-bird base PCG for running

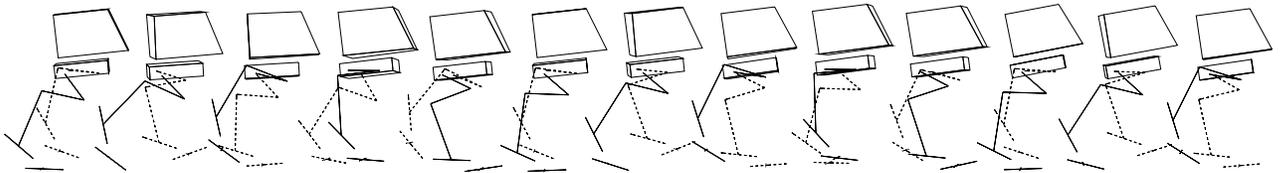


Figure 4.18 - A running robo-bird

While both the robo-bird and human models are bipeds, their dynamics are different. When standing and walking, the human model is supported by the rigidity of its straight stance leg, which typically has little or no freedom in the vertical direction. In theory, no actuator forces are required to prevent the human model from collapsing. The leg structure of the robo-bird model is such that actuator forces must constantly be applied even to remain standing upright. This means there is a significant amount of vertical compliance when moving, leading to “bouncy” motions. As well, the base of the robo-bird model is wide enough to make standing on one foot for any length of time difficult. These factors make a dynamic running motion more natural than a walk for this model.

## 4.6 Conclusions

In this chapter we have presented evidence that the discrete limit cycle control technique introduced in Chapter 3 can be used to generate balanced walking motions for the human model and a running motion for the robo-bird model. Two choices of RV, based on the up-vector and the swing-COM vector, were successfully applied to this end. A torso servo was shown to be useful in improving the aesthetics of the human model’s motion. Finally, a running motion for

the robo-bird motion was used to demonstrate the applicability of the control technique to a different motion and model. In the next chapter, we will present a number of interesting gait variations, all of which make use of the same basic form of balance control demonstrated here.

## 5. WALKING VARIATIONS

The ability to control walking characteristics such as speed, direction, stride rate and posture is important if a system is to be seriously considered as an animation tool. This chapter first introduces an approach to controlling the speed of a walk without requiring changes to the underlying base PCG. Next, the idea of using parameterized base PCGs is introduced and a number of interesting variations on the basic walk are described.

The idea of parameterizing an open-loop control structure such as PCGs is not entirely new [vKF94b]. However, the successful parameterization of dynamically unstable motions such as walking does represent an important advance.

### 5.1 Speed Control

The basic balance control discussed in Chapter 3 uses fixed values of  $Q^d$  for all steps of a walk and achieves constant walking speeds. One possible way to vary the speed of a walk is to provide a number of base PCGs which yield various speeds and interpolate between these. This approach might be somewhat cumbersome, however, since it requires the animator to provide suitable base PCGs. An alternative is to allow  $Q^d$  to vary from one step to the next instead of remaining fixed. By varying  $Q^d$  to cause the biped to speed up or slow down each step, a wide range of walking speeds is possible without requiring modifications to the base PCG. This approach requires that we be able to affect the speed of the walk in a consistent manner by varying  $Q^d$ . Whether or not this is possible depends on the choice of control system parameters (RVs, LPPs, etc.). With the choices described in Chapter 3, an intuitive form of control is available. In qualitative terms, an increase in speed can be achieved by using a more forward-leaning value for  $Q^d$  and a decrease in speed can be achieved using a more upright value for  $Q^d$ .

More formally, to control walking speed,  $\tilde{Q}^d$ , a new value for  $Q^d$  is computed each step by adding a velocity feedback term to a nominal value of  $Q^d$  as follows:

$$\tilde{Q}^d = Q_{bias}^d + k_p \cdot (v_d - v) \quad (5.1)$$

where

- $Q_{bias}^d$  is a nominal bias value
- $v_d$  is a desired speed
- $v$  is a measure of the speed of the current step
- $k_p$  is a proportional feedback gain

The bias value is required because the required  $Q^d$  for a stationary walk (i.e. in-place) is not necessarily 0. The feedback term adjusts  $\tilde{Q}^d$  in the appropriate direction (more forward or more upright) by an amount proportional to the error in speed. We choose to use the distance travelled by the pelvis over the current step as a velocity measure for  $v$  and  $v_d$ .

The speed of the walk can be varied using either  $v_d$  or  $Q_{bias}^d$ , since both contribute a constant term to the overall equation. Fixing  $v_d$  and varying  $Q_{bias}^d$  is akin to choosing a nominal speed and varying the actual speed about that point by leaning more forward or more upright.

To prevent excessively large accelerations that could cause a fall, the feedback term is limited in magnitude:

$$|k_p \Delta v_d| \leq \Delta Q_{max},$$

where  $\Delta Q_{max}$  is the maximum allowable change in  $\tilde{Q}^d$  from  $Q_{bias}^d$

$Q_{bias}^d$  can be taken from any successful walk. In theory, we could use the  $Q^d$  value which gives a stationary walk to reach any desired speed. In practice, it may be necessary to choose a few nominal bias values near the various desired speeds to achieve the broadest possible range of control. Other parameters require some trial-and-error.  $k_p$  must be chosen to relate a required change in speed to a suitable change in  $Q^d$ . If  $k_p$  is too large, the system will be too sensitive to small errors in speed and fall easily. If  $k_p$  is too small, the system may not respond quickly enough to prevent a fall. The size of the range of constant  $Q^d$  values which produce successful (constant speed) walks for the given base PCG is typically a reasonable value for  $\Delta Q_{max}$ .

Figure 5.1 shows the nominal RV range and hip plots of a series of trials using speed control with balance parameters based on Figure 4.15. The trials of Figure 5.1 use a composite choice of RVs, consisting of a forward up vector RV component and a lateral swing-COM RV component. This is done to avoid the tightrope walking and uneven step length associated with each, respectively. The hold times for each time-based transition of the base PCG have been increased from 0.2 to 0.25 to yield a natural stride rate of 1.0 steps/second. Speed control parameters for each control dimension are shown in Table 5.2. A nominal desired speed of 0.4 meters per step or 0.8 m/s is chosen.  $Q_{bias}^d$  is varied to generate various speeds around this operating point.

Table 5.2 - Speed control parameters

	$K_p$	$\Delta Q_{max}$	$Q_{bias}^d$	$v_d$
fwd	1.5	0.15	0.1 - 0.5	0.4
lat	0.5	0.05	$\pm 0.03$	0.0

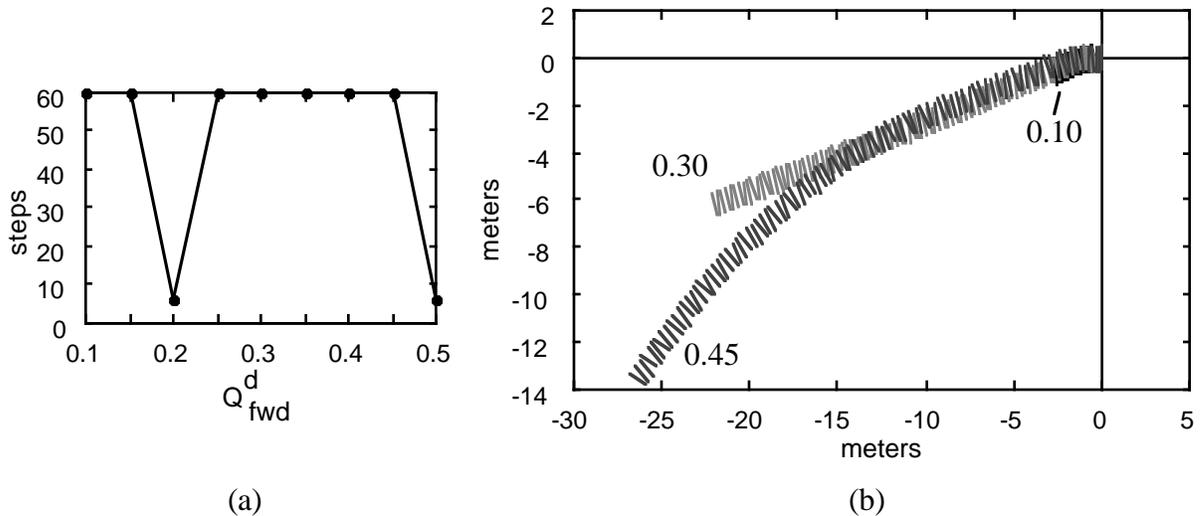


Figure 5.1 - Speed control using composite RV and stride time of 1.0 seconds. Only select  $Q^d$  values are shown on hip plot. Other paths are similar in direction with varying lengths. The biped initially faces left (i.e. in the -x direction).

The resulting walks exhibit a number of desirable characteristics:

1. Applying speed control results in a wider nominal range for  $Q^d$ .
2. The use of a composite RV and the desired lateral speed of 0 are both effective in eliminating a number of the artifacts in earlier walks. The walks generally follow a straight line for a large range of  $Q^d$ , with consistent step length from one step to another and no tightrope walking or leg crossing.
3. A wide range of speeds is possible, with linear control of the steady-state speed about the chosen nominal speed. All walks begin from a stop and reach steady-state speed within around 8 steps.
4. A very slow walk, almost in place, is possible ( $Q^d_{bias} = 0.1$ ). Note that this walk is outside the linear range of control.
5. Acceleration and deceleration is possible as shown in Figure 5.4.

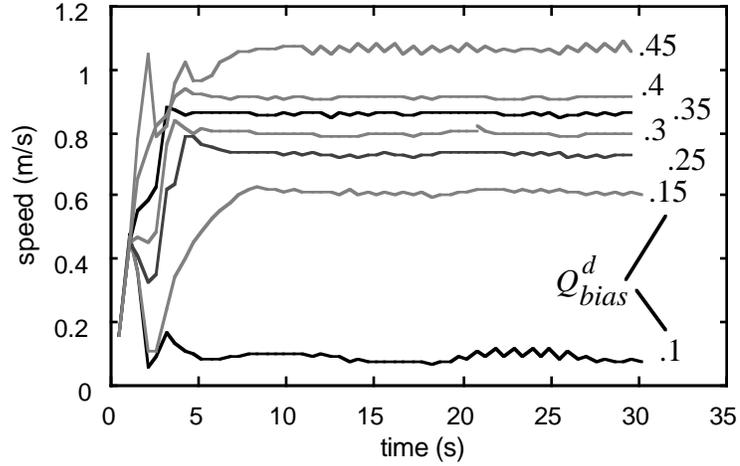


Figure 5.2 - Forward velocity (m/s) versus time for velocity feedback using various base RV target values.

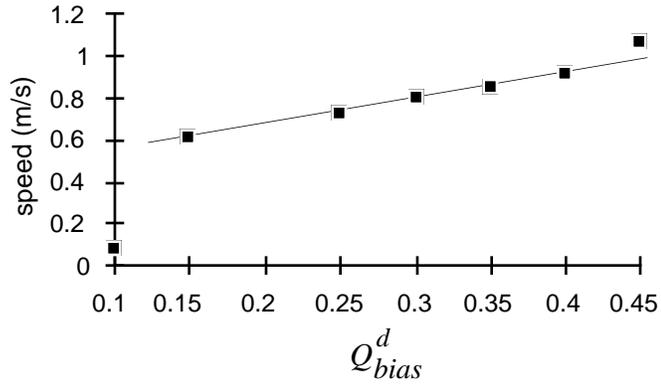


Figure 5.3 - Steady-state velocity vs  $Q_{bias}^d$

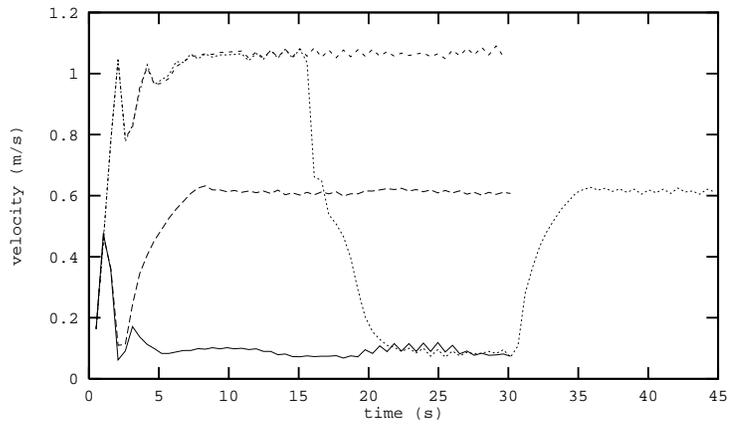


Figure 5.4 - Acceleration and deceleration

## 5.2 Base PCG Parameterization

By applying the balance control of Chapter 4 to various base PCGs, it is possible to achieve controlled variations on basic walking motions such as balanced turns, bent over walks, and different stride rates. A more powerful feature of these gait variations is that they can sometimes be predictably interpolated. That is, given two different styles of walking, based on PCGs  $B1$  and  $B2$ , it is possible to interpolate the control in order to obtain a motion that qualitatively interpolates the respective motions.

