Newcastle University
School of Computing Science

Real-Time Biped Character Stepping

Dissertation submitted in partial fulfillment
of the requirement for the degree of

Doctor of Philosophy

in

Computer Science

by

Benjamin Kenwright

January 2014

Supervisors:

Dr. Graham Morgan
Dr. Peter Andras

The Dissertation of Benjamin Kenwright is approved:

Newcastle University

Abstract of the Dissertation

Real-Time Biped Character Stepping

by

Benjamin Kenwright

Doctor of Philosophy, School of Computer Science
Newcastle University, January 2014

A rudimentary biped activity that is essential in interactive virtual worlds, such as video-games and training simulations, is stepping. For example, stepping is fundamental in everyday terrestrial activities that include walking and balance recovery. Therefore, an effective 3D stepping control algorithm that is computationally fast and easy to implement is extremely valuable and important to character animation research. This thesis focuses on generating real-time controllable stepping motions on-the-fly without key-framed data that are responsive and robust (e.g., can remain upright and balanced under a variety of conditions, such as pushes and dynamically changing terrain). In our approach, we control the character's direction and speed by means of varying the step position and duration. Our lightweight stepping model is used to create coordinated full-body motions, which produces directable steps to guide the character with specific goals (e.g., following a particular path while placing feet at viable locations). We also create protective steps in response to random disturbances (e.g., pushes). Whereby, the system automatically calculates where and when to place the foot to remedy the disruption. In conclusion, the inverted pendulum has a number of limitations that we address and resolve to produce an improved lightweight technique that provides better control and stability using approximate feature enhancements, for instance, ankle-torque and elongated-body.

Contents

List of Figures **xi**

1 Introduction **1**
- 1.1 Motivation . . . 3
- 1.2 Contributions . . . 4
- 1.3 Dissertation Outline (i.e., Road-Map) . . . 5
- 1.4 Publications . . . 6

2 Background **7**
- 2.1 Data-Driven Solutions . . . 8
- 2.2 Optimized Search . . . 10
- 2.3 Controlled Physics-Based Characters . . . 12
- 2.4 Balanced Biped Models . . . 14
- 2.5 Moving Away From Data-Driven Solutions . . . 16
- 2.6 Summary . . . 18

3 Inverted Pendulum (IP): A Low-Level Approach **19**
- 3.1 Introduction . . . 19
  - 3.1.1 Chapter Contribution . . . 21
- 3.2 Related Work . . . 22
- 3.3 What is the Inverted Pendulum (IP)? . . . 23
- 3.4 Motion Fundamentals . . . 24
  - 3.4.1 Inverted Pendulum Principle . . . 24
  - 3.4.2 Mathematics and Control . . . 25
  - 3.4.3 Step Direction . . . 28
  - 3.4.4 Walk Velocity . . . 34
    - 3.4.4.1 Constant Walk Velocity . . . 34
    - 3.4.4.2 Varying Walk Velocity . . . 34
  - 3.4.5 Disturbances . . . 34
  - 3.4.6 Non-Flat Ground . . . 35
    - 3.4.6.1 Slopping Terrain . . . 35
    - 3.4.6.2 Peg-Leg . . . 35
    - 3.4.6.3 Random Ground Height Deviations . . . 35
  - 3.4.7 Pulling-Pushing Objects . . . 35
  - 3.4.8 Terrain Gaps . . . 37
- 3.5 2D to 3D (Sagittal and Coronal Planes) . . . 37
  - 3.5.1 Simulation Analysis . . . 41
- 3.6 Discussion . . . 41
- 3.7 Conclusion . . . 42

4 Real-Time Dynamic Balancing and Walking: Elongated Rigid-Body 43
4.1 Introduction . . . 43
4.1.1 Chapter Contribution . . . 44
4.2 Related Work . . . 45
4.2.1 Inverted Pendulum . . . 45
4.3 Overview . . . 46
4.3.1 IP Mechanics . . . 46
4.3.2 Mechanics of Enhanced IP . . . 47
4.3.3 Calculations . . . 47
4.3.4 Managing Stairs and Slopes . . . 48
4.3.5 Controllable Motion . . . 49
4.4 Results . . . 51
4.4.1 Steering and Speed . . . 51
4.4.2 Robustness to Pushes . . . 52
4.5 Discussion . . . 53
4.6 Conclusion and Future Work . . . 53

5 Responsive Stepping: Foot Control 55
5.1 Introduction . . . 55
5.1.1 Chapter Contribution . . . 57
5.2 Related Work . . . 58
5.3 Overview . . . 60
5.4 Orientation and Feet . . . 61
5.4.1 Do We Need Feet? . . . 62
5.4.1.1 Determining the Support Polygon . . . 62
5.4.1.2 Feet Location Comfort Factor . . . 62
5.4.1.3 Foot Placement Constraints . . . 63
5.4.2 Body Orientation . . . 63
5.5 Base-Controller . . . 64
5.5.1 Controller Constraints . . . 65
5.5.2 Foot Swing . . . 66
5.5.3 Ankle Torque Feedback Control . . . 69
5.6 Static and Dynamic Balancing . . . 70
5.7 Passive and Active Walking . . . 71
5.7.1 Static and Dynamic Walking . . . 72
5.8 Foot Logic . . . 72
5.9 Inverse Kinematics . . . 74
5.9.1 Biped Model . . . 76
5.9.2 Articulated Rigid-Body Control . . . 76
5.10 Results . . . 79
5.11 Limitations . . . 80
5.12 Discussion . . . 81
5.13 Conclusion and Future Work . . . 82

6 Stepping Framework: Bringing it all Together 85
6.1 Introduction . . . 86
6.1.1 Chapter Contribution . . . 87
6.2 Overview . . . 89
6.3 Pendulum Stepping Model (Capture Height) . . . 91
6.3.1 Capture-Point . . . 93
6.3.2 Control Mechanisms . . . 94

6.3.3 Mapping: Bridging the Gap between Control and Kinematics (IP to Full-Body) 95
6.4 Results 96
6.5 Discussion 100
6.6 Conclusion 100

7 Conclusion **103**
7.1 Flexible Practical Approach 104
7.2 Natural Dynamics 104
7.3 Beyond Key-Frame Animations 104
7.4 Discussion 105
7.5 Summary and Future Work 105

A Appendix **109**
A.1 Articulated Skeleton Joint Parameters 109

References **111**

List of Figures

2.1 Model and Computational Complexity - Reducing the articulated model's complexity down to its fundamental components to solve targeted motions using low-dimensional models that can be less ambiguous and straightforward to solve both mathematically and computationally. 15

3.1 Time-Line - Research time-line focused on the inverted pendulum in terms of character-based solutions (e.g., biped balancing) over the past few decades (biomechanics, robotics, and graphics). [A] [VS72]; [B] [HGJ77]; [C] [MS84a]; [D] [KT91]; [E] [KT95]; [F] [PK98]; [G] [CPP99]; [H] [KKK*01a]; [I] [KKK*02]; [J] [KKK*03]; [K] [YLvdP07]; [L] [CBvdP10, MdLH10, TLC*10]; [M] [KDM11]; [N] [GOAS12]; [O] [Ken12a, Ken12d]. 21

3.2 Simplified Inverted Pendulum (IP) Biped Model Dynamics - The IP model provides key balancing information based on the foot placement and centre-of-mass location. 21

3.3 Fundamental Inverted Pendulum Stepping Logic - The continuous stepping motion for a simple IP model in ideal conditions by swapping the leg at a specific angle (i.e., we assume the IP begins with a constant velocity that it maintains through stepping). 25

3.4 Fundamental Inverted Pendulum Step Transition Paramaters - At any moment we can take a snap shot of the current IP model's paramaters (e.g., velocity, height) and use them to calculate the location of the new foot position to achieve the desired control (i.e., the velocity in the steered direction). 28

3.5 Stepping Direction - The body velocity and the direction from the foot position to the body, we can determine if we need to place a foot forward or backward to remain balanced. 29

3.6 Walking Up Stairs - Extending the basic model to include deviation for changes in terrain height (e.g., uneven terrain, such as stairs or sloping ground). The leg-lengths are extended back to their rest length when the centre-of-mass is above the support foot. 30

3.7 Walking Up Stairs - When walking up stairs, we vary the leg-length to mimic an active leg muscle. The leg-length is re-extended when it is vertically above the support foot (i.e., analogous to leg muscle actively pushing the body upwards for each step). 30

3.8 Steady Walk Simulation - The figure shows the centre-of-mass velocity step transitions, the velocity reducing and increasing for each step. (a) Walking at a steady constant pace, (b) following expected walk velocity (speeding up, then slowing down); (with time-step=0.01, mass=70kg, and leg-length=1.0m). The important information to recognize is that the velocity does not increase or decrease gradually - even when we alter the desired velocity to increase or decrease the walking speed. 31

3.9 Flat, Up-Hill, and Down-Hill Velocity - The velocity when walking on flat, up-hill, and down-hill ground, with (a) showing the centre-of-mass and ground height, and (b) the corresponding velocity. (time-step=0.01, mass=70kg, leg-length=1.0m, terrain height increment/decrement=0.1m, average step length 0.2m). . . . 31
3.10 Disturbance Force Applied While Walking - (a) Biped walking velocity during push force disturbances, and (b) disturbance force magnitude, direction, and time; (with time-step=0.01, mass=70kg, and leg-length=1.0m). . . . 32
3.11 Terrain Ground Gap Stepping - The model has to choose a suitable foot position or let the body continue moving forwards. If left to move, the model can re-attempt to find a more desirable foot position that is more adequate for preventing the character from falling into the gaps (Figure 3.12 shows the simulation graphs). . . . 32
3.12 Terrain Ground Gaps Simulation - (a) The height of the body and the terrain, and (b) the velocity of the body (gap 0.01m to 0.1m); (with time-step=0.01, mass=70kg, and leg-length=1.0m) . . . 33
3.13 Evaluating the Capabilities of the Stepping Model - Walking at a controllable speed under a variety of conditions, such as pulling/pushing forces and terrains height deviations that include inclines (approx. 25 degrees) and gaps (approx. 0.2m). (a) Flat terrain, (b), sloping terrain (e.g., stairs and hills), (c), uneven ground, (d) constant pulling or pushing force, and (e) terrain gaps. . . . 33
3.14 Pulling or Pushing an Object - The IP stepping equation can be modified so that it can maintain a balanced walking momentum while pull or push force is being applied to the body (i.e., simulating a pushing or pulling action). . . . 37
3.15 Constant Pulling Force (i.e., Pulling an Object) - A constant pulling force is applied to the body of varying magnitude (0, 10N, 100N) at different times. The simulation shows the velocity is relatively constant while stepping distances are reduced to compensate for the external pulling force. . . . 38
3.16 Pull Force vs Stepping Distance - Stepping distance varies with different constant pulling force (with leg-length=1.0m, mass=50kg,70kg,90kg,150kg, time-step=0.01). When stepping distance is 0 the pendulum is vertical and the foot must remain behind the centre-of-mass to keep the pendulum walking forward. Initially, when starting, the pendulum is pulled and needs to take a backward step before moving forwards. The step limit specifies the foot range - beyond which we are constantly pulled backwards if we cannot place our foot. As expected, increasing the mass shows we can pull more force due to gravity. For the unstable oscillating motion for the 50kg mass - this was due to the pendulum stepping back and forth at the threshold to counteract the pull force. . . . 38
3.17 Push Force vs Stepping Distance - Stepping distance varies with different constant pushing force (with leg-length=1.0m, mass=50kg,70kg,90kg,150kg, time-step=0.01). To counteract the push force the stepping distance has to increase. Limiting the maximum allowable step distance due to physical constraints (i.e., leg-length) as shown on the graph as step limit (allowable range before we are unable to counteract the push force and need to take a step backwards). . . . 39
3.18 3D Inverted Pendulum Movement - We can apply the basic 2D IP model to 3D. We divide the stepping movement between two perpendicular vectors (i.e., forwards and sideways). We can then control the forward and sideways speed (e.g., for steering). The figure shows the forward direction always facing towards the closes point on a circular path. The IP model continues to follow the path at a constant velocity. . . . 39

3.19 Simulation of Leg-Length vs Stepping Distance - We show the linear relationship for different leg-lengths (0.7m-1.3m) with stepping distance for a constant walking velocity on flat terrain. As expected, the feedback system corrects for minor errors from numerical approximations (i.e., minor oscillates around the ideal stepping distance $\delta d \approx 0.02$) to speed-up or slow-down so the final walking velocity is constant and stable. (with time-step=0.01, desired velocity 0.7m/s, and mass=70kg) . . . 40
3.20 Simulation of Force Disturbance vs Stepping Distance - We show the relationship of push disturbances with stepping distance. Where (a) shows the relationship for force and duration, and (b) the impulse force (i.e., force multiplied with the time) for the necessary stepping distance to bring the pendulum to a stop (i.e., upright with velocity of zero); We draw a dotted line to point out stepping limits (i.e., the maximum distance the pendulum can step based on leg-length deviations). (with time-step=0.01, leg-length=1.0m, and mass=70kg) 40

4.1 Adaptive Stepping - Predictive stepping and posture correction due to random force disturbances being applied to the body. . . . 43
4.2 Pendulum Motion - (Left) Illustrates IP following a basic elliptical trajectory with no body feedback, (Right) the elongated body following a similar ellipsoid-like trajectory during locomotion, with additional torque-body feedback. 47
4.3 Virtual Pivot Point Concept and Reasoning - A visual aid to help clarify the underlying principle of the Virtual Pivot Point (VPP) - consider a stepping cycle. The base and the centre of mass provide a means of calculating a control torque for the base to sway back and forth. However, we need to include the ground reaction force (GRF) magnitude and a Virtual Pivot Point (VPP) for multiple reasons - firstly to ensure the swaying motion is based on the physical situation (i.e., how hard we land on the ground with our foot) and a means of controlling the postural motion. . . . 48
4.4 Hip Torque - Hip torque calculated using the COP and VPP. . . . 49
4.5 Slopes and Stairs - Walking up stairs/steps or a gradual slope. . . . 49
4.6 Walking Up Slope - Offsetting the VPP enables the character to climbing gradual slopes. The screen capture shows a slope of approx 6 degrees with the VPP horizontal offset at 15cm. However, if the slope is too steep or the VPP is offset too much, it can lead to instability issues (i.e. oscillating motion leading to sharp pelvic torques so the character needs to take a step back to regain balance). 50
4.7 Steering - The Virtual Pivot Point (VPP) mechanism introduces an additional control feature. In 2D, as shown in (a), if we move the VPP outside the trunk we can add or remove energy from the system to increase or decrease the walking speed. However, as shown in (b), in 3D, this feature can also be used to steer the character left or right. . . . 50
4.8 Elongated Body Model - The enhanced inverted pendulum generates reference motions (feet, hip positions, and body orientation) that we use to control our five-link biped. . . . 51
4.9 Posture Steering - Alter the VPP-offset to control the steering locomotion (i.e., the arrow in the figure indicates the desired steering direction). The character follows the spline (i.e., steers to the left or right) using a penalty-based approach, such that we measure the error between the current position and the desired and offset the VPP accordingly to steer in that direction. . . . 52
4.10 Push Disturbances - Incremental screenshots of the biped model responding to a 100N push force being applied to the COM for 0.1s. . . . 52

5.1 Push Response - Illustrate the basic model standing before being pushed twice and having to take corrective steps before returning to a relaxed stance. 55
5.2 Inverted Pendulum Principle - The human-leg is analogous to a spring-damper system. We can represent the overall character's centre-of-mass and leg using an inverted pendulum system. The leg spring-damper force can be calculated using an uncomplicated penalty-based technique - we configure the parameters to produce a relatively stiff-knee joint which is desired during casual stepping motions. 60
5.3 Unnatural and Uncomfortable Stance Pose - Uncomfortable poses can arise during stepping, which are physically accurate and balancing yet look awkward and unnatural. For example, (a) and (b) show the character cross-legged and (c) shows the character with one foot in front of the other. 63
5.4 Foot Support Area - Inverted pendulum model showing the approximated circular shape foot support regions and the combined overall capsule shaped support region. 64
5.5 Reference Pelvis - Illustrating the comfortable positions the feet will return to after a disturbance or when idling. 65
5.6 Ankle Torque & Foot Trajectory - (a) Common foot swing trajectory for a human casually walking; and (b) ankle torque can provide minor steering and control abilities for the upper body (e.g., standing without requiring to constantly step). 66
5.7 Step Trajectory - Foot stepping trajectories are calculated using Bezier curves so they appear more natural, (a) trajectory paths for a simple forward step, and (b) blended motions following the trajectory. 67
5.8 Foot and Ankle Trajectory Correlation - As the foot transitions along its spline we orientate the foot based upon the distance along the route and the height from the ground. (a) Foot following symmetrical sinusoidal path, (b) while we can mimic more esoteric paths with dips that closely match a human by combining multiple sinusoidal functions, (c) we eventually settled on a Bezier path solution (of course, higher order curves could be used to give a more accurate representation), and (d) which we can use Bezier path to calculate the foot rotation angle, so the stepping motion is more aesthetically pleasing (also, the angle needs to be inverted, as it passes half way due to the way we lift our toe and land with our heel). 68
5.9 Procedural Path Single Foot Stride - Plot of a generated foot position for a single stride from Meredith and Maddock.[MM04]. 69
5.10 Motion Capture Walk Cycle - Interestingly, if we look closely at a pre-recorded human walk cycle (i.e., motion capture), we can visually see a number of crucial transitions (i.e., foot orientation and trajectory) - while we capture the significant components (i.e., the offset peak) - there are a number of minor discrepancies. 69
5.11 Static and Dynamic - Static walking the centre-of-mass transitions is constantly supported, while dynamic walking the centre-of-mass remains in a state of dynamic motion. 73
5.12 Walking - Draw the step transitions and the projected centre-of-mass on the ground (top, side, and perspective view) 73
5.13 Response Decision Flow Diagram - State flow diagram for balanced logic. . . . 75
5.14 Projected Ground Information - Projected COM provides essential balancing information (e.g., default mass of 75kg, leg-length 1.0m, foot radius 10cm, and outer capsule radius 25cm. 76

5.15 Skeleton Mapping - Simple spring-loaded inverted pendulum and full skeleton model used for testing. 77
5.16 Push Disturbance - Recovering from push disturbance to the side while walking. 77
5.17 Skeleton Mapping - The point-mass model generates balanced stepping motion when mapped onto an articulated skeleton. However, the point-mass does not capture posture information (i.e., the upper body posture reaction based on the corrective stepping motion). We address and solve this limitation in Chapter 6 by incorporating an elongated body into the lightweight model. 78
5.18 Holding a Box - Biped character holding a box that is gradually increasing in weight (left to right) while maintaining balance. 80
5.19 Inverse Kinematic (IK) Re-Rooting - (a) The joint configuration with the support foot set as the root of the IK (i.e., right foot in the figure); (b) the IK root swapping between the left and right foot during stepping. 81
5.20 Simulation Screen Captures - The uncomplicated inverted pendulum model (a), through to the extended version with a support area (b)-(c), and finally integrating it with a fully articulated character (d) to generate full-body postural information. 83

6.1 Combining Multiple System Components - We bring together previous chapters to create a unified system for solving multiple-problems (i.e., a synergistic stepping framework that combines different techniques to attain an overall solution that cannot be attained with the individual parts). 85
6.2 Unified Overview - Illustrating the combination of the different stepping components from Chapters 3, 4, and 5. 86
6.3 Stepping Model Components - While there are different flavors and approaches for generating stepping motions based upon the pendulum model, we illustrate and compare the logic and features that each component provides. (A) The inverted pendulum model (IPM) was originally a biomechanically inspired approach [VS72, HG77] that later gained recognition in robotics [MS84b, KT95] and later the graphics community [KKI06]. (B) Linear inverted pendulum model (LIPM) [KKK*01b]. (C) Spring-loaded inverted pendulum (SLIP) [GOAS12]. (D) Elongated-body (either for more life-like walking with upper-body posture as shown by Kenwright [KDM11] (i.e., Chapter 4), or as a means of counter-balance correction [PT06]. (E) Ankle-Torque Feedback [Ken12d] (i.e., Chapter 5). (F) Centre of Pressure Strategy [PT06]. (G) Variable leg-lengths [PCD06] with an IP model to compensate for push disturbances using the capture-point, while Ito and Sasaki [IS11] performed lateral stepping based on zero moment point feedback for adaptation to slopes. (H) The hip midpoint as the COM position similar to SIMBICON [YLvdP07] due to it being fast and simple, however, the model could be adapted to constantly update and track the full articulated body COM position synonymous with the approach by Tsai et al. [TLC*10]. . 90
6.4 Capture-Point Comparison - The capture-point concept for (a) capture-point "distance", and (b) capture-point "height". 92
6.5 Capture-Point "Distance" - Estimating the capture-point "distance" based on a rigid mass-less support leg. 93
6.6 Capture-Point "Height" - Estimating the capture-point "height" based on a rigid mass-less support leg (illustrate a linear transition). The principle focuses on trading energy to increase or reduce momentum by means of increasing or decreasing the leg-length between foot-placement transitions. 96

6.7 Feet and Ankle Trajectories - (a) ankle-angle between step transitions, (b) ankle-foot path trajectory, and (c) upper-body angle deviation). An important note, is methods, such as SIMBICON [YLvdP07], keeps their feet parallel to the ground during gait motions, producing a robot-like marching effect, while, in reality, humans tilt and swing their foot (i.e., toe-stubbing effect on take-off). 97
6.8 Controlled Stepping - We carefully control the stepping locations of feet while navigating a complex environment (i.e., we cannot always place our feet at desired locations and must work within the constraints of the environment). . 98
6.9 Balanced Standing - The elongated rigid body provides a lightweight model for upper body postural information (from Chapter 4) which is further combined with a controllable balance stepping mechanism (from Chapter 5 and 3). 98
6.10 Elongated IP Walk Cycle Side View - Motion from the elongated body IP walking at a constant speed. 99
6.11 Elongated IP Walk Cycle Perspective View - Motion from the elongated body IP walking at a constant speed. 99
6.12 Coordinated Stepping - Coordinated foot-placement stepping while navigating terrain with holes. 101
6.13 Unstable Terrain Stepping - Navigating an unstable terrain (i.e., a bridge formed of rigid body cubes fixed in place with spring-dampers). 101
6.14 Simulation Screen Captures - (a) the inverted pendulum with foot-support area, (b) elongated-body, (c) combined foot-support area and elongated-body, and finally (d) an integrating articulated character with body posture and foot-ankle feedback. 102

7.1 Crowd Simulations - Hundreds of Instances of the extended inverted pendulum (IP) model. Our model's uncluttered and straightforward approach of generating simple biped avatars means it is ideal for simulating large groups of interactive pedestrians. For example, the figure shows hundreds of instances of our model running in real-time. 106
7.2 Close Interaction - Nine biped characters repeatedly bumping into one another because while being forced to walk in random directions, back and forth within a small circular boundary. 107
7.3 Simplified Point-Mass Biped Model (Additional Controllers) - Possible future model enhancements: The arms and legs of the character are analogous with a spring-damper mechanism and can be combined to work in synergy to control the overall character's center-of-mass (COM) and produce other motions, such as climbing and get-up motions. 107

A.1 Character joint names, individual body parts names, and degrees-of-freedom (DOF). 110

Chapter 1

Introduction

Character animation is employed in a variety of disciplines (e.g., graphics, animation, and robotics). While used extensively in video-games and animated films, character-based animations are also becoming widespread in training simulations, such as the medical profession, scientific research, and military design. In recent years, the rendering of computer generated characters has become increasingly realistic, while constructing similarly life-like and realistic motions remains challenging. The challenges are down to a number of factors. Firstly, humans have the ability to spot unbelievable and unnatural motions since we see them constantly in our daily lives. Secondly, understanding how the human body moves is complex and difficult due to the intricate anatomical structure and stylistic deviations of humans (i.e., the large number of degrees-of-freedom (DOF) and diverse range of behavioral characteristics). The interest and importance of understanding and "reproducing" believable, life-like, and natural human movement is valuable to multiple disciplines.

More often than not, prerecorded animation libraries are used to create a character's movements. While prerecorded human movements (e.g., motion capture data) can be played back in virtual environments to synthesize these highly realistic and life-like character movements, they, nevertheless, can be inflexible and difficult to adapt to unforeseen circumstances in interactive environments. In retrospect, upright motions, such as standing, walking, and running are of crucial importance for any computer generated biped character, so that it can explore and navigate its virtual world. In this thesis, we focus primarily on synthesizing controlled stepping motions with an emphasis on developing flexible and straightforward techniques and algorithms that can run at real-time frame-rates, while being robust and interactive. We use low-dimensional approximations for key information (e.g., centre-of-mass dynamics and foot locations) to create coordinated full-body movements that are visually life-like and interactive (e.g., responding to random push disturbances and attempting to remain balanced and upright by means of intelligent foot placement logic). Our approach focuses on controlled stepping that can synthesize and produce new upright biped motions that run in real-time and require a minimum amount of customization or data (i.e., hand tweaking, offline pre-processing, or key-frame libraries). In conclusion, we presents a lightweight solution for generating biped stepping motions (e.g., standing, walking, and running) without any offline processing or key-framed animation data. We focus on low-dimensional approximation models, to produce full-body animations that can be used in time-critical virtual environments, such as games.

The key challenges that make it difficult to reproduce life-like human movements in "real-time" that mimic the real-world are:

- Realism: A particular character model gives rise to a large set of possible motions with different styles. Even if robust and stabilizing control laws can be found, it is strenuous to construct motions that reproduce the intricate and agile locomotions we observe in nature.
- High Dimensions: Characters have a relatively high number of degrees-of-freedom, making the search for the appropriate control parameters hard. Although continuous numerical optimizations can cope with large search spaces, the stringent demands of interactive applications make it clear that optimizations cannot solely be performed at the time control is needed.
- Contacts: Characters are restricted to move within a certain region of their three-dimensional environment, and these constraints are difficult to maintain with a real-time control system. Furthermore, frequent ground contacts create a highly discontinuous search space that makes most continuous controller synthesizing methods ineffective at planning over longer time horizons.

- Underactuation: Dynamically simulated characters are difficult to control because they have no direct control over their global position and orientation. Even staying upright is a challenge for large disturbances. In order to succeed, a control law must plan ahead to determine actions that can stabilize the body [MLPP09].

1.1 Motivation

Physics-based techniques result in motions that obey the laws of nature and allow us to imitate the real-world accurately. For example, applying a force to a virtual character's body leads to realistic and physically-correct responsive movement. However, generating the character dynamics to counteract and mimic the real-world reactive human behaviors (e.g., balancing, stepping, and postural changes) is difficult.

Walking, standing, and running, are common everyday human motions that are essential for any virtual environment to allow the character to move and explore its world. While these motions can be recorded from real-actors using motion capture or created by key-framed based techniques, they have problems adapting to disturbances, such as pushes and trips. Even adapting key-framed data (i.e., motion capture data) for character feature changes (e.g., height and walk stride) is arduous.

We wish to achieve a practical and stable real-time solution that has a low computational cost and can recover from problems gracefully (e.g., over multiple frames). We aim to handle numerical inaccuracies while producing reasonably life-like and aesthetically pleasing results through different approximation techniques. This thesis targets a scalable lightweight solution that can incorporate different levels of detail (e.g., single high detailed character movement or multiple less-accurate instances for crowds of characters).

This thesis creates an intelligent hierarchical solution enabling users to control high level motions, such as direction and speed, while low-level problems are solved automatically, such as balancing and foot placement. The final solution combines different components with varying priorities (e.g., balancing and walking) so they work together to complement the character's movements and produce a responsive and life-like solution.

Since humans are capable of producing a monumental assortment of different actions and styles, it would be unrealistic if this thesis aimed to attempt to synthesize and generate every possible motion in real-time without key-framed data while appearing life-like and natural within the allocated time limit. Hence, the focus of our work is on balanced upright stepping motions (e.g., walking and standing), since these are one of the most common actions in real-worlds and are indispensable so virtual characters can move around and explore their environment. Generate motions that are interactive (e.g., push and can pick-up objects), balanced (e.g., remain upright on two legs), adaptable (e.g., handle uneven terrain and character dimensions), controllable (e.g., steering, speed), recover from disturbances (e.g., pushes and trips), and customizable (e.g., style) is significant and valuable.

1.2 Contributions

The key contributions of this thesis evolve around the generation of upright balanced biped stepping motions (e.g., standing and walking) without any offline processing or artist intervention by extending the biomechanically inspired inverted pendulum model to include additional control approximations, such as postural upper-body feedback. That is, we develop a lightweight model to emulate full-body interactive animations for use in time-critical virtual environments, such as games.

The key contributions for this thesis based on chapter inclusion are:

- Chapter 3

  The contribution is a **lightweight** goal-directed 3D stepping control algorithm that runs in real-time, straightforward, and robust, for generating interactive, controllable, balanced upright biped standing and walking motions. We demonstrate a non data-driven (i.e., keyframless) system for biped stepping motions that has the ability to remain stable under random unforeseen disruptions (e.g., push forces) while providing a solution to compensate for linear exertions (e.g., return to desired speed or cancel out movement by stepping).

- Chapter 4

  We extend the work of the enhanced IP described by Maus et al. [MRS08]. The work by Maus et al. [MRS08] was 2D and only provisioned the calculated expectation of the simulation. In the field of robotics, this initial investigatory stage is usual as the end goal is to actually build a robot. However, for video-games we want a fully interactive 3D run-time implementation. This is what we achieved, allowing our character model to be exposed to forces and be manipulated by a player. The model itself has no data-driven aspects with all foot placements and body movements created dynamically. The fundamental contribution demonstrates the enhanced 3D IP (i.e., IP with elongated body) as an option for fully interactive realistic human locomotion in real-time without any data-driven requirements.

- Chapter 5

  The contribution focuses around simplifying the biped foot support region (e.g., using spheres and capsules) to produce approximate foot placement and balancing information (e.g., position, orientation, and path trajectories) in conjunction with an ankle-torque feedback mechanism to remedy oversimplifications in the inverted pendulum model (i.e., not having feet) and produce a more controllable solution. We use this approach to generate interactive character movements for controlling a fully articulated skeleton body.

- Chapter 6

  This chapter unifies the work presented in earlier chapters (e.g., elongated body for postural control from Chapter 4 and foot-ankle feedback from Chapter 5) into a single biped stepping framework.

1.3 Dissertation Outline (i.e., Road-Map)

This dissertation continues onto Chapter 2 to provide background information for the area of controlled character animation strategies. *This background chapter sets the scene for the thesis, while the related work is described specifically in each of the following chapters. Due to the breadth of the subject, this structure allows the reader to be led into each chapter's* *contribution by the appropriately organised related work.* Thereafter, the following chapters of the dissertation (i.e., Chapter 3 to Chapter 6) are divided into key self-contained components that focus on specific problems in the area of controlled character movements without key-framed data using physics-based methods; for example, foot-ankle control and upper-body postural feedback. In the final Chapter 7, we conclude with an overview of the dissertation's contributions, limitations, discussion, and future work.

In summary, this dissertation and the interest behind this research is aimed at creating more reactive, dynamic, and adaptable avatar character animations **without key-frame data** for use in randomly changing interactive virtual environments, such as games. This dissertation investigates and demonstrates different approaches of using uncomplicated low-dimensional techniques (e.g., an extended inverted pendulum) as a means of modeling fundamental upright full-body character motion; in particular, interactive and controlled balanced biped stepping movements, such as standing and walking.

1.4 Publications

Preliminary versions of the work presented in chapters 3, 4, 5, and 6 of the thesis are in the following publications:

- Kenwright B., Davison R., Morgan G.: Dynamic balancing and walking for real-time 3d characters. In Proceedings of the 4th international conference on Motion in Games (Berlin, Heidelberg, 2011), MIG'11, Springer-Verlag, pp. 63-73. [KDM11]
- Kenwright B.: Responsive biped character stepping: When push comes to shove. In Proceedings of the 2012 International Conference on Cyberworlds (2012), vol. 2012, pp. 151-156. [Ken12d]
- Kenwright B. and Huang C-C.: Beyond Keyframe Animations: A Controller Character-Based Stepping Approach. ACM Technical Brief Transactions on Graphics in SIGGRAPH Asia, 10:1–10:4 (2013). [KH13]
- Kenwright B.: Inverse Kinematics: Cyclic Coordinate Descent (CCD). Journal of Graphics Tools 16, 4, (2012), 177–217. [Ken12b]

Chapter 2

Background

This background chapter sets the scene for the thesis, while *specific* related work and principles are described separately in each chapter. Due to the breadth of the subject, this structure allows the reader to be led into each chapter's contribution by the appropriately organised material. Thereafter, **each subsequent chapter of the thesis (i.e., Chapter 3 to Chapter 6) provides key self-contained related work that introduce and solves specific problems in the area of controlled character stepping without key-framed data using a lightweight physics-based methods (e.g., foot-ankle control and upper-body postural feedback)**.

One of the biggest challenges in 3D character development, and probably one of the most rewarding and interesting, is mimicking the dynamic interactive properties of real-world humans. We provide a brief comparison between existing work and our approach while identifying and explaining the advantages and disadvantages of specific techniques. Since the topic is a multi-discipline problem and is an active area of research, we include a number of contributions from other fields, such as, graphics, robotics, and biomechanics, that have inspired algorithms and approaches used throughout this thesis, including simplified balancing mechanisms, locomotion models, and studies of human movement. It should be noted that our approach focuses on a lightweight physics-based model, for time critical systems, such as games. These systems typically desire visually plausible results using approximate methods (e.g., low-dimensional hybrid techniques), while robotics and biomechanics utilize accurate models that concentrate on physical precision.

2.1 Data-Driven Solutions

A recorded human's movements (i.e., motion-capture data) can be applied to an articulated virtual character and played back in real-time to create highly life-like and realistic motions. These prerecorded animations can be modified and adapted by joining, mixing, and reordering motion segments [RCB98, KG04a, LCR*02, TH00, AF02]. However, it can be difficult to find the right combinations of motion segments for the desired task, while the final motions are limited by the available library of animations and cannot generate or create truly unique and original motions. Data-driven methods have attempted to synthesize the physical responsiveness of a character by blending various motion segments to synthesize impacts [MP07, KLK04, ZH02, ZMCF05]. For example, Zordan et al. [ZMCF05] demonstrated a martial art test bed that adapted motion-capture data from humans to produce responsive physics-based motion segments that reacted to varying force disturbances. Although there has been a tremendous amount of work in this area, the approaches can broadly be categorized into two main groups: *motion-rearrangement* [AFO03, KGP02, LCR*02, AF02, LWS02] and *motion-parameterization* [KG04a, WH97, MK05a, RCB98, KS05] (i.e., blending and mixing).

Motion-rearrangement generates new animations by searching a motion database library for similar motion segments to meet a desired constraint (e.g., walking along a specified path). Most of the approaches use a *motion-graph* type structure [KGP02] to connect similar poses in a motion-stream. Systems for identifying and selecting similar frames [WB03, WB08]. Split-second reactions for high-quality motion transitions using multiple-frames [IAF07]. Tidying the final motion by detecting and ensuring the final foot-placements constraints are correct by means of inverse kinematic techniques [KSG02, IAF06, SLSG01]. Tools for modifying animations, so that they are physically-correct (i.e., balanced) by controlling the centre-of-mass or zero-moment-point [BMT96, TK05, BMT95, TSK00]. The algorithm presented in this thesis can be used to create motion transitions. However, while a motion-graph requires similar poses to create the transitions, our proposed techniques do not have this limitation, since the directed steps can be calculated before and after the transition to avoid falling over. Finally, our approach takes into account the primary physical properties of the character (i.e., the centre-of-mass and foot-placement location) during transitions to ensure physical plausibility.

Motion-parameterization blends labeled motion pieces together to synthesize specified behaviors (e.g., happy, sad, and angry). For example, Wiley and Hahn [WH97] blended various motion segments to create more expressive exaggerated character motions that possessed artistic qualities. Whereby, the motions were generated at run-time and demonstrated smoothly controlled motion transitions, such as walking and cycling. The motion segments needed to be labeled by an artist to specify animation sequences; however, Kwon and Shin [KS05] presented an approach for automating the motion labeling for walking sequences based on the centre-of-mass trajectory. The motion sequences were divided into classifications based on footstep similarities. The locomotion system could synthesize real-time walking motions with diverse speeds, turning angle, and accelerations. In contrast, this thesis focuses on a procedural approach for generating the controlled walking movements based upon intelligent foot-placement logic (i.e., control of a lightweight physics-based model through precise foot-placement stepping). While similar foot-placement research with a likewise goal [vBSE11, vBPE10, EvB10] attempted to solve the same problem, this dissertation generates the stepping motions through algorithmic techniques based upon the character's centre-of-mass and foot-balance information rather than relying on pre-canned motion-capture libraries that requires large numbers of stepping examples to produce the final blended motion solution. Importantly, our lightweight physics-based stepping model has no data driven aspects with lower-body motions being created on-the-fly to fit the situation. We present a number of different feature enhancement (e.g., an elongated upper-body and ankle-torque) to target specific problems (e.g., posture and steering).

