**Title**
Goal-Directed Biped Stepping and Push Recovery with Momentum Control



**Author**
Wu, Chun-Chih

**Publication Date**
2011

UNIVERSITY OF CALIFORNIA
RIVERSIDE

# Goal-Directed Biped Stepping and Push Recovery with Momentum Control

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Chun-Chih Wu

March 2011

Dissertation Committee:

Dr. Victor Zordan, Chairperson
Dr. Christian Shelton
Dr. Eamonn Keogh

The Dissertation of Chun-Chih Wu is approved:

Committee Chairperson

University of California, Riverside

## Acknowledgments

I would like to thank my advisor, Victor Zordan, for his advice and support. I would also like to thank everyone in Riverside Graphics Lab, especially Adriano Macchietto who let me build my system with his software and showed me how amazing and passionate a graduate student researcher can be. Without their help and motivation, this dissertation may have never happened. Additionally, I would like to thank my committee members, especially Christian Shelton who wrote me a recommendation letter along with Victor's that helped me receive UCR's Dissertation-Year Fellowship. I would also like to thank my department's former and current graduate student affair officers Terri Phonharath and Amy Ricks as well as staff in International Education Center and Graduate Division who gave me a lot of help at UCR. Finally, I would like to thank my family, friends here and in Taiwan for their encouragement and always being there for me.

This dissertation is composed of three publications: Simple Steps for Simply Stepping [91], Goal-directed Stepping with Momentum Control [92] and Anticipation from Example [102]. I would like to thank my co-authors: Victor Zordan, Adriano Macchietto, Jose Medina, Marc Soriano, Ron Metoyer and Robert Rose.

ABSTRACT OF THE DISSERTATION

Goal-Directed Biped Stepping and Push Recovery with Momentum Control

by

Chun-Chih Wu

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, March 2011
Dr. Victor Zordan, Chairperson

Stepping is a fundamental skill involved in common bipedal activities such as walking, foot repositioning and step recovery. Generating these stepping activities requires characters that are controllable and responsive. This dissertation describes a goal-directed controller and a momentum supervisor for characters that perform both believable and robust steps under a variety of conditions. The desired step is controlled by generic task goals, namely step position and step duration, which allow characters to step in arbitrary directions with various speeds. These high-level goals guide desired changes of a character's center of mass and swing foot over the duration of the step. To produce realistic and flexible steps, the desired time-varying values for the center of mass and the swing foot are derived from parametric curve generators which are built on empirical evidence extracted from motion capture data of stepping. Controlling these two values along with regulation of angular momentum in vertical axis produces characters with coordinated full-body movements including natural arm swings during stepping. The system can guide a character with purposeful, directable steps to precisely follow user-specified foot placements and to carefully control the character's position and orientation. Moreover, the same system can be used to create protective steps to maintain the character's balance in response to a perturbation. A novel supervisory routine automatically chooses when and where to step with a straightforward goal: removing all linear and angular momenta induced by a push. In contrast to previous methods for push recovery using the inverted pendulum, the proposed momentum supervisor introduces a nice clean formulation to determine when and where to step and provides better prediction of a character's stability under perturbations by considering both linear and angular momenta of the character. In addition to responding to a perturbation, this dissertation also presents an approach for characters that anticipate impending perturbations with examples taken from human motion capture data. I focus on the motion interpolation sysnthesis technique which allows a character to anticipate by blocking or dodging a threat coming from a variety of directions and targeting any part of the body.

# Contents

# List of Figures

# Chapter 1

# Introduction

3D character animation has applications in a wide variety of fields such as entertainment, education and communication. While currently being used extensively in animated films and electronic games, character animations are also becoming prevalent in military trainings, medical simulations, news reports as well as social skills learning just to name a few. However, as modeling and rendering of computer generated characters become more and more realistic, constructing believable motion remains a challenging task. The challenge can be attributed to two major reasons. First, we are especially good at discerning the nuances between natural and unnatural motion since we see each other's movements in our daily lives. Second, despite advances in current technology, our knowledge is still limited about how humans control their bodies which consist of hundreds of bones and muscles to move in a coordinated manner and perform various activities such as balance, response and locomotion under different circumstances. The interest of understanding and reproducing natural, efficient human movements has been the motivation for several disparate disciplines such as biomechanics, robotics and computer graphics.

In traditional computer graphics, expressive character movements are created by experienced animators through their manually specified key frames. The rest of frames can be automatically generated by computers with interpolation techniques. While keyframing provides animators with full control of the final motion, the technique itself is an off-line process and relies heavily on artistic skills and experiences to fine-tune a character's movements. To date, animation studios such as Pixar and DreamWorks still rely on this labor intensive approach to produce their movie blockbusters.

The advent of motion capture provides an alternative solution to creating realistic movements. The motions of human actors are recorded at a high sampling rate (120 to 240 Hz) and are mapped onto digital characters. Motion capture has been widely used in current video games and in increasing number of live-action films such as Avatar due to its strength of preserving every subtle nuance of an individual's motion style. Because of the ease of creating motions, aforementioned animation studios are also employing motion capture in their pre-visualization stage to help film directors experiment camera movement and placement. However, when the recorded motion data does not satisfy requirements of an application, it is often easier to capture a new motion rather than reusing the existent data. The reason is because motion capture only records kinematic aspects of movement, i.e. position and velocity information. Dynamic aspects of movement are not directly available in the data and need to be approximated, for example force and torque exerted by and acting on a character's body which contains mass and inertia information. Manipulating existing motion data without careful consideration of the required constraints often results in physically implausible motion. As a result, motion capture-driven animations resort to a large collection of examples for realism and other requirements such as interactivity and flexibility. However, this approach still fails to generalize to situations where the desired motions are too dissimilar or not belonging to the same category of the pre-recorded database. Real-time applications with unpredictable environments and user inputs suffer most dramatically from this limitation.