More precisely, the interpolation procedure can be described as follows:

Define  $\beta$  as the balance control process which operates on a base PCG to balance it as described in Chapter 3.

Given  $B1$  and  $B2$ , the parameter sets defining two base PCGs, and given that  $\beta(B1)$  and  $\beta(B2)$  are both successful (i.e. that they don't fall), we often find that the control interpolation

$$\beta(\alpha B1 + (1 - \alpha)B2)$$

varies smoothly with  $\alpha$  and is qualitatively similar to what we might imagine a physically correct motion interpolation to be, namely

$$\alpha\beta(B1) + (1 - \alpha)\beta(B2).$$

There is no guarantee that any such successful interpolation exists. In practice, however, smoothly interpolating parameterizations can be found often enough to make them useful.

Such parameterizations can equivalently be thought of as a *linear parametric perturbation* which modifies the open-loop behaviour of a base PCG. The perturbations are identical in form to the balance control LPPs of Chapter 3, but differ in function. Instead of providing step-to-step variations for balance, the perturbations presented in this chapter are applied over a number of steps and are used to vary the overall behaviour of a balanced base PCG in order to achieve a different motion from the original. By recasting the interpolation described above as an LPP, it can then be applied directly to other base PCGs to hopefully effect a similar qualitative change in the motion. For example, if  $\beta(B1)$  walks straight forward and  $\beta(B2)$  turns to the right, then

$$\Delta P = B2 - B1$$

might be called a turning LPP and could be applied to various other basic gaits to allow them to turn.

This is the interpretation we shall build on in the rest of this chapter.

### 5.3 Turning Perturbations

One important parameterized perturbation can be used to achieve turning motions. When applied to the base PCG of Section 3.4 and subsequently controlled for balance, this perturbation provides a means to smoothly vary the curvature of the biped's path. Furthermore, by applying a simple proportional control law to determine a suitable turning rate, the biped can be made to follow an arbitrary direction and thus the path of the biped can be controlled.

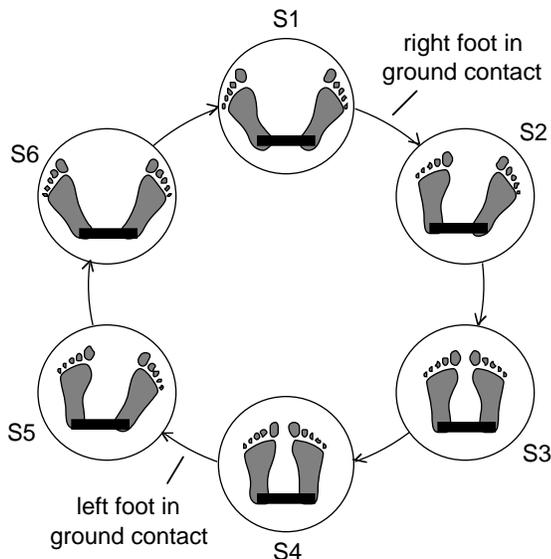
Figure 5.5 (a) and (c) show the PCG perturbation for a right turn for the human model. The perturbation affects only the left and right hip yaw DOFs (3:2 and 6:2). The conceptual operation of the perturbation is straightforward. In order to cause the biped to turn, the foot is placed on the ground at an angle which points toward the new direction of travel. During the stance phase, the hip realigns the body with the foot, which remains fixed relative to the ground.

DOF:		1	2	3:0	3:1	3:2	4	5:0	5:1	6:0	6:1	6:2	7	8:0	8:1	transition info
state	S1	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0
	S2	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
	S3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
	S6	0	0	0	0	1	0	0	0	0	0	0	-1	0	0	0

(a)

DOF:		1	2	3:0	3:1	3:2	4	5:0	5:1	6:0	6:1	6:2	7	8:0	8:1	transition info
state	S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	S3	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0
	S4	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0
	S5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	S6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)



(c)

Figure 5.5- Turning perturbations for human model with ball-and-socket hips (DOF 3:2 and DOF 6:3).

- (a) Unit-scaled right turning perturbation
- (b) Unit-scaled left turning perturbation
- (c) State diagram of relevant DOFs for right turning perturbation. Feet indicate hip twist with respect to pelvis.

For the right leg of the right turning perturbation, the initial foot rotation is performed during the swing phase of the right leg, as shown in state 5 of Figure 5.5 (c). During the right leg stance phase, this rotation is eliminated. The return rotation is done in the second stance state (S3), thereby ensuring that the foot is planted firmly on the ground (in S2) before attempting to turn the body. Ground friction keeps the foot position and orientation fixed, while the torso rotates about the leg. The left foot performs a similar motion in its stance phase to distribute the complete rotation evenly over both feet. A similar and symmetric set of perturbations is used for turning to the left. Figure 5.6 shows the typical action of the turning perturbation. For the sake of clarity, the motion is illustrated for the case of a stationary stepping motion.

Figure 5.7 shows the hip plots from a set of turning trials performed using a base PCG similar to that of Figure 3.7. The trials use up vector RVs with  $Q^d = [.25,0]$ . As can be seen, the turning control works well for each of the three sampling strategies. Each trial uses a different scaling of the turning perturbation in Figure 5.5. The largest scaling factor used which still yields forward motion corresponds to a hip yaw of 8 degrees for each leg, giving a turning radius of approximately 2.5 meters. All of the walks face slightly into the centre of the circle, an effect which is more pronounced for tighter curves.

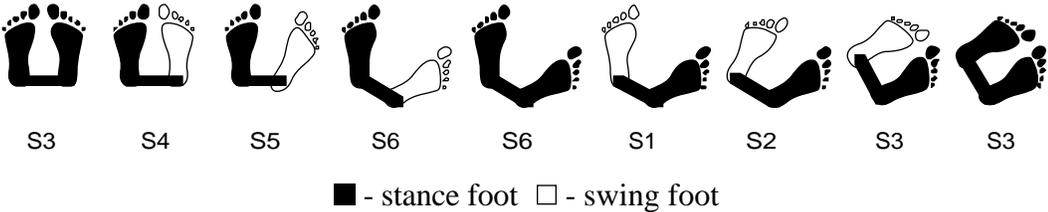


Figure 5.6 - Typical operation of a right turn perturbation for an in-place stepping motion. A top view of the feet and pelvis is shown, with the associated PCG state indicated below.

The largest scaling-factor trials attempted, corresponding to 16 degrees of right and left hip yaw, demonstrates an interesting behaviour. In such trials, the tendency to "lead" the turn with the

torso rotation causes the biped to rotate about its own vertical axis more quickly than it can move along a circular path. As a result, it turns to face the centre of the circle.

In the F-L and L-F sampling trials, the highest turning rates eventually cause the biped to turn around completely and walk almost straight backwards. The turning perturbation is ineffective in the backward walking cases because the model walks primarily on its heels. As a result, the foot never obtains sufficient ground reaction forces to fix the foot's orientation.

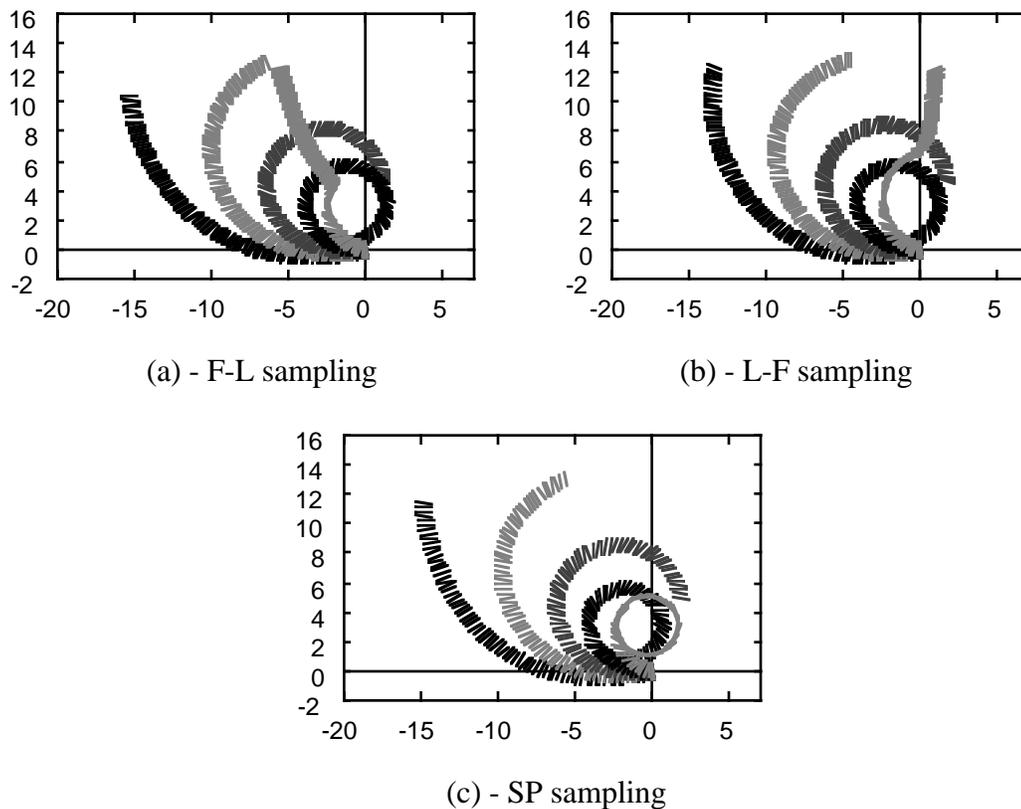


Figure 5.7 - Hip plots for the most successful turning perturbation trials over a range of scaling factors.  $Q^d = [.25,0]$  for all trials.

Not all values of  $Q^d$  work equally well as Figure 5.8 demonstrates. Linearly varying the perturbation scaling factor still yields smooth behaviour, but not the desired results of Figure 5.7. While this result makes it clear that some care must be taken in choosing parameters, it also serves to illustrate the fact that reasonable choices can often be made without exploring a large parameter

space through brute force experimentation. Here, the clue lies in the fact that the value of  $Q^d=[.25,0]$  is around the centre of the range of nominal  $Q^d$  values for the straight forward walks of Figure 4.3. This suggests that this value of  $Q^d$  might be best for other similar PCGs.

One deficiency of the walks of Figure 5.7 is that the smallest successful turning radius is approximately 2.5 meters. While this is adequate for navigating the biped in a relatively open environment, it is too large for motion in a constrained environment such as turning a corner in a narrow corridor.

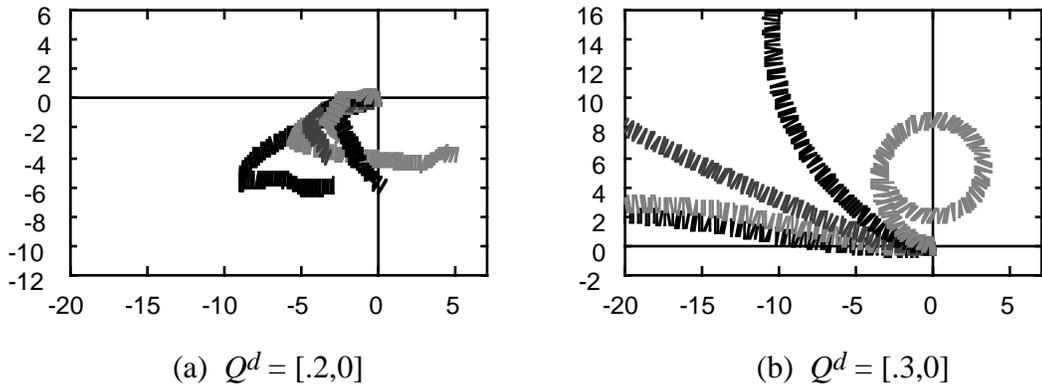


Figure 5.8 - Hip plots for less successful turning perturbation trials over a range of scaling factors.