In many instances, data-driven and controller-based approaches are combined (e.g., [ZMM*07, ZMCF05, CBvdP10]) to produce character animations. This has the advantage that fewer predetermined animation sequences may be required while affording a greater degree of animation.

For example, a character falling over on an uneven surface (e.g., stairs) has to be achieved via physics whereas upper-limb movement (e.g., reaching) could be achieved via artist directed animation. The popularity of this approach seems to indicate that a purely physics-based approach may not suffice on its own (realism is lacking). However, if games are to afford complete freedom of interaction, this is the only viable approach.

2.2 Optimized Search

Classical mechanics is the study of the motion of bodies under forces through physics-based laws. These physics-based laws can emulate the real-world properties of a character (i.e., mass, muscle strength, and frictional contacts) and hence, provide a method for synthesizing actual physical-world movement. Consequently, it should come as no shock that numerous researchers have investigated and explored physics-based techniques in conjunction with optimization search techniques [SC92a, FP03, LHP06, WK88]. Firstly, the work by Witkin and Kass [WK88] presented a spacetime constraint technique for creating character animation. The artist specifies what the character has to do, for instance, *jump from here to there, clearing a hurdle in between*; how the motion should be performed, for instance *don't waste energy*, or *come down hard enough to splatter whatever you land on*; the character's physical structure (i.e., geometry, mass, and connectivity) of the parts; and the physical resources' available to the character to accomplish the motion, for instance the character's muscles, a floor to push off from. The requirements contained in the description, together with Newton's laws, comprise a problem of constrained optimization. The solution was a physically valid motion satisfying the 'what' constraints and optimizing the 'how' criteria. Examples included a Luxo lamp performing a variety of coordinated motions - the realistic motions conformed to such principles of traditional animation as anticipation, squash-and-stretch, follow-through, and timing. Secondly, Stewart and Cremer [SC92b, SC92a] and Liu et al. [LHP06] created character motions for handling interactive situations by solving space-time optimization problems. While the animation search space is vast and possesses an enormous amount of ambiguity, it should come as no shock that optimisation techniques typically require artistic intervention to control and solve these multi-object optimization problems. Whereby, the solution is controlled by injecting motion-capture data into the problem to produce targeted character motions [MLPP09, DSAP08a, AdSP07, MZS09, dSAP08b]. Typically, for the optimisation problem of balanced locomotion and standing fixed frame reference motions are used, and hence disallow for the ability to generate unique stepping actions on-the-fly in any direction with varying step durations.

In contrast, to the lightweight physics-based model in this theses, to capture the fundamental characteristic of balanced biped stepping, there are a number of important related techniques that use more complex models to accomplish similar goals, such as statistical methods [KG04b, MK05b, GMHP04, WFH08] that have tried to synthesize and modify particular character motions. For example, a style-based IK technique presented by Grochow et al. [GMHP04] used a training model to create interactive character posing, trajectory keyframing, real-time motion capture with missing markers, and posing from a 2D image. The real-time technique required training data (e.g., motion capture data) and a set of constraints to find poses that most likely satisfied the situation . The model was trained for different input data leads to create IK style variations. The IK approach used a probability distribution (i.e., Scaled Gaussian Process Latent Variable) over the space of all possible poses that could generate any pose, but preferred poses that were most similar to the space of poses in the training data. The parameters for the model were all learned automatically; no manual tuning is required for the learning component of the system. Similarly, the system by Wang et al. [WFH08] used a Gaussian Process Dynamical Model (GPDM) for nonlinear time series analysis (i.e., a learning model analysing training data from human poses or high-dimensional motion capture data (e.g., 50-dimension)). The GPDM operated in a low-dimensional latent space with associated dynamics, and a mapped from the latent space to an observation space. The marginalized output model parameters in closed-form were created using the Gaussian process for both dynamic and observation mapping that resulted in non-parametric model for a dynamic systems. The GPDM presented an effective representation of the non-linear dynamics in these spaces. These methods require training data and work with intricate models that are capable of solving diverse tasks, compared to our lightweight stepping approach (i.e., see Figure 2.1).

Finally, it is worth mentioning interesting techniques that have focused on distinguishable approaches, such as the inspiring work by Jakobsen [Jak01] who modified the skeleton structure using a real-time particle-based solution to represent articulated bodies. Similarly, Kawachi et al. [KKS04] used a simple spring-mass system to model hands and feet. The interactive algorithm by Kawachi et al. [KKS04] created character animations using kinematic constraints with limited computational time. In order to reduce computations, the animations were not created by procedural algorithms but synthesized by deforming and concatenating short motion examples, each consisting of a sequence of postures. A keyframe placed between two consecutive motion examples was deformed using inverse kinematics so it satisfied given constraints. The motion examples between the keyframes were deformed to ensure continuity of postures in position and velocity. Each posture was parametrized by a set of particles in an orthogonal coordinate system. The inverse kinematics method with the particle representation realizes fast and stable deformation of keyframe postures, and the deformation of motion examples were calculated on a frame-by-frame basis by decomposing whole-body deformation into per-particle deformations. The approach presented examples of character animations being synthesized at interactive frame-rates. Lastly, the notable work by Shiratori and Hodgins [SH08] focused on creating more immersive character interaction by allowing the user to control a fully articulated physics-based character through the interface. All of these cases offer a variety of novel lightweight approaches for controlling characters based on simplifying the problem akin to this thesis direction (i.e., simplifying the problem down to key elements, in our case, the centre-of-mass and foot position for real-time controlled biped stepping that is interactive and dynamic).

2.3 Controlled Physics-Based Characters

Physics-based animated character approaches typically use dynamic models (e.g., articulated rigid body structures) to represent the character while forces and torques control the movement. However, it is challenging to generate coordinated controlled torques and forces that mimic the human muscle to produce natural life-like motions. One approach is to use a controller to generate particular motions (e.g., walking, running, diving) [HWBO95, LvdPF00, WH96]. Another approach for physics-based characters is a puppet-like control (i.e., joint-based tracking) that follows reference motions [ZH02, LKL10]. For example, the joint angles for the puppet can be provided by a key-framed animation library. Then again, simply following reference motions without taking into account balance can cause the character to fall-over. There are two fundamental problems of ignorantly tracking reference key-framed motions. Firstly, the recorded human motions may have different feature sizes compared to the physical model (e.g., height and weight). A solution to this problem was presented by Sok et al. [SKL07] with an optimization technique that corrects the motions in order to remove inconsistencies. Secondly, physics-based characters are *underactuated*, that is, their feet are not fixed to the ground. This means the biped character does not directly control their global position or orientation because the number of degrees-of-freedom is greater than the number of actuators (i.e., joints). Only by means of external forces (e.g., ground reaction force) can the character control the underactuated degrees-of-freedom (i.e., global position and orientation). Ignoring the underactuated degrees-of-freedom by naively tracking a motion will cause the character to deviate from the reference motion and eventually lose its balance and tip-over. In this thesis, we constantly analyze the character's physical properties and feedback and remedy balance deviations to control the underactuated degrees-of-freedom issue.

In the context of interactive controllers, a number of physics-based approaches have focused on protective stepping in the past [YLvdP07, FvdPT01a, JYL09, KKI06, YvdP06], such as Shiratori et al. [SCCH09] who explored the biomechanical principles of trip recovery during walking. However, our approach is most similar to [KKI06, YLvdP07] who calculates foot-placement information based on an inverted pendulum (IP) model (with [KKI06] extracting parameters from motion-capture data and [YLvdP07] calculating the step position so that the centre-of-mass lies within the centre of the support-polygon after stepping). The main distinction is that our pendulum model includes feature enhancements, such as, ankle-torque feedback, while step information is calculated without any motion-capture reference data. We do not extract parameters for the IP using key-frame information to decide foot-placement locations. The parameters can be modified to regulate and control walk velocity (i.e., steerable and controlled speed) while being able to automatically compensate for disturbances.

Without a physics-based solution, the generation of stepping motions in response to unforeseen disturbances depends upon data-driven approaches, which requires a large collection of pre-canned animations [YPvdP05, KHL05, AFO05]. The approach of this thesis generates trajectory motions through a procedural physics-based stepping technique. Our stepping technique uses knowledge about itself and its surroundings to accomplish the primary goal by controlling the centre-of-mass position and foot-placement locations to synthesize the physical attributes of an upright biped stepping motion. For example, Stewart and Cremer [SC92a, SC92b] demonstrated an optimisation solution using the end-effector planning and centre-of-mass for control. These approach of controlling high level goals, such as, the centre-of-mass, end-effectors, and angular momentum, has proven to be a popular method for generating realistic, flexible, and robust motions [YL10, WP10, CBvdP10, dLMH10, MdLH10]. The generated motions are a mix of data-driven techniques or through computationally expensive global optimization, such as covariance matrix adaptation (CMA) [Han06]. This thesis's approach focuses on approximate low-dimensional procedural technique for controlling the high level goals in real-time (e.g., see Figure 2.1).

2.4 Balanced Biped Models

Robotics has demonstrated techniques for automatically generating stepping motions for humanoid robots [SA10, AS07, HM04, Ste07, HHC07]. While the fundamental balancing mechanics have been defined [PH05, R*86], multiple low-dimensional control models for balance and locomotion exist. For example, in robotics, one approach is to treat the character as a linear IP [KKK*01b], and control the stepping based on massless legs and point-mass [KKK*03]. Another approach, uses a 'capture-point' that yields a single step to recover from disturbances [PCD06].

The popular biomechanically inspired IP model has been extended by multiple researchers in robotics to account for disturbances. For example, angular momentum pendulum [KLK04] and the IP plus flywheel [PCD06] to remedy changes in angular momentum error. This thesis takes a similar approach, and focuses on extending the basic IP model to remedy oversimplifications, such as constantly needing to step, and unable to steer from a still standing start. The goal of our stepping model is not to completely remove or cancel out disturbances but to place the foot at select locations to compensate for disturbances and maintain a control motion (e.g., walking at a fixed speed or standing still).

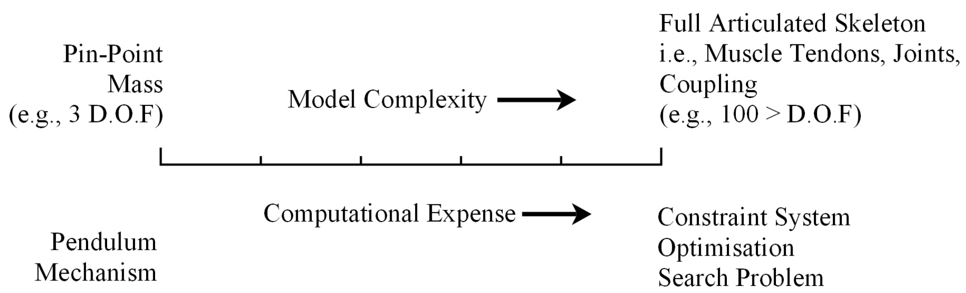


Figure 2.1: Model and Computational Complexity - Reducing the articulated model's complexity down to its fundamental components to solve targeted motions using low-dimensional models that can be less ambiguous and straightforward to solve both mathematically and computationally.

We emphasising important approaches in the field of robotics that have contributed to the development of responsive biped controllers. To begin with, Shih et al. [SGL93] developed a straightforward model for enabling characters to respond to small disturbances, while later Stephens [Ste07] and Pratt et al. [PCD06] developed controllers that could generate motions to recover from a range of push disturbances. Synonymous to our lightweight biped stepping approach are a number of important models that approach the problem from a like-wise direction (i.e., simplifying the problem down to key elements so that a particular motion can be created); For example, Popović and Witkin [PW99] - simplified models for jumping motion; Raibert [R*86] set the ground work for a number of well known principles and fundamental (i.e., stepping mechanisms); and Hodgins et al [HWBO95] and a number of related papers [SHP04, RH91, WH96, HP97, HJO*10, ZH02, SCCH09] who make use of certain fundamental techniques described in this thesis (e.g., inverted pendulum and PD control method). A major disparity between this thesis goals and those in robotics is that we focus on an approximate solution that is computationally fast while producing physically credible and aesthetically pleasing motions.

For a comprehensive introduction to legged motion mechanics with an emphasis on dynamic and active balance principles, see Raibert [R*86], who is one of the leading authority on legged locomotion. Raibert [R*86] presents a number of engineering theories for human motor control while explaining the fundamental groundwork for legged locomotion (e.g., static and dynamic balance), addressing *real-world legged robots that run and balance.* While the work studies *physical* machines that run and balance on **just one leg**, including analysis, computer simulation, and laboratory experiments, Raibert [R*86] explains how the principles of locomotion with one leg can be extended to systems with multiple legs. The studies focus on the issues of balance and dynamic control, while avoiding several problems that have dominated previous research on legged machines (e.g., physical limitations).

2.5 Moving Away From Data-Driven Solutions

We begin by reviewing some successful examples of where computer graphics research has focused specifically on creating interactively controlled characters based upon physics rather than motion-capture or kinematic data. After the initial work by Yin et al. [YLvdP07] (i.e., SIMBICON) who used an inverted pendulum model to generate dynamic locomotive motions in real-time that could be used to correct motion-capture data, followed a number of physics-based model approaches, including Wu and Popović [WP10] who combined the Jacobian-transpose (JT) method and optimization to solve control problems using a static resolution of forces based on the relations of pairs of action frames (such as feet) and reaction frames (such as the root body). Along the same lines Wang et al. [WFH10] used an inverse dynamic optimization approach to develop walking locomotion - note, a frame-rate of 2400Hz was required. A graph-based technique was presented by Lee et al. [LKL10] for robustly synthesizing natural animations in real-time - note, the final motions lacked the reactivity of the animated character with user interaction; then Tsai et al. [TLC*10] demonstrated the IP model for biped character animation using a balance filer to track motion-capture reference trajectories and lower-body stepping movements (i.e., adapted by the IP model), while we generate the character motions without any data-driven reference material; similarly, Coros et al. [CBvdP10] used the IP centre-of-mass to accelerate/decelerate and steer the character (i.e., with small ad-hoc control forces) - note, in essence, the approach employed an IP model in combination with a Jacobian-transpose control for tuning like Wu and Popovi(́c) [WP10]. Mordatch et al. [MdLH10] exploited a low-dimensional preview control for the character model to enable on-line look-ahead optimization - note, the low-dimensional system was a spring-loaded inverted pendulum model (SLIP) and provided on-line planning of locomotion tasks on constrained and uneven terrain. Finally, Lasa et al. [dLMH10] tracked a character's centre-of-mass and foot trajectories using low-level feature objectives to generate biped motions. The physical simulations generated interactive biped motions that could automatically adapted to environmental change, essentially, guiding a controller to behave in a specific manner. Our approach focuses on a similar goal but uses a lightweight physics-based model as the driving concept while avoiding any data-driven animations to create controllable upright stepping motions on-the-fly and in real-time that can handle disturbances, such as pushes and changing terrain.

While the IP model has been seen repeatedly in biomechanics, robotics, and computer animation, this thesis addresses a number of problems not addressed before with the IP model so that it can be extended to produce controllable biped stepping movements without key-framed data. We propose a number of novel methods to solve different problems that do not require any computationally expensive techniques or offline processing. In contrast, with previous methods that work around the IP model, they can require offline precomputed optimizations or run at non real-time frame-rates and do not address inherent oversimplifications. While our approach uses a greatly simplified structure for computational speed, we successfully demonstrate the mapping of the low-dimensional model onto a whole-body biped for balanced stepping motions. This thesis uses a framework similar to [TLC*10, MdLH10, YLvdP07], however, these proposed methods focused on combining the inverted pendulum model with animation data, while we focus on extending the IP model through feature enhancements to produce controlled life-like stepping motions without any data based using a procedural approach.

2.6 Summary

In conclusion, there is a demand for techniques and approaches that can identify, create, and adapt human-based character animations so they are more engaging, life-like, and physically-correct. Current computer graphics hardware and software make it possible to synthesize near photo-realistic images, but the simulation of natural-looking, physically-correct, interactive, character motions remains a difficult and challenging task. While rendering the animation of humans, animals, and robots can delight and move us, simulating and synthesizing their realistic motion in real-time holds great promise for many applications, including training simulations and video games.

This chapter has introduced the reader to respective approaches and techniques from different disciplines (e.g., computer graphics, robotics, and biomechanics) that have contributed to the mechanics, control, and animation of articulated biped characters. We have also emphasised a number of techniques and approaches that have attempted to create animated solutions using low-dimensional techniques (e.g., pendulum and particle-based systems). We go into greater depth on particular approaches in subsequent chapters that build upon synonymous techniques (e.g., posture) or solve similar problems using alternative methods (e.g., capture point). Each chapter addresses particular issues that we combine to create a lightweight physics-based technique to produce - adaptable, robust, and interactive stepping information with a minimalistic amount of tuning and data (i.e., key-framed libraries).

Chapter 3

Inverted Pendulum (IP): A Low-Level Approach

3.1 Introduction

This chapter explains how a virtual biped character's upright movements (i.e., walking, running, and standing) are essential in simulation environments (e.g., video-games) and allow the character to interact and explore its universe. While it is easy to generate basic character animations that mimic real-world humans (e.g., by means of key-framed animations, interpolated splines, and inverse kinematics) they can, however, produce implausible life-less looking movements that are non-interactive and lack real-world physical properties (e.g., momentum and dynamic stepping logic).

The challenge is humans possess a huge number of degrees of freedom and are capable of producing a vast assortment of diverse, original, and complex movements that are both physically bound (i.e., balanced and dynamic) and life-like. However, a limited number of parameters are common and crucial for balancing (e.g., centre-of-mass trajectory and foot support area), which we identify and synthesize to emulate fundamental biped character movements. These crucial parameters are combined with a low-dimensional physics-based model known as the inverted pendulum (IP) to create fundamental balancing information that we use to construct our adaptable dynamic biped motions.

We develop a lightweight model that synthesizes and generates dynamic biped motions by observing how real-world humans remain balanced during upright movements. For example, as mentioned the physics-based inverted pendulum model, which was originally developed in biomechanics by observing the logic behind human stepping provides a **low-dimensional approach for mimicking 'fundamental' upright balancing movements**. However, the goal is not to generate a biped with perfect balance, but to intelligently recover from it when it is lost in a realistic way, over and over again. Human movement is smooth, realistic, and life-like, and the character is always moving and is never static and perfectly still (e.g., small swaying movements). **A human's movements are typically graceful and come from dynamic rather than static stability. Merely steering the character by pushing it with forces in the desired direction will produce unrealistic motions.** For example, if we make the character timidly extend his free leg in the direction of navigation before committing any weight to it, while constantly maintaining balance, this produces movements that appear robot-like and unnatural. The motion does not feel fluid and never takes flight; making the character appear scared of losing balance. In reality, a **human character relishes his dynamic ability without any effort or worry.** The character allows his full-body weight to wander away from his point of static balance in any direction, and are able to recover and adapt to the situation. As the character falls further off-centre, he must push harder into the floor to keep the motion horizontal and stretch the anchored leg further and more quickly to compensate for his hypotenuse. Achieving this smoothly and in a life-like way, demands both active muscle power and careful control. Whereby, each horizontal fall-away from his old centre-of-mass towards his new one barely perceiving any imbalance, and produces smooth deliberate stepping motions.

A crucial aspect of generating physics-based character animations is adapting the motions to compensate for unforeseen disturbances (e.g., pushes or terrain height changes). For example, if a character's body is pushed slightly off-centre, it does not know or care whether it represents the beginning or end of a step and is able to shift the following foot placement to compensate. When a character's mass is off-centre, its weight will push down on the floor in a more horizontal direction. Whereby to maintain a motion that is truly balanced and avoid

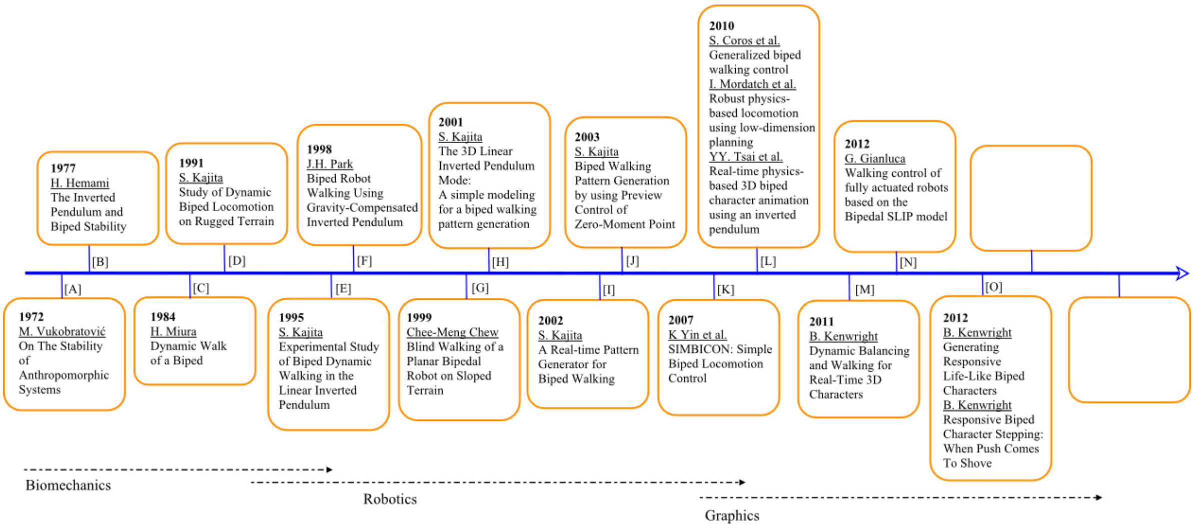


Figure 3.1: Time-Line - Research time-line focused on the inverted pendulum in terms of character-based solutions (e.g., biped balancing) over the past few decades (biomechanics, robotics, and graphics). [A] [VS72]; [B] [HGJ77]; [C] [MS84a]; [D] [KT91]; [E] [KT95]; [F] [PK98]; [G] [CPP99]; [H] [KKK*01a]; [I] [KKK*02]; [J] [KKK*03]; [K] [YLvdP07]; [L] [CBvdP10, MdLH10, TLC*10]; [M] [KDM11]; [N] [GOAS12]; [O] [Ken12a, Ken12d].

falling towards the floor, the character must pay Pythagoras his due by gradually extending and bending his free leg in an off-centre direction to compensate and remain balanced.

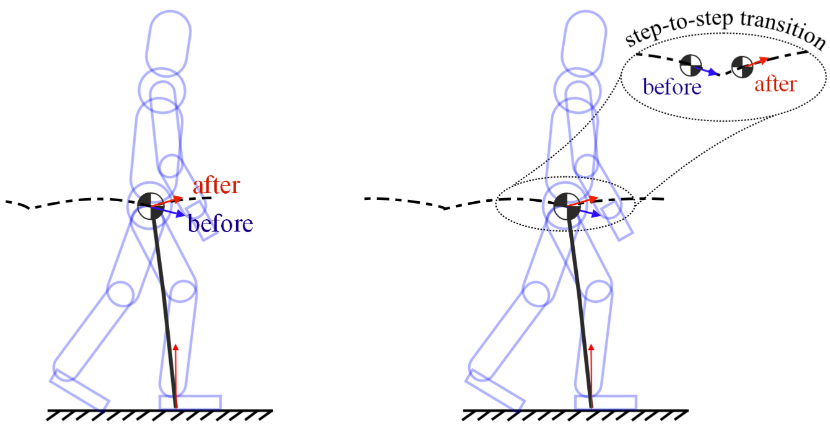


Figure 3.2: Simplified Inverted Pendulum (IP) Biped Model Dynamics - The IP model provides key balancing information based on the foot placement and centre-of-mass location.

3.1.1 Chapter Contribution

The contribution of this chapter is the explanation, demonstration, and evaluation of the inverted pendulum technique as a method for generating balanced upright motions for biped characters. In later chapters, we extend the technique to include additional feature enhancements to remedy oversimplifications while not compromising the computational speed and simplicity of the model. In summary, the important points that we address in this chapter, which are novel and interesting, are:

- Controlled balanced stepping (e.g., the ability to maintain a controlled steered velocity during disturbances and/or while pushing/pulling an object)
- Real-time stepping on complex and unstable movable terrain (e.g., stairs and bridges)
- 2D and 3D explanation with limitations and advantages

3.2 Related Work

This chapter and the motivation behind this thesis is aimed at creating responsive dynamic motions using a lightweight physics-based models without artistic intervention (i.e., key-framed animations). Some earlier examples of work that has approached this problem from a similar direction, include work by Panne [dP97] who demonstrated how character motions could be generated using only foot placement information and Jain et al. [JYL09] who used a similar method with a support polygon to position and control the centre-of-mass so it remained balanced during stepping.

Controllers are a popular approach for driving physics-based animations using forward-dynamics because, while challenging to design, their feedback nature is reflective of the life-like control strategies exhibited in humans, animals, and robots [YCBvdP08]. For example, early controllers include locomotive motions (i.e., not just biped) [GT95, SC92b, AFP*95, LvdPF96, RH91, SC92a, vdP96, SvdP05], human athletics [HWBO95, WH96], protective falls [FvdPT01b], *tracking motion-capture data* while maintaining balance [DSAP08a, YLvdP07, SKL07], accommodating and reacting to unpredicted perturbations [YCBvdP08], and more recently, mixing multiple controllers for navigation tasks [CBvdP09] and combining offline optimization in an attempt to reproduce key human features [WFH10]. While in short, *the goal of this chapter* is to demonstrate control strategies for automatically adapting to complex environments and interactive situations (e.g., slopes and pushes) without any motion-capture data or off-line optimization.

In part, this chapter uses the inspiring work by Kajita et al. [KNK04] and Sugihara [Sug08], since they showed that the IP model possesses the potential for mimicking dynamic upright human motions while providing a lightweight stepping mechanism. In retrospect, the IP was initially a biomechanically inspired approach that later gained recognition in robotics [PH04, SCCH09, HS99] and then computer graphics [YLvdP07, TLC*10] as a robust and straightforward technique for synthesizing how we, as humans, perform upright balanced foot movements, since the concept of intelligent foot placement is crucial for balance during stepping as shown by Pratt [PT06, PCD06] who demonstrated how the IP could also be used as a viable method for generating responsive biped humanoids (see Figure 3.2).

Our Work: In summary, this chapter's approach builds upon the popular IP technique, which provides a simplified dynamic model for generating crucial balancing information. For example, the IP model has been exploited time-and-time again due to its computationally efficiency and simplicity [dSAP08b, CBvdP10, MdLH10, TLC*10]. This chapter extends the IP model's ability without sacrificing the simplicity and speed upon which its popularity is built to incorporate workable features for dealing with specific situations (e.g., sloping terrain and pulling objects). While in later chapters (i.e., Chapter 4 to Chapter 6) we expand the pendulum-based model's powerful ability to addressing and solving particular oversimplification to construct full-body upright biped animations without animation libraries.

3.3 What is the Inverted Pendulum (IP)?

The inverted pendulum was originally a biomechanically inspired approach [VS72, HG77] that later gained recognition in robotics [MS84b, KT95] and then the graphics community [KKK*01b, KDM11]. The reason the IP model has gained a great deal of popularity is because it is a computationally fast and straightforward technique that is easy to understand and control and requires no implicit or explicit system dynamic inversion. Most importantly, however, is the IP model can effortlessly run at real-time frame-rates and does not require any demanding online or offline optimization processing.

While there are multiple deviations of the basic IP model, we broadly classified them into three main types:

- IP model with rigid-legs [DH94, GCRC98, KDR05, KMS01].
- Spring-Mass IP Model [Ale92, RBS10, GSB06].
- Wheel IP Model [CRTW05, McG90].

These three broad model types attempt to maintain walking and balancing under ideal conditions but cannot adapt to uneven terrain or large push disturbances (i.e., resulting in the IP model falling down over time).

When mapping the point-mass for the IP model to a *biped*, we can use an approximate or exact solution. For example, Kajita et al. [KKK*01b] and Yin et al. [YLvdP07] placed the point-mass at the "hip" position to maintain a constant height above the supported foot, while Coros et al. [CBvdP10] and Tsai et al. [TLC*10] used the true articulated character's centre-of-mass for the point-mass placement by analyzing the current character's dynamic state each frame.

3.4 Motion Fundamentals

3.4.1 Inverted Pendulum Principle

The concept behind the inverted pendulum is based upon the conservation of energy. If we visualize a point-mass sitting on a massless vertical fixed length-leg (i.e., with pin-sized feet). Eventually, the fixed length-leg will begin to tilt and fall to one side. This causes the point-mass to lose height and consequently lose potential energy while gaining kinetic energy (i.e., velocity). After some time the leg will reach an angle (e.g., 20 degrees) from vertical. At this point, we instantly flip the leg, so the pin-point foot is at a new positions (i.e., vertically mirroring the leg location) causing a 'pole-vault' like action. The point-mass will rise in height while losing velocity causing the kinetic energy to be converted back into potential energy, as shown in Figure 3.3.

In a perfect world, that is, on flat ground with no slipping or air resistance, fixed length-legs, and continuous sampling (i.e., no discrete time-step errors or numerical inaccuracies), the IP stepping model would remain in a "perpetual" locomotive motion (i.e., constant regular stepping), see Figure 3.2. The IP model's step transitions are uncomplicated and computationally fast to compute, while adjusting the stepping angle allows the IP model to gain or lose energy and hence speed during step transitions.

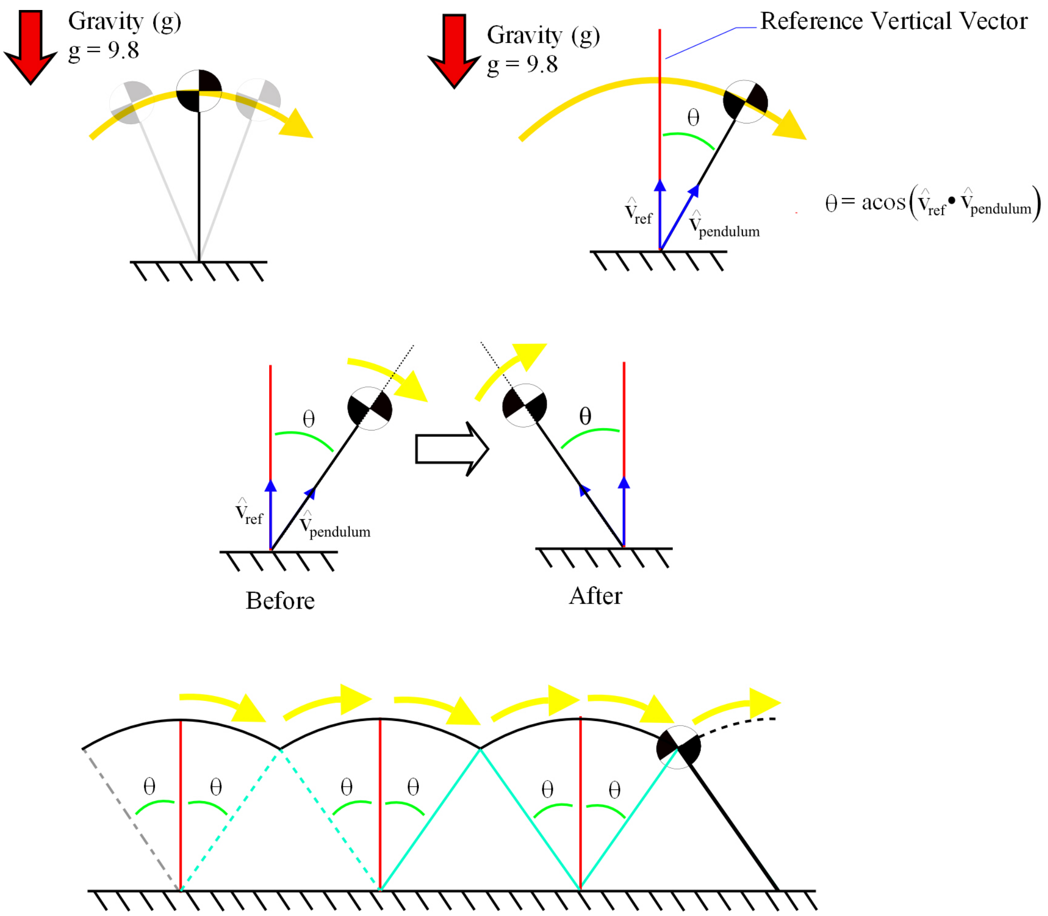


Figure 3.3: Fundamental Inverted Pendulum Stepping Logic - The continuous stepping motion for a simple IP model in ideal conditions by swapping the leg at a specific angle (i.e., we assume the IP begins with a constant velocity that it maintains through stepping).

3.4.2 Mathematics and Control

The IP model is able to sustain a constant stepping motion by conserving the total mechanical energy of the system. Whereby, as the mass falls and rises the IP model's *total energy* remains constant while the potential and kinematic energy is converted back and forth, as shown by Equation 3.1 with reference to Figure 3.4.

$$\begin{aligned} P_0 + K_0 &= P_1 + K_1 \\ \frac{1}{2}mv_0^2 + mgh_0 &= \frac{1}{2}mv_1^2 + mgh_1 \end{aligned} \tag{3.1}$$

where m is the mass, v is the velocity, and h is the height above the ground with the subscript indicating before and after the step transition (K and P represents the kinetic and potential energy).

We extend the IP model's ability to generate balanced biped walking motions for various body types (i.e., thin, fat, short). As shown by Pratt and Tedrake [PT06], we begin by assuming a constant leg-length to calculate an approximate foot placement location to attain a controllable upright stepping motion. We derive the equation for this foot placement location so that it can control the velocity (i.e., speed) for the IP model between step transitions as done and shown by Coros et al. [CBvdP10] to create generalized walking motions.

The formulation begins by calculating the foot placement distance d that will bring the mass to a complete stop (i.e., zero velocity) when it is vertically above the new foot location (see Figure 3.4), since **a 'zero walk' provides a logical starting point for editing the end-users walk**. We look at the *current* state of the IP model (i.e., *before* the foot is placed at a new location) and we assume an *after* velocity of zero (i.e., 'stop') and the geometric properties (i.e., Pythagoras), given below in Equation 3.2.

$$\begin{aligned} v_1 &= 0 \\ h_1 &= L = \sqrt{h_0^2 + d^2} \end{aligned} \tag{3.2}$$

We then substitute v_1 and h_1 from Equation 3.2 into Equation 3.1 and re-arrange for d, so we get Equation 3.3.

$$\begin{aligned} d &= v_0\sqrt{\frac{h_0}{g} + \frac{v_0^2}{4g}} \\ &= \frac{v_0}{2g}\sqrt{(v_0^2 + 4gh_0)} \end{aligned} \tag{3.3}$$

where g is the gravitational constant (i.e., $9.81m/s^2$).

The stepping distance using Equation 3.3 can cause initial leg-length deviations. For example, if the stepping distance is less, then the distance between the centre-of-mass and the new foot position will be less than the leg-length. As we place our new foot at the target location and begin to pole-fault forward, we must return the leg-length back to its original length. We accomplish this by linearly correcting the leg-length as the stepping motion moves forward to produce an elliptical arc. Since we only take into account the current and final velocity and neglect minor deviations in leg-length, we introduce a small error which must be corrected by a scaling factor. The error could, all the same, be reduced by sampling the position, height, and velocity multiple times along each step transition and incorporate the error into the final solution, as done by Liu et al. [LLGC12]. Alternatively, to keep our approach as computationally simple and uncomplicated as possible, we incorporate a basic scaling factor into the stepping distance (e.g., 0.9) to reduce the error to an acceptable tolerance. In practice, the scaling factor allows the simulation to produce consistent stable stepping motions, which we show throughout this chapter (e.g., velocity deviate around a desired velocity during step transitions).