Physically based simulation, on the other hand, holds promise for animating realistic controllable characters in interactive environments. Realistic motion is derived from two aspects: physical plausibility from the equations of motion and coordinated body movement from controller which dictates naturalness of motion. Through simulation, the interplay between characters, objects, and their surroundings is generated automatically by constraining the motion to follow the dynamic equations of motion. However, developing robust and flexible controllers for simulated characters that convey human-like motion qualities remains an open problem. Due to difficulties in creating such controllers, physics-based characters had only been used sparingly, mostly for creating the so-called ragdoll effect which employs little or zero control to simulate characters' unconscious or death sequences. Until recently, with the advance of computer hardware and research in control strategies, physically simulated characters are finally seen in commercial softwares such as NaturalMotion's endorphin and Euphoria [55].

## 1.1 Motivation

An ideal animation system should provide characters that preserve their styles and motion qualities while seamlessly interacting with the environments and responding to users' commands. In the last fifteen years, there has been a significant amount of research on motion capture, dynamical simulation and combination of both of them to try to solve the above problem. While much progress has been made, the problem has not been fully solved and is still an active research area in computer graphics.

In this dissertation, I chose stepping behavior as a testbed and present two control algorithms, one using a kinematic model and the other using a dynamic model, to address the problem of creating controllable, believable, responsive and robust characters. I elaborate on motivation for the selected behavior and the proposed algorithms next.

### 1.1.1 Stepping Activities

Stepping is a fundamental skill involved in common activities such as walking and full-body maneuvering such as foot repositioning and step recovery. As such, stepping is a critical behavior for applications involving virtual human avatars. For example, characters in electronic games must be able to change their stance and facing direction in response to a player's inputs. The ability of being able to precisely control foot placement is also important for characters to navigate in a constrained virtual environment and to avoid collisions. Furthermore, characters must also be able to take protective steps to regain their balance in response to external perturbations from players and the environments. Besides computer scientists, researchers from biomechanics and robotics have also shown great interest in stepping activities. By investigating such fundamental behavior, we hope to shed light on how humans control their bodies and perform more complex motion such as step recovery and walking.

In biomechanics, there has been in-depth investigation performed on both voluntary and compensatory steps for the purposes of rehabilitation and prevention of falling. For example, it is a necessary and a distinguishing postural adjustment for healthy humans to execute a self-initiated step by shifting body's weight toward the stance foot before lifting the swing foot. Studies have found that patients with Parkinson's disease suffer from deficits of such anticipatory postural adjustment. The stepping algorithms in this dissertation take into account the body's weight shift by controlling a character's center of mass (CM) to follow desired path based on insight from biomechanical observations of CM trajectories during gait and stepping. On the other hand, compensatory stepping, has been identified as a natural response for both young and older adults when they receive external perturbations. Biomechanists investigated age and gender differences of a single-step recovery with various perturbation techniques including waist pulls, lean-and-releases and platform movements. In this dissertation, both liberal and conservative step recoveries can be produced by the proposed momentum supervisor through adjusting its damping variables to decide how quickly momenta induced by a push should be removed.

In robotics, humanoid robots have the unique potential to cooperate with humans and operate in the environments made by humans. However, it is inevitable that these robots will encounter disturbances from humans and the uncertain environments. The occurrence of a fall is a devastating event for a humanoid robot because it can injure people and cause damage to the expensive robot as well as objects in the surroundings. As a result, push recovery by stepping is also an active research topic in robotics community in order for humanoid robots to safely work in these complex scenarios. In this dissertation I present a straightforward formulation to determine when and where to step under perturbations by considering both linear and angular momenta of a character. Considering both linear and angular momenta improves a character's robustness under perturbations in comparison to inverted pendulum which only considers a character's linear momentum.

### 1.1.2 Control Algorithms

I propose two control algorithms to address problems with animating biped stepping behaviors. The first algorithm employs a kinematic model which relies on a procedural approach in conjunction with a motion capture database of stepping to create characters that perform controllable and believable steps. Procedural animations refer to motions which are generated through a set of user specified rules or computational procedures. The advantage of procedural animation is that new motion can be easily produced by adjusting control parameters which come with the procedures. The control parameters of the first algorithm are generic task goals, namely step position and step duration, which allow human characters to step in a range of directions with different speeds. These two parameters provide high-level, goal-directed control and thus avoid adjustments in a character's individual joint level which might lead to uncoordinated body movements. The algorithm achieves the desired stepping goals by parameterizing and modifying a character's CM and swing foot. As mentioned in the first section, modifying existent motion data requires careful consideration of the physical constraints. CM and swing foot were chosen as the control parameters because they encapsulate many physical attributes of a step. In contrast to pure data-driven approach, procedural approach is more flexible because of its controllability. However, motion quality produced by procedural approach alone still lacks subtle details found in motion capture data. Therefore, the first algorithm combines procedural approach with motion capture data of stepping in order to get the best of both approaches: controllability and believability.