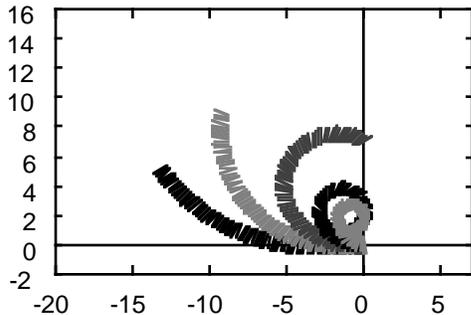


Figure 5.9 - Turning perturbation with torso servoing applied.

Applying torso servoing during turning helps significantly, as indicated by Figure 5.9. With torso servoing to an angle of 5 degrees from vertical, the walk is slower and the circles can be

tighter. The 16 degree turning perturbations fail, resulting in a turning limit cycle of approximately 0.75 meters in radius, a more useful lower limit in terms of the motion that can be animated. The walks still suffer from the biped facing too much toward the centre of the turn. For tight turns, the biped's outside swing leg interpenetrates the stance leg in a noticeable fashion. Nonetheless, the perturbations provide a good basis for higher level control of the biped's path.

### 5.3.1 Point and Path Following

By applying simple feedback control, the biped can be made to walk in a desired direction. Proportional control of the angle of the current facing direction and the desired direction as given by a target point is used to generate an appropriate turning rate.

The pose control applied each step including balance control is the same form as Equation 5.1.a:

$$( B + k_{turn}\Delta P_{turn} ) + k_1\Delta P_1 + k_2\Delta P_2$$

In the case of separate left and right turning control perturbations,

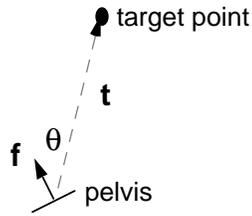
$$k_{turn} > 0 \text{ applies to the right turning perturbation and } \Delta P_{turn} = \Delta P_{turnright}$$

$$k_{turn} < 0 \text{ applies to the left turning perturbation and } \Delta P_{turn} = \Delta P_{turnleft}$$

and  $k_{turn}$  is computed according to the proportional control law:

$$k_{turn} = k_{\theta} \theta$$

where  $\theta$  is the angle between the biped's current facing direction and a vector to the target point, as illustrated in Figure 5.10.  $k_{\theta}$  is a gain constant which determines how tightly the biped turns for a given error in direction.  $k_{turn}$  is bounded to a predetermined maximum value in order to avoid large, unstable turn rates.



(TOP VIEW)

Figure 5.10 - Point-following. Vector  $f$  indicates the facing direction of the biped. Vector  $t$  indicates the desired direction.  $f$  and  $t$  are both in the horizontal plane.

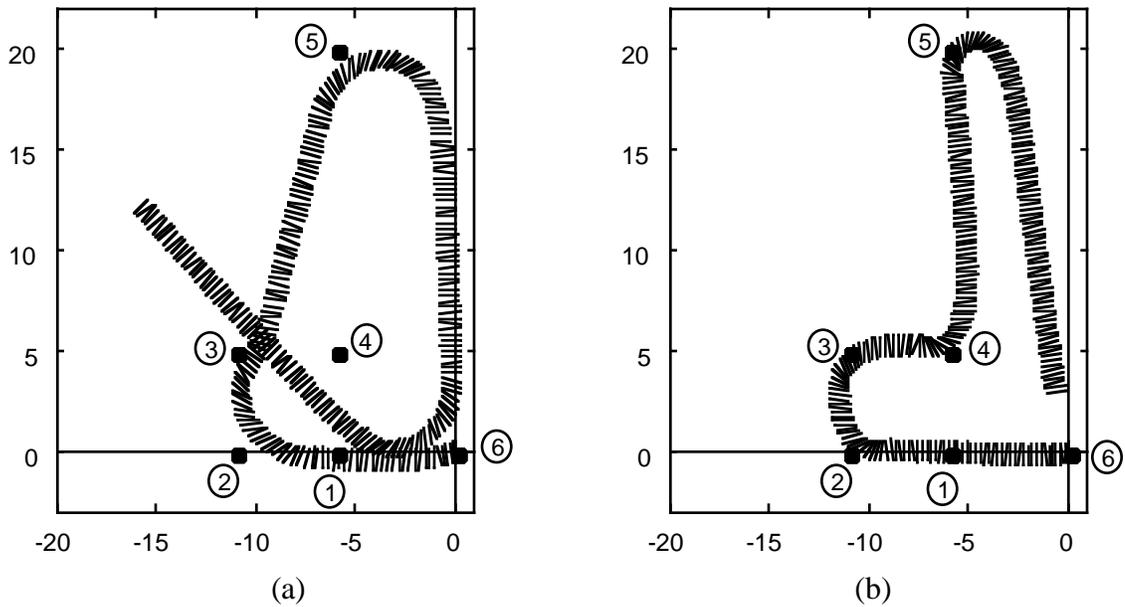


Figure 5.11 - Path following.  
 (a) without torso servoing  
 (b) with torso servoing

With this turning control, a simple form of path control can be achieved by following a suitable set of target points in sequence. Each time the biped approaches a target point to within a minimal distance, the next point in sequence becomes the current target point. Figure 5.11 shows the results of applying this algorithm to a set of target points, both with and without torso servoing. In both cases, the proportional constant,  $k_{\theta}$ , is chosen to give the minimum radius turn when the facing direction of the biped is 45 degrees from the target point and has a maximum value at that

angle. The minimum distance to the current target point before changing to the next target point in sequence is chosen to be slightly larger than the tightest turning diameter. This helps avoid scenarios where the biped indefinitely walks in circles around a target point.

DOF:		1	2	3:0	3:1	3:2	4	5:0	5:1	6:0	6:1	6:2	7	8:0	8:1	transition info
	S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05
state	S3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05
	S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05
	S6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05

Figure 5.12 -  $\Delta P_{striderate}$ , stride rate perturbation for human model with ball-and-socket hips and arms.

### 5.4 Stride Rate Perturbation

The stride rate perturbation can be used to vary the stride rate of the biped. The perturbation, shown in Figure 5.12, modifies only the hold times of particular states in the base PCG. Applying the perturbation with a positive scaling parameter increases the hold time in each state, resulting in a decrease in stride rate. Applying a negative scaling parameter has the opposite effect, decreasing the hold time in each state and the increasing stride rate. The unperturbed walk is a straight walk with a stride rate of 1.0 strides/second and uses torso servoing and speed control with a composite RV<sup>3</sup>.

Figure 5.13 (a), (b) and (c) show three distinct steps in which the perturbation is applied with a gain of -1.0, 0 and +1.0 respectively to obtain stride rates varying from 0.8 strides per second to approximately 1.25 strides per second. The three sequences are in fact taken from a single walk, during which the stride rate is dynamically varied. In this case, stride rate transitions use 3 steps for each unit of change in the perturbation gain.

---

<sup>3</sup> In this case, an up vector-based forward RV component and a swing-COM-based lateral RV component.

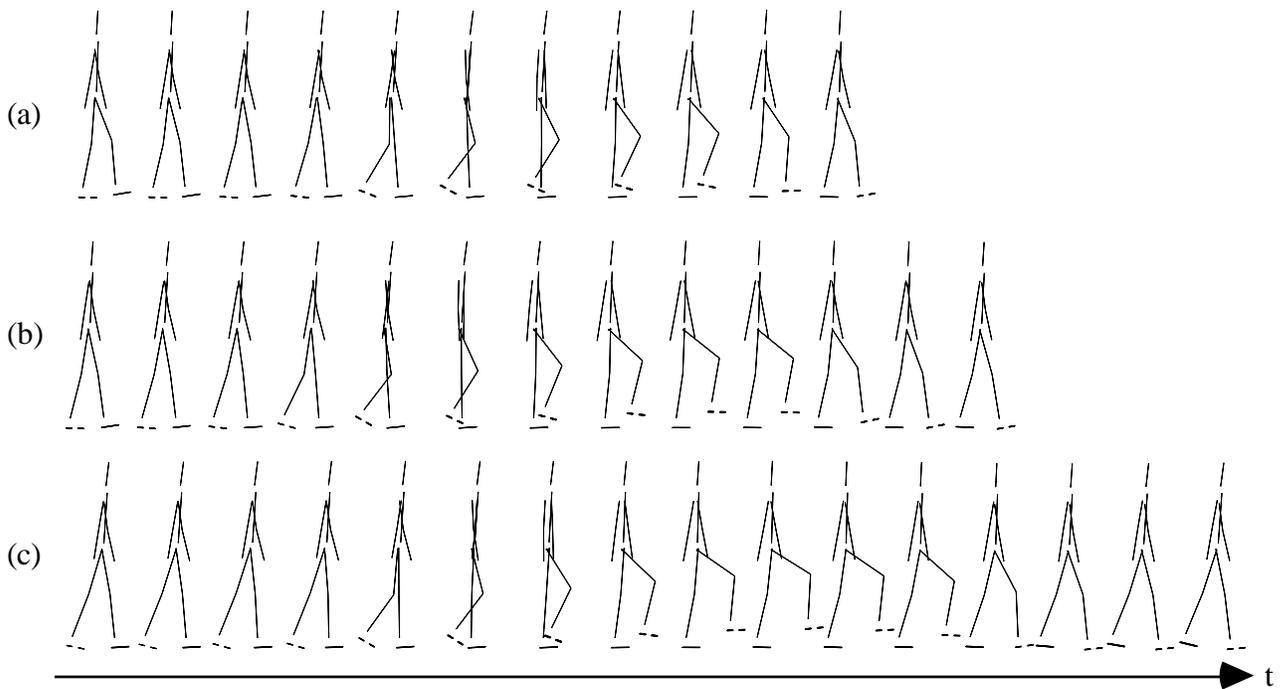


Figure 5.13 - Results of applying different stride rate perturbation scalings. Each sequence indicates one full step. Horizontal distance along the page indicates relative stride time.

- (a)  $k_{striderate} = -1$  i.e.  $B^* = B + (-1.0 \cdot \Delta P_{striderate})$
- (b)  $k_{striderate} = 0$
- (c)  $k_{striderate} = +1$

As the sequences of Figure 5.13 indicate, the peak height of the swing knee in each step increases with hold time. This is due to an associated increase in stance hip pitch. The increased hip pitch is automatically introduced by the balance control in order to meet  $Q^d$  at the end of each step. Without this action the biped would fall due to leaning too far forward with larger hold times or too far backward with smaller hold times. The increased hip pitch gives the walk a more mechanical, "marching" appearance at lower step rates than at higher step rates.

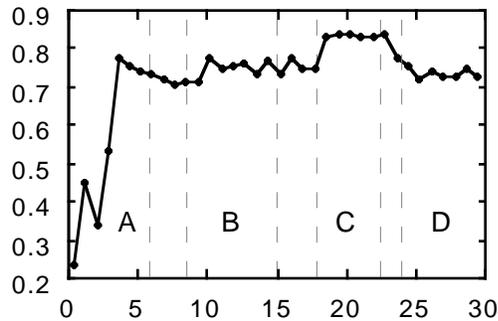


Figure 5.14 - Average speed (m/s) for varying stride rates

Region A:  $k_{striderate} = 0.0$

Region B:  $k_{striderate} = +1.0$

Region C:  $k_{striderate} = -1.0$

Region D:  $k_{striderate} = 0.0$

Other regions are transition regions.

Figure 5.14 shows the average speed of the pelvis over the course of the walk which is controlled to stay relatively constant while the biped's stride rate increases and decreases. A constant speed implies that the stride length increases inversely with stride rate (and hence linearly with hold time) for this particular perturbation.