Note, if the stepping distance is less than d, the velocity will increase, and if the stepping distance is greater than d the velocity will decrease (e.g., walk backwards). Based on this understanding, we can bias the stepping distance to control the final stepping motion as demonstrated by Coros et al. [CBvdP10] and shown in Equation 3.4.

$$
\begin{aligned}
d_{final} &= d - \alpha \, v_{diff} \\
v_{diff} &= v_0 - v_{desired}
\end{aligned}
\tag{3.4}
$$

where α is a *constant* (i.e., typically around 0.1), $v_{desired}$ is the desired velocity, and v_{diff} is the difference between the current and desired velocity magnitudes. Limiting the maximum step distance d to less than $0.6L$ (similar Coros et al. [CBvdP10]).

The error correction feedback mechanism in Equation 3.4 and show in Figure 3.8 allows speed up or slow down the model to match the desired velocity (i.e., a controlled walking speed). For example, if the pendulum velocity increases the error feedback causes a corresponding increase in step length which reduces the velocity - allowing us to create a stable and controllable walking speed.

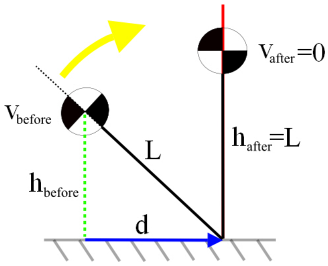


Figure 3.4: Fundamental Inverted Pendulum Step Transition Paramaters - At any moment we can take a snap shot of the current IP model's paramaters (e.g., velocity, height) and use them to calculate the location of the new foot position to achieve the desired control (i.e., the velocity in the steered direction).

The pendulum stepping mechanics are based on the conservation of energy (i.e., the conversion of kinetic and potential energy during foot support transitions). Equation 3.3 enables us to inject or remove energy from the system by deviating the stepping distance. For example, as mentioned, if we reduce the stepping distance less than d the velocity will increase (i.e., add energy to the system), while if the stepping distance is increased greater than d the velocity will decreate (i.e., loss energy from the system). This is valuable as it allows us to compensate for loss of momentum (i.e., energy) due to the step transition impulse (i.e., the abrupt change in direction of the centre-of-mass when changing leg).

3.4.3 Step Direction

During the step transition, the foot is moved to a new location so the mass can be supported without falling down. The foot step direction is critical for the biped to remain balanced. For example, if the body falls backwards due to disturbances, the step direction must move backwards to prevent the body from falling over. We use the body velocity and the direction from the foot to the body to determine the foot placement direction as shown in Figure 3.5. However, if the movement is in an undesirable direction, then we cancel out the movement in that direction. This allows us to control the direction and speed of movement.

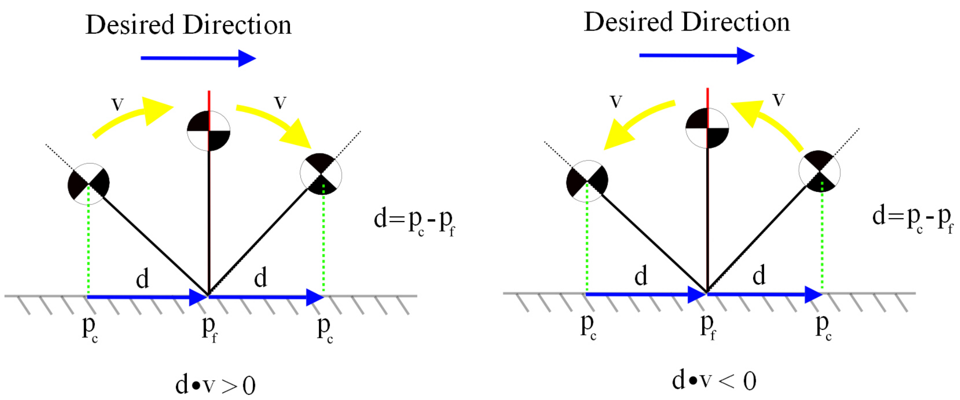


Figure 3.5: Stepping Direction - The body velocity and the direction from the foot position to the body, we can determine if we need to place a foot forward or backward to remain balanced.

Due to pin-point feet, the IP model must keep stepping (i.e., remain in a perpetual state of motion); however, due to the feedback velocity, the model can maintain a stably controlled stepping motion (e.g., walking speed) while compensating for inaccuracies from numerical errors, such as floating point approximations and discrete integration calculations.

We can accommodate *uneven terrain* without modifying Equation 3.3, by having the character extend their leg-length when it is above the foot support region as shown in Figure 3.7. Alternatively, we can modify Equation 3.3 to accommodate sloping terrain (e.g., stairs) and changes in potential energy by incorporating height deviation as shown in Figure 3.6 and Equation 3.5. For example, to maintain momentum when climbing in height a shorter step distance d would be calculated using Equation 3.3. Furthermore, for a continuous set of steps (i.e., stairs) the step distance must shift between short and long strides as shown in Figure 3.6 to gain kinetic energy due to an increase in height and hence an increase in potential energy. In retrospect, this enables us to solve the issue of increasing and decreasing terrain slopes.

$$\begin{aligned} v_1 &= 0 \\ h_{0s} &= h_0 - s && \text{(Subtract terrain height change)} \\ h_1 = L &= \sqrt{(h_{0s}^2 + d^2)} && \text{(Sub in Eq. 3.1)} \\ \frac{1}{2}m_0 v^2 + m_0 g h_{0s} &= mg(\sqrt{(h_{0s}^2 + d^2)}) && \text{(Solve for d)} \\ d &= \frac{v_0}{2g}\sqrt{(v_0^2 + 4gh_{0s})} \end{aligned} \tag{3.5}$$

where the subscript 0 and 1 indicate before and after with $s = h_{step}$=step height (as shown in

Figure 3.6). For when s is zero the formula simplifies down to Equation 3.3.

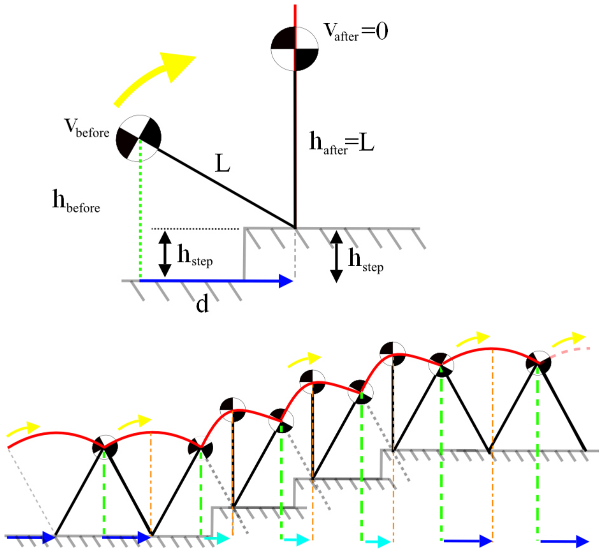


Figure 3.6: Walking Up Stairs - Extending the basic model to include deviation for changes in terrain height (e.g., uneven terrain, such as stairs or sloping ground). The leg-lengths are extended back to their rest length when the centre-of-mass is above the support foot.

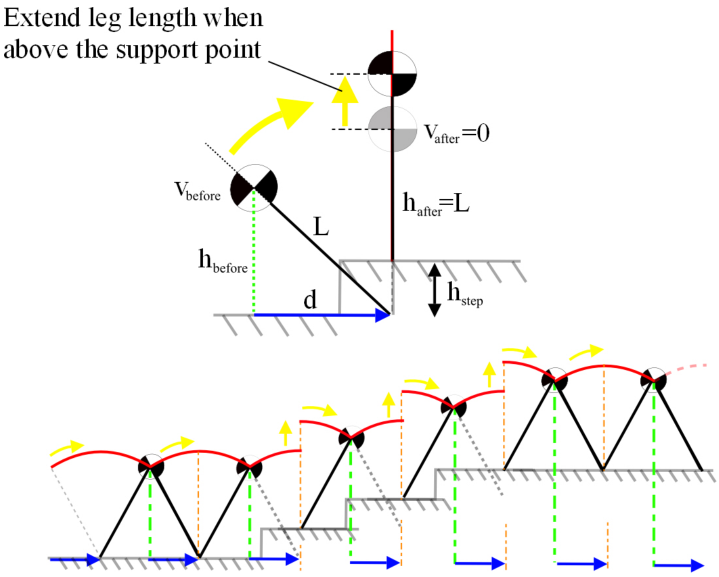


Figure 3.7: Walking Up Stairs - When walking up stairs, we vary the leg-length to mimic an active leg muscle. The leg-length is re-extended when it is vertically above the support foot (i.e., analogous to leg muscle actively pushing the body upwards for each step).

The approach of varying the leg-length to handle terrain height changes has been presented before. For example, Guocai et al. [LLGC12] (i.e., from robotics) showed how a simple 2D balancing biped could handle disturbances (i.e., push forces) while walking on uneven terrain and maintaining a controlled walk speed, by means of an extended IP model that changed its leg-length between foot support transitions.

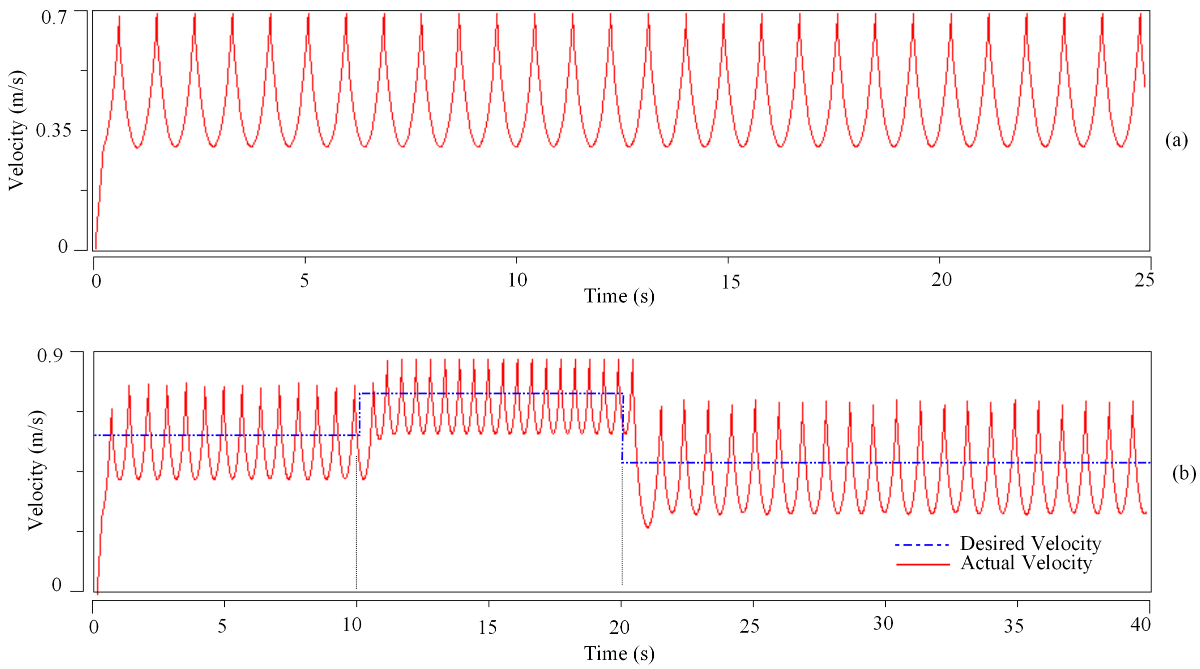


Figure 3.8: Steady Walk Simulation - The figure shows the centre-of-mass velocity step transitions, the velocity reducing and increasing for each step. (a) Walking at a steady constant pace, (b) following expected walk velocity (speeding up, then slowing down); (with time-step=0.01, mass=70kg, and leg-length=1.0m). The important information to recognize is that the velocity does not increase or decrease gradually - even when we alter the desired velocity to increase or decrease the walking speed.

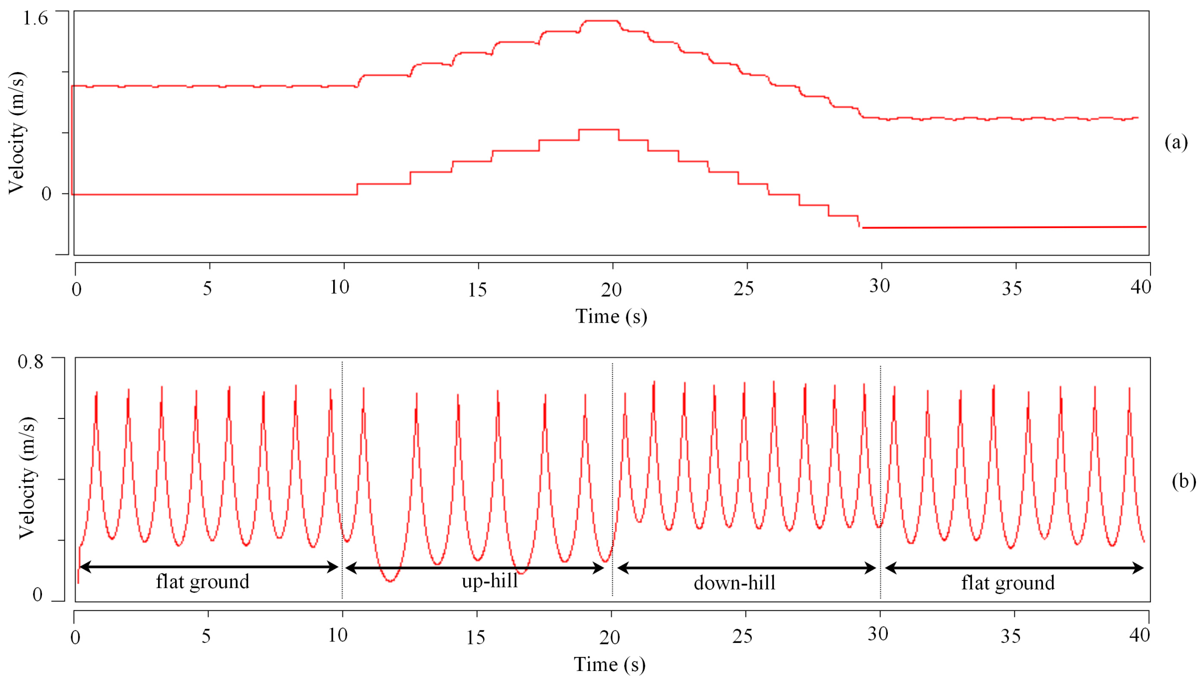


Figure 3.9: Flat, Up-Hill, and Down-Hill Velocity - The velocity when walking on flat, up-hill, and down-hill ground, with (a) showing the centre-of-mass and ground height, and (b) the corresponding velocity. (time-step=0.01, mass=70kg, leg-length=1.0m, terrain height increment/decrement=0.1m, average step length 0.2m).

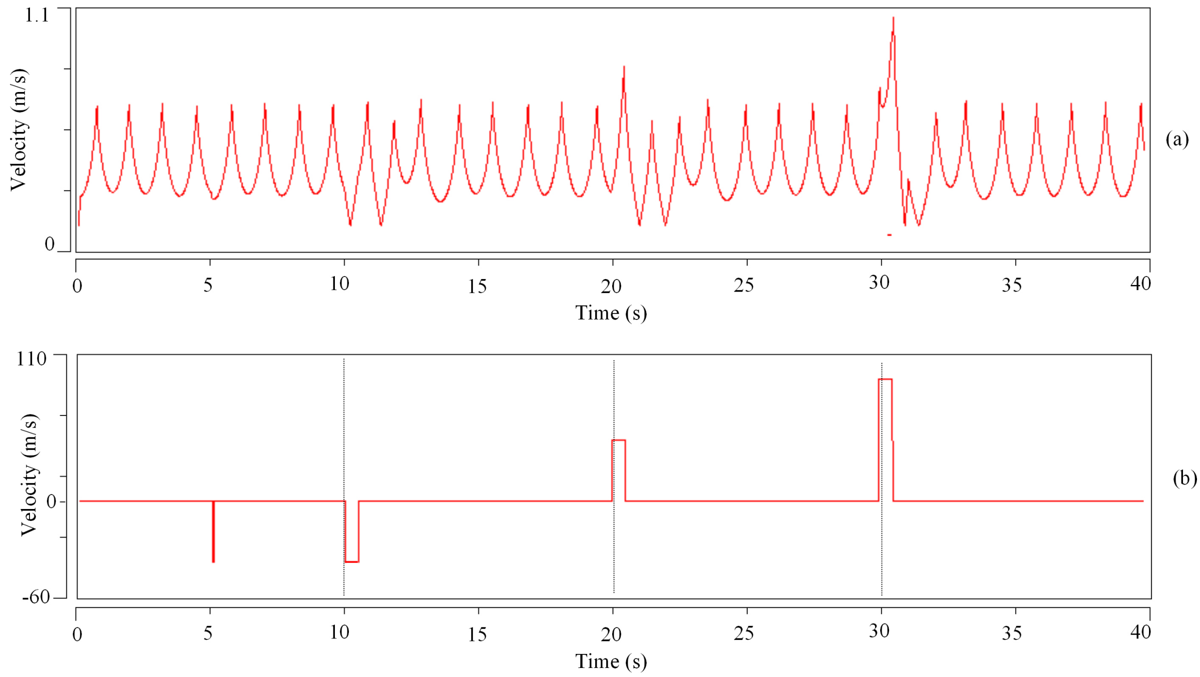


Figure 3.10: Disturbance Force Applied While Walking - (a) Biped walking velocity during push force disturbances, and (b) disturbance force magnitude, direction, and time; (with time-step=0.01, mass=70kg, and leg-length=1.0m).

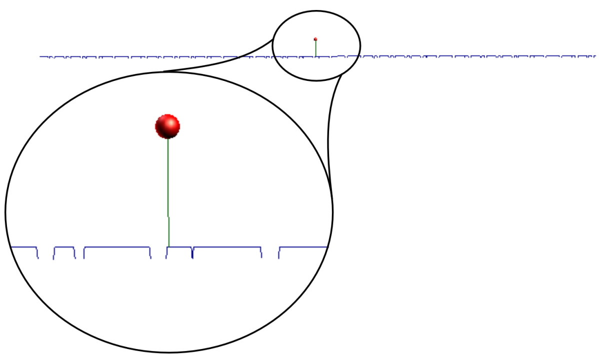


Figure 3.11: Terrain Ground Gap Stepping - The model has to choose a suitable foot position or let the body continue moving forwards. If left to move, the model can re-attempt to find a more desirable foot position that is more adequate for preventing the character from falling into the gaps (Figure 3.12 shows the simulation graphs).

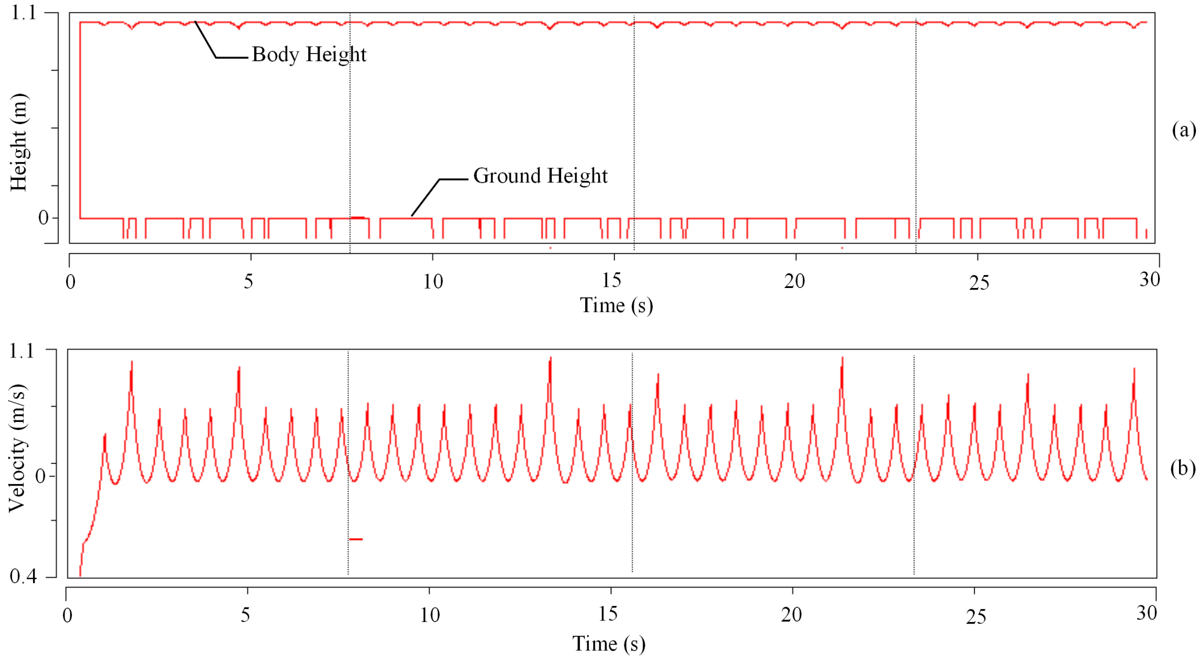


Figure 3.12: Terrain Ground Gaps Simulation - (a) The height of the body and the terrain, and (b) the velocity of the body (gap 0.01m to 0.1m); (with time-step=0.01, mass=70kg, and leg-length=1.0m)

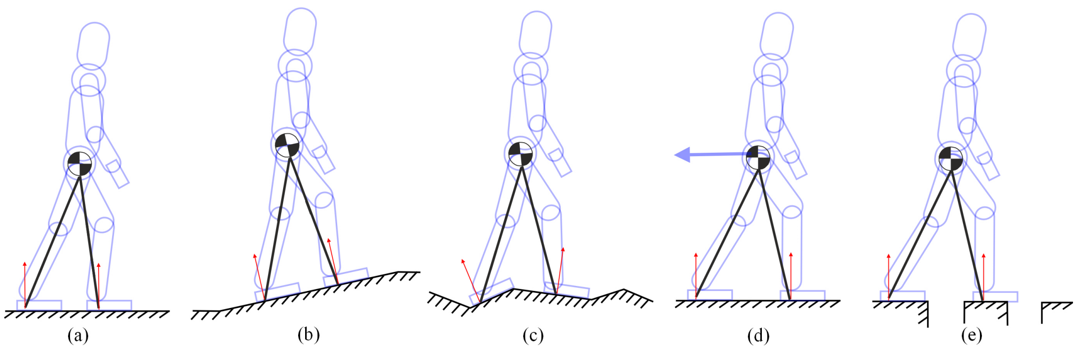


Figure 3.13: Evaluating the Capabilities of the Stepping Model - Walking at a controllable speed under a variety of conditions, such as pulling/pushing forces and terrains height deviations that include inclines (approx. 25 degrees) and gaps (approx. 0.2m). (a) Flat terrain, (b), sloping terrain (e.g., stairs and hills), (c), uneven ground, (d) constant pulling or pushing force, and (e) terrain gaps.

3.4.4 Walk Velocity

3.4.4.1 Constant Walk Velocity

Given a constant velocity our simulations observed four walk patterns:

1. when the desired velocity is zero, the IP will rock back and forth - while the IP model will remain balanced it will persistently take small steps
2. with a very small desired velocity (i.e., less than 0.3m/s) the IP model will walk but occasionally rock back and forth, because the walk velocity is too low
3. steady walk pattern
4. a large desired walk velocity (i.e., greater than 1.5m/s), we get an unstable walking pattern and can lead to the IP model eventually falling over

Hence, for a stable walking pattern, we try to keep the desired velocity within a certain range. Figure 3.8(a) shows walk velocity of the IP remains constant and consistent on a flat terrain.

3.4.4.2 Varying Walk Velocity

We control the IP model's desired walk speed by altering the desired velocity (i.e., in Equation 3.4). Figure 3.8(b) shows the model's speed adjusting through multiple steps to meet desired speed requirements (e.g., speeding up or slowing down).

3.4.5 Disturbances

The IP model possesses the valuable ability to adapt and recover from external force disturbances (e.g., pushes). For example, we applied forward and backward push disturbances between 50N to 100N for 0.1s to 0.5s causing sudden changes in walk velocity as shown in Figure 3.10. While small impulse forces (e.g., for 0.1s) would only cause minor velocity disruptions, larger forces of either greater magnitude or duration would result in the body falling back and requiring it to take one or many steps to correct and return to regular walking in the desired direction and speed. (In contrast, it should be noted, that Tsai et al. [TLC*10] model applied maximum external force of 700N for 0.2s).

3.4.6 Non-Flat Ground

3.4.6.1 Slopping Terrain

Walking on terrain that has a changing slope (e.g., up-hill, down-hill, or stairs). As shown in Figure 3.9, as the terrain slopes upwards and downwards the corrective stepping ensures the model remains balanced and continues walking at a consistent average speed. We experimented with an upward/downward slope of 26.5 degrees. In relation to existing publications based on the IP model, Tsai et al., [TLC*10] model walked on slopes of a maximum of 15 degrees, Coros et al., [CBvdP10] 0.15m high stairs (i.e., 26 degrees), and Mordatch et al., [MdLH10] maximum of 15 degrees).

3.4.6.2 Peg-Leg

The solution for terrain height deviates (i.e., stairs and sloping ground) can be applied to characters with unequal leg-lengths. For example, in the simulations, we set both the character's default leg-lengths to 1m; however, we could make the left-leg 1.0m and the right-leg 0.9m and treat the deviation as a terrain height difference, so we can maintain a controlled steered walking velocity.

3.4.6.3 Random Ground Height Deviations

The stepping model has the ability to adapt to uneven terrain. We experimented with ground height deviations of +/- 0.1m. In the simulations, the model could maintain a relatively constant walk speed with minor velocity fluctuation while, most of all, maintaining balance (i.e., the limitations are based upon the permissible leg-length changes).

3.4.7 Pulling-Pushing Objects

We can modify the stepping logic so that it can maintain balanced locomotion while pushing or pulling an object. We compute the ratio between the downward force from gravity so that cancels out the pull (or push) force and keeps the body in equilibrium (i.e., a state of not moving) as shown in Figure 3.14. We calculate the scale factor between the gravitational force magnitude and the pull force to correct the stepping distance from Equation 3.3.

$$\begin{aligned} (\hat{u} \cdot \hat{a})||F_g|| &\geq (\hat{b} \cdot F_p) \qquad \text{(halt `or' move in the walk direction)} \\ \beta &\geq \frac{(\hat{b} \cdot F_p)}{(\hat{u} \cdot \hat{a})||F_g||} \end{aligned} \tag{3.6}$$

where g is the gravitational constant (i.e., $9.8m/s^2$), m is the mass, F_p is the external pull or push force exerted on the body, and $\hat{u}, \hat{a}, \hat{b}$ are unit vectors as shown in Figure 3.14. Hence, we modify the stepping distance Equation 3.5 so that it can continue to remain balance and walk while under the influence of a push or pull force (e.g., while dragging a box) as shown below in Equation 3.6.

$$d_{push} = d + \beta k_p \tag{3.7}$$

where k_p is a scaling factor (e.g., 0.4), d is the stepping distance calculated using Equation 3.3. Whereby, the pull (or push) force reduces (or increases) the step distance sufficiently to compensate for the pull (or push) and maintain relatively constant walk speed in the desired direction. The stepping distance bias is based on an instantaneous state and does not include changes in potential and kinetic energy as the body travels through an elliptical arc trajectory. However, the approximation is close enough (i.e., $\geq 90\%$) and in practice, the scaling factor reduces the error sufficiently to ensure the desired speed and walk direction are ultimately achieved.

The step size is limited (e.g., 0.5m), so if the pull/push force becomes too excessive the body will be pulled/pushed backwards. While the body will still remain balanced, it will be unable to gain enough potential energy to pull/push the object. In our model, we neglect any ground frictional force, which could be included to gain greater pulling (or pushing) power. For example, with a maximum step size of 0.2m, leg-length of 1m, and a mass of 70kg the *horizontal* pull force should be much less than 137.2N. (Note, Coros et al. [CBvdP10] pulls/pushes an 80kg object with a ground friction coefficient of 0.2 and connects the hands to the handles

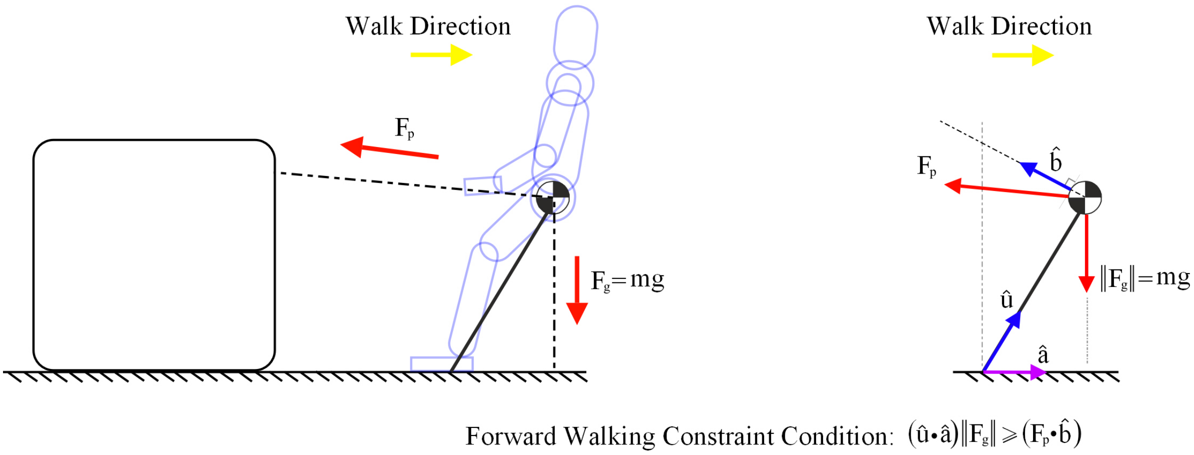


Figure 3.14: Pulling or Pushing an Object - The IP stepping equation can be modified so that it can maintain a balanced walking momentum while pull or push force is being applied to the body (i.e., simulating a pushing or pulling action).

using a spring-damper model).

3.4.8 Terrain Gaps

We add gaps (i.e., holes) to the terrain to limit where the foot can and cannot be placed. When it is time to calculate the desired step distance and foot placement position; if the foot placement position is unavailable due to a gap in the terrain, we have two choices. Either reduce the stepping distance and increase the walking velocity or keep the old foot placement and let the mass move marginally forward before trying again at calculating the foot placement position. Hence, we search for a foot placement position ahead and behind current stepping distance stride. If a foot placement position is available ahead of the stepping distance (i.e., within an acceptable range), we increment forward until we reach it and take the step. Otherwise, we use the lesser stepping distance. If neither ahead nor behind is available, we need to move forwards until, we find a position, or we fall over.

3.5 2D to 3D (Sagittal and Coronal Planes)

In Equation 3.3's simplest form, it does not provide any steering information (i.e., 3D direction). However, similar to Coros et al. [CBvdP10], we accomplish steering by splitting the velocity into the Sagittal and Coronal plane and calculating separate distance parameters for

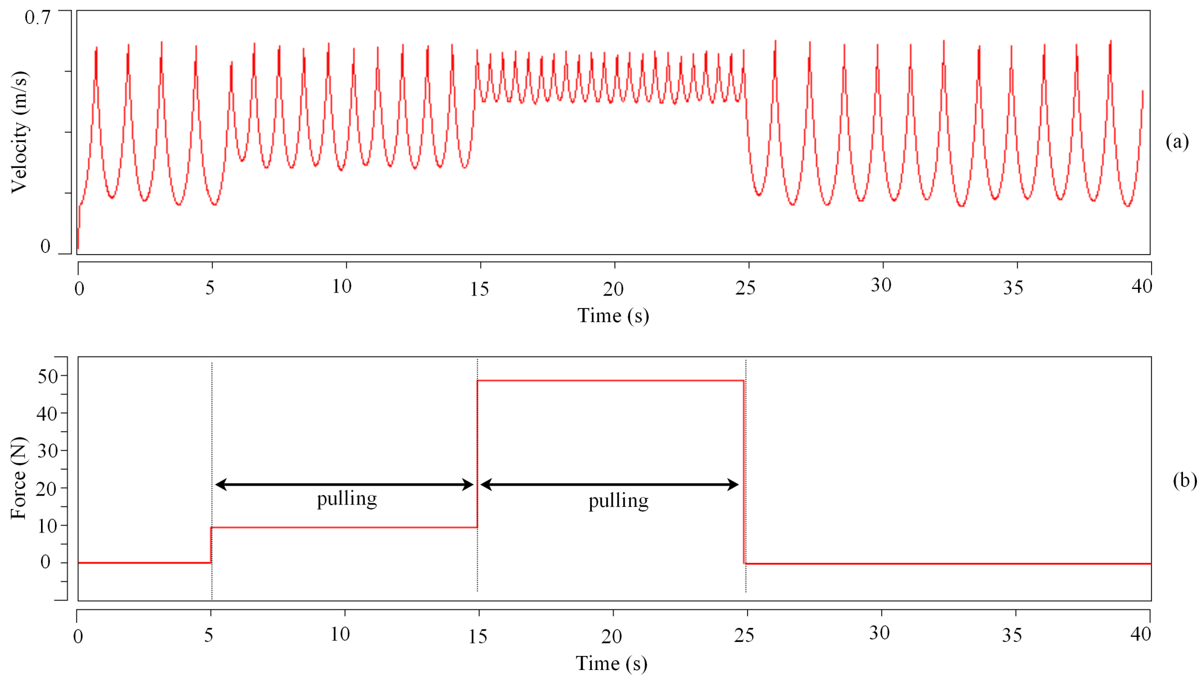


Figure 3.15: Constant Pulling Force (i.e., Pulling an Object) - A constant pulling force is applied to the body of varying magnitude (0, 10N, 100N) at different times. The simulation shows the velocity is relatively constant while stepping distances are reduced to compensate for the external pulling force.

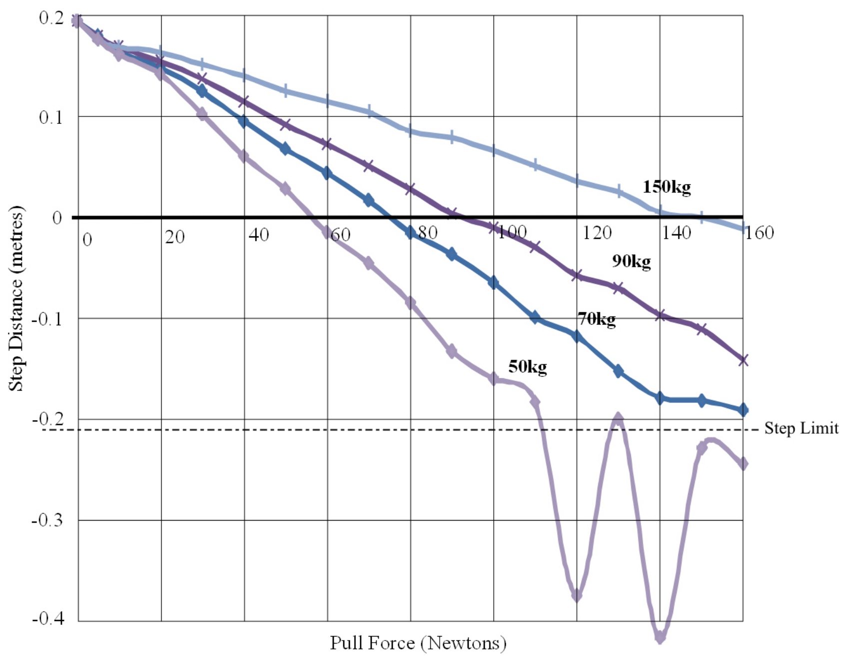


Figure 3.16: Pull Force vs Stepping Distance - Stepping distance varies with different constant pulling force (with leg-length=1.0m, mass=50kg,70kg,90kg,150kg, time-step=0.01). When stepping distance is 0 the pendulum is vertical and the foot must remain behind the centre-of-mass to keep the pendulum walking forward. Initially, when starting, the pendulum is pulled and needs to take a backward step before moving forwards. The step limit specifies the foot range - beyond which we are constantly pulled backwards if we cannot place our foot. As expected, increasing the mass shows we can pull more force due to gravity. For the unstable oscillating motion for the 50kg mass - this was due to the pendulum stepping back and forth at the threshold to counteract the pull force.