The second algorithm in this dissertation employs a physical model which relies on full-body momentum control and a momentum-based supervisor to create responsive robust characters that are able to follow user specified footprints and react to external perturbations by stepping. The algorithm generates realistic whole body movements by controlling three high level objectives: linear momentum, angular momentum and swing foot. Control of a character's linear and angular momenta creates coordinated full-body movements including natural arm swings. The supervisor improves a character's robustness under perturbations by assessing the character's momenta conditions to automatically decide when and where to step. Because only controlling high-level objectives, the algorithm uses no heuristics specific to humanoid characters and can be applied to any biped characters. Because no motion capture data is required, the characters are able to step in arbitrary directions both voluntarily and as a result of perturbation. Believable motion is still retained because desired trajectories of the character's CM and swing foot are guided by the parameterized model developed in the first algorithm which are empirical evidence extracted from motion capture data.

## 1.2 Contributions

As mentioned earlier, the goal of this dissertation is to animate biped characters that are controllable, responsive while able to perform robust, believable motions. I address this goal by solving the stepping problem and presenting two control algorithms to answer the core issues of stepping: how to step, when to step and where to step in a principled manner. I demonstrate the effectiveness of the proposed algorithms through exploration of three classes of stepping behaviors including directed step for navigation, reactive step for regaining balance as well as continuous steps for walking and multiple-step recovery. In summary, the primary contributions of this research include:

- a goal-directed controller which provides a character with flexible step position and duration.
- parametric models for CM and swing foot which are empirical evidence extracted from motion capture data and have shown to be critical for maintaining a character's dynamic balance while achieving the desired stepping goals.
- a momentum supervisor which improves prediction of a character's stability under

perturbations by considering both linear and angular momentum and provides a formulation to automatically decide when and where to step with a straightforward goal: removing all linear and angular momenta induced by a push.

## 1.3 Outline

This dissertation continues in the next chapter with a survey of related work in the areas of motion capture editing and control strategies for simulated characters as well as relevant background in robotics and biomechanics. In Chapter 3, I describe the first approach to creating characters that perform controllable and believable steps using a procedural approach in combination with motion capture data. In Chapter 4, I introduce a physics-based approach for simulating characters that perform directed steps for navigation and protective steps for maintaining balance in response to external perturbations. In Chapter 5, I present a method for characters generating anticipation to unexpected interactions with example taken from human motion capture data. Lastly, conclusions are given in Chapter 6.

# Chapter 2

# Related Work

Creating controllable responsive characters is an open problem in computer graphics. This chapter presents an overview of related work on motion editing techniques which add controllability to data-driven motion capture animations and control strategies which create responsive characters using physical simulation as well as hybrid approaches which combine the advantages of motion capture and dynamic simulation. Furthermore, because the proposed algorithms in this dissertation are inspired by domain knowledge in robotics and biomechanics, I summarize related work from each of these areas including simplified models for balance and locomotion as well as angular momentum studies of human activities.

## 2.1 Editing Techniques for Motion Capture Data

Motion capture data produces realistic animations but the data itself does not provide controllability and is best for playback. When used in interactive applications, motion capture data must be augmented with high-level control mechanisms in oder to accomplish new tasks and to smoothly create transitions between different motions.

There has been a great amount of work in this area and the approaches can be categorized into two groups: motion parameterization through blending [88, 67, 59, 39, 44, 54] and motion rearrangement [4, 40, 45, 47, 5, 22].

A desired motion or behavior can be parameterized and synthesized by blending labeled motion segments. For example, Rose et al. [67] proposed a framework for creating walking motions with various emotions such as happy, sad, angry, tired and etc. They recorded many different walking sequences and manually labeled them off-line. At runtime, they create walking motion with smooth transitions from one emotion to another using scattered data interpolation with radial basis functions. Later, Kwon and Shin [44] enhanced the approach by introducing an automatic motion labeling scheme. They presented an approach to decompose unlabeled motion sequence into segments based on CM trajectories and classified them into groups with identical footstep pattern. Their on-line locomotion system can synthesize parameterized walking motions with different speeds, turning angles and accelerations. Unlike their methods which focused on locomotion, the control algorithms in this dissertation emphasize the importance of precise control of foot placement through stepping. Other researchers have also addressed similar goal lately [57, 18]. However, I achieve parameterized step positions through a procedural approach which directly modifies a character's CM and swing foot rather than relying on motion blending techniques which require a large number of stepping examples. To be clear, stepping examples in my first algorithm are used to enhance a character's upper body movements because the procedural approach only modifies the character's lower body through CM and swing foot control. Stepping examples are not used in my second algorithm because controlling a character's aggregate linear and angular momenta creates full-body movements.

Motion rearrangement synthesizes new motions to follow user specified paths or constraints by searching for similar postures or motion segments that are easy to produce seamless transitions. Most of the approaches use variants of 'motion graph' data structure to represent connectivity between similar poses in a given motion stream. Since the introduction of motion graph, the concept of generating transitions has become increasingly popular. Some contributions highlight ideal lengths for transitions and similarity metrics for selecting similar frames [84, 85, 86]. Some use multiple frames to generate high quality transitions that have split-second reactivity [32]. Transitions are typically performed by blending character's joint angles as well as root position and orientation over time. Cleaning up transitions is usually done with algorithms for footplant detection and enforcement [71, 41, 31] with inverse kinematics. Tools for correcting balance ensure that motion transitions remain physically plausible by controlling the CM or zero moment point [8, 9, 81]. The algorithms presented in this dissertation can also be used to create motion transitions. Unlike motion graph which requires similar postures to create transitions, the proposed algorithms here do not have this limitation because characters can take directed steps to align foot placements before and after the transition to avoid foot sliding. Also, the proposed algorithms take into account a character's CM and swing foot during transition to maintain the physical plausibility.