## 5.5 Other Interesting Variations

PCGs varied in other ways can also be balanced successfully. Stylistic variations can be used to convey different moods or emotions and allow a broader range of abilities. In some cases, simpler variations can be combined successfully to form more complex motions. The next few sections provide some examples.

### 5.5.1 Bent-Knee Walking

The perturbation of Figure 5.15 generates a walk with bent knees. While this perturbation may not seem particularly useful on its own, it can be successfully combined with other perturbations to generate a ducking perturbation for the biped.

DOF:		1	2	3:0	3:1	3:2	4	5:0	5:1	6:0	6:1	6:2	7	8:0	8:1	9	10	11	12	transition info
state	S1	0	0	0	-20	0	20	-20	0	0	-20	0	20	-20	0	15	-30	15	-30	0
	S2	0	0	0	-20	0	20	-20	0	0	-20	0	20	-20	0	15	-30	15	-30	0
	S3	0	0	0	-20	0	20	-20	0	0	-20	0	20	-20	0	15	-30	15	-30	0
	S4	0	0	0	-20	0	20	-20	0	0	-20	0	20	-20	0	15	-30	15	-30	0
	S5	0	0	0	-20	0	20	-20	0	0	-20	0	20	-20	0	15	-30	15	-30	0
	S6	0	0	0	-20	0	20	-20	0	0	-20	0	20	-20	0	15	-30	15	-30	0

Figure 5.15 - Bent-knee perturbation for a human model

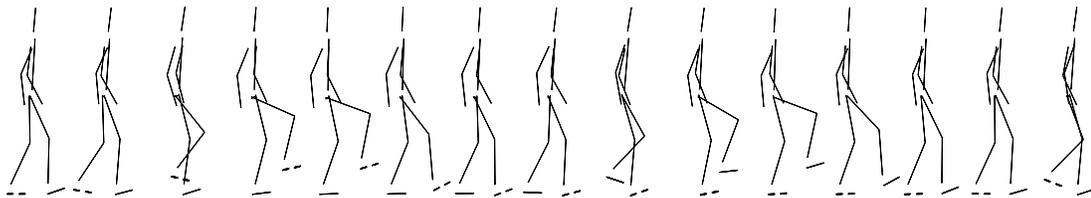


Figure 5.16 - Bent-knee walking with a parameter scaling of 1.0

Figure 5.16 shows a sequence of steps from a bent-knee walk generated by applying the perturbation to a basic forward walk. The transition from the normal walk to the fully perturbed (scaling parameter = 1.0) walk is performed smoothly over a series of 18 steps. Qualitatively speaking, the motion is lively and “bouncy”. The model spends most of each stance phase standing on its heel due to a raised stance toe (-20 degrees ankle pitch).

### 5.5.2 Bent-Over Walking

A bent-walking perturbation shown in Figure 5.17 provides yet another walking configuration. It consists of a relatively small bend at the waist and a larger bend in the torso, generating a slightly hunched over, forward leaning walk.

DOF:		1	2	3:0	3:1	3:2	4	5:0	5:1	6:0	6:1	6:2	7	8:0	8:1	9	10	11	12	13	transition info
state	S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-52	-5	-42	-5	15	0
	S2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-52	-5	-33	-5	15	0
	S3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-52	-5	-33	2	15	0
	S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-42	-5	-52	-5	15	0
	S5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-33	-5	-52	-5	15	0
	S6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-33	-2	-52	-5	15	0

Figure 5.17 - Bent-over walking perturbation

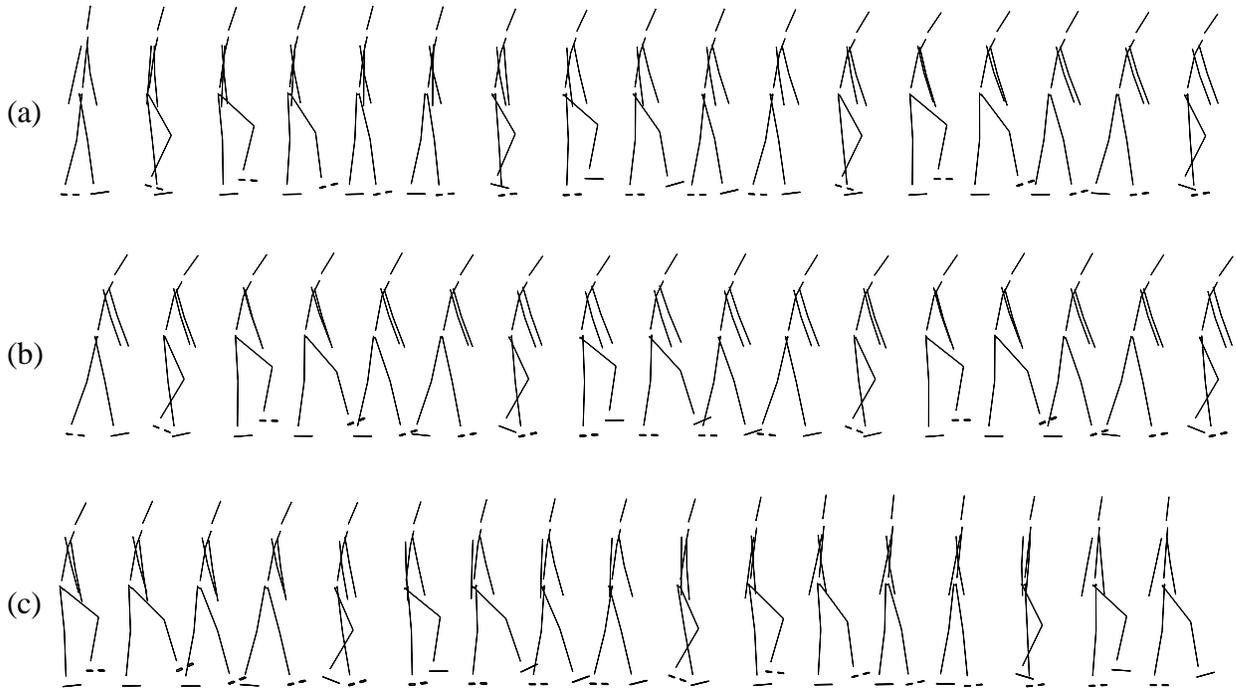


Figure 5.18 - Bent-over walking  
 (a) straight to bent transition  
 (b) walking bent-over  
 (c) bent to straight transition

Figure 5.18 shows the transitions into and out of the bent-over walk and a few steps of the bent-over walk. Both transitions are performed over two steps, with a different scaling parameter used for each intermediate step. The arm perturbations are included in order to make the biped's arms appear to hang by its side in a "natural" position while the body is bent-over. In retrospect, a better approach would be to reduce the stiffness and damping parameters of the shoulder actuators, letting physics determine a more appropriate arm position for the walk. This would require variable joint stiffness and damping parameters which are not currently supported.

### 5.5.3 Ducking

By combining the bent-knee and bent-over perturbations with an additional bend of the neck, it is possible to form a relatively simple composite "ducking" perturbation which could potentially be

used to avoid overhead obstacles. Figure 5.19 shows the composite perturbation for the human model used with the bent-over perturbation.

DOF:		1	2	3:0	3:1	3:2	4	5:0	5:1	6:0	6:1	6:2	7	8:0	8:1	9	10	11	12	13	transition info
state	S1	5	40	0	-20	0	20	0	0	0	-20	0	20	0	0	-52	-5	-42	-5	15	0
	S2	5	40	0	-20	0	20	0	0	0	-20	0	20	0	0	-52	-5	-33	-5	15	0
	S3	5	40	0	-20	0	20	0	0	0	-20	0	20	0	0	-52	-5	-33	2	15	0
	S4	5	40	0	-20	0	20	0	0	0	-20	0	20	0	0	-42	-5	-52	-5	15	0
	S5	5	40	0	-20	0	20	0	0	0	-20	0	20	0	0	-33	-5	-52	-5	15	0
	S6	5	40	0	-20	0	20	0	0	0	-20	0	20	0	0	-33	-2	-52	-5	15	0

Figure 5.19 - Composite bent-knee, bent-over and bent-neck perturbation

Figure 5.20 shows the transitions to and from "ducking". Note that while the bent-knee perturbation was originally designed for a slower rate walk which uses speed control, it also works well when applied to a different basic walk (bent-over) without speed control.

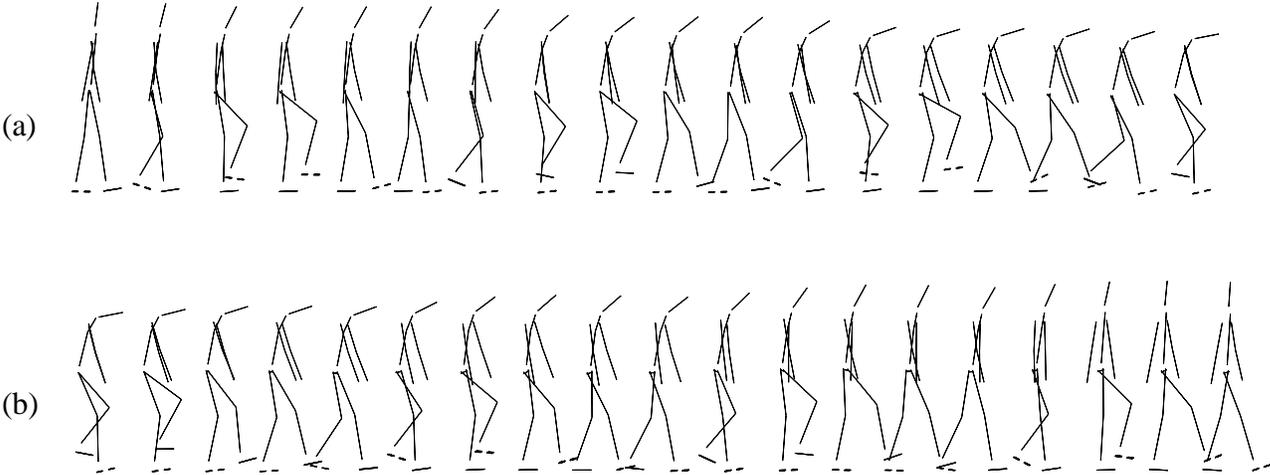


Figure 5.20 - Ducking  
 (a) straight to ducking transition  
 (b) ducking to straight transition

While the ducking perturbation is demonstrated here in an open loop configuration, one can imagine using feedback to provide the biped with the ability to automatically duck under obstacles in its path.

## 5.6 Conclusions

In this chapter, we have shown that parameterized PCG perturbations can be used to generate a variety of interesting gaits simply by modifying the underlying open loop control. Smooth transitions between motions were demonstrated and an example of the use of a simple feedback mechanism to increase creature autonomy was given. An approach to controlling the speed of a walk without requiring modification of the underlying base PCG was also provided. The basic balance control introduced in Chapter 3 required no modification to be used successfully with variable motions.

## 6. CONCLUSIONS AND FUTURE WORK

The initial goal of this thesis was to provide a general technique for the control of complex, statically-unstable 3D bipedal creatures for use in physically-based animation. For simplicity, only cyclic motions were considered. The proposed control formulation meets this goal by discretizing the periodic motion into cycles and simplifying the control through the use of a limited set of control perturbations which are used to stabilize a small set of observed variables.

The approach is general in the sense that the same control technique can be used to control a wide variety of walking gaits for a wide range of control parameter choices. Control of speed and direction for a biped has been demonstrated, as well as parametric variations of a number of walking characteristics. By using simple feedback to drive the parameterized controllers, it is possible to give the animated creature a greater degree of autonomy, providing the animator with high level control. The approach has been demonstrated for walking and limited forms of running, using both a human model and a bird-like robot model. We believe that the technique will also prove to be a suitable approach to animating many other types of periodic behaviours for a wide variety of articulated figures.

While the bipeds are not of full human complexity (which would require about 200 DOFs), they are sufficiently complex that many other forms of control would prove computationally infeasible. Many of the DOFs in the human body cannot act independently, for example, the vertebrae of the spine. The tens of DOFs in our models are sufficient to capture the gross motions of natural bipedal locomotion with reasonable fidelity.

The primary contribution of this thesis is to illustrate that control techniques can be successfully applied to animate creatures of high complexity.