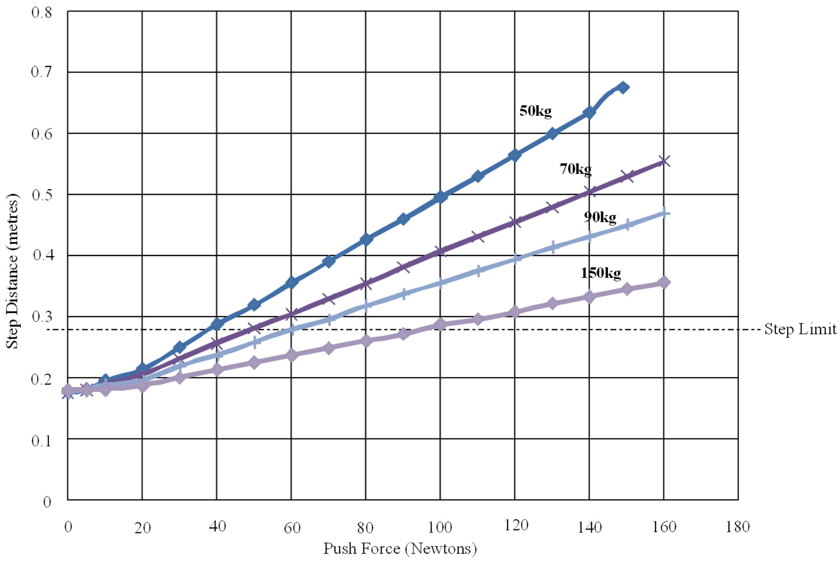


Figure 3.17: Push Force vs Stepping Distance - Stepping distance varies with different constant pushing force (with leg-length=1.0m, mass=50kg,70kg,90kg,150kg, time-step=0.01). To counteract the push force the stepping distance has to increase. Limiting the maximum allowable step distance due to physical constraints (i.e., leg-length) as shown on the graph as step limit (allowable range before we are unable to counteract the push force and need to take a step backwards).

them. Hence, we extended the simple 2D IP model into 3D and take the character's forward direction as the frame of reference; splitting the stepping mechanism into two independent tasks defined as the forward and sideways (i.e. sagittal and coronal planes) stepping distance calculations. Whereby, we can control the stepping motions in a 3D virtual environment (e.g., see Figure 3.18).

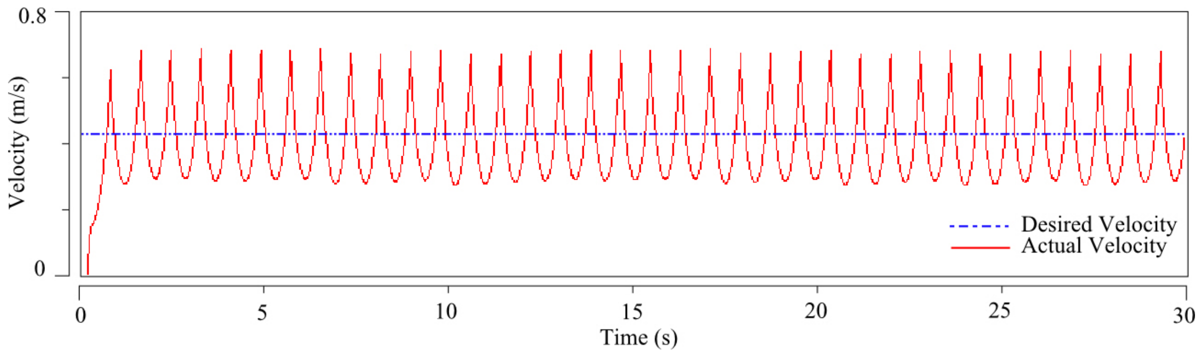


Figure 3.18: 3D Inverted Pendulum Movement - We can apply the basic 2D IP model to 3D. We divide the stepping movement between two perpendicular vectors (i.e., forwards and sideways). We can then control the forward and sideways speed (e.g., for steering). The figure shows the forward direction always facing towards the closes point on a circular path. The IP model continues to follow the path at a constant velocity.

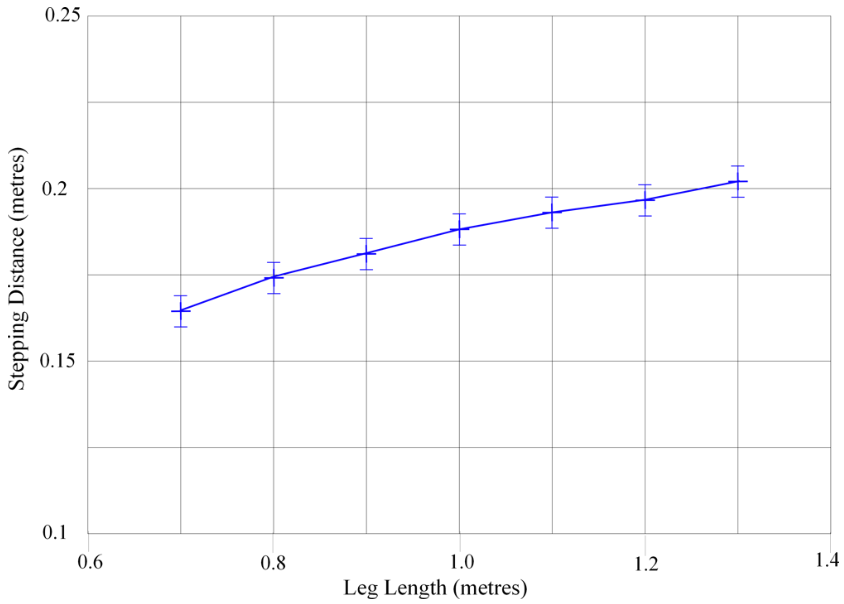


Figure 3.19: Simulation of Leg-Length vs Stepping Distance - We show the linear relationship for different leg-lengths (0.7m-1.3m) with stepping distance for a constant walking velocity on flat terrain. As expected, the feedback system corrects for minor errors from numerical approximations (i.e., minor oscillates around the ideal stepping distance $\delta d \approx 0.02$) to speed-up or slow-down so the final walking velocity is constant and stable. (with time-step=0.01, desired velocity 0.7m/s, and mass=70kg)

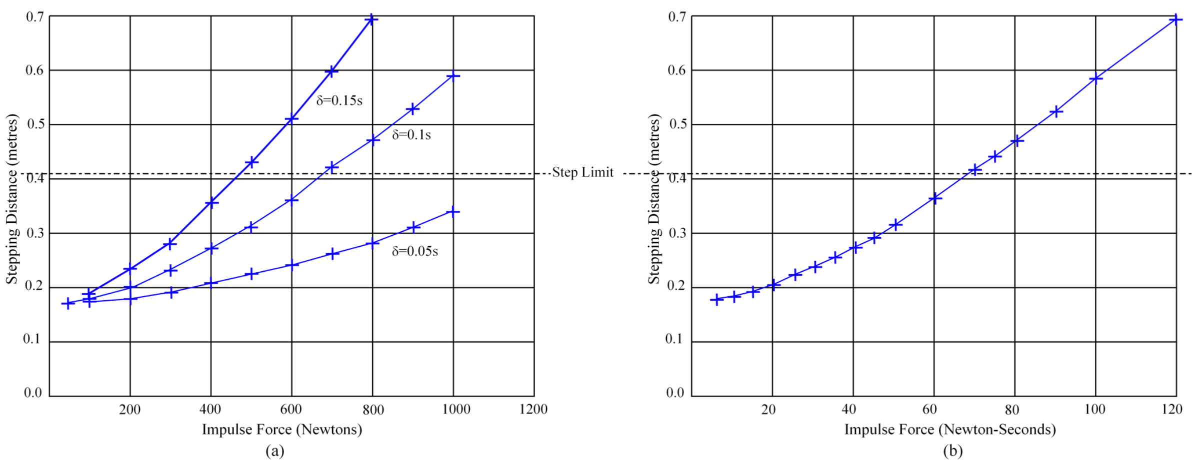


Figure 3.20: Simulation of Force Disturbance vs Stepping Distance - We show the relationship of push disturbances with stepping distance. Where (a) shows the relationship for force and duration, and (b) the impulse force (i.e., force multiplied with the time) for the necessary stepping distance to bring the pendulum to a stop (i.e., upright with velocity of zero); We draw a dotted line to point out stepping limits (i.e., the maximum distance the pendulum can step based on leg-length deviations). (with time-step=0.01, leg-length=1.0m, and mass=70kg)

3.5.1 Simulation Analysis

We simulate and plot changes in leg-length versus step-length in Figure 3.19 to demonstrate the stable stepping over changes in parameters. While we set the default mass to 70kg, however, simulations demonstrate, as expected that deviations in mass (e.g., +/- 50kg), do not effect the stepping distance. We compare and plot the push disturbance based on a single-step (as shown in Figure 3.20), with the assumption of limited stepping distance (i.e., leg-length deviation) - the factor of when a push is applied (e.g., at the start or half-way through a stepping motion) can require a different stepping distance - however, we compare deviations at the same point (i.e., when upright) to illustrate the limitations.

3.6 Discussion

The most significant advantage of this chapter's approach over prior work is: Our low-level model uses **no motion capture data, offline optimization, or pre-processing** while generating fundamental biped stepping information on-the-fly for dynamic environments; including, responding to disturbances, terrain height deviations, and pushing/pulling. In contrast, although some feedback controllers [SKL07, YLvdP07, HWBO95] run in real-time, they require offline model or task specific parameter computing, and while the work by Tsai et al. [TLC*10] overcame this by automatically estimating parameters, it required motion capture data. Whereby, we demonstrate how a simplified pendulum model can be customized and modified to capture significant stepping dynamics, including object interaction (e.g., pulling force). **Our proposed approach is constructed around approximate but effective estimates that can be used to create new motions with reasonable degrees of flexibility.**

This chapter's work shares essential features with a number of recent and impressive pendulum-based techniques. For example, to begin with, there was the inspiring work by SIMBICON [YLvdP07], which was later extended [YCBvdP08] to allow the model the ability to perform variations in style and step over objects. This later led to a number of novel improvements, such as Coros et al. [CBYvdP08] walking in a constrained environment, which was also integrated with multiple controllers for navigation tasks [CBvdP09]; Wang et al. [WFH10] offline optimization to try and reproduce key human features when walking in a straight line. However, in contrast with our approach, the most notable difference is the focus. Instead of taking animation data and mixing or adapting it with simple physics-based models to output character movements, we target the challenging problem of creating animations on-the-fly for interactive environments that possess basic life-like properties with no key-frame data, offline optimization, or pre-processing (i.e., a procedural physics-based approach). In summary, we believe **the pendulum model provides a powerful system for deriving primitive balanced biped stepping dynamics**. In later chapters (i.e., Chapter 4 to Chapter 6) we build-upon and explore enhancement strategies to address and solve distinctive short-comings surrounding the uncomplicated pendulum mechanism to make the solution more usable.

3.7 Conclusion

The low-dimensional IP model provides a computationally fast and straightforward technique for creating fundamental 2D and 3D balanced biped stepping motions. These upright motions hold vital dynamic information for generating or editing character motions, so they are more interactive (e.g., the ability to respond to pushes and terrain height changes), controllable (e.g., speed and steering), and physically-correct. As this chapter has shown, the IP can create controllable robust balanced biped stepping motions that can handle a variety of terrain situations (e.g., holes and sloping terrain), walk at different speeds, push and pull objects (e.g., drag a box), while possessing the ability to compensate for unpredictable disturbances, such as pushes.

Chapter 4

Real-Time Dynamic Balancing and Walking: Elongated Rigid-Body

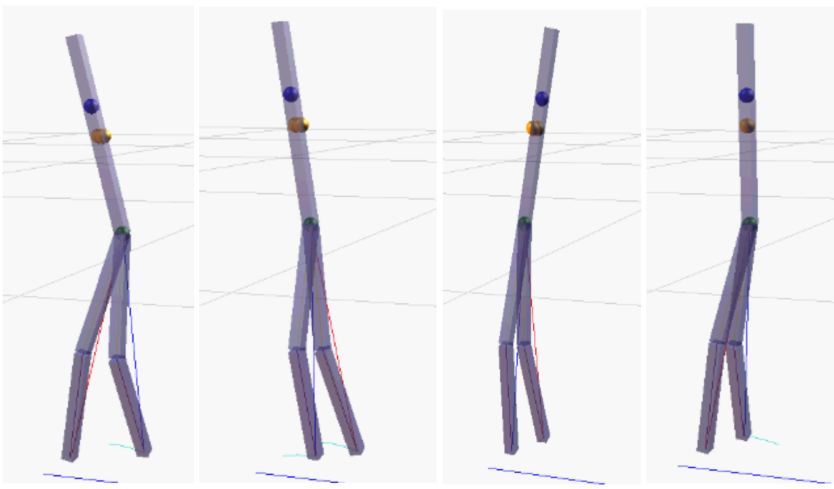


Figure 4.1: Adaptive Stepping - Predictive stepping and posture correction due to random force disturbances being applied to the body.

4.1 Introduction

This chapter extends the lightweight inverted pendulum with an elongated rigid-body for solving the problem of postural feedback and control. As in many immersive style games, there is a need for player-controlled characters to interact with the gaming arena. Such character provision player representations in a game usually take human form. As a player understands how humans may act and move given certain physical environments, unexpected or restricted behavior of a player-controlled character is noticeable. Therefore, a challenging research problem is to ensure player-controlled characters are sufficiently expressive to afford the desired level of game-play while maintaining naturally realistic movements and actions.

This chapter considers a purely controller-based approach for realistic human movement that is rendered in real-time. The key to mimicking human movement is not to attempt to recreate the appropriate physical motion alone, but to augment it with a balance controller for posture alignment. Our technique is prompted by work within theoretical robotics where an inverted pendulum is used for such purposes [MRS08]. We demonstrate that our approach favors a human-like approach to movement in terms of posture when presented with external forces. Furthermore, we extend the theoretical work described in [MRS08] by moving our model from 2D to 3D and providing a real-time implementation.

To overcome the problems of unnatural posture control in the IP approach an extended rigid-body may be used to represent the trunk (enhanced IP). The hip point would be located at the base of the trunk, the centre-of-mass (COM) at the mid-point of the trunk and a virtual pivot point (VPP) point located parallel to the extended body and positioned above the COM. This work was initially described in [MRS08] and demonstrates a correlation between natural and simulated ground reaction force GRF. In addition, [RS10] describes the resultant motion demonstrated in [MRS08] is more realistic than the IP without an extended rigid-body.

4.1.1 Chapter Contribution

In this chapter, we extend the work of the enhanced IP described in [MRS08]. The work in [MRS08] was 2D and only provisioned the calculated expectation of the simulation. In the field of robotics, this initial investigatory stage is usual as the end goal is to actually build a robot. However, for video-games, we want a fully interactive 3D run-time implementation. This is what we achieved, allowing our character model to be exposed to forces and be manipulated by a player. The model itself has no data-driven aspects with all foot placements and body movements created dynamically.

The fundamental contribution of this chapter is the demonstration of the enhanced 3D IP as an option for fully interactive realistic human locomotion in real-time without any data-driven requirements.

4.2 Related Work

In this section, we give a brief overview of the evolution of controller-based approaches to human motion. / We then describe an approach that provisions suitable locomotive control methods that are realistic. Finally, we describe the enhancements made by this chapter.

4.2.1 Inverted Pendulum

As show in Chapter 3, the inverted pendulum (IP) can generate predictive foot placement information for balancing characters in 2D and 3D (e.g., [TLC*10, KKK*01b]). IP can be used for characters balancing while standing still or during locomotion. The IP motion is constrained to move along an elliptical path. To push the character upright a force is exerted on a character's mass from the centre of the foot that is in contact with the ground. As demonstrated in [KT91, KT95], the character's mass is concentrated into a single point (centre-of-mass (COM)) for the required calculations. To encourage a more natural motion a spring may be associated with the balancing leg [BF93, FFW00]. This makes the foot trajectory resemble a bouncing ball that mimics, to a visually convincing level, a running motion. This approach is described as a spring loaded inverted pendulum (SLIP).

The IP has been used as a calculation of balance to form the basis on which other aspects of human motion may be constructed [SKL07, AdSP07, TLC*10, RH91, PCCDP01, KKK*03, MdLH10]. However, as the **standard IP model uses a single mass-point in its calculations, there is a lack of information present to realistically represent the posture of a character**. An approach to correct this issue would be to use a proportional derivative controller [R*86, ZH02, NBS06] to apply torque to ensure a character's body remains upright. Unfortunately, this corrective torque produces unnatural movements due to the body not receiving feedback from the character's feet, known as the ground reaction force (GRF). For in humans it is this force that causes people to sway and shift as they walk and change direction.

Our Work: Overcome the problems of unnatural posture control in the IP approach with an extended rigid-body for representing the trunk (i.e., an enhanced IP). The hip point would be located at the base of the trunk, the COM at the mid-point of the trunk, and a virtual pivot point (VPP) point located parallel to the extended body and positioned above the COM. This work was initially described in [MRS08] and demonstrates a correlation between natural and simulated GRF. In addition, [RS10] describes the resultant motion demonstrated in [MRS08] more realistic than the IP without an extended rigid-body.

4.3 Overview

In this section, we describe our method for achieving realistic character motion in real-time. As we build directly on suggestions found in [MRS08], we describe the IP model in more detail and then the enhanced version. This allows us to say quite specifically what enhancements we have made. We then describe how balancing, motion and the handling of uneven surfaces (e.g., stairs) may be achieved using our approach.

4.3.1 IP Mechanics

For the purposes of balance control, we consider a human body to exhibit similar qualities to an IP. A human standing on a single-leg would be equivalent to an IP assuming the body is represented by a single COM linked to the ground contact point (centre-of-pressure (COP)). In this model, we assume the legs have no mass. This trivializes the calculation significantly while still affording the desired motion. We assume that there is no ground slippage and a single point replaces the foot.

The IP provides a simple technique for predicting where a character should place its feet to remain balanced. The diagram on the left of Figure 4.2 shows how the different elements combine to produce a repeated motion suitable for modeling walking. The springs in the legs represent the fact that we are considering the SLIP approach to improve realism as described in [BF93, FFW00].

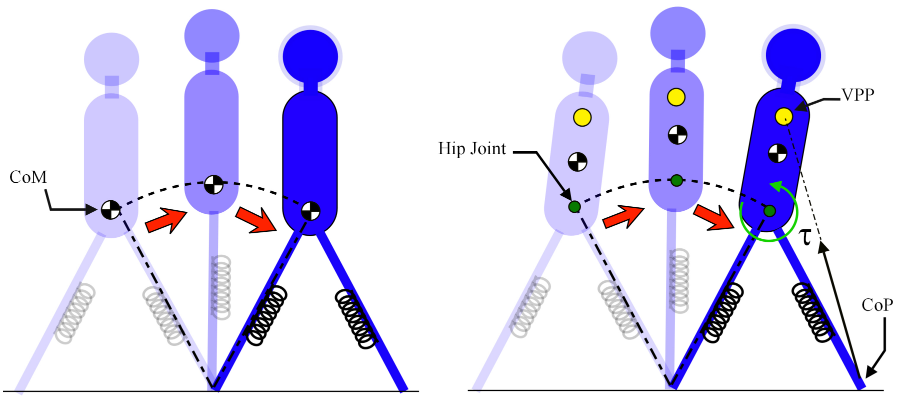


Figure 4.2: Pendulum Motion - (Left) Illustrates IP following a basic elliptical trajectory with no body feedback, (Right) the elongated body following a similar ellipsoid-like trajectory during locomotion, with additional torque-body feedback.

4.3.2 Mechanics of Enhanced IP

The enhanced IP model uses all the properties of the basic SLIP model with some important additional features. The enhanced IP is shown on the right of Figure 4.2. An elongated body is added that represents the trunk [MRS08]. A VPP is added above the hip and parallel to the trunk.

In [MRS08] the author provides a discussion where there is a suggestion that allowing the VPP to move outside the trunk will vary the energy in the model. Varying the energy in such a way would have the effect of speeding up or slowing down the walking motion in characters using the technique in [MRS08]. As [MRS08] dealt only in 2D speeding up and slowing down was all that could be accomplished. *However, we realized that if this approach could be extended to 3D, the VPP could ultimately be used for steering the character.* Therefore, to pursue this line of research our initial technical change made to [MRS08] was to allow the VPP to move outside the trunk to produce responsive balancing motions.

4.3.3 Calculations

The trunk torque is calculated by projecting the force from the leg (i.e., hip-ground spring) onto the VPP (i.e., ground-VPP) to induce a responsive self-balancing posture.

$$\hat{V}_C = \frac{(VPP - COP)}{||VPP - COP||} \tag{4.1}$$

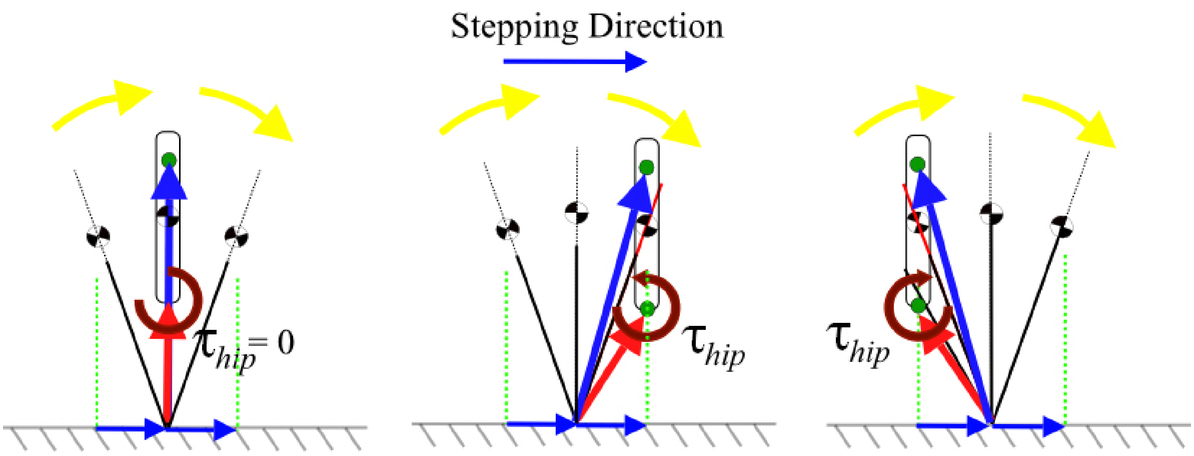


Figure 4.3: Virtual Pivot Point Concept and Reasoning - A visual aid to help clarify the underlying principle of the Virtual Pivot Point (VPP) - consider a stepping cycle. The base and the centre of mass provide a means of calculating a control torque for the base to sway back and forth. However, we need to include the ground reaction force (GRF) magnitude and a Virtual Pivot Point (VPP) for multiple reasons - firstly to ensure the swaying motion is based on the physical situation (i.e., how hard we land on the ground with our foot) and a means of controlling the postural motion.

$$\hat{V}_{TRUNK} = \frac{(VPP - COM)}{||VPP - COM||} \tag{4.2}$$

$$||GRF|| = F_L \cdot \hat{V}_C \tag{4.3}$$

$$GRF = ||GRF|| \; \hat{V}_C \tag{4.4}$$

$$\tau_{hip} = GRF \times \hat{V}_{TRUNK} \tag{4.5}$$

The directional force from the COP to the VPP produces the GRF. Using the equations above, we extract a magnitude and vector to apply a torque to the trunk that feeds back to the hip. This is shown in Figure 4.4.

4.3.4 Managing Stairs and Slopes

Our approach can be extended to allow a character to use stairs and traverse sloping terrain by offsetting the VPP position proportional to the change in step-size. In traditional approaches

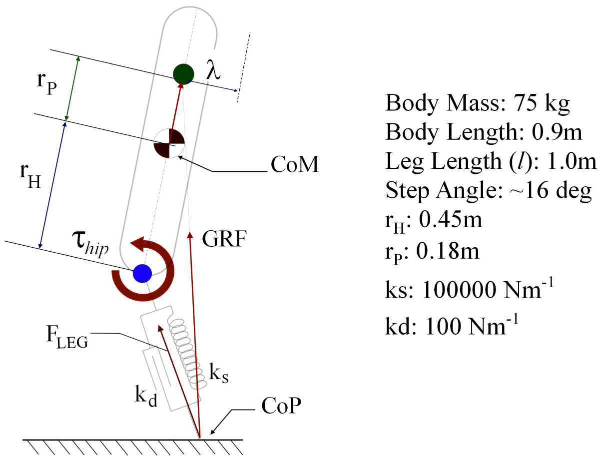


Figure 4.4: Hip Torque - Hip torque calculated using the COP and VPP.

foot placement would be a costly exercise (e.g., re-targeting motion capture data), whereas in our approach, it is a byproduct of the movement.

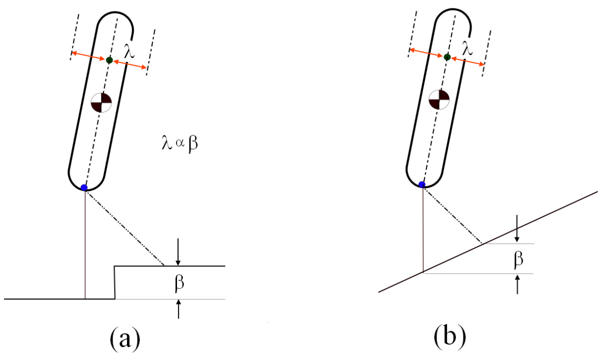


Figure 4.5: Slopes and Stairs - Walking up stairs/steps or a gradual slope.

$$\lambda \propto \beta \tag{4.6}$$

where λ is the offset of the VPP perpendicular to the trunk and β is the change in height from the current standing foot placement (as shown in Figure 4.5).

4.3.5 Controllable Motion

When the VPP is shifted to the left or right, the body sways in the appropriate direction (i.e., with the VPP). If the body is moving in a specific direction, the VPP can provide small temporary offsets to maintain, speed-up, or slow-down a character's overall velocity. As shown in Equation 4.7, we can make changes to the velocity by recognizing that the perpendicular

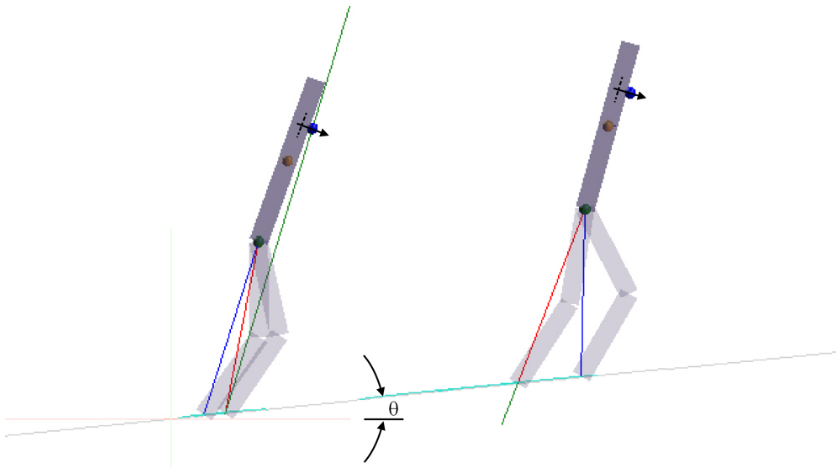


Figure 4.6: Walking Up Slope - Offsetting the VPP enables the character to climbing gradual slopes. The screen capture shows a slope of approx 6 degrees with the VPP horizontal offset at 15cm. However, if the slope is too steep or the VPP is offset too much, it can lead to instability issues (i.e. oscillating motion leading to sharp pelvic torques so the character needs to take a step back to regain balance).

offset relative to the trunk causes a proportional change in the desired speed (v_d).

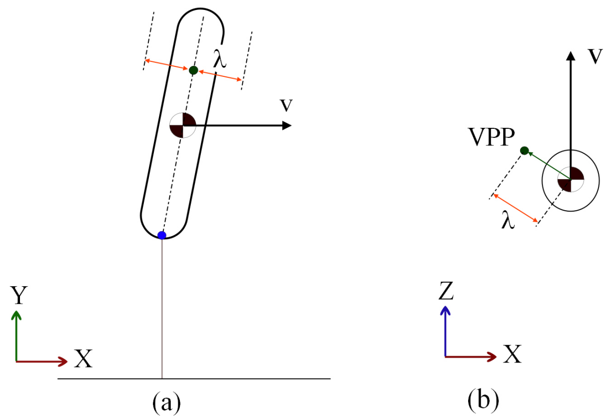


Figure 4.7: Steering - The Virtual Pivot Point (VPP) mechanism introduces an additional control feature. In 2D, as shown in (a), if we move the VPP outside the trunk we can add or remove energy from the system to increase or decrease the walking speed. However, as shown in (b), in 3D, this feature can also be used to steer the character left or right.

$$v_d \propto \lambda \tag{4.7}$$

where v_d is the desired velocity and λ is the perpendicular offset relative to the trunk. Hence, a penalty-based approach can be used to determine λ. The error between the current and desired velocity provide a feedback constant to calculate λ.

Possibly, the strongest reason for our approach is an ability to handle disturbances such as

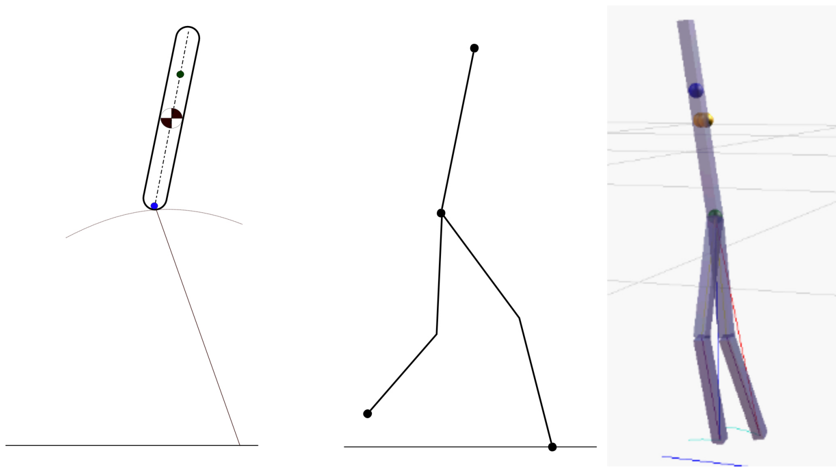


Figure 4.8: Elongated Body Model - The enhanced inverted pendulum generates reference motions (feet, hip positions, and body orientation) that we use to control our five-link biped.

wind or being pushed. A character will respond naturally to the disturbing forces and attempt to take corrective steps to remain balanced. The corrective stepping automatically provides feedback to the trunk orientation to reflect these dynamic changes.

4.4 Results

In this section, we present preliminary results for our method. Our method is used to control a five-link biped model. We consider steering, speed control, and balancing in the presence of pushing. Diagrams are provided to show the different snapshots of movement over time.

4.4.1 Steering and Speed

A turn angle less than 10 degrees kept our locomotion natural, realistic and stable. Spline paths were created for a character to follow and enabled us to evaluate the applicability of using the VPP for steering.

The model was able to produce various types of animation and movement. This included: standing still (balancing), starting to walk, taking 2-5 steps then stopping, and walking by following various paths. Changing the VPP enabled us to create varying gaits dynamically (slow/fast walking).

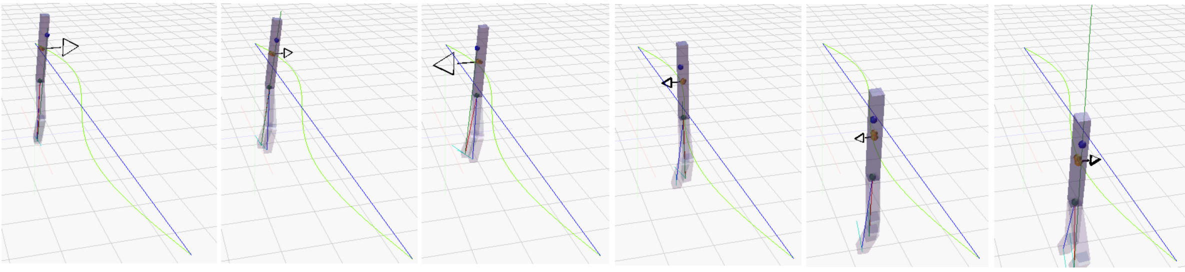


Figure 4.9: Posture Steering - Alter the VPP-offset to control the steering locomotion (i.e., the arrow in the figure indicates the desired steering direction). The character follows the spline (i.e., steers to the left or right) using a penalty-based approach, such that we measure the error between the current position and the desired and offset the VPP accordingly to steer in that direction.

4.4.2 Robustness to Pushes

The underlying stepping mechanism is based on the principles presented in Chapter 3 (e.g., where to place the foot to counteract external force disturbances). However, the stepping response now also contains postural feedback. We applied external disturbances to determine the robustness and viability of our extended pendulum model (an ability to balance realistically). We set our model walking at a stable gait then randomly applied forces between 50N to 100N for 0.1 to 0.4 seconds at the COM. The model passed the test, if the body remained upright. Altering body mass and leg parameters affects the robustness of the model. This is a desired outcome, as eventually, we would like to model a variety of human body types to gain realistic behaviors within crowds.

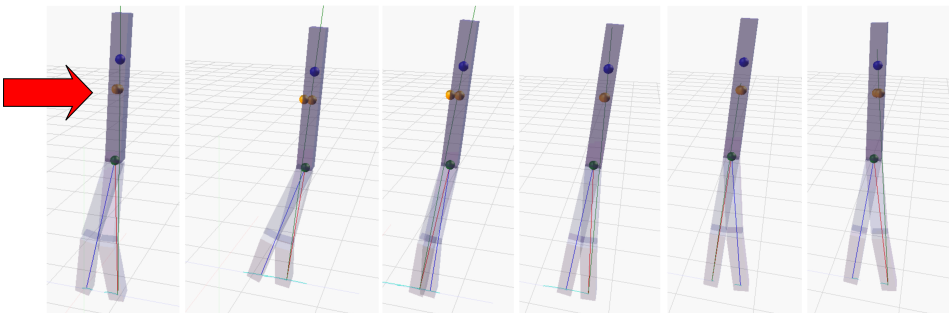


Figure 4.10: Push Disturbances - Incremental screenshots of the biped model responding to a 100N push force being applied to the COM for 0.1s.

4.5 Discussion

The most significant advantage of this chapter's approach over prior work is: While a single point-mass stepping mechanism is a popular approach [PCCDP01, TLC*10, YLvdP07], it does not explicitly deal with upper-body movements and neglects Coriolis and angular forces. We dynamically synthesize the upper-body posture information on-the-fly based on ground reaction forces compared to other popular methods, such as SIMBICON [YLvdP07] and Coros et al. [CBvdP10], that compute lower-body stepping information in real-time, but do not include any intelligent posture control. While upper-body movement has been integrated in with a point-mass stepping model by Tsai et al. [TLC*10] it required motion-capture data. Our simplified elongated body does not require detailed knowledge of inertia properties and has a modest computational speed while being straightforward to implement.

4.6 Conclusion and Future Work

We have described an approach for modeling mid to lower-body 3D human movement in real-time. We require no data-driven elements (e.g., key-frames) to achieve such movement and our approach exhibits robust self-balancing properties. Furthermore, the enhanced IP used in our model also introduces movement that resembles human motion by considering the trunk of the body and hip together in posture calculations. This is the first time a real-time 3D model has made use of this technique. Significantly, we have described a technique that is not only computationally constrained, making it suitable for use in video-games, but also exhibits motion that is human-like without any artist or key-frame intervention.

Chapter 5

Responsive Stepping: Foot Control

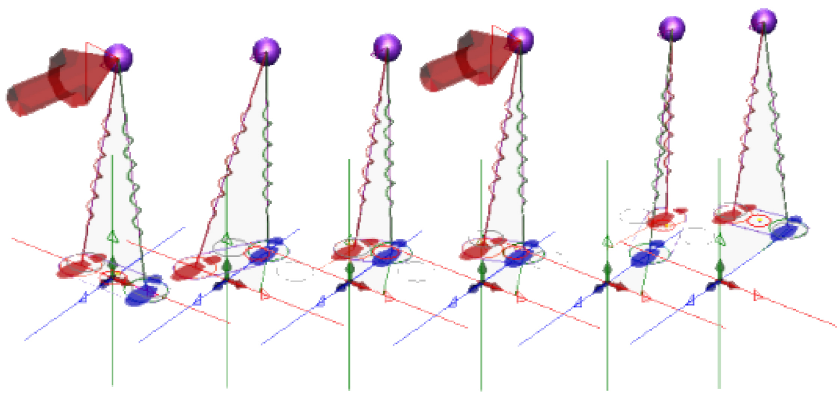


Figure 5.1: Push Response - Illustrate the basic model standing before being pushed twice and having to take corrective steps before returning to a relaxed stance.