## 2.2 Control Strategies for Simulated Characters

Despite advances in current technology, control of physically simulated character is still challenging because how humans control their muscles to generate coordinated body movements is still not well understood. Early work on simulated characters relied on hand-crafted controllers to generate specific behaviors such as running, handspring vaulting, diving and etc [28, 90]. The other common approach to simulate physics-based characters is to use joint-based tracking controller to follow a reference motion [101, 46]. Motion capture data provides desired joint angle information for all of the degrees of freedom. However, naively tracking the reference motion without considering balance usually causes the character to tip or fall over. There are two major problems regarding this naive tracking approach. First, there are discrepancies between physical models of the simulated character and the human actor whose motion is recorded. Sok et al. [74] used optimization to correct motion in order to remove these discrepancies. Second, biped characters are underactuated because their feet are not fixed on the ground. That is, they don't have direct control over their global position and orientation because the number of degrees of freedom of a character is larger than the number of its actuators (joints in this case). This is why the problem is called underactuation. Only through external forces such as ground reaction force can the characters regain control over the unactuated degrees of freedom: global position and orientation. If we do not pay attention to these unactuated degrees of freedom, any tracking error will cause the character to deviate from the reference motion and eventually to lose balance. In this dissertation, I control the aggregate linear and angular momenta of a character which in turn leads to control over those unactuated degrees of freedom.

Recently, several new motion control approaches used multiobjective optimization and optimal control theory to take advantage of the realism of data examples while employing simulation to create characters with controllable movements [2, 15, 16, 53, 48]. This dissertation uses a similar framework, most specifically [48]. However, of these efforts the proposed methods have focused on locomotion and standing, but none have focused on the problem of control for stepping. A distinction between these previous efforts and the research work in this dissertation is that a single fixed reference motion is not acceptable for stepping potentially in any direction at any time.

Other physics-based techniques have been proposed to generate protective steps [19, 43, 99, 97, 33]. Researchers at CMU graphics lab have focused on closely related biomechanical principles of trip recovery during walking [73]. Closest to this dissertation, Kudoh et al. [43] chose the desired foot placement using an inverted pendulum (IP) model with its parameters extracted from motion capture data. Also, Jain et al. [33] picked the desired step position such that the CM will lie in the center of the support polygon after stepping. The main distinction of this dissertation is that the choice of when and where to step is automatically computed by our supervisor based on assessment of momenta. We don't have to extract parameters for IP nor do we use heuristics to decide foot placements. Instead, our momentum supervisor relies on two straightforward parameters which control how fast the induced momenta should be dissipated to automatically decide when and where to step under perturbations.

Data-driven techniques for generating steps in response to unpredicted disturbances without physical simulation require the collection of a large database [6, 98, 38]. In this dissertation, I synthesize motion without a reference trajectory through the goal-directed stepping model. The stepping model is designed based on knowledge about the principle goals of the behavior. Controlling CM and swing foot have shown to encapsulates many physical attributes of a single step. Previous work of Stewart and Cremer [78, 79] has the same spirit of CM and end-effector planning. In fact, controlling high-level goals such as CM, end-effector and angular momentum has shown to be a very general approach which is able to generate realistic and flexible motions [17, 52, 93, 12, 94]. However, the control signals for these high-level goals in these SIGGRAPH 2010 papers are either manually specified or through expensive global optimization such as Covariance Matrix Adaptation (CMA) [25]. Instead, the control signals of these high-level goals in this dissertation are empirical evidence extracted from human motion capture data.

## 2.3 Simplified Models for Balance and Locomotion

Researchers in robotics have also proposed techniques for automatic generation of stepping motion for use in control of humanoid robots. Similar challenges in this area include choosing step location and maintaining balance [29, 75, 76, 69, 77]. Various numerical values have been introduced to define balance [60] and many simplified models for control of balance and locomotion have appeared. One simplifying model is to treat the dynamics of character as a linear IP [36] and to control the robot to perform stepping and walking [34] based on a point mass and massless leg. Another group of researchers introduced the concept of 'capture point' which is the step point which yields a single step to recover from a perturbation [64].

Several roboticists have extended the IP model to account for change in angular momentum due to disturbances. In particular, the angular momentum pendulum model [37] and the IP plus flywheel [64] are close models to the proposed momentum-based supervisor in this dissertation. One difference is that the IP focuses on single stance while the supervisor proposed here considers double stance for step recovery. Moreover, the IP plus flywheel uses a constant inertia as an approximation of the entire body but the proposed supervisor here does not make this assumption since human body can have different moment of inertia based on different postures. Lastly, the goal of step recovery in this dissertation is to place the foot at the proper position to remove all linear and angular momenta induced by a push. This strategy is supported by study of fall recovery in the biomechanics literature [50]. On the other hand, the idea of IP plus flywheel is that the angular momentum can be stored in the flywheel. The stored angular momentum can later be released to produce angular momentum when there is a difference between the actual step position and the capture point to enhance a character's robustness. If we follow the classification of the use of angular momentum by Zordan in [100], IP plus flywheel uses non-zero spin (NZS) strategy while the momentum supervisor proposed here uses zero-spin (ZS) strategy to regulate angular momentum.

## 2.4 Angular Momentum Control

Several researchers from biomechanics have studied the role of angular momentum in human activities such as walking, turning and maintaining balance [61, 62, 26, 20, 30] and found that angular momentum is carefully regulated in the above activities. Robotics researchers have also proposed ways to control angular momentum to improve robustness of their humanoid robots. Kajita et al. [35] introduced 'resolved momentum control' to simultaneous control robot's linear and angular momentum to create coordinated motions for activities such as kicking and walking. Goswami and Kallen [24] suggested to keep angular momentum change to be zero as a robust way to maintain biped robot's balance. Abdallah and Goswami [1] proposed a two-phase strategy to absorb disturbance by first preserving momentum and then returning to the upright posture.