## **6.1 Future Work**

A number of items remain for future work. In particular, improved performance, other forms of locomotion and more natural looking motion remain to be addressed. As well, extension of the control formulation to non-periodic motions and the possibility of automating much of the design process stand out as worthwhile avenues to pursue.

### **6.1.1 Better Discrete System Models**

One of the drawbacks of the current control approach is the high computational expense of reconstructing the discrete system model each cycle. In the case of our bipedal control, this results in a four-fold increase in the time required to generate the final motion. Two reasonable possibilities exist to reduce this cost. Both are based on the reuse of previously computed models rather than blind reconstruction of the model each cycle.

One approach would be to reconstruct the model only when necessary. Once a reasonable limit cycle has been reached, the model parameters determined through sampling remain relatively constant from one cycle to the next. In such cases, a fixed model may be sufficient. By assuming fixed model parameters and monitoring the final RV values for deviations from the limit cycle, it seems likely that direct balance control can be achieved for much of the desired motion. When a limit cycle terminates, for example due to a change in base PCG, a new system model could be constructed.

A second approach might be to construct a general discrete system model which is parameterized with respect to the creature's initial state at the start of a cycle. Such a model could be constructed by generating a number of walks from various initial conditions and recording the model parameters and initial state for each. Once a large enough number of models have been generated, they could be used in the form of a lookup table. Particular models could be chosen using a nearest neighbour approach based on the initial state of the current cycle. The number of different

models generated should be sufficiently large to span a reasonable domain of initial states. A general model like this could incorporate varying terrain or other environmental state information in order to allow a wide range of behaviours.

### **6.1.2 Additional Forms of Locomotion**

A number of features need to be added to the system for it to be truly useful as a generic biped animation system. First, other common forms of bipedal locomotion such as skipping and hopping would be necessary, as well as transitions between the various types of gaits. A second desirable feature not yet explored is the ability to generate robust locomotion over varying terrain. Finally, it should be possible to parameterize a controller with respect to various model properties such as mass and dimensions. We believe that it is possible to implement such features within the proposed control structure.

### **6.1.3 Natural Motion**

The motions obtained to date using our technique do not yet represent convincing human motion. This is primarily due to the use of simple base PCGs. One possible way to achieve more natural motion might be to fine tune a suitable open-loop motion based on motion capture data. While we are convinced that more natural looking motion can be attained with a reasonable amount of extra effort, this remains to be demonstrated.

### **6.1.4 Extension to Aperiodic Motions and Further Generalization**

While this thesis has focused on cyclic motions, a similar approach might be suitable for controlling aperiodic motions. Such motions would include standing up, sitting down or throwing a ball at a target. Other useful acyclic motions might include transitions into and out of cyclic motion and dynamic balancing in place, stepping only when necessary. A unified control technique for both periodic and aperiodic motions would be quite useful, since animators typically require the ability to freely move between the two as needed.

### 6.1.5 Automatic Synthesis

Another possible direction for future work is toward automatic controller synthesis. While the base PCGs, RVs and LPPs for the motions given throughout this thesis were designed by hand, they should ideally be automatically generated. Previous use of pose control has been directed primarily toward such automatic synthesis of motion controllers [vKF94] [vKF94b] [vL95].

Most of these attempts deal with relatively simple, statically-stable creatures. The problem of automatic synthesis of control for statically unstable 3D systems is a difficult one. In [vL95], van de Panne and Lamouret propose the use of external guiding forces to initially generate cyclic PCGs, followed by two subsequent phases which reduce the forces and then attempt to eliminate them respectively. While the motions generated by the second phase are significantly more natural looking than the initial PCGs, they are not fully realistic in the sense that the creature's actuators do not drive the motion unassisted. The final phase, removal of external forces, is not always successful and can be very computationally expensive.

Rather than attempting to directly synthesize the entire PCG in this way, it might be possible to automatically synthesize various control components for our system. The two most likely candidates are the base PCG and the fixed PCG perturbations (LPPs). Base PCGs for various interesting motions might be synthesized using van de Panne's first two phases and then stabilized using the balance control presented in this thesis. LPPs might be synthesized using a more traditional generate, test, and refine process such as that used for simple, statically stable systems.

## REFERENCES

- [Alias] *Learning Alias Level One*, R.R.Donnelley publisher, 1995.
- [Ale84] R. McN. Alexander. The Gaits of Bipedal and Quadrupedal Animals. *International Journal or Robotics Research*, 3(2):49-59, Summer 1984.
- [AGL86] W. W. Armstrong, M. Green and R. Lake. Near-real-time control of human figure models. *Graphics Interface '86*, pages 147-151, 1986.
- [A+95] J. Auslanderet, A. Fukunaga, H. Partovi, J. Christensen, L. Hsu, P. Reiss, A. Shuman, J. Marks, J. T. Ngo. Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics*, pages 311-336, October 1995.
- [BMW87] N. Badler, K. H. Manoocherhri and G. Walters. Articulated figure positioning by multiple constraints. *IEEE Computer Graphics and Applications*, 7(6):28-38, June 1987.
- [BPW93] N. I. Badler, C. B. Phillips and B. L. Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, 1993.
- [BMTT90] R. Boulic, N. Magnenat-Thalmann and D. Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6:344-358, 1990.
- [BN88] L. S. Brotman and A. N. Netravali. Motion interpolation by optimal control. *Proceedings of SIGGRAPH 88*. In *ACM Computer Graphics*, 22(4):309-315, August 1988.
- [BC89] A. Bruderlin and T. W. Calvert. Goal-Directed, Dynamic Animation of Human Walking. *Proceedings of SIGGRAPH 89*. In *ACM Computer Graphics*, 23(3):233-242, July 1989.

- [BC93] A. Bruderlin and T. W. Calvert. Interactive Animation of Personalized Human Locomotion. *Graphics Interface '93*, pages 17-23, 1993.
- [BC96] A. Bruderlin and T. W. Calvert. Knowledge-Driven, Interactive Animation of Human Running. *Graphics Interface '96*, pages 213-221, 1996.
- [BW95] A. Bruderlin and L. Williams. Motion Signal Processing. *Proceedings of SIGGRAPH 95*. In *ACM Computer Graphics*, pages 97-104, August 1995.
- [BW71] N. Burtnyk and M. Wein. Computer generated keyframe animation. *Journal of the Society of Motion Picture and Television Engineers*, 80(3):149-153, March 1971.
- [CHP92] P. H. Channon, S. H. Hopkins and D. T. Pham. Derivation of optimal walking motions for a bipedal walking robot. *Robotica*, 10:165-172, 1992.
- [Coh92] M. F. Cohen. Interactive spacetime control for animation. *Proceedings of SIGGRAPH 92*. In *ACM Computer Graphics*, 26(2):293-302, July 1992.
- [Csu71] C. Csurí. Real time film animation. *IEEE Convention Digest*, pages 42-43, March 1971.
- [FM87] J. Furusho and M. Masubuchi. A Theoretically Motivated Reduced Order Model for the Control of Dynamic Biped Locomotion. *Journal of Dynamic Systems, Measurement and Control*, 109:155-163, June 1987.
- [Fri86] B. Friedland. *Control System Design: An Introduction to State-Space Methods*. McGraw-Hill, 1986.
- [FS90] J. Furusho and A. Sano. Sensor-Based Control of a Nine-Link Biped. *International Journal of Robotics Research*, 9(2):83-98, April 1990.
- [GM85] M. Girard and A. A. Maciejewski. Computational modeling for the computer animation of legged figures. *Proceedings of SIGGRAPH '85*. In *Computer Graphics*, pages 263-270, July 1985.

- [GT95] R. Grzeszczuk and D. Terzopoulos. Automated Learning of Muscle-Actuated Locomotion through Control Abstraction. *Proceedings of SIGGRAPH '95*. In *Computer Graphics*. pages 63-70, August 1995.
- [HF77] H. Hemami and R. L. Farnsworth. Postural and Gait Stability of a Planar Five Link Biped by Simulation. *IEEE Transactions on Automatic Control*, pages 452-458, June 1977.
- [Hod91] J. K. Hodgins. Biped Gait Transitions. *Proceedings of the International Conference on Robotics and Automation*, pages 2092-2097, April 1991.
- [HL91] H. M. Hmam and D. A. Lawrence. Biped Control Via Nonlinear Dynamics. *29th Allerton Conference on Communication, Control, and Computing*, October, 1991.
- [HR86] D. Halliday and R. Resnick. *Fundamentals of Physics*. John Wiley & Sons, 1986.
- [HSL92] J. K. Hodgins, P. K. Sweeney and D. G. Lawrence. Generating Natural-looking Motion for Computer Animation. *Graphics Interface '92*, pages 265-272, 1992.
- [H+95] J. K. Hodgins, W. L. Wooten, D. C. Brogan and J. F. O'Brien. Animating Human Athletics. *Proceedings of SIGGRAPH 95*. In *ACM Computer Graphics*. pages 71-78, August 1995.
- [IRT81] V. T. Inman, H. J. Ralston and F. Todd. *Human walking*. Williams & Wilkins, Baltimore, 1981.
- [GM85] M. Girard and A. A. Maciejewski. Computational modelling for the computer animation of legged figures. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1405-1411, Sacramento, CA, April 1991.
- [KT91] S. Kajita and K. Tani. Study of Dynamic Biped Locomotion on Rugged Terrain. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1405-1411, Sacramento, CA, April 1991.
- [Kim94] R. Kim. *Automatic Motion Synthesis using Pose Controllers*. M.A.Sc. Thesis, University of Toronto, Department of Electrical Engineering, 1994.

- [KKI90] S. Kitamura, Y. Kurematsu and M. Iwata. Motion generation of a biped locomotive robot using an inverted pendulum model and neural networks. *IEEE*, pages 3308-3312, 1990.
- [KB84] D. H. Kochanek and R. H. Bartels. Interpolating splines for keyframe animation. *Graphics Interface*, pages 41-42, 1984.
- [KB93] H. Ko and N. I. Badler. Straight Line Walking Animation Based on Kinematic Generalization that Preserves the Original Characteristics. *Graphics Interface '93*, pages 9-16, 1993.
- [KB82] J. U. Korein and N. I. Badler. Techniques for Generating the Goal-directed Motion of Articulated Structures. *IEEE Computer Graphics and Applications*, pages 71-81, November 1982.
- [Las87] J. Lasseter. Principles of traditional animation applied to 3D computer animation. *Proceedings of SIGGRAPH '87*. In *ACM Computer Graphics*. 21(4):35-44, July 1987.
- [LGC94] Z. Liu, S. J. Gortler and M. Cohen. Hierarchical Spacetime Control. *Proceedings of SIGGRAPH '94*. In *ACM Computer Graphics*. pages 35-42, 1994.
- [MTT85b] N. Magnenat-Thalmann and D. Thalmann. *Computer Animation: Theory and Practice*. Springer-Verlag, 1985.
- [McG89] T. McGeer. Powered flight, child's play, silly wheels and walking machines. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1592-1597, 1989.
- [McG90] T. McGeer. Passive walking with knees. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1640-1645, 1990.
- [McG90b] T. McGeer. Passive bipedal running. *Proceedings of the Royal Society of London*, Vol B.240, 1990. pp. 107-134.

- [MZ90] M. McKenna and D. Zeltzer. Dynamic Simulation of Autonomous Legged Locomotion. *Proceedings of SIGGRAPH 90*. In *ACM Computer Graphics*, 24(4):29-38, August 1990.
- [McM84] T. A. McMahon. Mechanics of Locomotion. *International Journal of Robotics Research*, 3(2):4-28, Summer 1984.
- [Mez68] L. Mezie. Sparta, a procedure-oriented programming language for the manipulation of arbitrary line drawings. *Fourth Congress of the International Federation for Information Processing*, pages 96-102, August 1968.
- [Mil88] G. S. P. Miller. The Motion Dynamic of Snakes and Worms. *Proceedings of SIGGRAPH 88*. In *ACM Computer Graphics*, 22(4):169-187, August 1988.
- [MM80] S. Mochon and T. A. McMahon. Ballistic walking. *Journal of Biomechanics*, 13:49-57, 1980.
- [MS84] H. Miura and I. Shimoyama. Dynamic Walk of a Biped. *International Journal of Robotics Research*, 3(2):60-74, Summer 1984.
- [NM93] J. T. Ngo and J. Marks. Spacetime constraints revisited. *Proceedings of SIGGRAPH 93*. In *ACM Computer Graphics*. ??(?):343-350, August 1993.
- [PB89] M. G. Pandy and N. Berme. Quantitative assessment of gait determinants during single stance via a three-dimensional model – Part 1. Normal gait. *Journal of Biomechanics*. 22(6-7):717-724, 1989.
- [Rai86] M. H. Raibert. Running With Symmetry. *International Journal of Robotics Research*. 5(4):3-19, Winter 1986.
- [Rai+84] M. H. Raibert, H. B. Brown and M. Chepponis. Experiments in Balance with a 3D One-Legged Hopping Machine. *International Journal or Robotics Research*. 3(2):75-92, Summer 1984.