5.1 Introduction

In this chapter, addresses the problem of why we need feet and a number of inherent problems of the uncomplicated inverted pendulum stepping model. We provide a novel solutions without sacrificing the computational speed or uncomplicated nature of the underlying stepping mechanism. We solve the problems by incorporating different feature enhancements, such as a simplified support region, foot-ankle feedback, and a pelvis reference direction. The simplifies foot support region (e.g., using spheres and capsules) solves a number of inherent shortcomings and provides additional foot and balancing information (e.g., position, orientation, and path trajectories) in conjunction with an ankle torque feedback mechanism to remedy issues, such as, pin-point ground contacts and allow us to create a more controllable and robust solution.

We use this approach to generate controlled interactive character movements for a fully articulated skeleton body. While the inverted pendulum model provides a fast robust stepping solution, we address the question of *control*, so the solution can be of practical importance and useful in future implementations (e.g., carefully placing the foot at specific locations while walking at different speeds and remaining balanced and in control - even during changing terrain and random external environmental disturbances).

The low-dimensional inverted pendulum model on its own has a number of limitations (e.g., pin-point feet and steering inability) and must be combined with a control mechanism (e.g., foot, hip, or ad-hoc feedback forces) to make the model a viable solution for generating controllable stepping motions. The stand-alone pendulum stepping model limitations, include, a continual state of motion (i.e., always needs to keep stepping to remain upright and balanced); pin-point feet (i.e., no support area or ankle torque); no feet or pelvis orientation information; no postural information (i.e., upper-body orientation); mass-less legs (i.e., no inertia or drag when positioning feet); no feet trajectory information (e.g., height, speed, direction); requires multiple steps for steering (i.e., cannot start locomotion from a stop and needs to wait for gravity to pull it forwards, which can be the wrong desired direction) - no steering control; and does not account for double-support foot-placement (i.e., when both feet are on the ground supporting the body).

An essential biped character action that is needed for any virtual environment is upright balancing (i.e., the ability to remain upright during either standing or during locomotion). Accordingly, this chapter focuses specifically on creating upright balancing motions without key-frame data. Whereby, we use lightweight physics-based approximation methods and intelligent stepping logic to generate the final movements. We target, in particular, the dynamic and adaptable nature of upright biped characters and the creation of interactive and responsive animations that are controllable and physically realistic (e.g., less Spider-man like). We approach the problem using a simple and straightforward methodology by means of simplified estimations allowing us to create a solution that is computationally fast, robust, and practical, for used in time critical environments, such as games.

We demonstrate our approach using numerous simulation situations (e.g., being pushed, holding objects of varying weight, and following paths) to show the potential, dynamic nature, and robustness of our method for creating a more engaging and interactive character solution. Our results show how a character can generate physically realistic motions for balancing and stepping that can recover from force disturbances, such as hits. While our approach is limited to upright motions and focuses purely on balancing and stepping logic, we believe that our method can easily be combined with other techniques (e.g., motion-capture data, or random coherent motions [Per95, Ken12a]) to produce a hybrid solution. This hybrid solution would present a more complete character system with a large repertoire of actions that is both physically accurate and interactive while possessing highly realistic human characteristics.

As we have emphasised in previous chapters, the inverted pendulum (IP) has long been a popular solution for providing dynamic and responsive physics-based information for character systems [TLC*10, KDM11]. The IP method is a low-dimensional mechanism for approximating dynamic characteristics of a biped. We use this model in place of a complex articulated character structure to decouple and focus on a specific motion (i.e., balanced stepping). The IP model gives us a lightweight and robust way to generate reliable dynamic information for balancing. However, the IP basic model does not provide any decisive data about the feet (e.g., how they transition, or how they handle constraint conditions). Nevertheless, we use the basic IP model and extend it to include additional foot information to produce a more stable and physical accurate biped character stepping model (e.g., include foot orientation, foot torque, and position transition paths). Hence, our approach builds on developing a robust and efficient foot placement controller that can generate or correct character motions, so they appear more natural and responsive.

5.1.1 Chapter Contribution

The contributions of this chapter are numerous approximation techniques for creating a practical, computationally efficient, and robust balancing biped character system (e.g, ankle-foot feedback) to produce physically responsive character stepping motions. The novel approach focuses around simplifying the biped foot support region (e.g., using circles and capsules) to produce approximate foot placement and balancing information (e.g., position, orientation, path trajectories) in conjunction with an lightweight physics-based model to generate interactive character movements that we can use to control a fully articulated skeleton body.

5.2 Related Work

Creating stable, responsive, balancing biped characters via physics-based techniques has been investigated across numerous fields (e.g., computer graphics and robotics). In retrospect, we briefly acknowledge and review some of the most recent and influential developments over the past few years that have contributed to our work and the creation of our lightweight physics-based dynamic stepping solution.

Controller-based approaches have been used, such as the method by Shiratori et al. [SCCH09] who exploited biomechanical principles to generate responsive balancing actions, while Arikan et al. [AFO05] did similar work on generating how people would respond to being pushed around. It is interesting to note, when a character could not recover from a disturbance, the work by Tang et al. [TPZZ06] concentrated specifically on creating animations that would emulate the realistic nature of falling based upon the unexpected forces. In the same way, a method similar to the one we present in this chapter was also offered by Wu et al. [WMZ08] who demonstrated an accurate method for controlling animated character stepping motions by modifying their foot placement information so that it was physically correct. Comparatively, Singh et al. [SKRF11] offered a simplified footstep model akin to ours, i.e., a circular foot approximation, that simulated foot positions for navigating crowds of characters. Our scheme seeks an approximate solution for real-time interactive environments by incorporating a foot-ankle feedback force to make the solution more controllable and overcome simplifications in the low-dimensional model without sacrificing the computational speed or uncomplicated nature of our design.

Emphasising some of the relevant work in the field of robotics that contributed to the development of responsive biped controllers, we outline a few interesting and important papers.

To begin with, Shih et al. [SGL93] developed a straightforward model for enabling characters to respond to small disturbances, while later Stephens [Ste07] and Pratt et al. [PCD06] developed controllers that could generate motions to recover from a range of push disturbances. A major disparity between this chapter's goals and those in robotics is that we focus more on an uncomplicated solution that is less precise and computationally faster while producing aesthetically pleasing motions that are physically plausible.

Our Work: In this chapter, we use the simple inverted pendulum approach to dictate stepping actions, as seen in SIMBICON [YLvdP07] and Arikan et al. [AFO05], however, while this approach produces robust and responsive stepping information in real-time, the model constantly needs to step in a marching-like gait to remain upright and balanced, since the model did not include any intelligent foot-ankle feedback control. While one solution is to add small virtual forces to the centre-of-mass in order to steer and accelerate or decelerate the model towards the desired velocity and direction [TLC*10, CBvdP10], we use an uncomplicated foot-ankle approximation feedback to remedy the pendulum-based systems shortcomings without sacrificing the model's simplicity or computational speed. Finally, **this chapter introduces the ankle-torque feedback scheme to add control to the low-dimensional inverted pendulum model, such as standing still and steering. However, a different solution that accomplished the same goal was presented by Pratt and Tedrake [PT06] known as the Centre of Pressure (COP) strategy and worked by shifting the ground contact point around (e.g., shifting COP around within the foot support region to keep the inverted pendulum standing on the spot).** We also note, that the Zero Moment Point (ZMP) concept [VS72] uses a similar approach to the Centre of Pressure strategy to control and steer the pendulum mass (i.e., the ZMP point on the ground is where the net moment of inertia and gravity force has no component along the axes parallel to the ground).

5.3 Overview

In our approach, we use a low-dimensional base-controller for estimating key information for the highly complex articulated characters that enable us to determine intelligent foot place information to remain balanced and upright. The low-dimensional controller calculates information on where to place the character's foot to produce the desired upright motion. The controller can be iteratively updated to give corrective feedback information and ensure the resulting motion is achieved (e.g., due to minor force disturbances and numerical inaccuracies). We take the basic inverted pendulum model and incorporate additional information to gain greater control through feedback approximations from the feet. However, while this chapter primarily focuses on the feet to gain additional control, alternative research has been done to extend other areas of the inverted pendulum. For example, we showed in Chapter 4 that extending the inverted pendulum model to include an elongated 3D rigid-body produces additional postural information in collaboration with further control possibilities.

The inverted pendulum model presents an ideal method for emulating a character's leg since *the human muscle is mechanically analogous to a spring-damper system*; consequently, stiffness and damping factors can be calculated to mimic a person's limbs and how they would respond (see Figure 5.2).

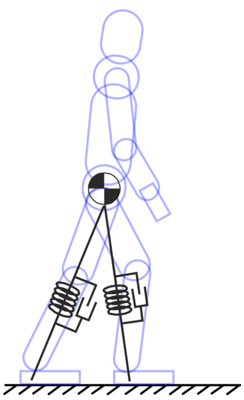


Figure 5.2: Inverted Pendulum Principle - The human-leg is analogous to a spring-damper system. We can represent the overall character's centre-of-mass and leg using an inverted pendulum system. The leg spring-damper force can be calculated using an uncomplicated penalty-based technique - we configure the parameters to produce a relatively stiff-knee joint which is desired during casual stepping motions.

Our method produces responsive character motions, which have the added advantage of being computationally simple and robust. We represent each foots support area by a circle that is projected onto the ground. When both feet are in contact with the ground the support area changes to a capsule shape. Projecting the centre-of-mass (COM) onto the ground, we can use these simplified support regions to give us essential balancing information. Hence, as the inverted pendulum changes between a single and double support stance during stepping transitions, we can gather extra intelligence to give corrective balancing and control feedback values (see Figure 5.4).

Furthermore, since we add a support region to the feet of the inverted pendulum, this allows us to *induce an ankle torque* to correct small disturbances without needing to take corrective foot-steps. This ankle torque provides a means of correcting minor disturbances due to any approximation errors (e.g., ankle torque can introduce corrective balancing and steering parameters).

The logic is managed using a finite state-machine. The state-machine examines the information from the inverted pendulum model to determine the next state of action that needs to be performed (e.g., apply ankle torque, take corrective step, or continue walking). The state-machine has three primary logic components shown in Figure 5.13.

The inverted pendulum on its own is a very minimalistic physics-based controller that has little overhead and is capable of producing practical, robust, and reliable data for balancing and locomotion; hence, it is ideal for time critical systems (e.g., games). Furthermore, the inverted pendulum is able to handle uneven terrain (e.g., stairs and slopes), as discussed in Chapter 3. Finally, by altering the placement of the foot position and the urgency that the foot reaches its target position, we can produce numerous styles of walking (e.g., relaxed or urgent).

5.4 Orientation and Feet

The inverted pendulum (IP) model at its heart provides us with crucial balancing information that ensures our biped character remains upright while performing various actions (e.g., such as standing, walking, or running). However, we briefly examine the shortcomings of an IP model without feet or body orientation and the justification and advantages for including them:

5.4.1 Do We Need Feet?

The fundamental IP model does not tell us how to move or orientate the feet during foot transitions, since it is only able to calculate the desired final foot position from the current position and velocity of the point mass. In essence, for a character to be useful, it should possess feet; since, it is impossible for a passive platform, such as a skeleton body, to stand in a single, stable position, if only two points are supporting it. However, a dynamic system can balance on two points like stilts if the supporting points are allowed to move and are controlled by a sufficiently sophisticated control system. The stiff-legged stilt character must remain in a continual state of motion to maintain balance (see Sias [SJZ90] for further details on why we need feet).

5.4.1.1 Determining the Support Polygon

The support polygon represents the support area for the feet used to keep the character balancing and upright. Without the support polygon, the character would constantly need to move to remain balanced. Exact approaches exist for calculating the feet contact area. These exact methods use complex contact polygon constructions or simplified rectangles. Our method uses a simplified approximation of circles to represent each foot's support region and a capsule when both feet are in contact with the ground. The circle-capsule method of calculating the support region is a computationally simple approach of generating valid foot approximation information (see Yin [YZX08] for detailed explanation of support regions and more exact methods).

5.4.1.2 Feet Location Comfort Factor

The dynamic model determines the necessary foot placement information to remedy any force disturbances and remain upright. However, the resulting foot placement movement can result in the character's feet being left in an uncomfortable and unnatural looking pose. To remedy this, we include an additional logic step to determine if the character has reached a stable state and needs to take a corrective step to return the feet to more comfortable positions.

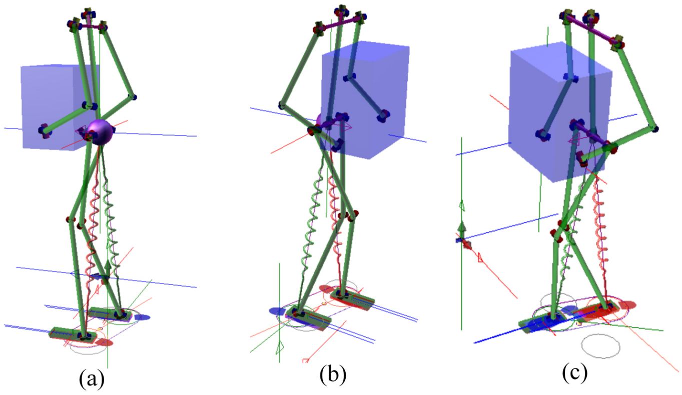


Figure 5.3: Unnatural and Uncomfortable Stance Pose - Uncomfortable poses can arise during stepping, which are physically accurate and balancing yet look awkward and unnatural. For example, (a) and (b) show the character cross-legged and (c) shows the character with one foot in front of the other.

5.4.1.3 Foot Placement Constraints

The final calculated foot position and orientation had limiting constraints imposed upon them. This ensured they never stepped on another foot or in an undesired location. For example, if we wanted to avoid the foot being placed in a hole, we would select the next closest point. The corrective step would then go ahead using the alternative position. However, if the corrective step was not able to balance the model, then the state logic would again repeat the corrective step calculation based upon the new position of the body and feet. This process would automatically repeat until it reaches a stable balancing state.

5.4.2 Body Orientation

The feet and centre-of-mass are moving around to keep the character upright and balanced. This simplified model has no concept of forward or sideways (e.g., see Figure 5.4). We rectify this problem by adding orientation information to the body. This body orientation provides reference information for other calculations, such as comfort factors and foot direction. For example, Figure 5.5, shows a top down view of the model with the pelvis possessing the reference direction. As the pelvis rotates, the new foot position for the left and right foot can be calculated (i.e., tangentially to the left and right of the pelvis) causing a stepping action to position the foot at the desired locations and with the desired direction (i.e., to match the pelvis). We set the default comfort factor to always want to have the feet orientated in the same direction as the pelvis. When the pelvis turns, it would cause the character to identify that the a step is necessary to reposition the foot. For example, when the character is standing on the spot, if rotate the character's main body, the desired foot locations (i.e., to the left and right of the pelvis) is modified, causing the feet to take corrective steps and align with the pelvis orientation.

5.5 Base-Controller

The base-controller for determining where to place the character's foot to achieve walking or halt motion is based on the spring-loaded inverted pendulum (SLIP). The SLIP model approximation represents the mass properties of the character as a single-particle point mass (as discussed in Chapter 4). This single-point mass is balanced upon mass-less spring-damper sticks that mimic the legs and muscles of a human. The SLIP model provides a computationally efficient approach for modelling the rigid-leg. The spring-damper coefficients are chosen so that the knee-joint remains sufficiently stiff while avoiding oscillatory motions - we do not store and release energy within the spring - such as in highly dynamic motions (e.g., running and jumping).

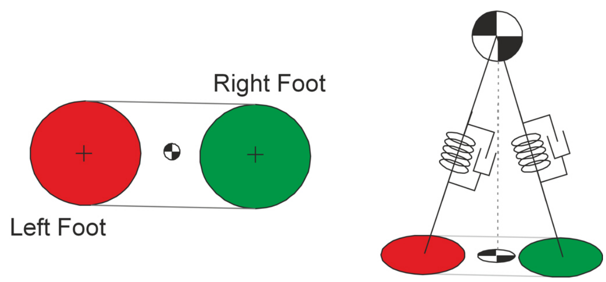


Figure 5.4: Foot Support Area - Inverted pendulum model showing the approximated circular shape foot support regions and the combined overall capsule shaped support region.

Figure 5.4 shows the base-controller model that generates the crucial balancing and locomotion information for our biped character. The key pieces of information are the pelvis positions and feet positions. The inverted pendulum can predict where to move the feet to maintain a persistent stable walking motion or to halt movement in any direction.

For statically balanced stepping motions (e.g., standing or raising a leg) that need to be controlled - the centre-of-mass needs to be shifted between each foot support area. Initially, it positions the centre-of-mass above the centre-of-pressure (COP), from there on, it lifts its front body up, while compensating with the lower-body to maintain the COM above the foot position. Due to the dynamic feedback from the model, any disturbances that might arise (e.g., pushes, trips, and uneven terrain), will be fed back into the base-controller, which will attempt to compensate for them in further steps.

5.5.1 Controller Constraints

We impose a number of additional constraints on the base-controller to achieve a reliable upright posture. It must be possible for the controller to place its centre-of-mass above its foot position. Since the inverted pendulum model has massless legs, it means we can move the leg and hence the final foot position to its target destination instantly to achieve the desired task (e.g., walking and stopping falling). The path taken or movement of the leg does not affect the default motion. While we can instantaneously move the leg to its target location, in practice, we interpolated the final foot movement along a trajectory spline path over a specific time to mimic human-like stepping more closely.

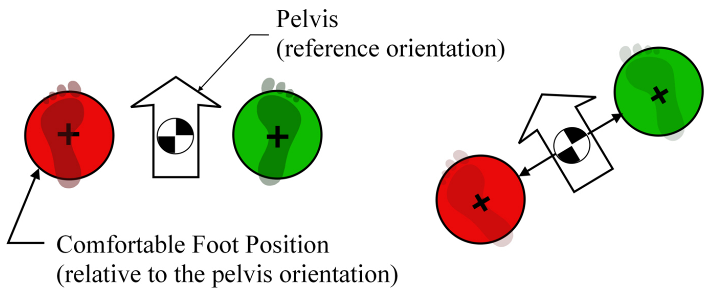
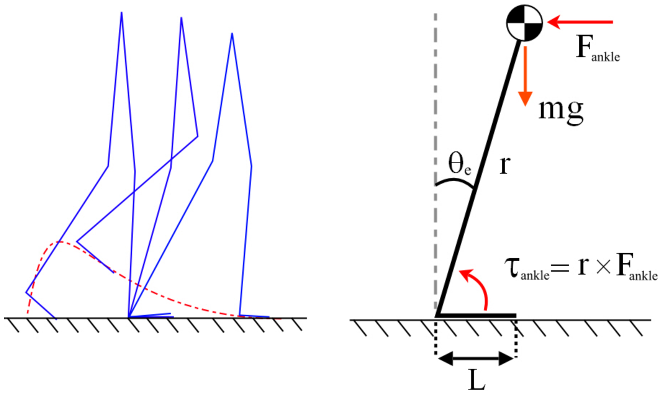


Figure 5.5: Reference Pelvis - Illustrating the comfortable positions the feet will return to after a disturbance or when idling.

5.5.2 Foot Swing

Creating life-like stepping motions makes the character more believable and life-like, while helping to maintain the connection between the player and the avatar. Artifacts, such as foot slipping and floating can break this connection and produce unnatural and bizarre movements. Hence, we generate foot-placement information for balancing and locomotion and combine them with Bezier splines to create smooth natural-looking stepping transitions.

Only after the calculated desired position and orientation for the foot can we begin to interpolate the foot to its new target. It is also necessary to generate a path for the foot to travel to reach its target. While initially, an elliptical arc was used based on an uncomplicated Bezier curve, it produced an unnatural looking stepping action. However, when we analyzed the walking motion of a real-world human, we found that the foot trajectory path would shoot up and exponentially decays towards the target location, which is due to the toe and heal (see Figure 5.6). Hence, the trajectory path was modified to mimic human stepping motions and present a more human-like and realistic effect. The trajectory was calculated using a Bezier curve approximation with the peak curve shifted towards the beginning to matching the path shown in Figure 5.6.



(a) (b)

Figure 5.6: Ankle Torque & Foot Trajectory - (a) Common foot swing trajectory for a human casually walking; and (b) ankle torque can provide minor steering and control abilities for the upper body (e.g., standing without requiring to constantly step).

The foot trajectory was calculated using the foot start and end position as the starting guide-lines. Since the inverted pendulum model has provides no trajectory information due to the legs being massless and only providing the start and end information, we must specify certain stepping parameters, i.e., height, offset, and time to reach the target. For example, limiting the angle deviation to 0 and setting the offset to zero would produce an arc-like motion with a constant horizontal foot (Figure 5.8(a)) and while physically-plausible it appears unnatural and out of the normal. Furthermore, our specified step parameters must take into account the articulated skeleton onto which the movement will be mapped, e.g., if the necessary muscle response time and strength can follow the calculated trajectory.

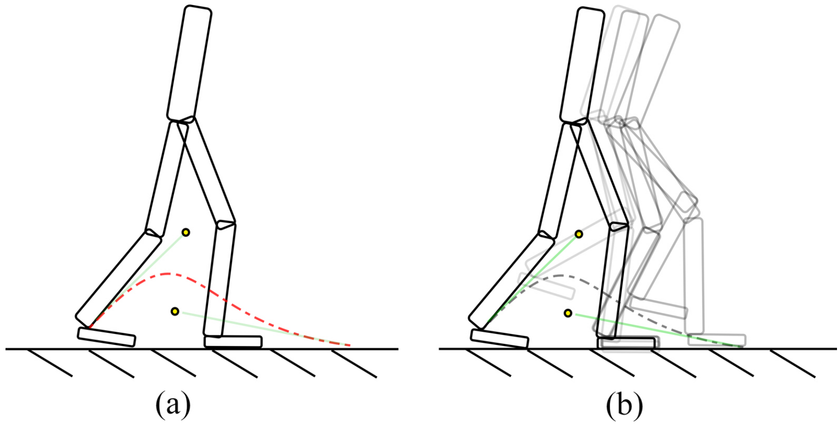


Figure 5.7: Step Trajectory - Foot stepping trajectories are calculated using Bezier curves so they appear more natural, (a) trajectory paths for a simple forward step, and (b) blended motions following the trajectory.

The ‘foot-angle’ and ’foot-trajectory’ are calculated separately. The implementation starts with the start and end foot position. We specify the hight and the interpolation time. This is defined using a Bezier curve. However, we also need to orientate the foot to correlate with the foot transition. The position of the foot (i.e., distance from the ground and the ratio along the arc trajectory) are used to calculate the foot orientation. We limit the foot ankle and orientate. The foot initially has the tow face towards the ground but is turned up as the foot approaches the trajectory end (e.g., see Figure 5.7 and Figure 5.8). The trajectory provides aesthetic satisfactory and can be generated from different trigonometric methods [BC89] - we chose Bezier splines as they are uncomplicated and provided a good starting point for exploring different walking styles.

We should point out, the work by Bruderlin and Calvert [BC89] also explored synthesizing human stepping trajectory information based on geometric data (e.g., foot height and distance)

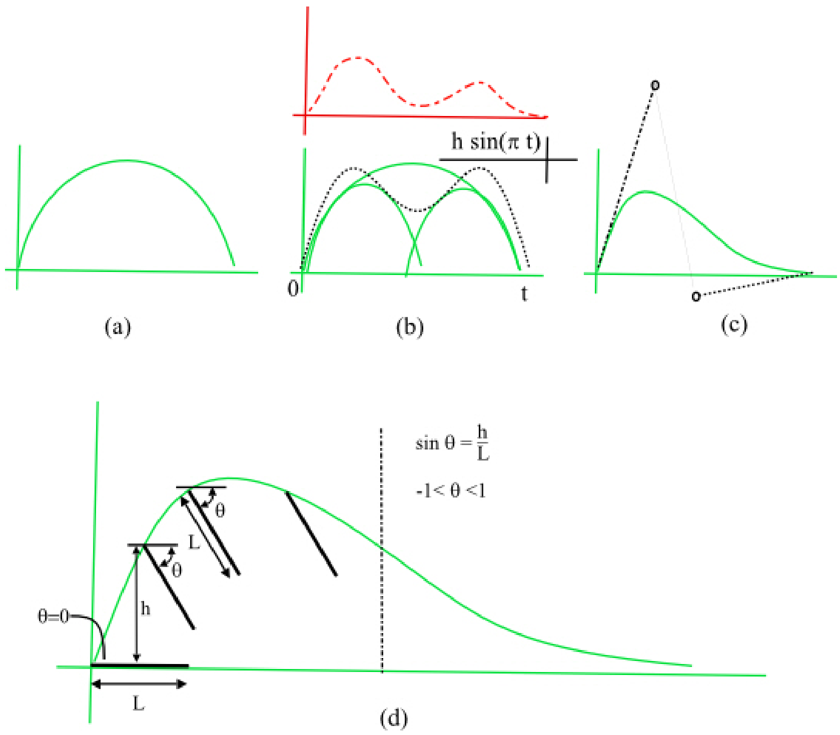


Figure 5.8: Foot and Ankle Trajectory Correlation - As the foot transitions along its spline we orientate the foot based upon the distance along the route and the height from the ground. (a) Foot following symmetrical sinusoidal path, (b) while we can mimic more esoteric paths with dips that closely match a human by combining multiple sinusoidal functions, (c) we eventually settled on a Bezier path solution (of course, higher order curves could be used to give a more accurate representation), and (d) which we can use Bezier path to calculate the foot rotation angle, so the stepping motion is more aesthetically pleasing (also, the angle needs to be inverted, as it passes half way due to the way we lift our toe and land with our heel).

- demonstrating a reasonably accurate model. In addition, Meredith and Maddock [MM04] produced foot transition trajectories based on a trigonometric approximation (e.g., as shown in Figure 5.9).

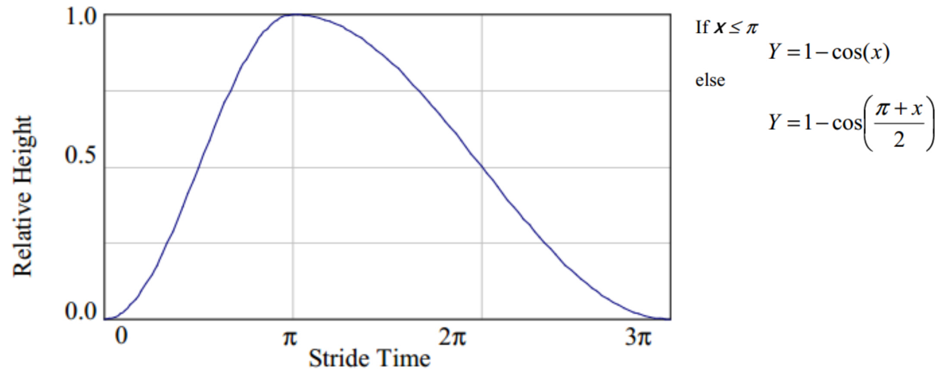


Figure 5.9: Procedural Path Single Foot Stride - Plot of a generated foot position for a single stride from Meredith and Maddock.[MM04].

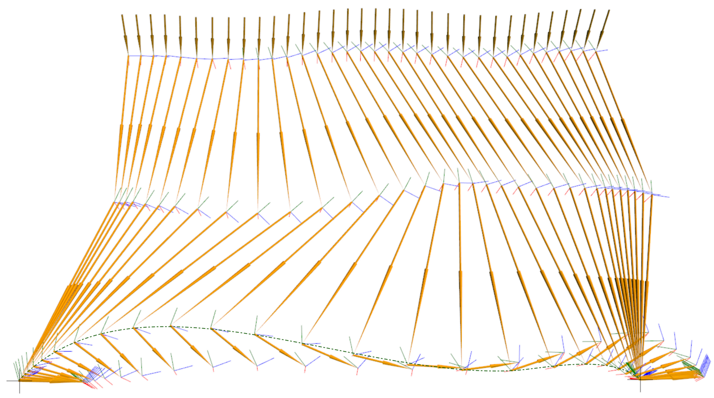


Figure 5.10: Motion Capture Walk Cycle - Interestingly, if we look closely at a pre-recorded human walk cycle (i.e., motion capture), we can visually see a number of crucial transitions (i.e., foot orientation and trajectory) - while we capture the significant components (i.e., the offset peak) - there are a number of minor discrepancies.

5.5.3 Ankle Torque Feedback Control

Additional control and stability can be gained by adding a simple ankle torque feedback force to the body. This ankle torque can remedy the problem of constantly stepping or needing to take multiple steps to move in a specific direction. We can derive the basic numerical formula for the ankle torque by analyzing the model in stance mode (as shown in Figure 5.6(b)). If we look at the geometric layout for the inverted pendulum, we can deduce Equation 5.1 and simplify it for small-angle deviations (i.e., $sin(\theta) \approx \theta$).

$$mgr\ sin(\theta_e) \approx mgr\theta_e \tag{5.1}$$

Since the human muscle (i.e., ankle) is analogous to a spring-damper mechanism, we use a simple PD feedback controller (i.e., an angular penalty error) to calculate the ankle torque magnitude. For example, if we want the centre-of-mass remain directly above the support foot (i.e., $\theta_d = 0$) and the current centre-of-mass is off to the side (i.e., θ_c) we can calculate the angular error (θ_e) using simple trigonometric functions (e.g., dot and cross product). Multiplying the angular error by the spring coefficient (k_p) gives the ankle torque $\tau_{ankle} = k_p\theta_e$. Making the coefficient k_p larger than mgr causes the body to remain upright (i.e., the centre-of-mass will oscillate around the support point). The damping coefficient k_d should be small enough to ensure the body's oscillations eventual converge and die-away. However, with a foot size of L, as shown in Figure 5.6, the maximum ankle torque (τ_{ankle}) should be less than mgL to prevent the ankle torque lifting the body up. For our experiments, we set the foot size to 25cm with a leg-length of 1.5m.

$$mgr > ||\tau_{ankle}|| < mgL \tag{5.2}$$

For standing the equilibrium position of the projected COM is ideally situated above the centre of the foot support area. However, in reality, it should be mentioned that the ankle is located towards the end of the foot, and not the centre of the support area as we approximate.

$$\begin{aligned}\tau_{ankle} &= r \times F_{ankle} \\ &= k_p\theta_e - k_d\dot{\theta}\end{aligned} \tag{5.3}$$

5.6 Static and Dynamic Balancing

Static balance and centre-of-mass have been understood for a very long time. Essentially, if the centre-of-mass is above the support region it will remain upright and balanced (e.g., a table lamp). However, the inverted pendulum, with pin-point feet, does not, in fact have a support region, so cannot remain in a statically balanced pose.

Dynamic balancing is not required to return to a statically balanced state at any point during motion. Where "dynamic balancing" is sometimes referred to as "actively balancing", since during "dynamic" movement the control system must constantly take actions to keep the body from falling over. In effect, dynamic balancing is achieved by shifting the body into a state of a continuous, controlled fall (i.e., similar to the inverted pendulum - constantly stepping to recover).

Quasi-dynamic control solutions attempt to solve dynamic problems using static system approximations. They provide extra flexibility over basic static balancing solution with the ability to break the rule of always being continuously statically balanced. However, they can require long static periods of time to recover from balance disturbances. The solutions are not "truly" physically correct.

The advantages dynamically balancing over static and quasi-dynamic solutions stem primarily from increased versatility and speed. For example, our lightweight inverted pendulum model, is not bound by the constraints of maintaining static balance and can adapt itself to environmental conditions (e.g., repeatedly stepping and regaining balance).

5.7 Passive and Active Walking

Passive walking which is also known as cyclic walking uses gravity to generate perpetual walking motions (e.g., a downhill slope or constant perpetual motion without intervention) typically by means of an under-actuated legged model (i.e., a passive interaction of gravity and inertia). An early example of this from Robotics is a walking toy with curved feet by McGeer [McG90]. The mechanical toy motions, if tuned correctly, produced a simple fluid pendulum motion that mimicked human-like walking. Our initial inverted pendulum model, introduced in Chapter 3 is analogous to a passive stepping system.

Active walking involves actuated joints (i.e., the joints are torque driven). The walking movements take into account the dynamics properties (i.e., velocity, inertia, gravity) and stability for each step transition. Different models generate the walking motions from different input parameters (e.g., the centre-of-mass, the swing foot placement, and ground reaction force) to produce smooth, stable, natural-looking walk movements. Our model does not include active joint muscles during foot support transitions.

5.7.1 Static and Dynamic Walking

Static walking is known as *slow walking*. During foot support transitions the centre-of-mass (COM) is always within the foot support area. That is, while the next foot is being placed at a new location the COM remains with the support foot region. Only once the new foot has been placed does the COM move towards the newly placed foot (staying within the foot support region of both feet). The dynamics of the body does not help the stability since the COM remains within the foot support area. If the walking motion is done at slow speeds it is referred to as *static walking*, while at the speed of a typical human walk or faster it is referred to as *dynamic walking* as shown in Figure 5.11. *Dynamic walking* is known as *fast walking*. During step transitions the COM is **not** inside the foot support as shown in Figure 5.11.

In the previous Chapter (i.e., Chapter 3), we demonstrated a dynamic balancing model using the inverted pendulum and its ability to remain upright and balanced during a variety of situations (e.g., push disturbances) - this Chapter enables us to also create statically balanced motions due to the ankle-foot feedback. While the ankle-foot feedback provides the ability to move the COP inside the foot support area it also allows us to make up for errors due to the use of approximations. For dynamic stepping cases - while we base the initial calculations on bringing the model to a stop (i.e., the capture point), in practice, dynamic walking is just a matter of not bringing the COM to a stop but allowing it to oscillate, as shown in Figure 5.11 and 5.12.

5.8 Foot Logic

We iteratively update the decision logic for our model based on the current state of the system (e.g., position of the COM, foot location, and comfort reasoning). A state-machine logic

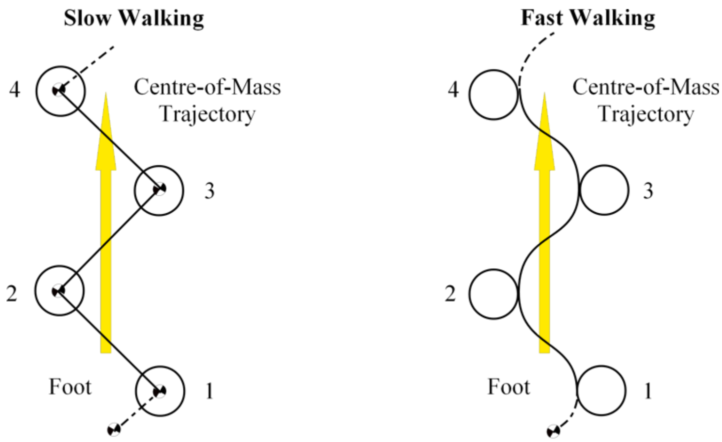


Figure 5.11: Static and Dynamic - Static walking the centre-of-mass transitions is constantly supported, while dynamic walking the centre-of-mass remains in a state of dynamic motion.

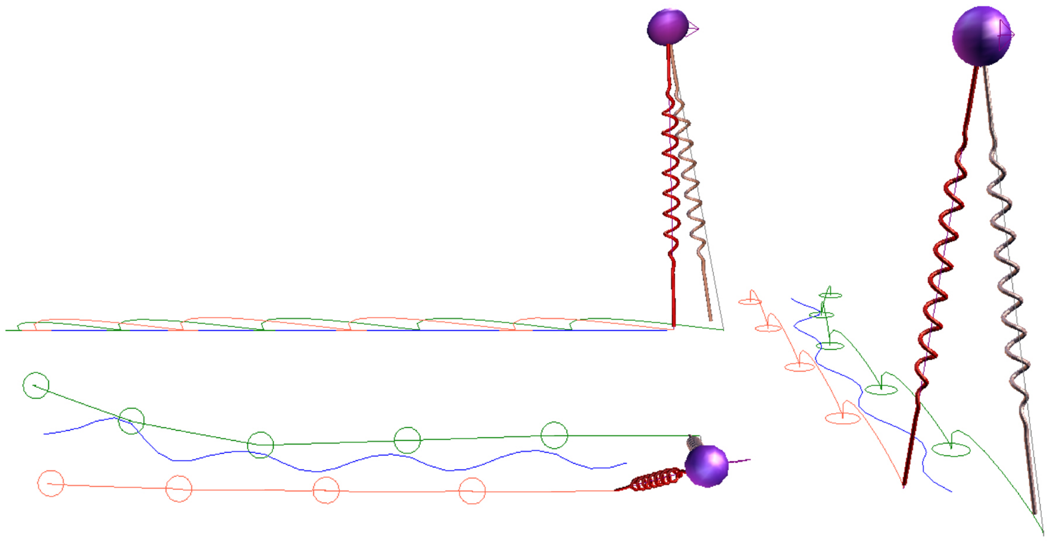


Figure 5.12: Walking - Draw the step transitions and the projected centre-of-mass on the ground (top, side, and perspective view)

analysis the current state of the biped and decides upon the necessary corrective actions (see Figure 5.13). The foot logic is recursive in nature because it might take a number of corrective steps to remedy a push disturbance and regain stable balancing control. Furthermore, when taking foot placement steps to gain a more desirable and comfortable posture it can again in certain situations take a number of steps to accomplish this. This constantly correcting recursive nature is highly desirable. For example, if we impose constraints on where the foot can be placed due to terrain holes or objects being in its way, we can calculate the next closes target and have our model try again the next time around.