Biomechanists have also investigated the underlying mechanics of arm swing during human walking. Herr and Popovic [26] found that the movement of arm swing cancels out the leg angular momentum in the vertical axis. Collins et al. [11] suggested that normal arm swing also reduces the vertical ground reaction moment acting on the support foot. Based on these findings, I show that by continuously damping out the angular momentum in a character's vertical axis leads to natural arm swings.

# Chapter 3

# Goal-Directed Stepping

In this chapter we introduce a general method to animate controlled stepping motion for use in combination of pre-recorded motion capture sequences. We use examples from a stepping motion database or from given motion segments to create an initial interpolation and then modify it to uphold characteristics of stepping. Stepping in our system is characterized by two simple models which idealize the movement of stepping foot and projected center of mass (CM) based on observations of examples in the database of stepping. Taking as basic features about the desired action, our system computes path and speed profiles from each model and adapts the initial interpolation to account for models' results. We show that our animation can be enriched by choosing a close example from the step motion database. Alternatively, we can synthesize stepping to create transitions between two given motion segments. We demonstrate that we are able to synthesize precise, realistic stepping for a number of scenarios.

## 3.1 Introduction

Motion capture blending, editing and reordering have become the standard suite of motion synthesis tools used for animating video games and increasing number of feature films. A standard practice associated with such applications is the generation of 'transitions' which combine two motion sequences together to create one longer sequence. Transition also plays an important role in improving a character's responsiveness to a player's inputs such as changing the character's walking direction or changing its behavior. However, the quality of the resulting motion depends on the choice of method(s) used to create the transition. Since motion capture-driven animations do not explicitly model physics, tell-tale artifacts of a poor transition generally include unnatural foot sliding and physically implausible motion. Groups of researchers have addressed these issues by explicitly removing the so-called foot skate [41, 31] and by enforcing various physical characteristics during motion transition [68, 72].

In video games especially, foot skate often appears when a character is in transition from one standing configuration to another. Unless the feet are perfectly aligned, naive interpolation techniques will induce unnatural foot sliding as the character goes from the beginning to the ending stance. A difficult problem in this scenario is to create a transition which accounts for the differences in the placements of the feet while also taking into account the movement of the body in a realistic manner. A human actor would tackle similar scenarios by shifting the weight of the body and taking steps to re-position the feet. We propose that a similar mechanism for stepping is necessary to generate a plausible transition in character animation. Based on this insight, a new problem arises during the special conditions of transitions where a character begins and ends in double stance.

In this chapter we introduce a general method to synthesize stepping actions for humanoid characters. While the technique is showcased in conjunction with a motion database of step examples to enrich the final motion, the power of the approach comes from the control models which drive the character using idealized, parametric trajectories for the swing foot and projected CM.

We show that these two trajectories can be simple mathematical functions built empirically from observations of example stepping movements and parameterized to be controlled by key features of the desired action such as step position and step duration. The result is a stepping system that allows a character to create transition from one double-stance pose to another automatically by stepping. The simplicity of the approach lends itself to being adopted easily and immediately by game developers and technical animators alike. To show the generalness of the results, we demonstrated example stepping animations for a variety of scenarios.

## 3.2 Stepping Algorithm

The algorithm we employed has two particular components. First, we control the swing foot, both its path and its speed along that path. And second, we control the CM, again both position and velocity. We choose to control the CM in order to create the visible weight shift that corresponds to stepping actions in humans. We demonstrate that these two factors alone encapsulate many of physical attributes of a single step. While we include motion examples of stepping to enrich the final motion, the main hypothesis is that by moving the stepping foot and CM realistically we can generate believable stepping simply. In the next chapter, we further demonstrate that these two components really play an important role in maintaining a character's dynamic balance during stepping and walking.

Our technique incorporates these two components into an animation transition system using optimization. First, we create a starting blend by naively interpolating the character's current motion with an example extracted from the stepping database. We use this sequence as the initial paths for the foot and CM. Second, we employed a per-frame based optimizer which takes a frame from starting blend as input and produces a modified posture that enforces the desired swing foot and CM trajectories. The choice of timing and position for the swing foot is provided by user as input to the system.

We break the description of the algorithm into two phases, preprocessing and step generation. In the preprocessing stage, we determine the necessary inputs to the stepping algorithm, specifically:

1. Input (from user) the final swing foot position and step duration

2. Select a step example in database which has the closest swing foot position to the final foot position in Step 1

3. Adjust the ending pose from the selected example using inverse kinematics (IK) to precisely place the foot at the final position

4. Extract the CM ending place from the (adjusted) end pose

Steps 3 and 4 are used solely to determine the final position of the CM based on the motion sequence selected. Alternatively, we can force the system to generate transition to a specific motion sequence. In this case, Steps 2 and 3 can be skipped.

Once we have the required parameters, the stepping algorithm follows a straightforward sequence:

1. Compute the stepping foot path, $P_f$, and speed profile, $V_f$

2. Compute the CM path, $P_c$ and speed profile, $V_c$
3. Blend to selected example with support foot as root
4. Modify the blend with optimizer to meet CM/foot trajectories

The starting blend from Step 3 is treated as the input to the optimizer. To keep the stance foot from moving during transition, the starting blend is performed by treating the stance foot as the fixed root of the branching chain for the entire body. All other parts of the body move accordingly by smoothly interpolating the included joint angles. More details about each step are described in the following sections.