- [Rai86b] M. H. Raibert, M. Chepponis and H. B. Brown. Running on Four Legs As Though They Were One. *IEEE Journal of Robotics and Automation*. RA-2(2):70-82, June 1986.
- [RH91] M. H. Raibert and J. K. Hodgins. Animation of Dynamic Legged Locomotion. *Proceedings of SIGGRAPH 91*. In *ACM Computer Graphics*. 25(4):349-358, August 1991.
- [SZ92] A. W. Salatian and Y. F. Zheng. Gait Synthesis for a Biped Robot Climbing Sloping Surfaces Using Neural Networks, Part I: Static Learning. *Proceedings of the 1992 International Conference on Robotics and Automation*. pages 2601-2606, Nice, France, May 1992.
- [SSH82] S. Seigler, R. Seliktar and W. Hyman. Simulation of human gait with the aid of a simple mechanical model. *Journal of Biomechanics*. 15:415-425, 1982.
- [Sim94] K. Sims. Evolving Virtual Creatures. *Proceedings of SIGGRAPH '94*. In *Computer Graphics*. pages 43-50, July 1994.
- [Sni95] Burr Snider. Rocket Science. *Wired Magazine*, November 1995.
- [SB85] S. N. Steketee and N. I. Badler. Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing. *Proceedings of SIGGRAPH '85*. In *Computer Graphics*. 19(3):255-262, July 1985.
- [SC92] A. J. Stewart and J. F. Cremer. Beyond Keyframing: An Algorithmic Approach to Animation. *Graphics Interface '92*. pages 273-281, 1992.
- [Stu84] D. Sturman. Interactive keyframe animation of 3-d artificial models. *Graphics Interface*. pages 35-40, 1984.
- [SCD80] D. H. Sutherland L. Cooper and D. Daniel. The role of the ankle plantarflexors in normal walking. *Journal of Bone and Joint Surgery*. (A) 62:354-363, 1980.
- [SV89] M.W. Spong and M. Vidyasagar. *Robot Dynamics & Control*. John Wiley & Sons, 1989.

- [TIYK85] A. Takanishi, M. Ishida, Y., Yamazaki and I. Kato. The realization of dynamic walking by the biped walking robot WL-10RD. *Proceedings of the 1985 ICAR*. pages 459-466, 1985.
- [Tow85] M. A. Townsend. Biped gait stabilization via foot placement. *Journal of Biomechanics*. 18(1):21-38, 1985.
- [Tow72] M. A. Townsend and A. Seireg. The synthesis of bipedal locomotion. *Journal of Biomechanics*. 5:71-83, 1972.
- [Tow76] M. A. Townsend and T. C. Tai. Biomechanics and modelling of bipedal climbing and descending. *Journal of Biomechanics*. 9:227-239, 1976.
- [UAT95] M. Unuma, K. Anjyo and R. Takeuchi. Fourier Principles for Emotion-based Human Figure Animation. *Proceedings of SIGGRAPH '95*. In *Computer Graphics*. pages 91-96, August 1995.
- [van89] M. van de Panne. *Motion Synthesis for Simulation-Based Animation*. M.A.Sc. Thesis, University of Toronto, Department of Electrical Engineering, 1989.
- [vF93] M. van de Panne and E. Fiume. Sensor-Actuator Networks. *Proceedings of SIGGRAPH 93*. In *ACM Computer Graphics*. pages 335-342, August 1993.
- [vL95] M. van de Panne, A. Lamouret. Guided Optimization for Balanced Locomotion, *Sixth Eurographics Workshop on Animation and Simulation*, Sept. 2-3, 1995.
- [vFV92] M. van de Panne, E. Fiume and Z. G. Vranesic. A Controller for the Dynamic Walk of a Biped Across Variable Terrain. *Proceedings of 31st Conference on Decision and Control*. pages 2668-2673, December 1992.
- [van96] M. van de Panne. Parameterized gait synthesis, *IEEE Computer Graphics and Applications*, v16 (Mar. 1996), pages 40-49.
- [vKF94b] M. van de Panne, R. Kim, and E. Fiume. Synthesizing Parameterized Motions, *Fifth Eurographics Workshop on Animation and Simulation*, Sept. 17-18, 1994.

- [vKF94] M. van de Panne, R. Kim, and E. Fiume. Virtual Wind-Up Toys, *Proceedings of Graphics Interface '94*, May 1994, 208-215.
- [VJ69] M. Vukobratovic and D. Juricic. Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*. 16:1-6, 1969.
- [VS72] M. Vukobratovic and Y. Stepanenko. On the stability of anthropomorphic systems. *Math. Biosci.* 15:1-37, 1972.
- [WH95] W. L. Wooten, J. K. Hodgins. Simulation of Human Diving. *Proceedings of Graphics Interface 95*. pages 1-9, 1995.
- [Wil86] J. Wilhelms. Virya - A motion control editor for kinematic and dynamic animation. *Proceedings of Graphics Interface '86*. pages 141-146, 1986.
- [WK88] A. Witkin and M. Kass. Spacetime Constraints. *Proceedings of SIGGRAPH 88*. In *ACM Computer Graphics*, 22(4):159–168, August 1988.
- [WP95] A. Witkin and Z. Popovic. Motion Warping. *Proceedings of SIGGRAPH 95*. In *ACM Computer Graphics*, pages 105-108, 159–168, August 1995.
- [Zel82] D. Zeltzer. Motor Control Techniques for Figure Animation. *IEEE Computer Graphics and Applications*, 2(9):53-59, November 1982.
- [ZS90] Y. F. Zheng and J. Shen. Gait Synthesis for the SD-2 Biped robot to Climb Sloping Surface. *IEEE Transactions on Robotics and Automation*, 6(1):53-59, February 1990.

## APPENDIX A – TERMS AND DEFINITIONS

### Robotics and Systems Control

**Articulated Figure** – A set of rigid body segments connected by rotary joints.

**Degrees of Freedom (DOFs)** – The set of variables required to fully specify a creature's position and orientation in free space. A single rigid body in free space has six degrees of freedom, three for position and three for orientation.

**Kinematics** – Computation of the absolute position of all parts of a creature, given the relative joint angles. In animation, kinematics typically refers to the specification or manipulation of the joint angles and velocities without regard to physics.

**Inverse Kinematics** – Computation of the intermediate joint angles from absolute positions. For example, computing the angles of leg joints needed to place a foot in a particular position.

**Dynamics** – Computation of the accelerations of the links of an articulated figure using the laws of physics.

**Inverse Dynamics** – Computation of the torques and forces required to be applied to a body to achieve desired accelerations.

**Centre of mass (COM)** – The single point on a body through which a linear force can be considered to act, defined as:

$$\bar{x} = \frac{\int x \, dm}{M}$$

where M is the total mass.

**System state** – The set of variables required to fully specify the position and velocity of every point on an object. The system state of an object consists of its degrees of freedom and their derivatives. Figure A–1 shows the two components of the state for

a simple swinging pendulum plotted with respect to time. The two near-sinusoidal curves are out of phase since the peak joint velocity occurs when the joint angle is zero and the peak angle occurs when the velocity is zero.

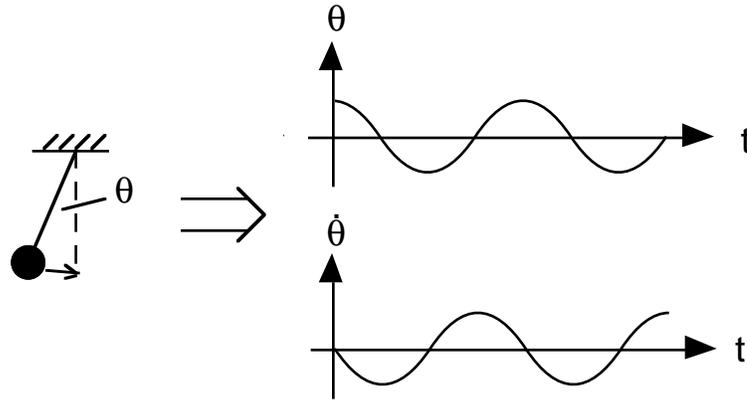


Figure A-1 - State vs time for a 1 degree-of-freedom swinging pendulum plotted vs time.

**State space** – The set of all possible values of the state of an object. A trajectory through the state space of an object describes its motion. Figure A-2 shows the state space representation of the trajectories of Figure A-1.

**Limit cycle** – A periodic, cyclic trajectory through state space. The trajectory in Figure A-2 is an example of a limit cycle which represents the periodic motion of a simple pendulum. Throughout this thesis, the term limit cycle is used to refer to cyclic, periodic motion in part of the state space rather than strictly applying to the full state space.

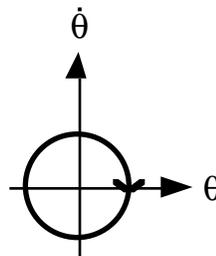


Figure A-2 - State-space trajectory of a simple swinging pendulum.



**Double-stance** – The phase of a walk during which both feet are in contact with the ground.

Figure A-4 shows the phases of a bipedal walk.

**Single-stance** – The phase of a walk or run during which only one foot is in contact with the ground. See Figure A-4.

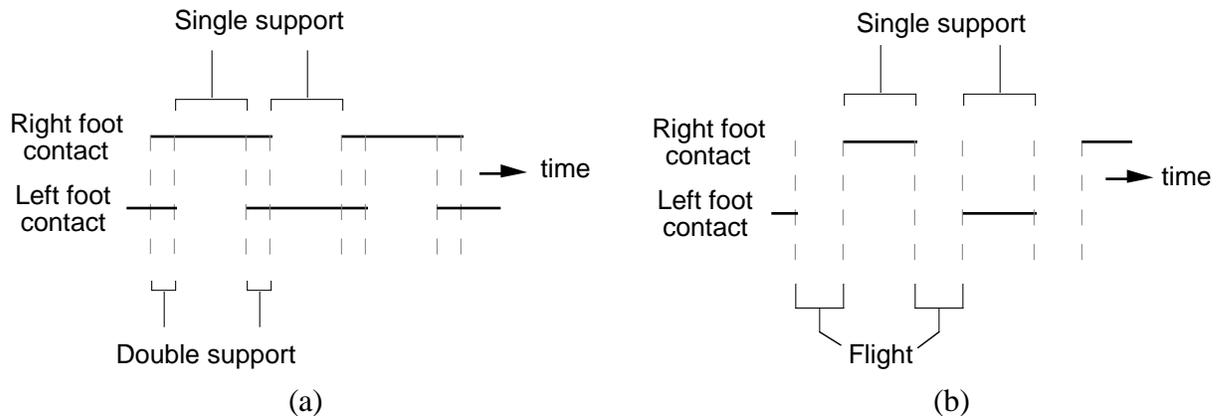


Figure A-4 - Phases of bipedal walking and running. Solid lines indicate the times that the associated foot is in contact with the ground.

(a) walking

(b) running

## Computer Animation

**Motion capture** - A technique, used to animate computer generated characters, in which the motion of the animated character is taken from a real world source such as a person walking. Modern forms of motion capture typically use markers attached to the subject's body which are tracked by a sensing device (e.g. an IR camera).

**Rotoscoping** - A form of motion capture in which motion data is taken from pre-recorded live images such as film or video.

## Thesis Related

**Pose Control Graph (PCG)** – A type of finite state machine which specifies a set of desired poses over time. Each desired pose specifies a set of desired joint angles for the articulated figure being animated.