The centre-of-mass is projected onto the simplified support region to provide information on how the character should proceed at each step as shown in Figure 5.14. The three main balancing choices are: firstly, through ankle torque; secondly, through corrective stepping; and, finally, identification of unrecoverable loss of balance.

5.9 Inverse Kinematics

The inverse kinematics (IK) solver takes the key information from the low-dimensional model to create the final articulated character poses (e.g., including the knee joints, pelvis, and arms) as shown in Figure 5.19. The IK solver generates the joint angles for the character model. Furthermore, it imposes constraints to ensure we always produce legal human poses. The key information from the controller guarantees the character remains balanced and upright. We pass the feet and body information (i.e., positions and orientations) along to the IK solver to generate the final pose. We use a real-time Jacobian inverse matrix method to solve IK problem [Ken12c, MM04].

There is a large amount of ambiguity between the initial low-dimensional model and the highly articulated character mode. This ambiguity largely comes from a lack of information for describing the arms, neck, and posture. However, this additional redundancy means that behaviors, such as waving and looking around, can be added from other sources (e.g., stored key-framed animations) and combined as secondary priority movements so the final motion incorporates more human characteristics.

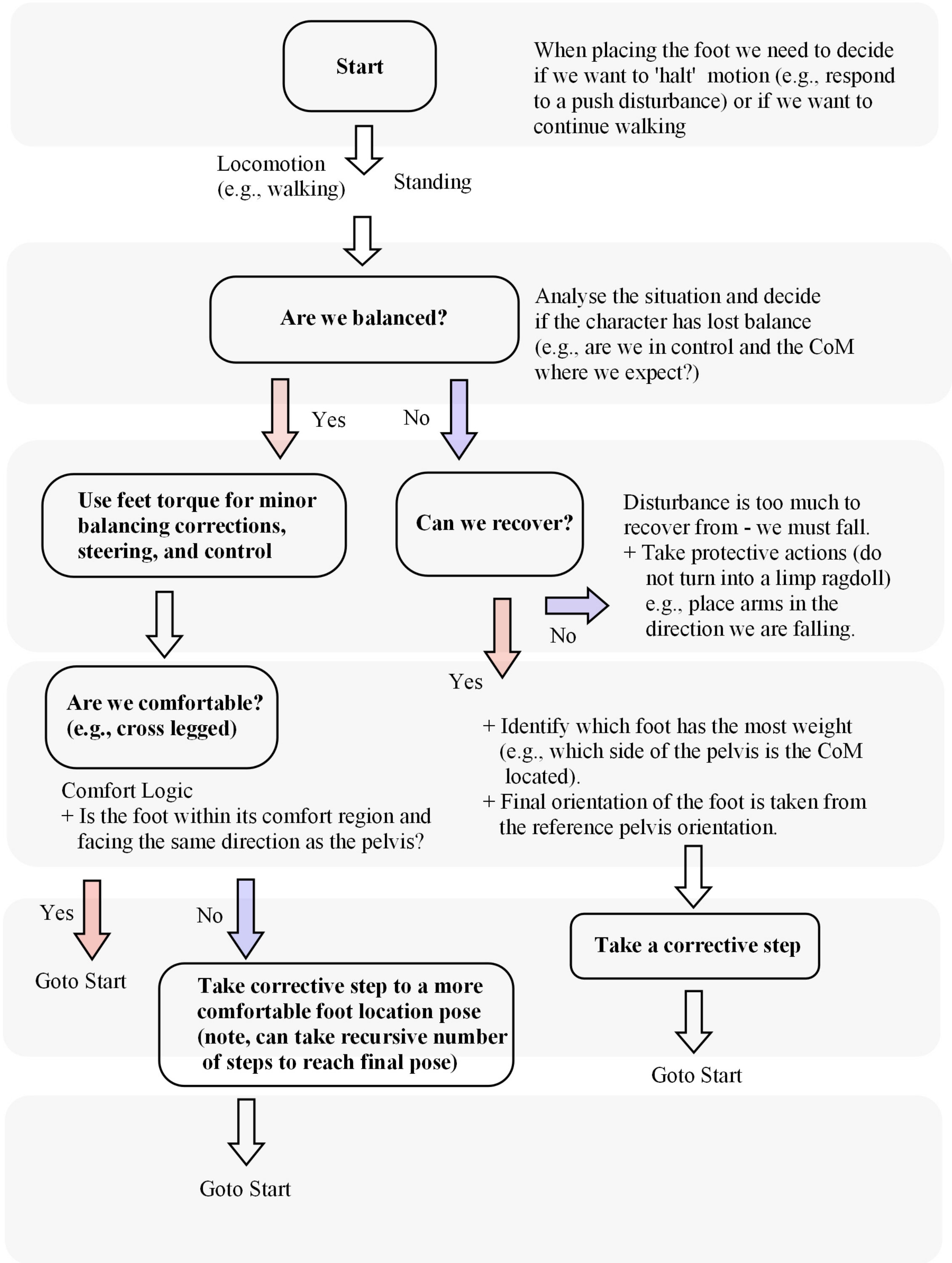


Figure 5.13: Response Decision Flow Diagram - State flow diagram for balanced logic.

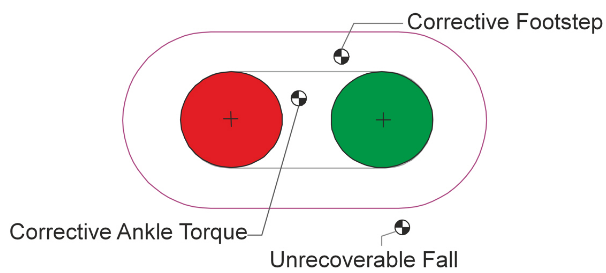


Figure 5.14: Projected Ground Information - Projected COM provides essential balancing information (e.g., default mass of 75kg, leg-length 1.0m, foot radius 10cm, and outer capsule radius 25cm.

5.9.1 Biped Model

The physical biped mechanism is modeled as a series of rigid segments (or links) connected by joints. These interconnected elements also called kinematic chains. As shown in Figure 5.15(b), we represent the character's body using 16 body segments connected using 16 links. The character gives us 36 degrees-of-freedom (DOF). Joints such as the shoulder have three-DOF corresponding to abduction/adduction, flexion/extension and internal/external rotation (e.g., rotation around the x, y and z-axis). Furthermore, it is convenient to note that a joint with n-DOF is equivalent to n-joints of 1-DOF connected by n-1 links of zero-length. Thus, the shoulder joint can be described as a sequence of 3-joints. Each separate joint has 1-DOF and 2 of the joints are connected with zero-length links.

Figure 5.19 shows the skeleton pose and Figure 5.15(b) shows the model combined with the inverse pendulum model. The foot was set as the root of the IK solver. The IK system had five end-effectors (i.e., head, pelvis, right-hand, left-hand, and left-foot). The base-controller would feed information to the feet and pelvis.

5.9.2 Articulated Rigid-Body Control

The inverse kinematic solver provides joint angles that we use to calculate joint torques to control the articulated rigid-body skeleton structure. The approach is analogous to a puppet on a string since the rigid-body structure emulates the inverse kinematic solution through angular springs (i.e., proportional derivative servos). However, since the final motions were generated using an articulated rigid-body structure, the movements were smoother while still

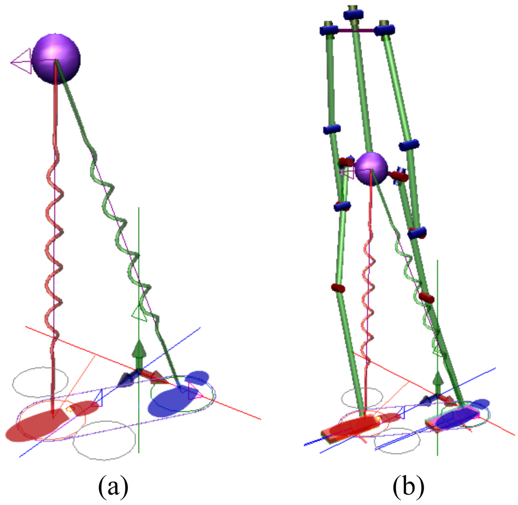


Figure 5.15: Skeleton Mapping - Simple spring-loaded inverted pendulum and full skeleton model used for testing.

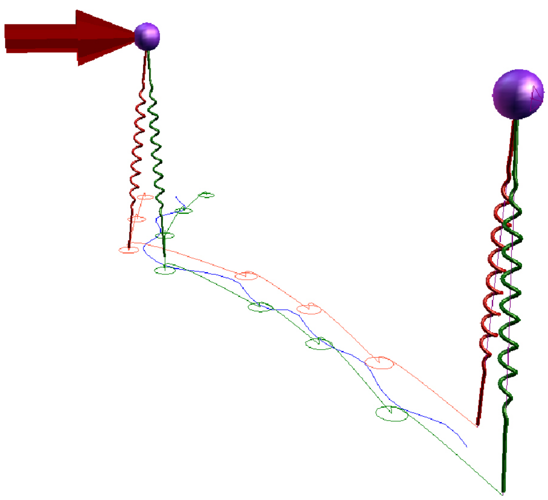


Figure 5.16: Push Disturbance - Recovering from push disturbance to the side while walking.

possessing their responsive and interactive properties. The joint torques for the articulated character were generated using a proportional derivative (PD) controller, shown in Equation 5.4.

$$\tau = k_p(\theta_d - \theta) - k_d\dot{\theta} \tag{5.4}$$

where τ is the torque, θ_d and θ is the desired and current joint angle, $\dot{\theta}$ is the angular velocity, k_p and k_d are the gain and damping coefficients. The gain and damping coefficients are crucial for the character's motions to appear responsive and fluid; however, calculating reliable, robust coefficients that result in the pose reaching its desired target within a specific time reliably and safely is difficult due to the highly dynamic model. Hence, it was necessary to hand-tune the coefficients to achieve visually pleasing motions (i.e., perceived as balanced and responsive) (see Appendix A for PD coefficients).

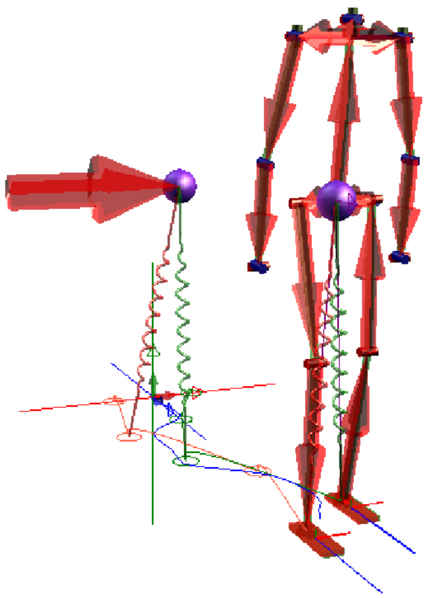


Figure 5.17: Skeleton Mapping - The point-mass model generates balanced stepping motion when mapped onto an articulated skeleton. However, the point-mass does not capture posture information (i.e., the upper body posture reaction based on the corrective stepping motion). We address and solve this limitation in Chapter 6 by incorporating an elongated body into the lightweight model.

5.10 Results

The controller generates essential information for the biped character to remain upright. This information is passed to the IK solver end-effector to produce the full character biped motion. Furthermore, an additional step is used to make the final motions more fluid by applying the generated full-body movement to an articulated rigid-body skeleton.

Producing interactive and dynamic motions in real-time by purely inverse kinematic and data-driven methods is difficult. While data-driven approaches can produce life-like results that can engage the environment using inverse kinematics, they usually fail to emulate responsive balancing motions realistically from the virtual world. Furthermore, for a physics-based solution, we can accommodate physical changes (e.g., size, strength, and weight) and reflect them in the final character animations.

The preliminary work shows promising results and great potential for simulating crowds of characters. The model is minimalistic and computationally efficient. The largest computational overhead was generating the inverse kinematic skeleton pose from the end-effectors information generated by our inverse pendulum model and foot logic algorithm (see Figure 5.15, Figure 5.1, and Figure 5.20 for simulation screen captures).

Figure 5.20(a) shows the uncomplicated inverted pendulum model being pushed to illustrate its responsive balancing nature and provides us with computationally fast model for generating crucial stepping information for remaining upright even during disturbances, such as being pushed (however, with just needle-like points for the feet the model constantly needs to keep stepping). Then Figure 5.20(b) extends the uncomplicated inverted pendulum to include a minimalistic foot support region using a circle-capsule approximation. This foot support region enables us to inject upper steering forces for additional balancing and steering control. Following on, Figure 5.20(c) goes on to show comfort logic to prevent the character being left in an uncomfortable and unnatural looking pose (e.g., cross-legged). Finally, Figure 5.20(d) integrates the low-dimensional inverted pendulum model with a fully articulated skeleton.

Due to the dynamic nature of our balancing biped model, it automatically compensates for changes. For example, in Figure 5.19, we have our biped character hold a box that adds to the overall weight of our character. Furthermore, as we increase the weight of the box, our character remains balanced but shows the strain of the extra weight by bending his legs.

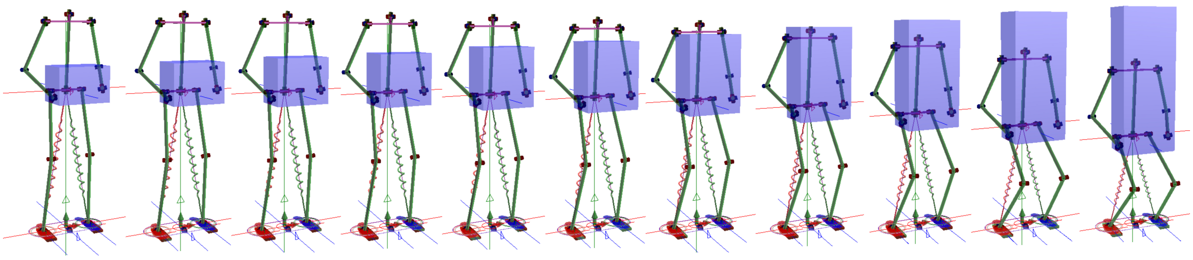
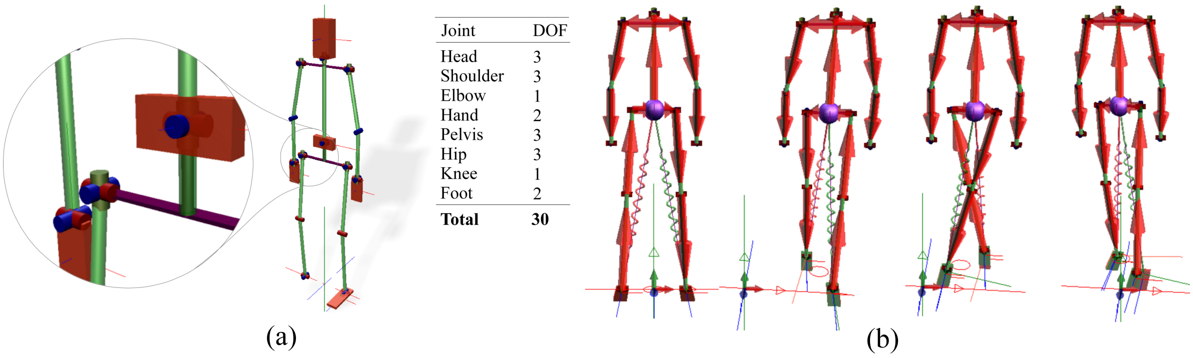


Figure 5.18: Holding a Box - Biped character holding a box that is gradually increasing in weight (left to right) while maintaining balance.

Performance We performed our simulation tests on a desktop computer (3.4 GHz, Intel i7-2600 CPU, 16 GB ram, Windows 7 64-bit). Due to the minimalistic nature of our mode and its computational efficiency, we were effortlessly able to run at real-time speeds. The physical simulation frame-rate was set at 100 fps. The inverse kinematic solver consumed the majority of the frame-rate time, while the stepping model and logic consumed very little due to their simplicity. (Note, we illustrate the low computational overhead of our system in later chapters by instancing large numbers of our model, i.e., in Chapter 7 - Figure 7.1).

5.11 Limitations

Our model focused on a lightweight physics-based model for generating balanced biped character stepping motions. The key emphasised in this chapter was foot placement and control. Hence, we did not address how the model could be used in a practical situation. In retrospect, our approach only addressed upright stepping actions, whereas for our approach to be a viable option for a real-world virtual environment, such as games, we would need to include a much larger repertoire of actions, such as, get-up, climb, punch, and dance. While motions such as dancing can require artistic intervention, other fundamental actions, such as climbing and getup, could be approached within our framework using the same technique set out in this chapter. For example, the articulated skeleton could be modelled as a point-mass and contacts (e.g., legs and arms could be modelled as spring-dampers) simplifying the problem to create solutions for specific tasks (e.g., when getting up we shift our balance to our feet and extend our body).



Joint	DOF
Head	3
Shoulder	3
Elbow	1
Hand	2
Pelvis	3
Hip	3
Knee	1
Foot	2
Total	**30**

Figure 5.19: Inverse Kinematic (IK) Re-Rooting - (a) The joint configuration with the support foot set as the root of the IK (i.e., right foot in the figure); (b) the IK root swapping between the left and right foot during stepping.

The lightweight model makes a number of approximations and discards certain information, such as arm movement (e.g., arms swinging or being used to modify the body's inertia when trying to avoid falling). Likewise, we place the foot at the centre of the foot support area (i.e., a human has the ankle at the back of the foot).

5.12 Discussion

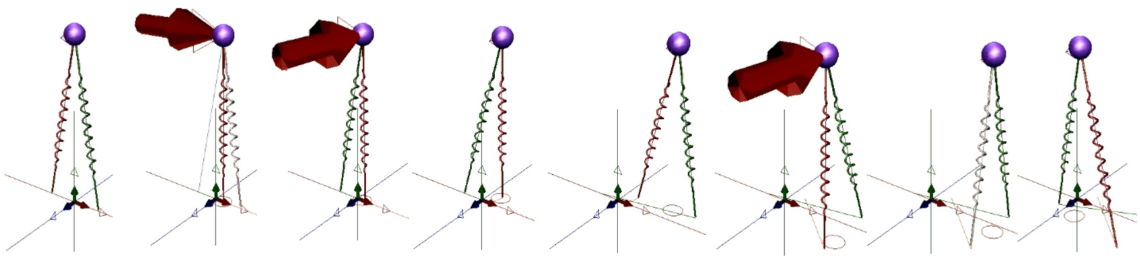
The most significant advantage of this chapter's approach over prior work is: Foot support details allowed us to solve a number of oversimplifications within the pendulum stepping mechanism, such as seen in SIMBICON [YLvdP07] who computed lower-body stepping information in real-time but would need to constantly step in a marching-like gait to remain upright and balanced, since the model did not include any intelligent foot-ankle feedback control. In retrospect, Tsai et al. [TLC*10] and Coros et al. [CBvdP10] added small ad-hoc virtual forces to the centre-of-mass in order to steer and accelerate or decelerate the model towards the desired velocity and direction.

Most importantly, pendulum-based models, such as the SIMBICON controller [YLvdP07] rely significantly on hip control for forward momentum, while human walking, in fact, relies heavily on ankle-foot control [Nov98, SJZ90]. While Wang et al. [WFH09] created an offline optimization method that incorporated objective functions that included, toe-off, passive knee-swing and leg extensions to produce more human-like walking styles for different bodies, our scheme follows a more approximate solution for real-time interactive environments.

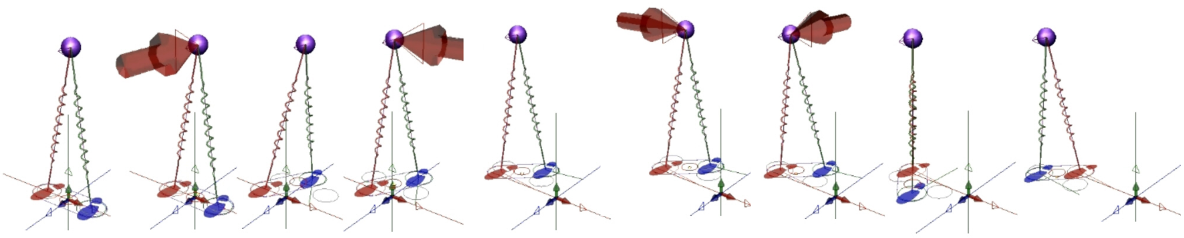
5.13 Conclusion and Future Work

In this chapter, we present a real-time biped character model for generating autonomous responsive balancing motions. We exploited numerous approximation techniques to create a straightforward, robust, and computationally efficient model that could be used in either, real-time environments, such as games, or in offline tools (e.g., for editing and correcting existing animations, so they possess dynamic interactive qualities). Additionally, by controlling an articulated rigid-body skeleton by means of the generated inverse kinematic joint angles allowed us to produce more fluid and life-like full-body movements.

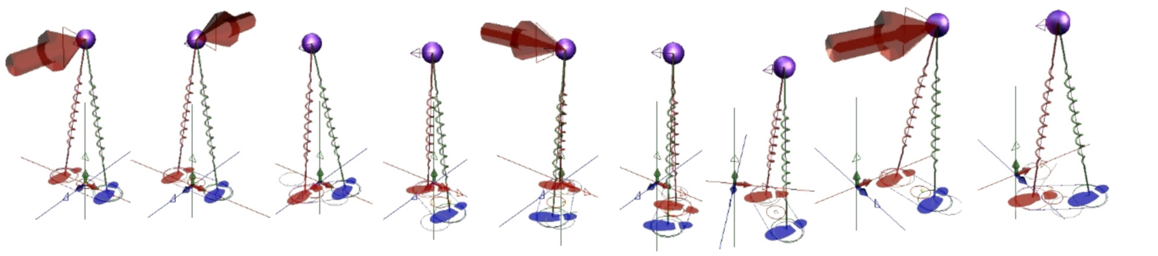
The dynamic nature of our model means that it has the ability to recover from a variety of different disturbances, such as changing uneven terrain (e.g., bridges, stairs, and obstacle avoidance), and foot placement constraints (e.g., does not have to be the desired foot placement target). Our approach solves numerous problems that allow us to create dynamic biped animations by means of an intelligent foot placement system. Finally, since the feet provide crucial information for a biped character to remain upright and balanced, it shows potential for further study in extending the model for greater realism (e.g., adding in heal-toe shifting during landing for better steering control).



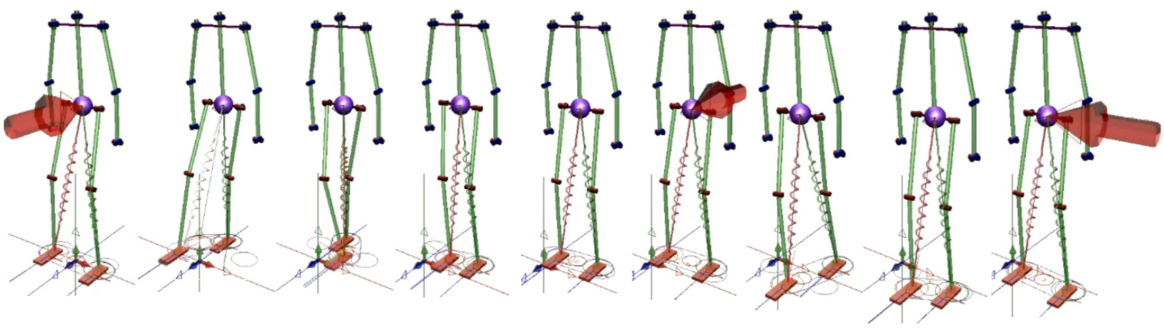
(a) Basic Inverted Pendulum Stepping



(b) Inverted Pendulum with Foot Support Area Approximation (no-comfort)



(c) Inverted Pendulum with Foot Support Area Approximation (comfort stepping)



(d) Inverse Kinematics Solver and Inverted Pendulum Stepping Model Working Together

Figure 5.20: Simulation Screen Captures - The uncomplicated inverted pendulum model (a), through to the extended version with a support area (b)-(c), and finally integrating it with a fully articulated character (d) to generate full-body postural information.

Chapter 6

Stepping Framework: Bringing it all Together

Figure 6.1: Combining Multiple System Components - We bring together previous chapters to create a unified system for solving multiple-problems (i.e., a synergistic stepping framework that combines different techniques to attain an overall solution that cannot be attained with the individual parts).

6.1 Introduction

This chapter is the culmination of previous chapters by synergistically connecting the different approaches to create a unified framework for generating controlled biped stepping movements without key-framed data (e.g., motion capture libraries). Each of the previous chapters (i.e., Chapter 3, Chapter 4, and Chapter 5)) addressed one specific problem (e.g., foot-ankle control and upper-body posture). This chapter, hence, presents a unified animation framework for creating coordinated stepping motions that can navigate complex terrain and handle unforeseen disturbances in real-time.

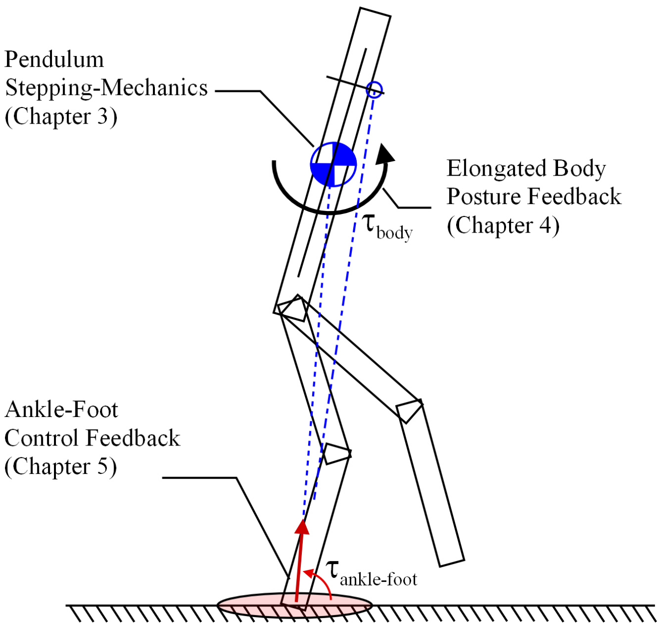


Figure 6.2: Unified Overview - Illustrating the combination of the different stepping components from Chapters 3, 4, and 5.

To recap, the previous chapters focused on:

- Chapter 3 - low-dimensional balanced stepping mechanism
- Chapter 4 - elongated-body postural feedback
- Chapter 5 - foot-ankle torque control

The unified framework solves problems that cannot be attained with the individual components. For example, while Chapter 4 provides a method for generating postural movement, it suffers from constant stepping due to pin-point feet. Nevertheless, we can remedy this issue by combining it with Chapter 5's foot-ankle control technique. Hence, this chapter demonstrates the effectiveness of creating controlled interactive stepping motions by combining a number of computationally efficient and uncomplicated techniques that are responsive and dynamic. We perform a variety of simulation experiments (e.g., pushes, complex terrain, with slopes, potholes, stairs, and bridges) to support and establish the practical viability of our approach.

6.1.1 Chapter Contribution

The contribution of this chapter is a stepping framework that combines multiple controller mechanisms in a coordinated synergistic fashion to create a computationally fast, responsive, and dynamic solution for producing balanced biped character movements. The simplified model components are shown in Figure 6.2. Hence, we summarize the key contributions of this chapter as:

- We evaluate and compare the different character-based pendulum stepping models and their associated control mechanisms for complex terrain (e.g., slopes and pot-holes)
- While the IP model provides a fast robust stepping solution, we address the highly crucial factor of *control* so the solution can be of practical importance and useful in future state-of-the-art implementations (e.g., carefully placing the foot at specific locations while walking at different speeds and remaining balanced and in control - even during changing terrain and random external environmental disturbances)
- We produce a customizable pendulum stepping system that provides better control and stability, using different feature enhancements, such as a variable leg-length and ankle-torque control

Balanced Stepping Motion We should point out, that the goal is not to generate a biped with perfect balance (e.g., a statically balanced statue), but to intelligently recover from it when it is lost in a realistic way, time and time again. Human stepping movement is smooth, realistic, and life-like; since, in reality, a human is always moving and is never statically still (e.g., they possess small swaying movements). In retrospect, a human's movement is typically graceful and comes from "dynamic" rather than "static" stability. *Merely steering a character* *by pushing it with forces in the desired direction will produce unrealistic motions.* For example, if we make a virtual character timidly extend his free leg in the direction of navigation before committing any weight to it, while constantly maintaining balance, will produce movements that appear robot-like and unnatural. This motion does not feel fluid and never takes flight; making the character appear scared of losing balance. *In reality, a human character relishes its dynamic ability without any effort or worry.* A life-like character allows their full-body weight to wonder away from their point of static balance in any direction, and is able to recover and adapt to the situation. As the character falls further off-centre, he must push harder into the floor to keep the motion horizontal and stretch the anchored leg further and more quickly to compensate for his hypotenuse. Achieving this smoothly and in a life-like way, demands both active muscle power and precise control. For example, as a character steps during walking he will fall away from his horizontal centre-of-mass towards his new one and barely show any imbalance while delivering a deliberate fluid stepping movement.

Challenges So why is it so challenging to reproduce life-like human movements in real-time 'and' without key-frame data? Why has it eluded us for so long? To begin with, realism is especially difficult, as a particular character model gives rise to a large set of possible motions with different styles. Even if a robust and stabilizing control law can be found, it is challenging to construct those that reproduce the intricate and agile movements we observe in nature. Then there is model complexity, since a character can have an extremely high number of degrees-of-freedom, it makes the search for the appropriate control parameters hard. Although continuous numerical optimizations techniques can cope with large search spaces, the stringent demands of interactive applications make it clear that optimization cannot solely be performed at the time control is needed. Also, the discontinuous non-linear character work-space (e.g., joint limits and contacts) restrict movement within a certain region of three-dimensional space; these constraints are difficult to maintain in real-time simulation systems, such as games. Furthermore, frequent ground contacts create a highly discontinuous search space rendering most continuous controller synthesis methods ineffective at planning over longer time horizons.

Finally, dynamically simulated characters are difficult to control because they have no direct control over their global position and orientation (i.e., underactuation). Even staying upright is a challenge for large disturbances. In order to succeed, a control law must plan ahead to determine actions that can stabilize the body.

At the heart of our stepping framework is the pendulum mechanism. Hence, this chapter begins by comparing and explaining the main character-based pendulum stepping models and their associated control approaches. We then present a novel real-time stepping framework for generating full-body biped motions on-the-fly without key-framed data that can be carefully controlled while remaining responsive and robust (e.g., the ability to move the foot to a new support region at a controlled speed and trajectory). We demonstrate our frameworks ability through various simulation situations, such as, push disturbances, walking on uneven terrain, walking on stepping stones, and walking up or down stairs. We extend the low-level stepping framework to create coordinated full-body motions. Our system produces directable steps that guide a character with specific goals (e.g., follow a particular path, or place feet at specific locations).

6.2 Overview

The inverted pendulum (IP) in the context of character-based systems with its various modifications and enhancements is a popular technique that has been exploited across different fields of research since it provides a computationally fast and simple balancing mechanism. We illustrate and explain the different character-based pendulum stepping techniques, what they provide, and their advantages and disadvantages in conjunction with their associated control strategy (e.g., how to steer or remain standing still). Figure 6.3 shows a comparison view of the most common pendulum-based techniques and control mechanisms.

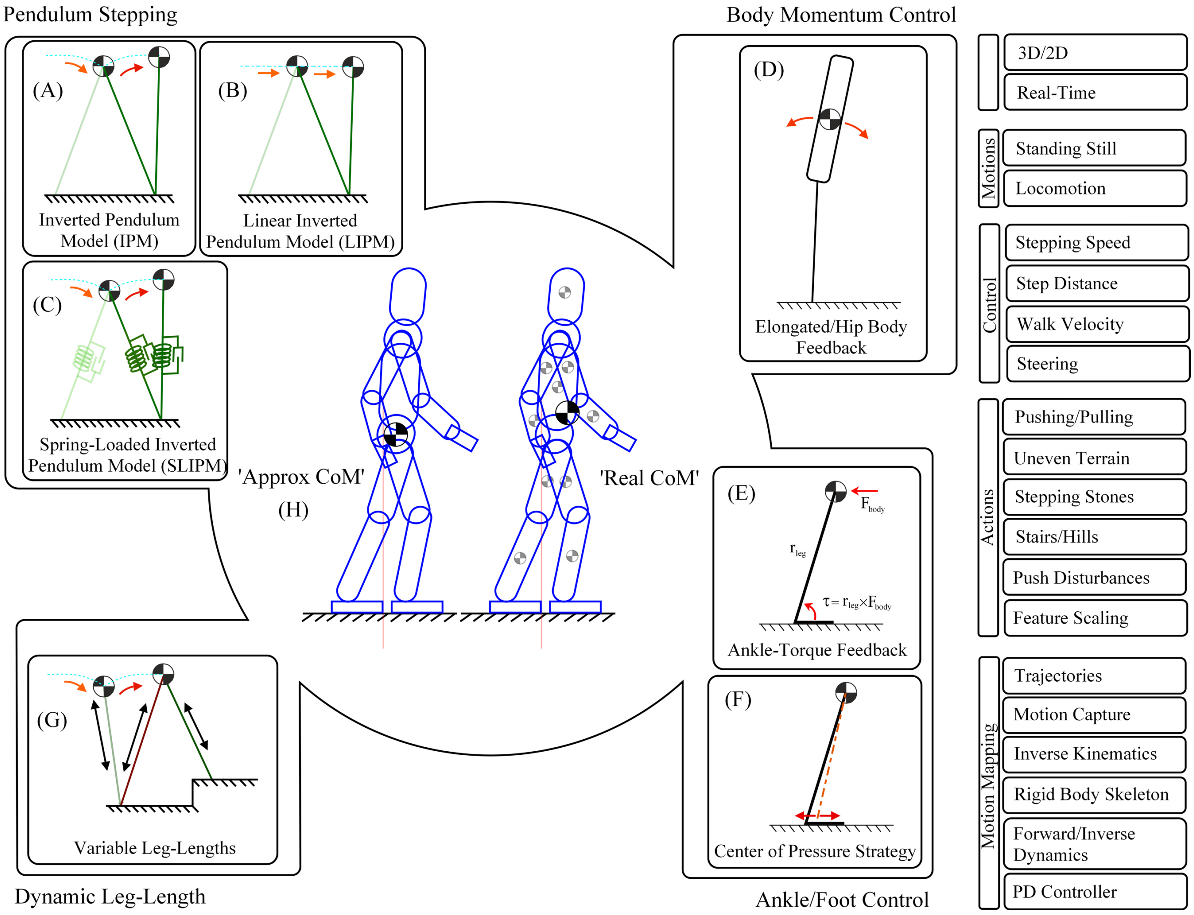


Figure 6.3: Stepping Model Components - While there are different flavors and approaches for generating stepping motions based upon the pendulum model, we illustrate and compare the logic and features that each component provides. (A) The inverted pendulum model (IPM) was originally a biomechanically inspired approach [VS72, HG77] that later gained recognition in robotics [MS84b, KT95] and later the graphics community [KKI06]. (B) Linear inverted pendulum model (LIPM) [KKK*01b]. (C) Spring-loaded inverted pendulum (SLIP) [GOAS12]. (D) Elongated-body (either for more life-like walking with upper-body posture as shown by Kenwright [KDM11] (i.e., Chapter 4), or as a means of counter-balance correction [PT06]. (E) Ankle-Torque Feedback [Ken12d] (i.e., Chapter 5). (F) Centre of Pressure Strategy [PT06]. (G) Variable leg-lengths [PCD06] with an IP model to compensate for push disturbances using the capture-point, while Ito and Sasaki [IS11] performed lateral stepping based on zero moment point feedback for adaptation to slopes. (H) The hip midpoint as the COM position similar to SIMBICON [YLvdP07] due to it being fast and simple, however, the model could be adapted to constantly update and track the full articulated body COM position synonymous with the approach by Tsai et al. [TLC*10].