## 3.3 Stepping Foot Control

To define the swing foot motion appropriate for the desired step/transition, we determine the foot's path and its speed along that path. We assume that the path and speed profile are related by the distance covered from start to finish. That is, the total path displacement must equal the integral of the function chosen for the speed. We also assume that distance covered is monotonically increasing along the path. We follow a similar set of definitions and assumptions for CM control.

**We model stepping as if it is a point-to-point reach.** Upon inspection of our database of examples, we found remarkable uniformity - nearly-linear, point-to-point paths for each stepping foot. There has been in-depth investigation performed on hand point-to-point movement for reaching tasks [3, 21]) and, in this body of work, it is commonly accepted that the hand traverses an approximately straight-line path with a bell-shaped speed profile. For our foot model, we adopt a similar estimate for the foot trajectory, $P_f$, by forcing the foot to traverse the line segment formed by its starting

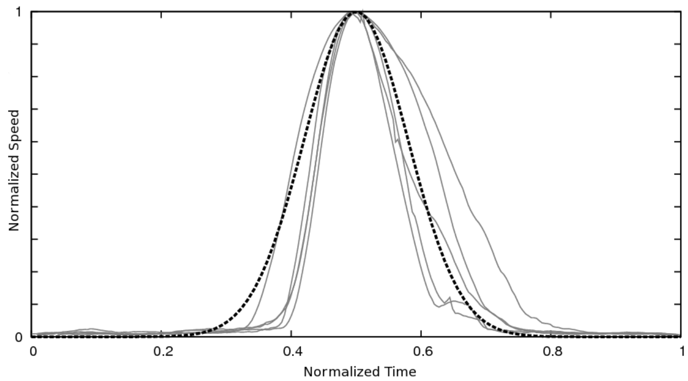


Figure 3.1: Foot speed, $V_f$. The black, dashed curve is the idealized normal speed fit using a Gaussian centered at 0.5. The rest are normalized sample profiles taken from various examples in our step database.

and ending position while using a normal Gaussian to serve as the bell-shaped speed curve:

$$V_f = ae^{\frac{-(x-0.5)^2}{2w^2}} \tag{3.1}$$

which define the speed along $P_f$.

This idealized speed curve is plotted in comparison to several speed profiles taken from our database in Figure 3.1. When the recorded curves are normalized in both time and amplitude, they show remarkable similarity, *independent of the stepping direction, the length of the step, or the duration.* We adjusted the shape (width) of our normalized Gaussian shown by manually setting the constant, $w$, to be 0.08. This value is used for all our results using stepping examples. For results without using examples, we found that adjusting the width is sometimes necessary to avoid abrupt movement of stepping foot.

To align the speed profile with the path, we must control the area under the curve to be equal to the distance from the start to the end of the footstep. We automatically tune the Gaussian by scaling amplitude, $a$, after integrating the curve for the normalized amplitude shown in Figure 3.1. Note this integration need only be done once and can then be scaled by $a$ to match the specific (known) distance covered in the to-be-synthesized motion.

## 3.4 Center of Mass Control

As with the foot, to control the CM we define a simplified model which captures the features of the human examples. Again the path and speed are both idealized from observations about the stepping examples recorded in our database. The CM path follows a parabola-like trajectory starting and ending at known points and moving toward and away from the support (pivot) foot. For $P_c$, we found empirically that a simple quadratic Bezier curve which uses the start and end of CM positions as well as pivot as control points reasonably maps out the path of the CM found in examples in our database. Comparisons appear in the results section.

For the speed, we observe consistent trends in the recorded motions that the CM velocity which can be broken down into three phases. 1) Push off. In this phase, *before* the foot is lifted, the CM begins to accelerate at a fairly constant rate toward the support foot (pivot). 2) Free fall. The second stage has the swing foot off the ground and we see a trajectory that mimics an unactuated inverted pendulum with the center of mass accelerating uniformly away from the support foot (now out of static balance.) 3) Landing. The swing foot reaches the ground and the motion induced in the second stage is dissipated with a slow changing acceleration toward the (original) support foot. What we infer from these observations is that three stages with constant acceleration reasonably describe the observed velocity profiles. Note, these phenomena are described

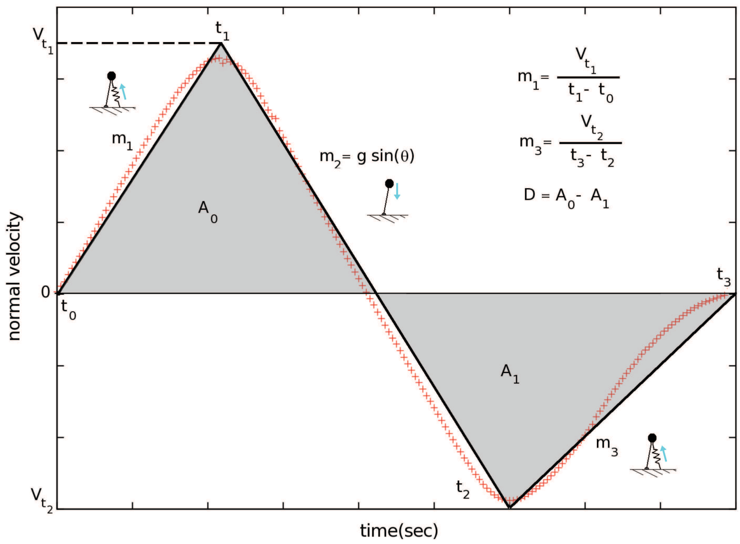


Figure 3.2: CM speed, $V_c$. Ideal and actual speed profiles in the direction of the support foot. That is, only motion toward and away from the pivot foot contribute to the data plotted, plus signs are derived from a real example. The timing information, $t_0 - t_3$, which delimit the stages (push off, free fall, and landing, respectively) can be estimated from the motion example by detecting when the stepping foot leaves and touches the ground again. Based on the pendulum model, $m_2$ is set to $gsin(\theta)$ where $\theta$ is the lean angle between vertical plane and support leg and $g$ is gravity. Areas, $A_0$ and $A_1$, link $D$, the displacement of the CM derived from $P_c$, to the slopes $m_1$ and $m_3$.