**Base Pose Control Graph** – A pose control graph used to provide the basic cyclic control for a desired periodic motion.

**Regulation Variable (RV)** – A scalar function of system state chosen to represent a key feature in the overall motion of an articulated figure. A set of regulation variables essentially form a reduced order model of the system. Regulation variables are controlled in order to achieve some desired change in the overall motion for a figure.  $Q^d$  is used to denote the desired (or target) value of an RV.  $Q^*$  represents the final controlled value of the RV.

**Linear Parametric Perturbation (LPP)** – A fixed PCG multiplied by a scaling factor and added to a base PCG. LPPs are used to achieve desired changes in the regulation variables or variations in the overall motion of the base PCG.

**Superposition (SP) sampling** – A sampling strategy for constructing a model to balance one step of a walk. Using SP sampling, the biped is balanced in the sagittal and coronal planes independently and then the results are combined.

**Forward-then-lateral (F-L) sampling** – A strategy for constructing a model to balance one step of a walk. Using F-L sampling, the biped is first balanced in the sagittal plane and then the results of this operation are used to balance the biped in the coronal plane.

**Lateral-then-forward (L-F) sampling** – A strategy for constructing a model to balance one step of a walk. L-F sampling is similar to F-L sampling except that the biped is balanced first in the coronal plane and then in the sagittal plane.

## APPENDIX B – MODEL PARAMETER SCRIPTS

```

#-----
#
# file: human.bones
#
# Skeleton for human model with feet, arms and 2-link torso.
# All joints are 1 DOF, except for hips (3 DOF) and ankles (2 DOF).
# Mass and inertia data from Wooten & Hodgens
#
# Body axes:  forward -> -1,0,0   (-ve x-axis)
#              right  ->  0,0,-1   (-ve z-axis)
#              up     ->  0,1,0    (+ve y-axis)
#
# Joints:  (z - pitch, x - roll, y - yaw)
#  waist-z, spine-z, neck-z, lhip-x, lhip-z, lhip-y, lknee-z, lankle-z, lankle-x
#              rhip-x, rhip-z, rhip-y, rknee-z, rankle-z, rankle-x
#              lshoulder-z, lelbow-z, rshoulder-z, relbow-z
#
#
# Model construction commands:
# -----
#
# bone <#>
#     Creatues a new bone.  Bones can have many rigidly attached segments.
# seg <x1>, <y1>, <z1> <x2>, <y2>, <z2>
#     Adds a segment to the current bone.
# cofm <x> <y> <z> <rx> <ry> <rz>
#     Specifies the centre of mass for the current bone.
# mass <mass>
#     Specifies the mass for the current bone.
# iner <ix> <iy> <iz>
#     Specifies the moment of inertia for the current bone about major axes
# mon <x>, <y>, <z>
#     Specifies a monitor point
# hinge <bone#> <x>, <y>, <z> <ax>, <ay>, <az>
#     Specifies a 1 DOF hinge at x,y,z with axis ax,ay,az
# uhinge and ghinge
#     Similar to hinge but specify 2 DOF and 3 DOF respectively
#     and take 2 and 3 hinge axis arguments respectively.
# kpd hinge <parent bone #>:<DOF#> <Kp> <Kd>
# mon <x>, <y>, <z>
#     Adds a new monitor point at x,y,z.  Monitor points are points of
#     interest on the creature.  They are the only parts of a creature
#     which are experience ground forces.
#
#-----

### STRUCTURE + LINK PARAMETERS -----

bone 1                                # pelvis
seg 0,0.95,0.12 0,0.95,-0.12         # - add a bone segment
mass 16.61                             # - bone mass
iner 0.23 0.16 0.18                   # - major axis inertias
cofm 0 0.98 0                          # - bone COM position

```

```

bone 2                                     # lower torso
seg 0,1.02,0 0,1.35,0
mass 19.5133
iner 0.4867 0.17427 0.42
cofm -0.03 1.3112 0
hinge 1 0,1.0202,0 0,0,1                 # waist - 1 DOF (pitch)

bone 3                                     # upper torso
seg 0,1.35,0 0,1.505,0
mass 9.7567
iner 0.2433 0.08713 0.21
cofm -0.03 1.4275 0
hinge 2 0,1.35,0 0,0,1                 # spine - 1 DOF (pitch)

bone 4                                     # head
seg 0.02,1.57,0 0.02,1.80,0
mass 5.89
iner 0.03 0.023 0.033
cofm -0.01 1.70 0
hinge 3 0.0478,1.5052,0 0,0,1          # neck - 1 DOF (pitch)

bone 5                                     # left upper leg
seg 0.00478,0.9629,0.0932 0.00478,0.5197,0.09091
mass 8.35
iner 0.15 0.025 0.16
cofm 0 0.80 0.09
ghinge 1 0.00478,0.9629,0.0932 1,0,0 0,0,1 0,1,0 # hip - 3 DOF
# (roll, pitch, yaw)

bone 6                                     # left lower leg
seg 0.00478,0.5197,0.09091 0.03348,0.09164,0.0837
mass 4.16
iner 0.055 0.007 0.056
cofm 0 0.35 0.07
hinge 5 0.00478,0.5197,0.09091 0,0,1    # knee - 1 DOF (pitch)

bone 7                                     # left foot
seg -0.10,0,0.0837 0.07,0,0.0837
mass 1.20
iner 0.0018 0.0070 0.0075
cofm -0.03 0.02 0.08
uhinge 6 0.03348,0.09164,0.0837 0,0,1 1,0,0 # ankle - 2 DOF
# (pitch, roll)

bone 8                                     # right upper leg
seg 0.00478,0.9629,-0.0932 0.00478,0.5197,-0.09091
mass 8.35
iner 0.15 0.025 0.16
cofm 0 0.80 -0.09
ghinge 1 0.00478,0.9629,-0.0932 1,0,0 0,0,1 0,1,0 # hip - 3 DOF
# (roll, pitch, yaw)

bone 9                                     # right lower leg
seg 0.00478,0.5197,-0.09091 0.03348,0.09164,-0.0837
mass 4.16
iner 0.055 0.007 0.056
cofm 0 0.35 -0.07
hinge 8 0.00478,0.5197,-0.09091 0,0,1    # knee - 1 DOF (pitch)

bone 10                                    # right foot
seg -0.10,0,-0.0837 0.07,0,-0.0837
mass 1.20
iner 0.0018 0.0070 0.0075
cofm -0.03 0.02 -0.08

```

```

uhinge 9 0.03348,0.09164,-0.0837 0,0,1 1,0,0 # ankle - 2 DOF ankle
# (pitch, roll)

bone 11 # left upper arm
seg 0.0289,1.4203,0.1847 0.0287,1.1491,0.2102
mass 2.79
iner 0.025 0.005 0.025
cofm 0 1.34 0.21
hinge 3 0.02,1.4203,0.1847 0,0,1 # shoulder - 1 DOF (pitch)

bone 12 # left lower arm
seg 0.0287,1.1491,0.2102 0.0287,0.85,0.2302
mass 1.761
iner 0.0066 0.0018 0.0080
cofm 0.0287 1.03 0.222
hinge 11 0.0287,1.1491,0.2102 0,0,1 # elbow - 1 DOF (pitch)

bone 13 # right upper arm
seg 0.0289,1.4203,-0.1847 0.0287,1.1491,-0.2102
mass 2.79
iner 0.025 0.005 0.025
cofm 0 1.34 -0.21
hinge 3 0.02,1.4203,-0.1847 0,0,1 # shoulder - 1 DOF (pitch)

bone 14 # right lower arm
seg 0.0287,1.1491,-0.2102 0.0287,0.85,-0.2302
mass 1.761
iner 0.0066 0.0018 0.0080
cofm 0.0287 1.03 -0.222
hinge 13 0.0287,1.1491,-0.2102 0,0,1 # elbow - 1 DOF (pitch)

### JOINT STRENGTH (Kp & Kd) PARAMETERS -----
kpd hinge 2:0 2000 100 # waist pitch (z)
kpd hinge 3:0 2000 100 # spine pitch (z)
kpd hinge 4:0 100 10 # neck pitch (z)

kpd hinge 5:0 900 30 # left hip roll (x)
kpd hinge 5:1 1275 60 # left hip pitch (z)
kpd hinge 5:2 1275 60 # left hip yaw (y)

kpd hinge 6:0 1275 60 # left knee pitch (z)

kpd hinge 7:0 170 17 # left ankle pitch (z)
kpd hinge 7:1 170 17 # left ankle roll (x)

kpd hinge 8:0 900 30 # right hip roll (x)
kpd hinge 8:1 1275 60 # right hip pitch (z)
kpd hinge 8:2 1275 60 # right hip yaw (y)

kpd hinge 9:0 1275 60 # right knee pitch (z)

kpd hinge 10:0 170 17 # right ankle pitch (z)
kpd hinge 10:1 170 17 # right ankle roll (x)

kpd hinge 11:0 100 10 # left shoulder pitch (z)
kpd hinge 12:0 50 5 # left elbow pitch (z)

kpd hinge 13:0 100 10 # right shoulder pitch (z)
kpd hinge 14:0 50 5 # right elbow pitch (z)

```

```

### MONITOR POINTS -----
# up vector monitor
# points (lower torso):
# 1 - origin
# 2 - "up"
# 3 - "right"
mon 0,1.1,0
mon 0,1.3,0
mon 0,1.1,-.05

# swing/stance COM points:
# 4 - left mid-foot
# 5 - right mid-foot
mon 0.03348,0,0.0837
mon 0.03348,0,-0.0837

# left foot:
# 6 - inside ball of foot
# 7 - inside heel
# 8 - outside heel
# 9 - little toe
mon -0.10,0,0.05
mon 0.07,0,0.05
mon 0.07,0,0.13
mon -0.10,0,0.13

# right foot:
# 10 - inside ball of foot
# 11 - inside heel
# 12 - outside heel
# 13 - little toe
mon -0.10,0,-0.05
mon 0.07,0,-0.05
mon 0.07,0,-0.13
mon -0.10,0,-0.13

# 14 - left knee
# 15 - right knee
mon 0.00478,0.5197,0.09091
mon 0.00478,0.5197,-0.09091

# 16 - head
mon 0.02,1.80,0

# pelvis-based up vector &
# torso servo mon points:
# 17 - origin
# 18 - "up"
# 19 - "right"
# 20 - servo axis (local)
# 21 - torso axis
mon 0,0.95,0
mon 0,0.97,0
mon 0,0.95,-.02
mon 0,0.95,+.02
mon 0,1.1,+.05

set up_mon 1,2,3
set cofm_mon 5,4
# up vect mon points
# initial swing,stance feet