6.3 Pendulum Stepping Model (Capture Height)

As presented in previous chapters, the biomechanically inspired inverted pendulum (IP) is at the **heart** of our character balancing model's mechanism, since it is an intuitive, computationally fast, algorithmically simple, and robust technique for providing dynamic, interactive, and controllable stepping information. The uncomplicated IP model is a point-mass supported by a single telescopic mass-less leg (as shown in Figure 6.3). While there are numerous extensions, for example, with double-support feet and multi-mass body parts, we begin by focusing primarily on the elementary model (e.g., single leg point-mass rigid/spring leg model).

The stepping motion of a rigid-leg pendulum model on flat ground under ideal situations can maintain a perpetual (i.e., constant) walking motion by converting energy between kinetic and potential energy (i.e., point-mass continually pole-vaulting over the supporting leg). The basic pendulum stepping motion is **passive** by default; however, an **active** system allows us to add controlling feedback forces (and torques) into the stepping mechanism to gain greater control (e.g., speed and steering).

The low-dimensional IP model on its own has a number of limitations (e.g., pin-point feet and steering inability) and must be combined with a control mechanism (e.g., foot, hip, or ad-hoc feedback forces) to make the model a viable solution for generating controllable character-based motions. The stand-alone pendulum stepping model limitations are:

- Continual state of motion (i.e., always needs to keep stepping to remain upright and balanced)
- Pin-Point Feet (i.e., no support area or ankle torque)
- No feet or pelvis orientation information
- No postural information (i.e., upper-body orientation)
- Mass-less legs
- No feet trajectory information (e.g., height, speed, direction)
- Requires multiple steps for steering (i.e., cannot start locomotion from a stop and needs to wait for gravity to pull it forwards, which can be the wrong desired direction) - no

steering control

- Does not account for double-support foot-placement (i.e., when both feet are on the ground supporting the body)
- No friction or ground-feet slipping

While there are different techniques for solving the simple analytical IP problem mechanics to accomplish specific goals (e.g., continual locomotive stepping), we use a "velocity-based" approach for solving the IP model's equations (i.e., capture-point approach). We control the direction and speed of the pendulum-based model by means of different control mechanisms (i.e., variable leg-length and ankle-torque) to enable the user to vary the step position and duration while remaining balanced and upright. We then take the low-level stepping model information and map it onto an articulated character to create full-body coordinated motions.

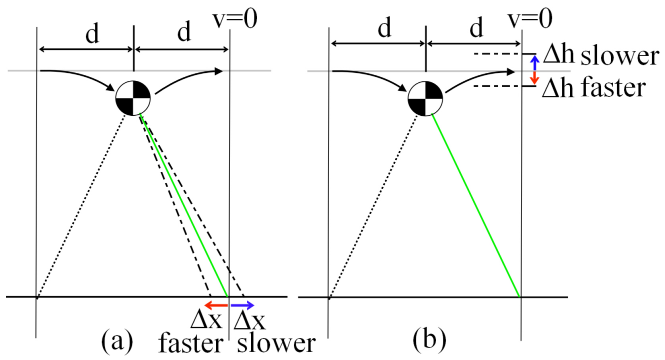


Figure 6.4: Capture-Point Comparison - The capture-point concept for (a) capture-point "distance", and (b) capture-point "height".

The model is made as simple as possible (i.e., a low-dimensional model) and gives us the following advantages:

- The balancing motion can be decoupled from the overall movement
- We can focus specifically on one crucial motion
- The full-body movement can be reconstructed around a simple model (we can take advantage of the redundancy as a secondary priority means of mixing in behavioral emotions, such as tired and happy)

6.3.1 Capture-Point

The capture-point defined by Pratt and Tedrake [PT06] is a position on the ground that would bring the final pendulum model to a complete stop when vertically upright (i.e., final velocity equal to zero). We define this as a capture-point "distance", since it calculates the unknown step-distance based on a fixed length-leg approximation. However, we define a capture-point "height" based upon the same principle; however, step-distance is known and the final leg-length height is what we calculate. The reasoning behind this is that in complex virtual environments, a character's stepping location can be limited or constrained (e.g., in stepping stones). For a pin-point rigid-leg pendulum model, we calculate the destination leg-length for the step transitions necessary based upon the foot-placement distance that would result in the mass reaching a zero velocity when vertical. We illustrate the capture-point distance and capture-point height in Figure 6.4.

Capture-Point "Distance": The capture-point "distance" is the specific foot position from the current projected location on the ground that will bring the pendulum to a stop (i.e., velocity will reach zero when the pendulum is standing vertically upright and straight), as shown in Figure 6.5. This method was proposed by Pratt and Tedrake [PT06] who applied it to both a pendulum model (i.e., arc like trajectory) and linear-pendulum model (i.e., flat fixed height trajectory).

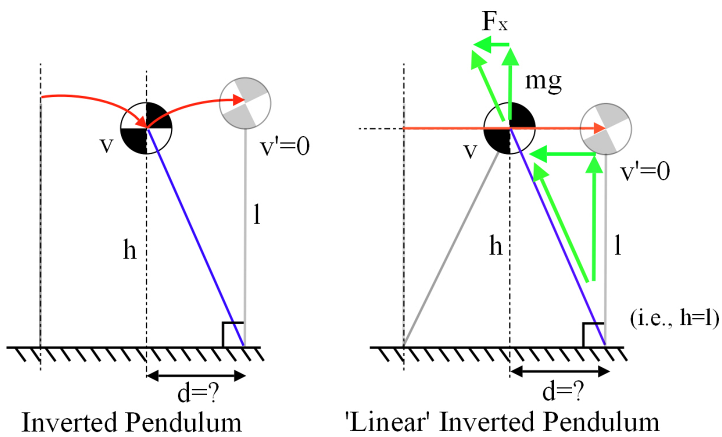


Figure 6.5: Capture-Point "Distance" - Estimating the capture-point "distance" based on a rigid mass-less support leg.

Capture-Point "Height": In contrast to the capture-point "distance", which focused on finding the unknown stepping distance necessary to bring the pendulum mass to a vertical upright stop, the capture-point 'height' focuses on finding the final leg-height given a specific stepping distance to achieve the same task. *If we specify a specific foot-placement location it means we can carefully control and navigate the pendulum stepping model in complex virtual environments.* However, the formula for calculating the leg-length is not as elegant and straightforward as the capture-point "distance" approach.

We solved the low-dimensional problem using a heuristic search method. Our lightweight model was computationally efficient enough to iteratively search for the best fitting leg-length for the controlled step. For cases where no solution could not be found, we put this down to the mass not possessing enough momentum to pole vault itself to the target. As a workaround for these cases, we keep both feet on the ground and shift the centre-of-mass towards the target foot (i.e., centre-of-mass kept above the support region). Once the centre-of-mass is above the support foot of the target foot, the leg can be returned to the rest-length and pushed in the desired direction by the ankle-foot torque to resume the pendulum like stepping motion.

6.3.2 Control Mechanisms

The capture-point does not provide a means of 'control' and, hence, must be combined with an additional control mechanism (e.g., body-momentum or ankle-torque) so that we can steer and guide the pendulum during foot transitions. The control mechanism keeps the pendulum balanced and allows us to direct the movement in a controlled manner. In summary, combining the IP model with a feedback control mechanism fixes a number of inherent oversimplification limitations to produce a viable practical solution that is robust and controllable.

The three fundamental control mechanisms we focus on are:

- Ankle-Torque Feedback (e.g., to avoid constant stepping, provide additional control, and static balancing data, as discussed in Chapter 5)
- Elongated-body (e.g., hip-joint torque, steering, and postural feedback information)
- Variable Leg-Lengths (e.g., walking up stairs and changing terrain heights)

Our approach exploits the ankle-torque control mechanism in conjunction with the capture-point height. The approach provides an intuitive solution that is computationally efficient and controllable. The ankle-torque control keeps the centre-of-pressure (COP) at a fixed location, as discussed in Chapter 5 compared to other techniques, such as the Centre of Pressure strategy [PT06] or Zero Moment Point (ZMP) concept [VS72].

6.3.3 Mapping: Bridging the Gap between Control and Kinematics (IP to Full-Body)

We address the issue of mapping the low-dimensional model onto a fully articulated biped skeleton. There are a number of *unknowns* that must be addressed, such as foot and arm trajectories. The inverse kinematic (IK) solver maps a solution between our IP model and our highly articulated biped skeleton hierarchy. While the highly articulated skeleton contains a huge amount of flexibility and ambiguity (i.e., multiple solutions for achieving the same goal), in comparison to the simplified low-dimensional model which is minimalistic, computationally efficient, and straightforward to solve. The simplified model, however, possesses multiple attributes (i.e., overall centre-of-mass position and feet locations) that are common to the highly articulated skeleton, which are fundamental for generating physically correct balanced biped stepping poses. To accomplish the mapping efficiently, we subdivided the IK problem into two separate parts (i.e., upper and lower body). This made solving the IK problem faster and more robust. Moreover, our adaptive stepping technique solves balancing logic while the upper-body motions are left free for alternative actions, such as personality and style (e.g., looking around, arms' swaying). The elongated body, as discussed in Chapter 4, provides postural information for the pelvis and torso, as shown in Figure 6.14.

We focused on lower body movements since they are the most crucial for upright balancing motions [TLC*10] compared to the upper-body. While, foot trajectories were generated by linearly interpolating along Bezier splines, between the current and desired foot transition positions.

The final motions **did not** use any motion capture or key-framed libraries. Hence, some

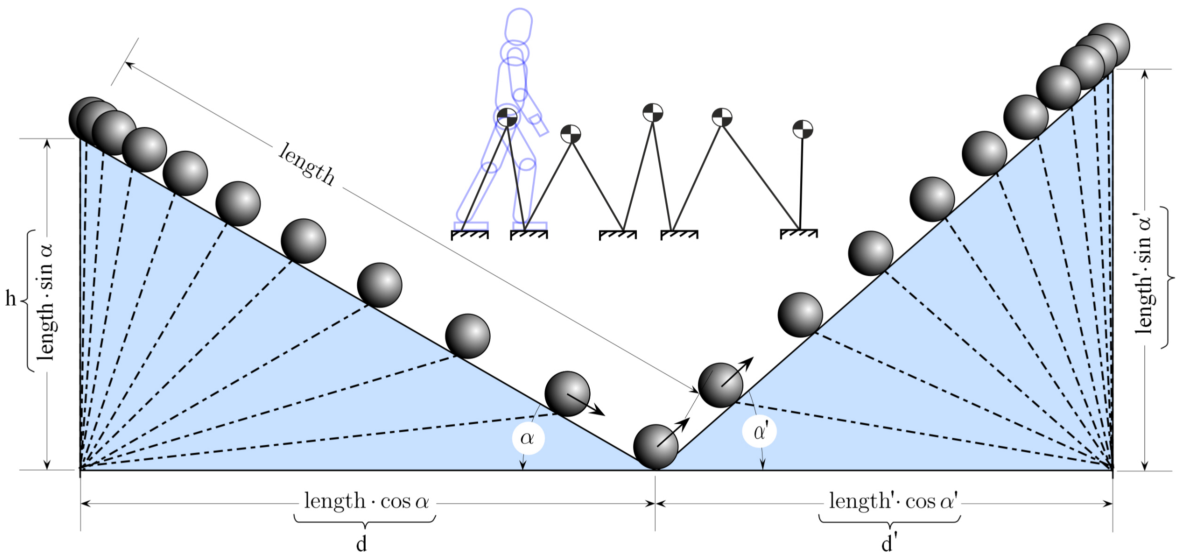


Figure 6.6: Capture-Point "Height" - Estimating the capture-point "height" based on a rigid mass-less support leg (illustrate a linear transition). The principle focuses on trading energy to increase or reduce momentum by means of increasing or decreasing the leg-length between foot-placement transitions.

of the motions may have appeared to look a bit robotic. This approach can be remedied by combining the generated motions with a multiple priority IK solution (i.e., with a primary and secondary goal). Whereby, the primary balanced physically correct motions are always enforced, while the secondary less crucial aesthetically pleasing life-like motions are combined on top from sources, such as key-framed libraries or random motion generators.

6.4 Results

The stepping framework was mapped on an uncomplicated mannequin character model shown in Figure 6.8, Figure 6.12, and Figure 6.13. The low-dimensional model maps the essential elements (i.e., the elongated-body, left and right foot positions/orientations) onto the mannequin limbs to generation the final character movements.

- Walking/exploring a virtual world (i.e., various terrain). The stepping motions are directed and controlled by the user (e.g., direction and speed).
- Push disturbances (e.g., various pushes at different parts of the body of different magnitude and direction).

We simulated a virtual agent that explored a complex environment (i.e., walking around on flat terrain, stairs, slopes, and stepping over objects) while numerous disturbances were applied (e.g., random push forces). All the generated motions were produced on-the-fly. All the simulations were carried out on an Intel Core i7-2600 CPU with 16-GB of memory running Windows-7 64-bit on a desktop PC. We used a simulation time-step of 0.01s and gravitational constant of 9.81. All the simulations ran in real-time and were written as a single-threaded application. Our implementation was written in C# using Visual Studio 2010.

In the simulation experiments, we ignore self-collision for faster performance (e.g, arms and body). However, self-collision can be an important factor during push disturbances, since it can cause the arms to intersect with the body.

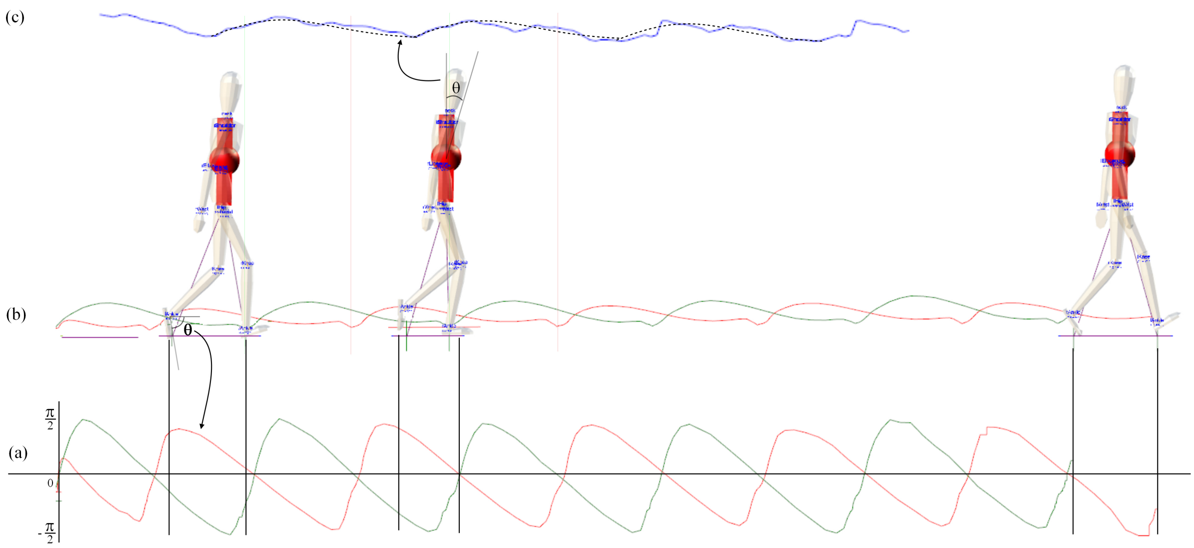


Figure 6.7: Feet and Ankle Trajectories - (a) ankle-angle between step transitions, (b) ankle-foot path trajectory, and (c) upper-body angle deviation). An important note, is methods, such as SIMBICON [YLvdP07], keeps their feet parallel to the ground during gait motions, producing a robot-like marching effect, while, in reality, humans tilt and swing their foot (i.e., toe-stubbing effect on take-off).

The flexible nature of our stepping model is shown in Figure 6.8 and shows screen captures of our pendulum-based approach mapped onto a simple biped rig. The model performs a variety of controlled stepping motions under different conditions (e.g., slopes, stairs, pushes, and avoiding holes).

We have presented and demonstrated a novel stepping model based on the combination of different techniques (i.e., capture-point height and ankle-torque) that is flexible, robust, and computationally efficient. The final biped stepping motions remained balanced against perturbations, such as random sudden pushes, and generated movements similar to those observed in humans.

We generated the fundamental stepping actions without any motion capture data. The basic model for maintaining balance was based on a pendulum-based technique and required a minimum number of tunable parameters. While we explained and compared the different control mechanisms, we settled on the uncomplicated ankle-torque feedback mechanism in conjunction with a variable leg-length system. All in all, the generated low-dimensional model can be mapped onto a whole-body biped character to create common upright balanced stepping movements (e.g., walking and standing).

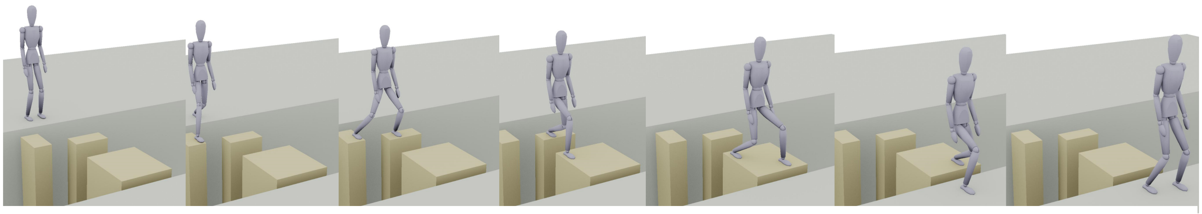


Figure 6.8: Controlled Stepping - We carefully control the stepping locations of feet while navigating a complex environment (i.e., we cannot always place our feet at desired locations and must work within the constraints of the environment).

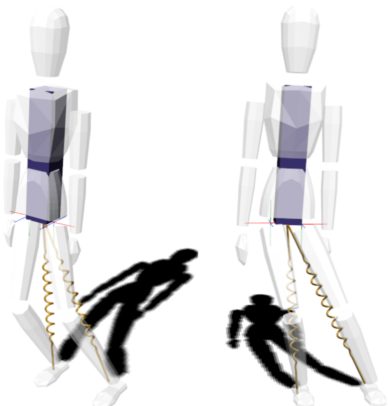


Figure 6.9: Balanced Standing - The elongated rigid body provides a lightweight model for upper body postural information (from Chapter 4) which is further combined with a controllable balance stepping mechanism (from Chapter 5 and 3).

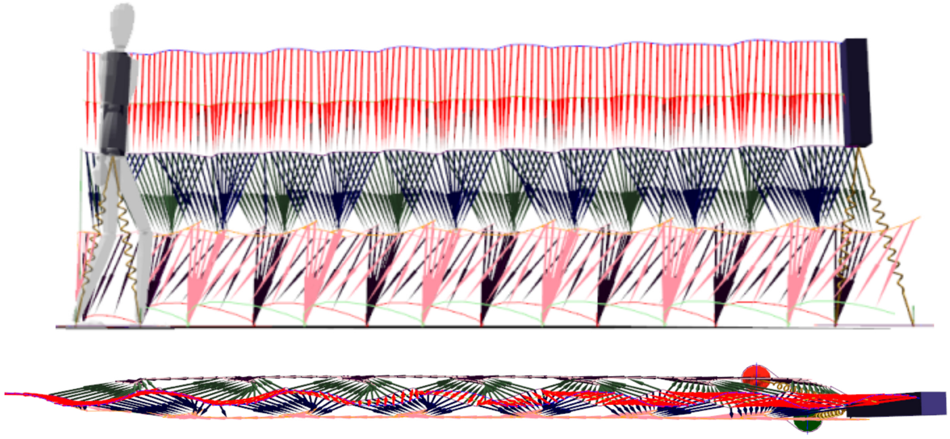


Figure 6.10: Elongated IP Walk Cycle Side View - Motion from the elongated body IP walking at a constant speed.

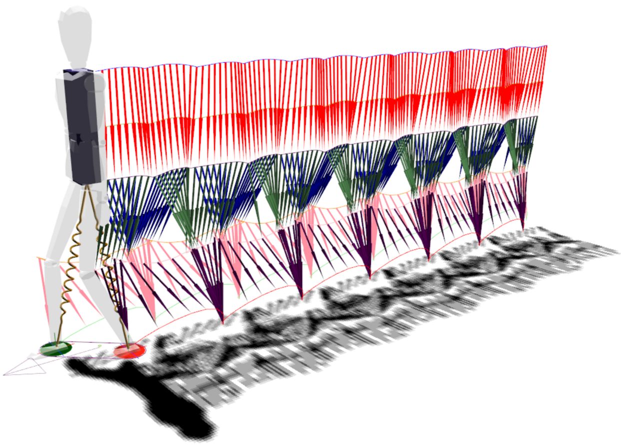


Figure 6.11: Elongated IP Walk Cycle Perspective View - Motion from the elongated body IP walking at a constant speed.

6.5 Discussion

The most significant advantage of this chapter's approach over prior work is: Our low-level method requires no offline optimization, motion capture data, or pre-processing and allows the generation of fundamental character stepping motions for real-time dynamic environment.

We focused on a lightweight physics-based model that decoupled balancing information overall articulated skeleton. This enabled us to focus specifically on one crucial motion, i.e., stepping. The full-body movement was reconstructed around our lightweight model. The mapping of the low-dimensional model onto the articulated biped introduces a lot of redundancy (e.g., arm positions). However, the advantage of the redundancy allows a means to incorporate secondary priority means of mixing in behavioral emotions, such as tired and happy.

There are a number of limitations of our method, which provide opportunities for future work. Since our goal was to generate biped stepping animations without animation data that were interactive and dynamic, on occasion the animations did not look as human-like as other methods, which used animation data [MLPP09, TLC*10, dSAP08b]. Since motion-capture data allows controllers to be automatically tuned [dSAP08b, MLPP09] [SKL07, TLC*10] to create motion styles, that are very true-to-life, as recently demonstrated by Muico et al. [MLPP09] who simulated an assortment of locomotive skills.

The simulation results did demonstrate specific feature differences, such as our model's ability to compensate for larger push disturbances due to the compensating foot-ankle feedback control force, in addition, to the ability to handle greater sloping terrains due to actively adjusting leg-lengths. We did not include upper body control (e.g., arms), however, this offers a further avenue of research and would increase the naturalness of the motion.

6.6 Conclusion

This chapter has presented a simulation and control framework for balanced biped character stepping motions. Our novel approach generated upright motions without key-framed data. We took three key components developed in previous chapters (i.e., stepping mechanics, postural feedback, and ankle-foot control), to produce a synergistic framework that encapsulates an overall more practical solution compared to the individual parts.

We took an uncomplicated approach so that our system was fast and relatively easy to implement. We manually defined the specifications of the low-dimensional model (i.e., legs and body dimensions) for testing. The generated motions, however, occasionally appeared unnatural. The final motions **do not** include arm movement and only focus on upright stepping motions. We do not include foot sliding in our model, which is important in highly dynamic models, such as jumping kicks.

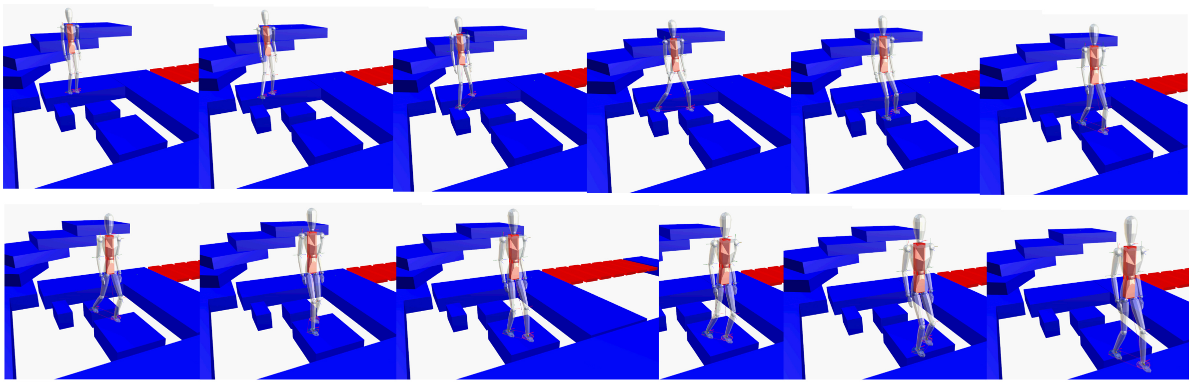


Figure 6.12: Coordinated Stepping - Coordinated foot-placement stepping while navigating terrain with holes.

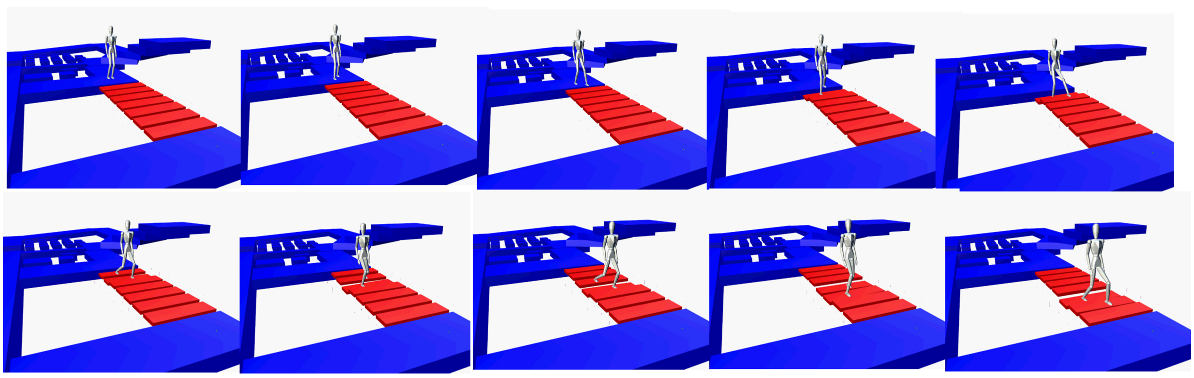
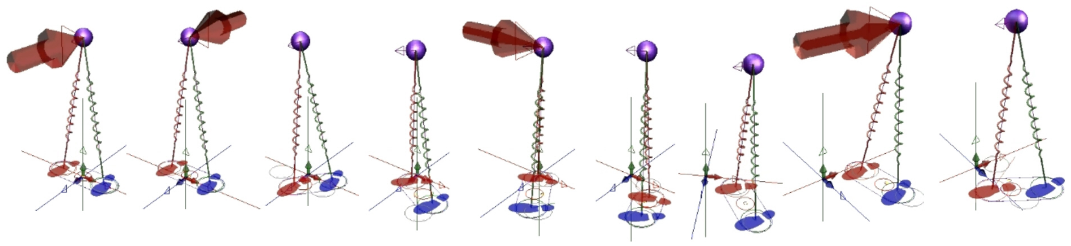
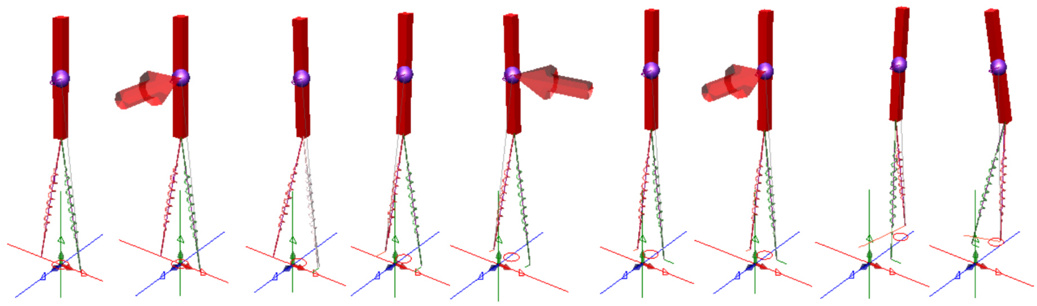


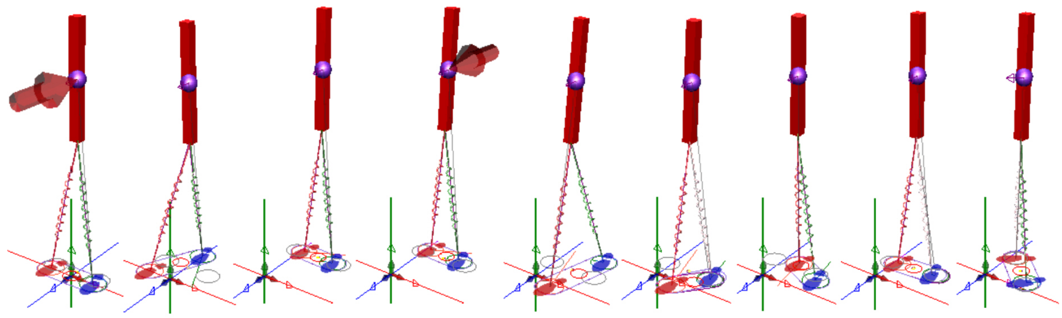
Figure 6.13: Unstable Terrain Stepping - Navigating an unstable terrain (i.e., a bridge formed of rigid body cubes fixed in place with spring-dampers).



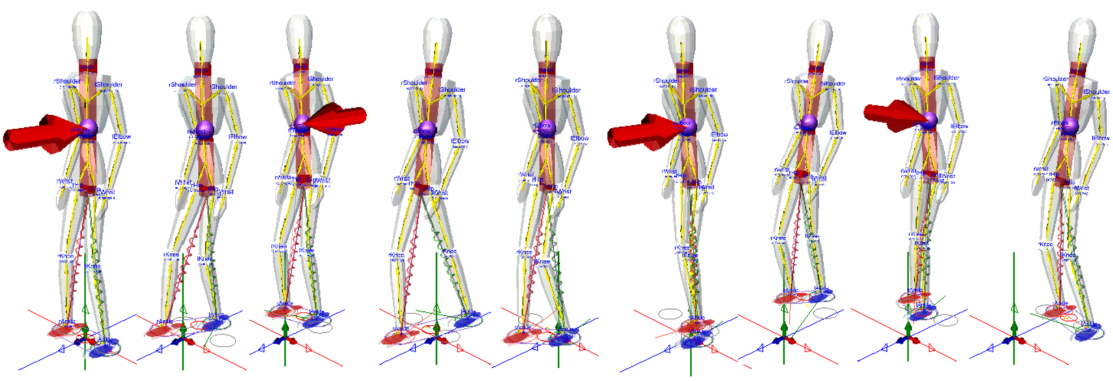
(a) Inverted Pendulum with Foot Support Area Approximation (comfort stepping)



(b) Extended Inverted Pendulum with Elongated Body (Posture Movement)



(c) Elongated Body with Foot Support Area Approximation



(d) Map Pendulum Stepping Model onto an Articulated Skeleton

Figure 6.14: Simulation Screen Captures - (a) the inverted pendulum with foot-support area, (b) elongated-body, (c) combined foot-support area and elongated-body, and finally (d) an integrating articulated character with body posture and foot-ankle feedback.

Chapter 7

Conclusion

An effective 3D stepping control algorithm that is computationally fast, robust, and easy to implement is extremely valuable and important to character animation research. In this thesis, we explained and demonstrated a novel lightweight physics-based technique for procedurally generating upright 3D biped character stepping animations. Our low-dimensional approach focused on controlling the COM, ankle-torque, elongated-body, and foot placement position, to produce coordinated stepping actions. The final solution consumed little computational time, memory, and bandwidth. Furthermore, because we focused on controlling high-level goals the complexity and difficulties of directing the articulated character was reduced.

A character's basic features (e.g., COM and linear momentum with respect to the support foot) form a low-dimensional balancing problem that we solved using the popular IP based technique. During stepping movements, the IP model pivots the COM about the support foot similar to a human and produces a pole-vault like motion. We demonstrated that extending the oversimplified IP model (e.g., with an ankle-torque feedback force) enhances a character's robustness to disturbances and creates a more controllable solution. Our low-level technique automatically computes where and when to step under disturbances while maintaining specific goals (e.g., walk speed and direction).

7.1 Flexible Practical Approach

Our balanced biped stepping approximation model is simple and flexible enough to be easily integrated within a character-based animation-system to create fundamental avatar motions (i.e., standing and walking). The model's uncomplicated set of control strategies can effortlessly be understood, implemented, or extended without difficulty by the user. The basic upright motions are generated without key-frame data and can handle various disturbances (e.g., pushes) and diverse terrains (e.g., ramps). Furthermore, the user is able to intuitively modify and tune the different control parameters by hand to create the final controlled motions.

7.2 Natural Dynamics

While the stepping movements are achieved through the natural dynamics of the balancing mechanism, the kinematic model ensures joint limits are always enforced (e.g., the knee inverting backwards). The work in this dissertation helps bridge the gap between more automatic non data-driven approaches for generating natural dynamic balanced motions that use an intelligent self-driven solution to accomplish the goal (e.g., standing, walking, and running).

7.3 Beyond Key-Frame Animations

Initially, at the start of the dissertation (i.e., in Section 2.5), we discussed various successful examples of where computer graphics had focused specifically on creating interactively controlled characters based upon physics rather than motion-capture. While this dissertation follows a similar goal of generating biped motions by means of intelligent physics-based methods, we primarily focus on a procedural approach without key-framed data that can run in real-time and generate physically plausible, balanced upright biped motions on-the-fly that can handle a variety of unforeseen situations (e.g., pushes, uneven terrain, and different feature sizes).

7.4 Discussion

While it is difficult to compare different approaches, and quantify certain aspects, such as, realism, flexibility, and computational cost, we should note that other methods have demonstrated successfully the creation of upright biped animations that are responsive and dynamic [BC89, CBvdP10, dLMH10, TLC*10]. Each of these methods exemplify in one thing or another, such as accuracy, flexibility, control, or realism. This dissertation focused on a novel lightweight physics-based system that synthesized balanced biped stepping motions (i.e., possessing key elements, such as, running in real-time, 3D/2D, handling sloping terrain, and being mapped onto a complex skeleton, i.e., 36 DOF).

In conclusion, we have presented a lightweight model that generates a controllable upright biped stepping system, which can adapt to unforeseen circumstances, such as pushes and terrain height deviations. Our approach has the advantage of being intuitive, simple to program, and easy to extend. Unlike key-framed approaches, our algorithm does not require the animator to repeatedly work on creating new key-frames or adapting motion-capture data for every walk or step sequence. Our real-time balance-aware physics-based solution uses a simple feedback mechanism that can synthesize poses and produce new motions "on-the-fly" without artist intervention (i.e., animation data).

7.5 Summary and Future Work

There are a few areas of improvement and exploration in future research, such as mixing our approach with different controller techniques to make it a viable solution for a real-world virtual gaming environment (e.g., including actions such as kicking, climbing, and crawling).

While this thesis showed one approach for generating upright balanced motions, there are many other approaches, and it is hard to say, which is the best. There are many avenues for further work in producing character motions that are interactive and dynamic. It is hoped that this work will eventually lead to a better understanding of generating and synthesizing more intelligent, self-driven, and competent character animation solutions. However, while we attempt to mimic human movements with basic controller models, there will always be more things to develop. For example, once walking, standing, and running has been developed, there is the dancing, climbing, aerobatics, movements to study.

Future extensions to the research presented in this thesis. For example, one direction would be crowds, as shown in Figure 7.1, lots of instances of the model can be spawned in real-time due to the model's minimalistic overhead. Multiple autonomous characters can navigating confined areas and interactive (i.e., collide and walk around) as shown in Figure 7.2. Exploiting paralization, such as the Graphical Processing Unit (GPU), it would be possible to spawn dense numbers of interactive crowd characters. Alternatively, we could further build upon the model to include additional features and actions, such as the inclusion of arms analogous to the legs spring-damper system, as shown in Figure 7.3, to model actions, such as climbing and rising animations.