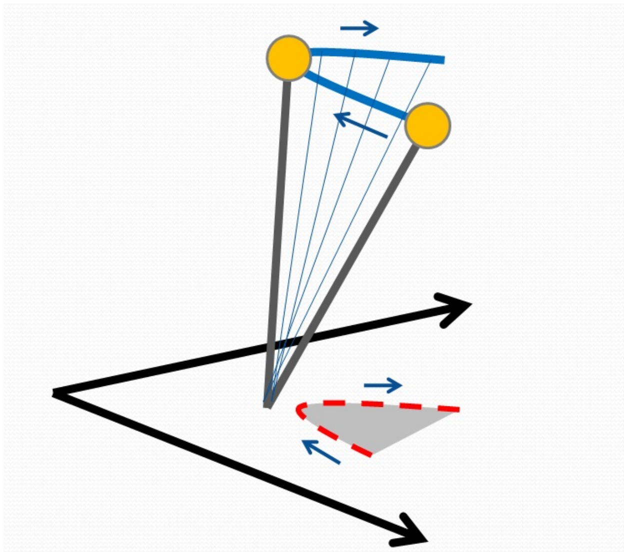

Figure 3.3: An idealized inverted pendulum which pivots about the support foot models path and speed observations reasonably.

in the coordinate frame oriented toward the support foot.

**An idealized inverted pendulum which pivots about the support foot models both our path and speed observations reasonably.** Based on an inverted pendulum model, our choice of path trajectory, $P_c$, is sensible since an idealized inverted pendulum moves its body on a ballistic, quadratic path. To fit the velocity characteristics for the CM, we could approximate the effects of the stepping leg as applying a uniform 'push-off' and 'landing' force before and after the step. (The *minimum jerk* theory for reaching partially supports this proposition [21].) An idealized constant force would yield a constant acceleration for push-off and landing. Constant acceleration is also reasonable for the middle phase when the body feels only the effects of gravity. Thus, for the model of our CM speed profile, $V_c$, we choose the piecewise linear function shown in Figure 3.2. We derive the terms of the velocity segments shown from known (or approximate) values for timing, $t_0 - t_3$, and the CM displacement, $D$, which is extracted from the Bezier curve, $P_c$.

## 3.5 Interpolation Synthesis for Stepping

To modify the starting blend given the stepping action parameters, we propose a simple, but effective interpolation synthesis technique. The problem here is to concatenate the motion the character is currently doing followed by the stepping motion in the example. To be successful, the transition should not introduce any unwanted artifacts. The most straightforward solution is to align the stepping motion globally to the character's current position and facing direction and then to blend the root position and orientation as well as the joint angles. However, this would introduce unnatural artifacts such as foot sliding if we do not specifically keep the feet in place. Instead of using this naive blend, we align the support foot of the before and after motion and use this foot as the fixed root for the stepping action. The system performs a simple blend using the new root (support foot) by interpolating over the errors for the root orientation and the joint angles of all other parts of the body across the transition sequence. In our implementation, our system interpolates by 'slerp'-ing quaternions, with a simple ease-in/ease-out (EIEO) time-based weighting across the transition. Note we do allow the support foot to rotate across the transition. This rotation is usually small if the facing direction of the two motions are closely aligned and acts to pivot the foot if there is a larger discrepancy. We show that such rotations appear natural looking in our results.

## 3.6 Optimization

Once we have the starting blend, we must modify it to uphold the stepping foot and CM trajectories determined for the transition. To accomplish this goal, we first apply IK to place the stepping foot at the desired position and then use an optimization which has an objective function of reaching the desired CM. The optimization works by moving the pelvis position in the horizontal plane and using an IK sub-routine [82] to generate adjustments for each leg which enforce the proper foot placement given the pelvis' new position. We found it necessary to constrain the height of the pelvis so as to not cause the change of the length of legs. To accomplish this goal, we limit the distance between the pelvis and each of the foot by lowering the pelvis automatically such that the leg is not stretched beyond its length. Therefore, the solver only needs to consider the placement of the pelvis in the horizontal plane.

Our implementation uses Numerical Recipes' BFGS [66] routine which employs a quasi-Newton gradient-based search to determine the pelvis location. The starting location is taken from the corresponding frame in the starting blend. The objective function is a weighted sum of the CM error in horizontal plane. The position of the desired CM is extracted from $P_c$ by moving along the Bezier curve until the normal displacement satisfies $V_c$. Likewise, the stepping foot location is set each frame to follow $P_f$ while also satisfying the rate determined from $V_f$.

## 3.7 Implementation and Results

Our final implementation includes additional details that need to be described. First, our stepping database was recorded by systematically creating a series of examples of normal steps taken in each of eight directions on the horizontal plane. Each example beginning and ending with neutral double-stance covers two steps which could be both for the same foot, i.e. taking a step and returning to neutral stance, or one step for each foot. This setting allows us to easily create continuous steps alternating with left and right feet. In total, we include twenty examples in our database. We use our system in two modes, starting from a known stance and transitioning to a modification of one of these examples or by combining two known clips, i.e. without using an example from the stepping database. When generating a step animation that includes an example, we select the example in the database which is closest to the desired step location. We note the running time of our system is fast enough to be used at interactive rates.