```

```

#-----
#
# file: robo-bird.bones
#
# Skeleton for bird-like robot model with feet, body and large head.
# All joints are 1 DOF, except for hips (2 DOF) and ankles (2 DOF).
# Mass and inertia parameters calculated automatically using density of
# 1.0 g/cm^3
#
# Body axes:  forward -> -1,0,0   (-ve x-axis)
#             right  ->  0,0,-1   (-ve z-axis)
#             up     ->  0,1,0    (+ve y-axis)
#
# Joints:    (z - pitch, x - roll, y - yaw)
# neck-z, rhip-x, rhip-z, rknee1-z, rknee2-z, rknee3-z, rankle-z, rankle-x
#           lhip-x, lhip-z, lknee1-z, lknee2-z, lknee3-z, lankle-z, lankle-x
#
#-----

### STRUCTURE + LINK PARAMETERS -----

# the words "left" and "right" may be backwards in the descriptions below.

bone 1                                # body
  # x-aligned segments
  seg 1.0,-0.4,-1.5 -4,-0.4,-1.5
  seg 1.0,-0.4,1.5 -4,-0.4,1.5
  seg 1.0,0.8,-1.5 -4,0.8,-1.5
  seg 1.0,0.8,1.5 -4,0.8,1.5
  # z-aligned segments
  seg 1.0,0.8,-1.5 1.0,0.8,1.5
  seg 1.0,-0.4,-1.5 1.0,-0.4,1.5
  seg -4,-0.4,-1.5 -4,-0.4,1.5
  seg -4,0.8,-1.5 -4,0.8,1.5
  # y-aligned segments
  seg 1.0,-0.4,-1.5 1.0,0.8,-1.5
  seg 1.0,-0.4,1.5 1.0,0.8,1.5
  seg -4,-0.4,-1.5 -4,0.8,-1.5
  seg -4,-0.4,1.5 -4,0.8,1.5

bone 2                                # head
  # x-aligned segments
  seg 0.5,1.5,-1.75 -6.0,1.5,-0.9
  seg 0.5,1.5,1.75 -6.0,1.5,0.9
  seg 0.5,5.5,-2.25 -4.5,5.5,-1.25
  seg 0.5,5.5,2.25 -4.5,5.5,1.25
  # z-aligned segments
  seg 0.5,1.5,-1.75 0.5,1.5,1.75
  seg 0.5,5.5,-2.25 0.5,5.5,2.25
  seg -6.0,1.5,-0.9 -6.0,1.5,0.9
  seg -4.5,5.5,-1.25 -4.5,5.5,1.25
  # y-aligned segments
  seg 0.5,1.5,-1.75 0.5,5.5,-2.25
  seg 0.5,1.5,1.75 0.5,5.5,2.25
  seg -6.0,1.5,-0.9 -4.5,5.5,-1.25
  seg -6.0,1.5,0.9 -4.5,5.5,1.25
  hinge 1 -2.3,1.0,0 0,1,0           # neck - 1 DOF (yaw)

bone 3                                # right leg bone1 (upper)
  seg 0,0,-2.5 -3,0,-2.5
  uhinge 1 0,0,-2.5 1,0,0 0,0,1    # hip - 2 DOF
                                     # (roll, pitch)

```

```

bone 4                                     # right leg bone2 (mid1)
seg -3,0,-2.5 -3,-4,-2.5
hinge 3 -3,0,-2.5 0,0,1                   # kneel - 1 DOF (pitch)

bone 5                                     # right leg bone3 (mid2)
seg -3,-4,-2.5 -3,-9.5,-2.5
hinge 4 -3,-4,-2.5 0,0,1                   # kneel2 - 1 DOF (pitch)

bone 6                                     # right leg bone4 (lower)
seg -4.8,-9.5,-2.5 -1.3,-9.5,-2.5
hinge 5 -3,-9.5,-2.5 0,0,1                 # kneel3 - 1 DOF (pitch)

bone 7                                     # right foot
seg -4.1,-10.4,-2.5 -7.6,-10.4,-2.5
seg -5.6,-10.4,-3.9 -5.6,-10.4,-1.1
uhinge 6 -4.8,-9.5,-2.5 0,0,1 1,0,0      # ankle - 2 DOF
#                                         # (pitch, roll)

bone 8                                     # left leg bone1 (upper)
seg 0,0,2.4 -3,0,2.4
uhinge 1 0,0,2.5 -1,0,0 0,0,1             # hip - 2 DOF
#                                         # (roll, pitch)

bone 9                                     # right leg bone2 (mid1)
seg -3,0,2.5 -3,-4,2.5
hinge 8 -3,0,2.5 0,0,1                     # kneel - 1 DOF (pitch)

bone 10                                    # left leg bone3 (mid2)
seg -3,-4,2.5 -3,-9.5,2.5
hinge 9 -3,-4,2.5 0,0,1                     # kneel2 - 1 DOF (pitch)

bone 11                                    # left leg bone4 (lower)
seg -4.8,-9.5,2.5 -1.3,-9.5,2.5
hinge 10 -3,-9.5,2.5 0,0,1                 # kneel3 - 1 DOF (pitch)

bone 12                                    # left foot
seg -4.1,-10.4,2.5 -7.6,-10.4,2.5
seg -5.6,-10.4,1.1 -5.6,-10.4,3.9
uhinge 11 -4.8,-9.5,2.5 0,0,1 1,0,0      # ankle - 2 DOF
#                                         # (pitch, roll)

### JOINT STRENGTH PARAMETERS -----
kpd hinge 2      1000 33                    # neck pitch (z)

kpd hinge 3:0    1000 33                    # right hip roll (x)
kpd hinge 3:1    1000 33                    # right hip pitch (z)

kpd hinge 4      1000 33                    # right kneel pitch (z)
kpd hinge 5      1000 33                    # right kneel2 pitch (z)
kpd hinge 6      1000 33                    # right kneel3 pitch (z)

kpd hinge 7:0    100  1.5                   # right ankle pitch (z)
kpd hinge 7:1    100  1.5                   # right ankle roll (x)

kpd hinge 8:0    1000 33                    # left hip roll (x)
kpd hinge 8:1    1000 33                    # left hip pitch (z)

kpd hinge 9      1000 33                    # left kneel pitch (z)
kpd hinge 10     1000 33                    # left kneel2 pitch (z)
kpd hinge 11     1000 33                    # left kneel3 pitch (z)

```

```

kpd hinge 12:0 100 1.5 # left ankle pitch (z)
kpd hinge 12:1 100 1.5 # left ankle roll (x)

### MONITOR POINTS -----

mon -3.8,-10.4,-3.9 # right foot
mon -3.8,-10.4,-1.1 # 1 - right/back
mon -7.6,-10.4,-3.9 # 2 - left/back
mon -7.6,-10.4,-1.1 # 3 - right/front
# 4 - left/back

mon -3.8,-10.4,3.9 # left foot
mon -3.8,-10.4,1.1 # 5 - left/back
mon -7.6,-10.4,3.9 # 6 - right/back
mon -7.6,-10.4,1.1 # 7 - left/front
# 8 - right/front

mon -5.5,2,0 # falling & up vect mons:
mon 1,2,0 # 9 - nose
mon 1,3,0 # 10 - up vect origin
mon 2,2,0 # 11 - up vect "up"
mon 1,2,0 # 12 - up vect "right"
# 13 - body

mon -4.8,-10.4,-2.5 # swing/stance COM points:
mon -4.8,-10.4,2.5 # 14 - right foot COM mon
# 15 - left foot COM mon

set up_mon 10,11,13 # up vect monitor points
set cofm_mon 15,14 # initial swing,stance feet

kpd mon 14 0 0 # no gnd force on cofm_mon
kpd mon 15 0 0 # no gnd force on cofm_mon

```

## APPENDIX C – SAMPLE ANIMATION SCRIPT

```

#-----
#
# file: human-walk-trials.script
#
# Script to perform basic walking trials
#
# Perform balance trials with up vector RVs
# Trials use Q_d from [0.1,0] to [0.5,0] (Q_d = [Q_d_fwd,Q_d_lat])
#
# balance <#strides> <Q_tol1> <K1_max> <K1_sample> <RV1>:<elem1> \
#           <Q_tol2> <K2_max> <K2_sample> <RV2>:<elem2>
#
#   <max_strides>    - desired # of strides
#
#   <Q1_tol>         - max. tolerable error in Q for 1st control dimension
#   <K1_max>         - max final perturbation scaling (dim 1)
#   <K1_sample>      - sample scalings (+/- K1_sample) (dim 1)
#   <RV>:<elem>      - RV type and component
#                     type:
#                       0 -> up vector
#                       1 -> stance-COM vector
#                       2 -> swing-COM vector
#
#                     component of RV
#                       0 -> forward component
#                       1 -> lateral component
#
#   <Q_tol2>, <K2_max>, <K2_sample>, <RV2>, <elem2> as above but for 2nd
#   control dimension.
#-----

< human.bones

set kfspring 25000          # floor stiffness (Kp)
set kfdamp 1500            # floor stiffness (Kd)

set sddt 0.00025          # simulation time-step

# base PCG for walking

set pcstates 6

posecycle poses 0
5 0 0 0 -50 0 60 -5 0 0 -10 0 0 0 0 7 0 -3 0 5 5 # r-foot
5 0 0 0 -50 0 60 -5 0 0 -10 0 0 0 0 7 0 -12 0 .2
5 0 0 0 -20 0 0 5 0 0 -10 0 0 0 0 7 0 -12 -7 .2
5 0 0 0 -10 0 0 0 0 0 -50 0 60 -5 0 -3 0 7 0 4 5 # l-foot
5 0 0 0 -10 0 0 0 0 0 -50 0 60 -5 0 -12 0 7 0 .2
5 0 0 0 -10 0 0 0 0 0 -20 0 0 5 0 -12 -7 7 0 .2

set hinit 0,0,0,0,0,0,0,0,0,-5.1,-20,0,50,0,0,0,0,0 # initial pose
# all velocities are 0

```

```

# ----- forward then lateral (F-L) 2d control

# hip pitch perturbations:

posecycle poses 1                                     # right hip pitch
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

posecycle poses 2                                     # left hip pitch
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

# hip roll perturbations:

posecycle poses 3                                     # right hip roll
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

posecycle poses 4                                     # left hip roll
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

# balance trials

set showfile basic-walk_upvect_FL_Q=[.1,0].out
set Q_d 0.1,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.15,0].out
set Q_d 0.15,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.2,0].out
set Q_d 0.2,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.25,0].out
set Q_d 0.25,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.3,0].out
set Q_d 0.3,0

```

```

balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.35,0].out
set Q_d 0.35,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.4,0].out
set Q_d 0.4,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.45,0].out
set Q_d 0.45,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_FL_Q=[.5,0].out
set Q_d 0.5,0
balance 30 1 180 5 0:0 1 180 1 0:1

# ----- superposition (SP) 2d control
#
# Use same perturbations as F-L since order of
# perturbations doesn't matter for superpos.

# balance trials

set superposition true

set showfile basic-walk_upvect_SP_Q=[.1,0].out
set Q_d 0.1,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.15,0].out
set Q_d 0.15,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.2,0].out
set Q_d 0.2,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.25,0].out
set Q_d 0.25,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.3,0].out
set Q_d 0.3,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.35,0].out
set Q_d 0.35,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.4,0].out
set Q_d 0.4,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.45,0].out
set Q_d 0.45,0
balance 30 1 180 5 0:0 1 180 1 0:1

set showfile basic-walk_upvect_SP_Q=[.5,0].out
set Q_d 0.5,0

```

```

balance 30 1 180 5 0:0 1 180 1 0:1

set superposition false

# ----- lateral then forward (L-F) 2D control

# hip roll perturbations:

posecycle poses 1                                     # right hip roll
0 0 0    0 0 0 0 0 0    1 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    1 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    1 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0

posecycle poses 2                                     # left hip roll
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    1 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    1 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    1 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0

# hip roll perturbations:

posecycle poses 3                                     # right hip pitch
0 0 0    0 0 0 0 0 0    0 1 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 1 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 1 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0

posecycle poses 4                                     # left hip pitch
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 0 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 1 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 1 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0
0 0 0    0 1 0 0 0 0    0 0 0 0 0 0    0 0    0 0    0

# balance trials

set showfile basic-walk_upvect_LF_Q=[.1,0].out
set Q_d 0.1,0
balance 30 1 180 1 0:1 1 180 5 0:0

set showfile tbasic-walk_upvect_LF_Q=[.15,0].out
set Q_d 0.15,0
balance 30 1 180 1 0:1 1 180 5 0:0

set showfile basic-walk_upvect_LF_Q=[.2,0].out
set Q_d 0.2,0
balance 30 1 180 1 0:1 1 180 5 0:0

set showfile basic-walk_upvect_LF_Q=[.25,0].out
set Q_d 0.25,0
balance 30 1 180 1 0:1 1 180 5 0:0

```

```
set showfile tbasic-walk_upvect_LF_Q=[.3,0].out
set Q_d 0.3,0
balance 30 1 180 1 0:1 1 180 5 0:0
```

```
set showfile basic-walk_upvect_LF_Q=[.35,0].out
set Q_d 0.35,0
balance 30 1 180 1 0:1 1 180 5 0:0
```

```
set showfile basic-walk_upvect_LF_Q=[.4,0].out
set Q_d 0.4,0
balance 30 1 180 1 0:1 1 180 5 0:0
```

```
set showfile basic-walk_upvect_LF_Q=[.45,0].out
set Q_d 0.45,0
balance 30 1 180 1 0:1 1 180 5 0:0
```

```
set showfile basic-walk_upvect_LF_Q=[.5,0].out
set Q_d 0.5,0
balance 30 1 180 1 0:1 1 180 5 0:0
```