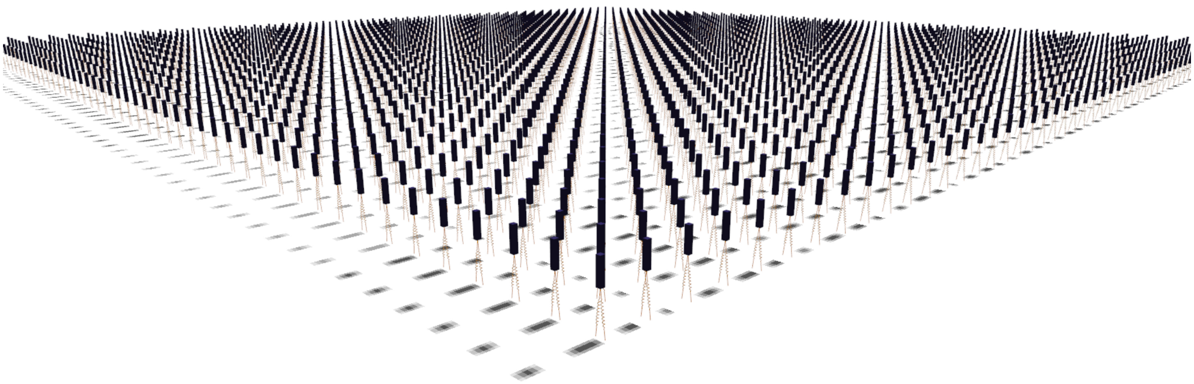


Figure 7.1: Crowd Simulations - Hundreds of Instances of the extended inverted pendulum (IP) model. Our model's uncluttered and straightforward approach of generating simple biped avatars means it is ideal for simulating large groups of interactive pedestrians. For example, the figure shows hundreds of instances of our model running in real-time.

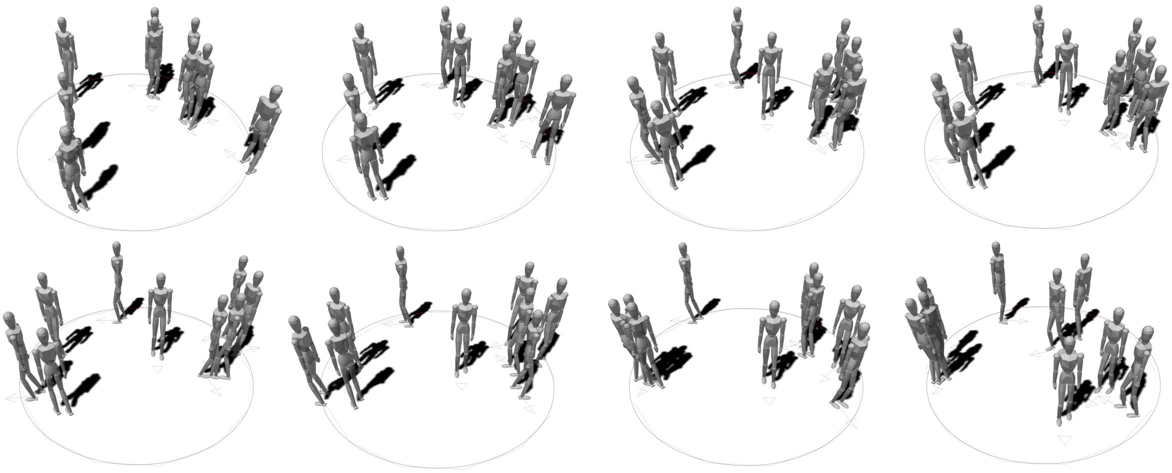


Figure 7.2: Close Interaction - Nine biped characters repeatedly bumping into one another because while being forced to walk in random directions, back and forth within a small circular boundary.

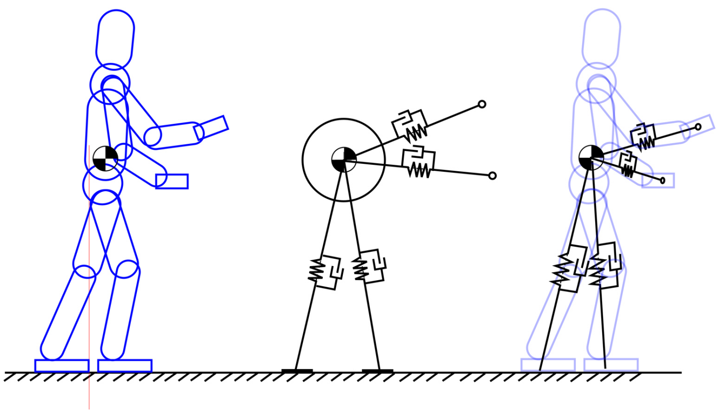


Figure 7.3: Simplified Point-Mass Biped Model (Additional Controllers) - Possible future model enhancements: The arms and legs of the character are analogous with a spring-damper mechanism and can be combined to work in synergy to control the overall character's center-of-mass (COM) and produce other motions, such as climbing and get-up motions.

7. CONCLUSION

Appendix A

Appendix

A.1 Articulated Skeleton Joint Parameters

For the tracking controller mechanism, we use the popular proportional-derivative (PD) method to calculate the joint torques by minimizing the displacement between the current and reference joint angles. Whereby, this approach has been successfully used in the past to create walking motions [vdP96, LvdPF96], athletic motions [HWBO95], and reactive motions [OM01, ZH02]. The PD controller uses the generated skeleton pose and elastic-damper coefficients to determine the necessary virtual torques to apply to the articulated rigid body biped that emulates muscles to create the final movements. Finally, the joint torque for each of the character's joints was limited to a maximum of 500Nm.

The torque produced by the PD controller is linearly proportional to the displacement error. The elastic-damper gains for each DOF joint of the articulated skeleton were determined through human intervention to achieve the necessary responsiveness (see Table A.1). Another approach, that we *do not* use, could be to perform an off-line search optimization to find the spring-damper coefficients, as done by Geijtenbeek et al. [GPvdS12].

A. APPENDIX

Joint Name	k_p	k_d	Scale (xyz)		
Spine	900	90	0.6	1.0	1.0
Hip	350	35	1.0	0.5	1.0
Knee	320	32	1.0	1.0	1.0
Ankle	120	12	1.0	1.0	0.5
Shoulder	110	11	0.5	1.0	1.0
Wrist	30	3	0.5	1.0	1.0
Neck	110	11	1.0	0.5	1.0
Elbow	105	10	0.5	1.0	1.0

Table A.1: The character model's default joint PD coefficients (shown in Figure A.1).

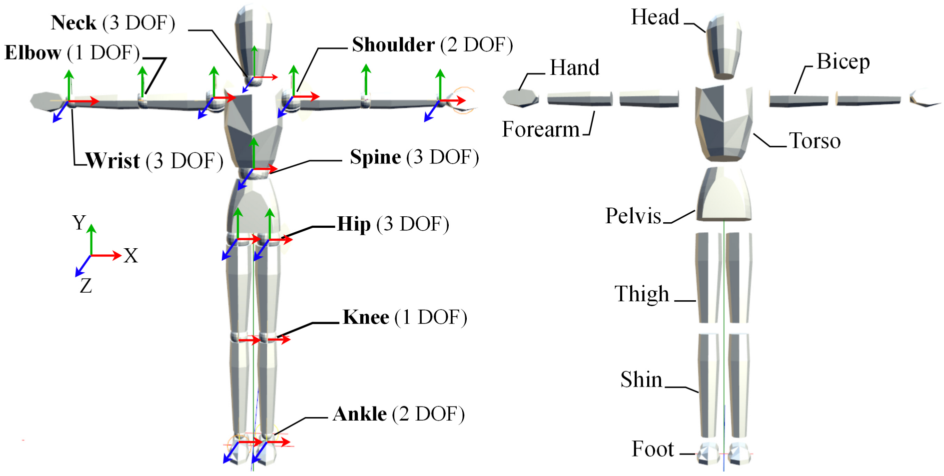


Figure A.1: Character joint names, individual body parts names, and degrees-of-freedom (DOF).

Joint Name	Rotational DOF		Limits					
		No	x-axis Lower	x-axis Upper	y-axis Lower	y-axis Upper	z-axis Lower	z-axis Upper
Spine	3	-	-40	85	-45	45	-50	50
Hip	3	x2	-170	40	-130	30	-25	25
Knee	1	x2	0	160	-	-	-	-
Ankle	2	x2	-40	40	-	-	0	30
Shoulder	2	x2	-	-	-75	150	-85	85
Wrist	3	x2	-85	45	-40	40	-85	85
Neck	3	-	-45	90	-55	55	-85	85
Elbow	1	x2	-	-	0	110	-	-

3D Skeleton Model 30 DOF Total

Body Part Name	No	Mass (kg)	Centre of Mass (x,y,z in metres)		
Head	1	4.80	0.00,	1.64,	0.00
Pelvis	1	21.00	0.00,	1.01,	0.00
Torso	1	25.50	0.00,	1.31,	0.01
Bicep	2	3.00	+/-0.30,	1.41,	0.00
Forearm	2	1.60	+/-0.57,	1.41,	0.00
Hand	2	0.70	+/-0.80,	1.41,	0.00
Thigh	2	8.70	+/-0.06,	0.68,	0.00
Shin	2	3.70	+/-0.06,	0.02,	0.06
Foot	2	1.40	+/-0.06,	0.26,	0.00
	15	89.5			

Table A.2: A list of the local 3D skeletons 30 degrees-of-freedom (DOF) limits (note an additional 6 DOF from the world root, and angular limits are shown in degrees) (shown in Figure A.1).

References

[AdSP07] Abe Y., da Silva M., Popović J.: Multiobjective control with frictional contacts. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), SCA '07, Eurographics Association, pp. 249–258. 10, 45

[AF02] Arikan O., Forsyth D. A.: Interactive motion generation from examples. *ACM Trans. Graph. 21*, 3 (jul 2002), 483–490. 8

[AFO03] Arikan O., Forsyth D. A., O'Brien J. F.: Motion synthesis from annotations. *ACM Trans. Graph. 22*, 3 (jul 2003), 402–408. 8

[AFO05] Arikan O., Forsyth D. A., O'Brien J. F.: Pushing people around. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), SCA '05, ACM, pp. 59–66. 14, 58, 59

[AFP*95] Auslander J., Fukunaga A., Partovi H., Christensen J., Hsu L., Reiss P., Shuman A., Marks J., Ngo J. T.: Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics (TOG) 14*, 4 (1995), 311–336. 22

[Ale92] Alexander R. M.: A model of bipedal locomotion on compliant legs. *Philos Trans R Soc Lond B Biol Sci. 338*, 1284 (oct 1992), 189–198. 24

[AS07] Atkeson C. G., Stephens B. J.: Multiple balance strategies from one optimization criterion. In *Humanoids* (2007), pp. 57–64. 14

[BC89] Bruderlin A., Calvert T. W.: Goal-directed, dynamic animation of human walking. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1989), SIGGRAPH '89, ACM, pp. 233–242. 67, 105

[BF93] Blickhan R., Full R.: Similarity in multilegged locomotion: bouncing like a monopode. *Journal of Comparative Physiology A 173*, 5 (1993), 509–517. 45, 46

[BMT95] Boulic R., Mas R., Thalmann D.: Position control of the center of mass for articulated figures in multiple support. *Computer Animation and Simulation 95* (1995), 130–143. 8

[BMT96] Boulic R., Mas R., Thalmann D.: A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics 20*, 5 (1996), 693 – 701. 8

[CBvdP09] Coros S., Beaudoin P., van de Panne M.: Robust task-based control policies for physics-based characters. In *ACM SIGGRAPH Asia 2009 papers* (New York, NY, USA, 2009), SIGGRAPH Asia '09, ACM, pp. 170:1–170:9. 22, 42

[CBvdP10] Coros S., Beaudoin P., van de Panne M.: Generalized biped walking control. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 130:1–130:9. xi, 9, 14, 17, 21, 23, 24, 26, 27, 35, 36, 37, 53, 59, 81, 105

[CBYvdP08] Coros S., Beaudoin P., Yin K. K., van de Pann M.: Synthesis of constrained walking skills. In *ACM SIGGRAPH Asia 2008 papers* (New York, NY, USA, 2008), SIGGRAPH Asia '08, ACM, pp. 113:1–113:9. 42

[CPP99] Chew C.-M., Pratt J., Pratt G.: Blind walking of a planar bipedal robot on sloped terrain. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* (1999), vol. 1, IEEE, pp. 381–386. xi, 21

[CRTW05] Collins S., Ruina A., Tedrake R., Wisse M.: Efficient bipedal robots based on passive-dynamic walkers. *Science 307*, 5712 (2005), 1082–1085. 24

[DH94] Dunn E. R., Howe R. D.: Towards smooth bipedal walking. *Robotics and Automation IEEE International Conference 3* (1994), 2489–94. 24

[dLMH10] de Lasa M., Mordatch I., Hertzmann A.: Feature-based locomotion controllers. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 131:1–131:10. 14, 17, 105

[dP97] de Panne M. V.: From footprints to animation. *Computer Graphics Forum 16* (1997), 211–223. 22

[DSAP08a] Da Silva M., Abe Y., Popovi J.: Simulation of human motion data using short-horizon model-predictive control. *Computer Graphics Forum 27*, 2 (2008), 371–380. 10, 22

[dSAP08b] da Silva M., Abe Y., Popović J.: Interactive simulation of stylized human locomotion. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 82:1–82:10. 10, 23, 100

[EvB10] Egges A., van Basten B.: One step at a time: animating virtual characters based on foot placement. *Vis. Comput. 26*, 6-8 (jun 2010), 497–503. 9

[FFW00] Full R. J., Farley C. T., Winters J. M.: Musculoskeletal dynamics in rhythmic systems: a comparative approach to legged locomotion. In *Biomechanics and neural control of posture and movement*. Springer, 2000, pp. 192–205. 45, 46

[FP03] Fang A. C., Pollard N. S.: Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics (TOG) 22*, 3 (2003), 417–426. 10

[FvdPT01a] Faloutsos P., van de Panne M., Terzopoulos D.: Composable controllers for physics-based character animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 251–260. 13

[FvdPT01b] Faloutsos P., van de Panne M., Terzopoulos D.: The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers & Graphics 25*, 6 (2001), 933–953. 22

[GCRC98] Garcia M., Chatterjee A., Ruina A., Coleman M.: The simplest walking model: Stability, complexity, and scaling. *Journal of Biomechanical Engineering Transactions of the Asme 120* (April 1998), 281–288. 24

[GMHP04] Grochow K., Martin S. L., Hertzmann A., Popović' Z.: Style-based inverse kinematics. *ACM Transactions on Graphics (TOG) 22*, 3 (August 2004). 11

[GOAS12] Garofalo G., Ott C., Albu-Schaffer A.: Walking control of fully actuated robots based on the bipedal slip model. In *2012 IEEE International Conference on Robotics and Automation RiverCentre, Saint Paul, Minnesota, USA May 14-18, 2012* (May 2012). xi, xv, 21, 90

[GPvdS12] Geijtenbeek T., Pronost N., van der Stappen A. F.: Simple data-driven control for simulated bipeds. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2012), SCA '12, Eurographics Association, pp. 211–219. 109

[GSB06] Geyer H., Seyfarth A., Blickhan R.: Compliant leg behaviour explains basic dynamics of walking and running. *Proc. R. Soc. B 273*, 1603 (nov 2006), 2861–2867. 24

[GT95] Grzeszczuk R., Terzopoulos D.: Automated learning of muscle-actuated locomotion through control abstraction. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 63–70. 22

[Han06] Hansen N.: The cma evolution strategy: A comparing review. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms* (2006), 75–102. 14

[HG77] Hemami H., Golliday C.: The inverted pendulum and biped stability. *Mathematical Biosciences 34*, 95–110 (1977), 95–110. xv, 23, 90

[HGJ77] Hemami H., Golliday Jr C.: The inverted pendulum and biped stability. *Mathematical Biosciences 34*, 1 (1977), 95–110. xi, 21

[HHC07] Hyon S., Hale J., Cheng G.: Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *Robotics, IEEE Transactions on 23*, 5 (2007), 884–898. 14

[HJO*10] Hodgins J., Jörg S., O'Sullivan C., Park S. I., Mahler M.: The saliency of anomalies in animated human characters. *ACM Trans. Appl. Percept. 7*, 4 (July 2010), 22:1–22:14. 15

[HM04] Hofmann A., Massaquoi S.: A sliding controller for bipedal balancing using integrated movement of contact and non-contact limbs. In *Proc. International Conference on Intelligent Robots and Systems (IROS* (2004), pp. 1952–1959. 14

[HP97] Hodgins J. K., Pollard N. S.: Adapting simulated behaviors for new characters. In *Proceedings of the 24th annual conference on Computer graphics* *and interactive techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 153–162. 15

[HS99] Hsiao T Elizabeth, Stephen R. N.: Biomechanical influences on balance recovery by stepping. *Journal of Biomechanics 32*, 10 (1999), 1099–1106. 23

[HWBO95] Hodgins J. K., Wooten W. L., Brogan D. C., O'Brien J. F.: Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 71–78. 13, 15, 22, 41, 109

[IAF06] Ikemoto L., Arikan O., Forsyth D.: Knowing when to put your foot down. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), I3D '06, ACM, pp. 49–53. 8

[IAF07] Ikemoto L., Arikan O., Forsyth D.: Quick transitions with cached multi-way blends. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2007), I3D '07, ACM, pp. 145–151. 8

[IS11] Ito S., Sasaki M.: Motion control of biped lateral stepping based on zero moment point feedback for adaptation to slopes. *A.C. Pina Filho (Eds.), Biped Robots, Rijeka: InTech* (2011), 15–34. xv, 90

[Jak01] Jakobsen T.: Advanced character physics. *In Game Developers Conference Proceedings* (2001), 383–401. 12

[JYL09] Jain S., Ye Y., Liu C. K.: Optimization-based interactive motion synthesis. *ACM Trans. Graph. 28*, 1 (Feb. 2009), 10:1–10:12. 13, 22

[KDM11] Kenwright B., Davison R., Morgan G.: Dynamic balancing and walking for real-time 3d characters. In *Proceedings of the 4th international conference on Motion in Games* (Berlin, Heidelberg, 2011), MIG'11, Springer-Verlag, pp. 63–73. xi, xv, 6, 21, 23, 57, 90

[KDR05] Kuo A. D., Donelan M., Ruina A.: Energetic consequences of walking like an inverted pendulum: Step-to-step transitions. *Exercise and Sport Sciences 33* (apr 2005), 88–97. 24

[Ken12a] Kenwright B.: Generating responsive life-like biped characters. *The Third Workshop on Procedural Content Generation in Games (PCG 2012)* (may 2012), 1–8. xi, 21, 57

[Ken12b] Kenwright B.: Inverse kinematics–cyclic coordinate descent (ccd). *Journal of Graphics Tools 16*, 4 (2012), 177–217. 6

[Ken12c] Kenwright B.: Real-time character inverse kinematics using the gauss-seidel iterative approximation method. *CONTENT 2012, The Fourth International Conference on Creative Content Technologies* (2012), 63–68. 74

[Ken12d] Kenwright B.: Responsive biped character stepping: When push comes to shove. In *Proceedings of the 2012 International Conference on Cyberworlds* (2012), vol. 2012, pp. 151–156. xi, xv, 6, 21, 90

[KG04a] Kovar L., Gleicher M.: Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph. 23*, 3 (aug 2004), 559–568. 8

[KG04b] Kovar L., Gleicher M.: Automated extraction and parameterization of motions in large data sets. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 559–568. 11

[KGP02] Kovar L., Gleicher M., Pighin F.: Motion graphs. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), SIGGRAPH '02, ACM, pp. 473–482. 8

[KH13] Kenwright B., Huang C.-C.: Beyond keyframe animations: A controller character-based stepping approach. In *SIGGRAPH Asia 2013 Technical Briefs* (New York, NY, USA, 2013), SA '13, ACM, pp. 10:1–10:4. 6

[KHL05] Komura T., Ho E. S. L., Lau R. W. H.: Animating reactive motion using momentum-based inverse kinematics: Motion capture and retrieval. *Comput. Animat. Virtual Worlds 16*, 3-4 (jul 2005), 213–223. 14

[KKI06] Kudoh S., Komura T., Ikeuchi K.: Stepping motion for a human-like character to maintain balance against large perturbations. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida* (may 2006), pp. 2661–2666. xv, 13, 90

[KKK*01a] Kajita S., Kanehiro F., Kaneko K., Yokoi K., Hirukawa H.: The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on* (2001), vol. 1, IEEE, pp. 239–246. xi, 21

[KKK*01b] Kajita S., Kanehiro F., Kaneko K., Yokoi K., Hirukawa H.: The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on* (2001), vol. 1, pp. 239–246. xv, 14, 23, 24, 45, 90

[KKK*02] Kajita S., Kanehiro F., Kaneko K., Fujiwara K., Yokoi K., Hirukawa H.: A realtime pattern generator for biped walking. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* (2002), vol. 1, IEEE, pp. 31–37. xi, 21

[KKK*03] Kajita S., Kanehiro F., Kaneko K., Fujiwara K., Harada K., Yokoi K., Hirukawa H.: Biped walking pattern generation by using preview control of zero-moment point. *Proceedings of the 2003 IEEE International Conference on Robotics & Automation* (sep 2003). xi, 14, 21, 45

[KKS04] Kawachi K., Kanade T., Suzuki H.: A lightweight algorithm for real-time motion synthesis. In *VRST* (2004), pp. 49–56. 12

[KLK04] Komura T., Leung H., Kuffner J.: Animating reactive motions for biped locomotion. In *Proceedings of the 2004 ACM Symposium on Virtual Reality Software and Technology (VRST)* (2004), vol. 1, pp. 32–40. 8, 15

[KMS01] Kajita S., Matsumoto O., Saigo M.: Real-time 3d walking pattern generation for a biped robot with telescopic legs. *In Proc. ICRA* (2001). 24

[KNK04] Kajita S., Nagasaki T., Kaneko K.: A hop towards running humanoid biped. *and Automation, 2004*, April (2004), 629–635. 23

[KS05] Kwon T., Shin S. Y.: Motion modeling for on-line locomotion synthesis. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), SCA '05, ACM, pp. 29–38. 8, 9

[KSG02] Kovar L., Schreiner J., Gleicher M.: Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), SCA '02, ACM, pp. 97–104. 8

[KT91] Kajita S., Tani K.: Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* (1991), IEEE, pp. 1405–1411. xi, 21, 45

[KT95] Kajita S., Tani K.: Experimental study of biped dynamic walking in the linear inverted pendulum mode. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on* (may 1995), vol. 3, pp. 2885–2891. xi, xv, 21, 23, 45, 90

[LCR*02] Lee J., Chai J., Reitsma P. S., Hodgins J. K., Pollard N. S.: Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics (TOG) 21*, 3 (2002), 491–500. 8

[LHP06] Liu C. K., Hertzmann A., Popović Z.: Composition of complex optimal multi-character motions. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), SCA '06, Eurographics Association, pp. 215–222. 10

[LKL10] Lee Y., Kim S., Lee J.: Data-driven biped control. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 129:1–129:8. 13, 16

[LLGC12] Liu G., Li M., Guo W., Cai H.: Control of a biped walking with dynamic balance. In *Mechatronics and Automation (ICMA), 2012 International Conference on* (aug 2012), pp. 261–267. 27, 30

[LvdPF96] Laszlo J., van de Panne M., Fiume E.: Limit cycle control and its application to the animation of balancing and walking. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 155–162. 22, 109

[LvdPF00] Laszlo J., van de Panne M., Fiume E.: Interactive control for physically-based animation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 201–208. 13

[LWS02] Li Y., Wang T., Shum H.-Y.: Motion texture: a two-level statistical model for character motion synthesis. *ACM Trans. Graph. 21*, 3 (jul 2002), 465–472. 8

[McG90] McGeer T.: Passive dynamic walking. *International Journal of Robotics Research 9* (apr 1990), 62–82. 24, 71

[MdLH10] MORDATCH I., DE LASA M., HERTZMANN A.: Robust physics-based locomotion using low-dimensional planning. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 71:1–71:8. xi, 14, 17, 21, 23, 35, 45

[MK05a] MUKAI T., KURIYAMA S.: Geostatistical motion interpolation. *ACM Trans. Graph. 24*, 3 (July 2005), 1062–1070. 8

[MK05b] MUKAI T., KURIYAMA S.: Geostatistical motion interpolation. *ACM Trans. Graph. 24*, 3 (2005), 1062–1070. 11

[MLPP09] MUICO U., LEE Y., POPOVIĆ J., POPOVIĆ Z.: Contact-aware nonlinear control of dynamic characters. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 81:1–81:9. 3, 10, 100

[MM04] MEREDITH M., MADDOCK S.: *Real-time inverse kinematics: The return of the Jacobian.* Tech. rep., Technical Report No. CS-04-06, Department of Computer Science, University of Sheffield, 2004. xiv, 69, 74

[MP07] MCCANN J., POLLARD N.: Responsive characters from motion fragments. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 6. 8

[MRS08] MAUS H.-M., RUMMEL J., SEYFARTH A.: Stable upright walking and running using a simple pendulum based control scheme. In *Advances in Mobile Robotics: Proc. 11th Int. Conf. Climbing and Walking Robots. Coimbra, Portugal: World Scientific* (2008), pp. 623–629. 5, 44, 46, 47

[MS84a] MIURA H., SHIMOYAMA I.: Dynamic walk of a biped. *The International Journal of Robotics Research 3*, 2 (1984), 60–74. xi, 21

[MS84b] MIURA H., SHIMOYAMA I.: Dynamic walk of a biped. *The International Journal of Robotics Research 3*, 2 (jun 1984), 60–74. xv, 23, 90

[MZS09] MACCHIETTO A., ZORDAN V., SHELTON C. R.: Momentum control for balance. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 80:1–80:8. 10

[NBS06] NEVILLE N., BUEHLER M., SHARF I.: A bipedal running robot with one actuator per leg. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (2006), IEEE, pp. 848–853. 45

[Nov98] NOVACHECK T. F.: The biomechanics of running. *Gait & posture 7*, 1 (1998), 77–95. 82

[OM01] OSHITA M., MAKINOUCHI A.: A dynamic motion control technique for human-like articulated figures. In *Computer Graphics Forum* (2001), vol. 20:3, pp. 192–202. 109

[PCCDP01] PRATT J., CHEW C. TORRES A., DILWORTH P., PRATT G.: Virtual model control: An intuitive approach for bipedal locomotion. *Intl J. Robotics Research 20*, 2 (feb 2001), 129–43. 45, 53

[PCD06] PRATT J., CARFF J., DRAKUNOV S.: Capture point: A step toward humanoid push recovery. *Proc. Humanoid Robots* (2006), 200–207. xv, 14, 15, 23, 59, 90

[Per95] Perlin K.: Real time responsive animation with personality. *Visualization and Computer Graphics, IEEE Transactions on 1*, 1 (1995), 5–15. 57

[PH04] Popovic M., Hofmann A.: Angular momentum regulation during human walking: biomechanics and control. *Robotics and Automation,*, April (2004), 2405–2411. 23

[PH05] Popovic M. B., Herr H.: Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *Int. J. Robot. Res 24* (2005), 2005. 14

[PK98] Park J. H., Kim K. D.: Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on* (1998), vol. 4, IEEE, pp. 3528–3533. xi, 21

[PT06] Pratt J., Tedrake R.: Velocity-based stability margins for fast bipedal walking. In *Fast Motions in Biomechanics and Robotics* (2006), Diehl M., Mombaur K., (Eds.), vol. 340 of *Lecture Notes in Control and Information Sciences*, Springer Berlin Heidelberg, pp. 299–324. xv, 23, 26, 59, 90, 93, 95

[PW99] Popović Z., Witkin A.: Physically based motion transformation. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 11–20. 15

[R*86] Raibert M. H., et al.: *Legged robots that balance*, vol. 3. MIT press Cambridge, MA, 1986. 14, 15, 16, 45

[RBS10] Rummel J., Blum Y., Seyfarth A.: Robust and efficient walking with spring-like legs. *Bioinspiration and Biomimetics 5*, 4 (2010), 046004. 24

[RCB98] Rose C., Cohen M., Bodenheimer B.: Verbs and adverbs: multidimensional motion interpolation. *Computer Graphics and Applications, IEEE 18*, 5 (1998), 32–40. 8

[RH91] Raibert M. H., Hodgins J. K.: Animation of dynamic legged locomotion. *ACM SIGGRAPH Computer Graphics 25*, 4 (1991), 349–358. 15, 22, 45

[RS10] Rummel J., Seyfarth A.: Passive stabilization of the trunk in walking. *In Proceedings of Int. Conf. on Simulation, Modeling and Programming for Aut. Rob. Darmstadt, Germany, Nov 15-16* (Nov 2010), 127–136. 44, 46

[SA10] Stephens B., Atkeson C.: Push recovery by stepping for humanoid robots with force controlled joints. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on* (2010), pp. 52–59. 14

[SC92a] Stewart A. J., Cremer J. F.: Animation of 3d human locomotion: Climbing stairs and descending stairs. In *In Eurographics Workshop on Animation and Simulation* (1992). 10, 14, 22

[SC92b] Stewart A. J., Cremer J. F.: Beyond keyframing: an algorithmic approach to animation. In *Proceedings of the conference on Graphics interface '92* (San Francisco, CA, USA, 1992), Morgan Kaufmann Publishers Inc., pp. 273–281. 10, 14, 22

[SCCH09] SHIRATORI T., COLEY B., CHAM R., HODGINS J. K.: Simulating balance recovery responses to trips based on biomechanical principles. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), SCA '09, ACM, pp. 37–46. 13, 15, 23, 58

[SGL93] SHIH C. L., GRUVER W. A., LEE T. T.: Inverse kinematics and inverse dynamics for control of a biped walking machine. *Journal of Robotic Systems 10*, 4 (1993), 531–555. 15, 59

[SH08] SHIRATORI T., HODGINS J. K.: Accelerometer-based user interfaces for the control of a physically simulated character. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 123. 12

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 514–521. 15

[SJZ90] SIAS JR F. R., ZHENG Y. F.: How many degrees-of-freedom does a biped need? In *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on* (1990), IEEE, pp. 297–302. 62, 82

[SKL07] SOK K. W., KIM M., LEE J.: Simulating biped behaviors from human motion data. In *ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 13, 22, 41, 45, 100

[SKRF11] SINGH S., KAPADIA M., REINMAN G., FALOUTSOS P.: Footstep navigation for dynamic crowds. *Computer Animation and Virtual Worlds 22*, 2-3 (2011), 151–158. 58

[SLSG01] SHIN H. J., LEE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Trans. Graph. 20*, 2 (apr 2001), 67–94. 8

[Ste07] STEPHENS B.: Humanoid push recovery. *7th IEEE-RAS International Conference on Humanoid Robots* (nov 2007), 589–595. 14, 15, 59

[Sug08] SUGIHARA T.: Simulated regulator to synthesize ZMP manipulation and foot location for autonomous control of biped robots. *Robotics and Automation, 2008. ICRA 2008. IEEE* (2008). 23

[SvdP05] SHARON D., VAN DE PANNE M.: Synthesis of controllers for stylized planar bipedal walking. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), IEEE, pp. 2387–2392. 22

[TH00] TANCO L. M., HILTON A.: Realistic synthesis of novel human movements from a database of motion capture examples. In *Human Motion, 2000. Proceedings. Workshop on* (2000), IEEE, pp. 137–142. 8

[TK05] TAK S., KO H.-S.: A physically-based motion retargeting filter. *ACM Trans. Graph. 24*, 1 (Jan. 2005), 98–117. 8

[TLC*10] TSAI Y.-Y., LIN W.-C., CHENG K. B., LEE J., LEE T.-Y.: Real-time physics-based 3d biped character animation using an inverted pendulum model. In *IEEE Transactions on Visualization and Computer Graphics* (2010), vol. 16:2, pp. 325–337. xi, xv, 16, 17, 21, 23, 24, 34, 35, 41, 45, 53, 57, 59, 81, 90, 95, 100, 105

[TPZZ06] TANG B., PAN Z., ZHENG L., ZHANG M.: Interactive generation of falling motions. *Computer Animation and Virtual Worlds 17*, 3-4 (2006), 271–279. 58

[TSK00] TAK S., SONG O.-Y., KO H.-S.: Motion balance filtering. *Computer Graphics Forum 19*, 3 (2000), 437–446. 8

[vBPE10] VAN BASTEN B. J. H., PEETERS P. W. A. M., EGGES A.: The step space: example-based footprint-driven motion synthesis. *Comput. Animat. Virtual Worlds 21*, 3/4 (may 2010), 433–441. 9

[vBSE11] VAN BASTEN B. J. H., STÜVEL S. A., EGGES A.: A hybrid interpolation scheme for footprint-driven walking synthesis. In *Proceedings of Graphics Interface 2011* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2011), GI '11, Canadian Human-Computer Communications Society, pp. 9–16. 9

[vdP96] VAN DE PANNE M.: Parameterized gait synthesis. *Computer Graphics and Applications, IEEE 16*, 2 (1996), 40–49. 22, 109

[VS72] VUKOBRATOVIĆ M., STEPANENKO J.: On the stability of anthropomorphic systems. *Mathematical Biosciences 15*, 1–2 (oct 1972), 1–37. xi, xv, 21, 23, 59, 90, 95

[WB03] WANG J., BODENHEIMER B.: An evaluation of a cost metric for selecting transitions between motion segments. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 232–238. 8

[WB08] WANG J., BODENHEIMER B.: Synthesis and evaluation of linear motion transitions. *ACM Trans. Graph. 27*, 1 (mar 2008), 1:1–1:15. 8

[WFH08] WANG J. M., FLEET D. J., HERTZMANN A.: Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Recognition and Machine Intelligence* (2008), 283–298. 11

[WFH09] WANG J. M., FLEET D. J., HERTZMANN A.: Optimizing walking controllers. In *ACM SIGGRAPH Asia 2009 papers* (New York, NY, USA, 2009), SIGGRAPH Asia '09, ACM, pp. 168:1–168:8. 82

[WFH10] WANG J. M., FLEET D. J., HERTZMANN A.: Optimizing walking controllers for uncertain inputs and environments. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 73:1–73:8. 16, 22, 42

[WH96] WOOTEN W. L., HODGINS J. K.: Animation of human diving. *Computer Graphics Forum 15*, 1 (1996), 3–13. 13, 15, 22

[WH97] WILEY D., HAHN J.: Interpolation synthesis for articulated figure motion. In *Virtual Reality Annual International Symposium, 1997., IEEE 1997* (1997), pp. 156–160. 8, 9

[WK88] WITKIN A., KASS M.: Spacetime constraints. In *ACM Siggraph Computer Graphics* (1988), vol. 22, ACM, pp. 159–168. 10

[WMZ08] Wu C.-C., Medina J., Zordan V. B.: Simple steps for simply stepping. In *Advances in Visual Computing*. Springer, 2008, pp. 97–106. 58

[WP10] Wu J.-c., Popović Z.: Terrain-adaptive bipedal locomotion control. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 72:1–72:10. 14, 16, 17

[YCBvdP08] Yin K., Coros S., Beaudoin P., van de Panne M.: Continuation methods for adapting simulated skills. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 81:1–81:7. 22, 41

[YL10] Ye Y., Liu C. K.: Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph. 29*, 4 (July 2010), 74:1–74:9. 14

[YLvdP07] Yin K., Loken K., van de Panne M.: Simbicon: simple biped locomotion control. In *ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. xi, xv, xvi, 13, 16, 17, 21, 22, 23, 24, 41, 53, 59, 81, 90, 97

[YPvdP05] Yin K., Pai D. K., van de Panne M.: Data-driven interactive balancing behaviors. *Pacific Graphics 05* (2005), 118–121. 14

[YvdP06] Yin K., van de Panne M.: Omnidirectional humanoid balance control: Multiple strategies for reacting to a push. *University of British Columbia, Computer Science Technical Report* (2006). 13

[YZX08] Yin C., Zhu J., Xu H.: Walking gait planning and stability control. *Power Engineering* (2008). 62

[ZH02] Zordan V. B., Hodgins J. K.: Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), SCA '02, ACM, pp. 89–96. 8, 13, 15, 45, 109

[ZMCF05] Zordan V. B., Majkowska A., Chiu B., Fast M.: Dynamic response for motion capture animation. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 697–701. 8, 9

[ZMM*07] Zordan V., Macchietto A., Medina J., Soriano M., Wu C.-C.: Interactive dynamic response for games. In *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games* (2007), ACM, pp. 9–14. 9

Declaration

I herewith declare that I have produced this thesis without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This thesis has not previously been presented in identical or similar form to any other English or foreign examination board.