**Results.** We show two types of results in the accompanying video to illustrate the animations which are possible using this technique. First, we show examples which use our stepping database and include an animation of a series of steps with the left and right feet alternately to create a careful navigation (see Figure 3.4) We compare the

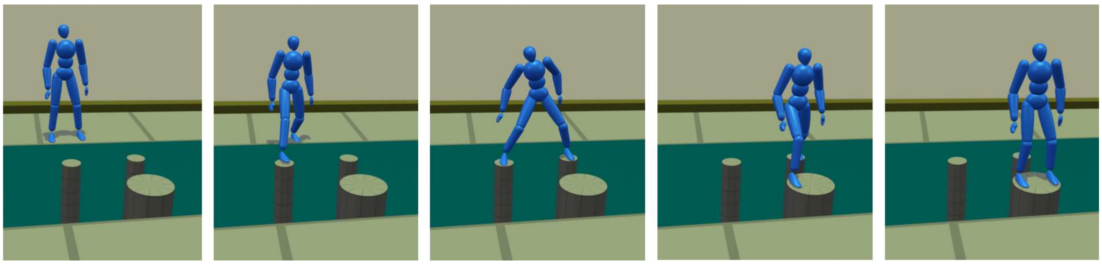

Figure 3.4: Careful stepping. Navigating an extreme environment by precisely placing steps shows off a series of four steps completely synthesized by our system.

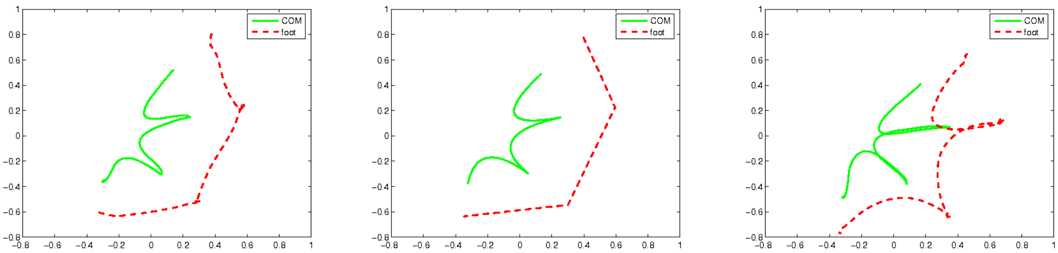


Figure 3.5: Comparisons for stepping. Cartesian plots for the foot and CM paths shown in red and green lines respectively over three consecutive steps. On the left is a contiguous motion capture example held out of the database but used as target input for foot placement and timing. In the middle is motion resulting from our model (also shown in Figure 3.6). On the right is the starting blend, as described in the text.

quality of a second synthesized series with a continuous motion sequence of three steps held out of the database (see Figures 3.5 and 3.6) Next, we include two animation results that are generated without the examples from database. The goal here is to breakdown the contributions of each component of our system and to show off the power of our technique for creating seamless transitions by stepping. In the video, we show a turning task which is derived from simply rotating a contiguous motion of a “ready-stance” in martial arts to certain degrees. We contrast the optimized result with the starting blend. Next, we modify a series of fighting attacks to control the direction of one kick by changing the stepping motion prior to the attack.

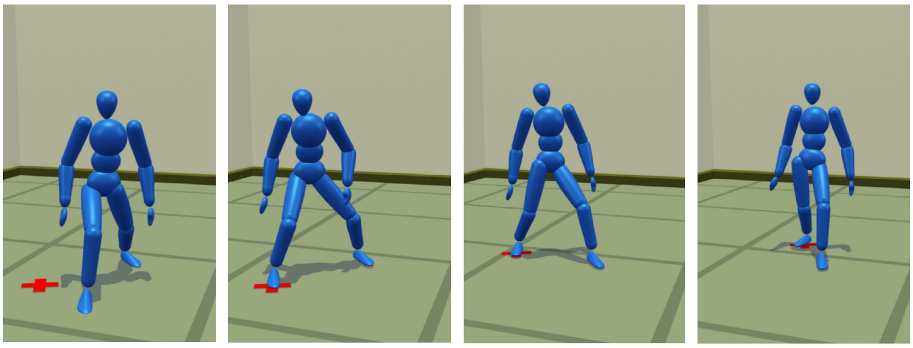

Figure 3.6: Animation of a series of steps pivoting on the left foot. The red X marks the consecutive end locations of the steps synthesized, the X values were taken from the motion capture sequence shown on the left in Figure 3.5.

## 3.8 Discussion and Conclusions

We have demonstrated the power of our simple method for generating controlled stepping movement. The underlying assumptions in our system are motivated by motor theorists and are supported by comparisons with motion capture examples of stepping. While the technique is very simple, used in combination with a stepping motion database we can generate rich motion that is comparable to unmodified stepping motion.

Our approach does include certain limitations. First, the system does not make any modifications to the upper body. While we know the upper body will respond to the movement of the lower body during stepping, we rely on the upper-body response embedded in the stepping example. When we remove the use of example, the motion of the upper body is computed solely from the interpolants and their blend. There is no guarantee that this will result in realistic motion. Likewise, the pivoting of the support foot is derived solely from the starting blend and we feel it is acceptable but not truly reflective of what we see in the motion database. In the next chapter, we will show that by controlling a character's aggregate linear and angular momenta we are able to generate full-body movements including arm swings without motion capture examples. Second, the piecewise linear model for the velocity of the CM is likely too over-simplified to match human motion tightly, although we found it acceptable for our purposes. And finally, if user inputs a desired stepping position whose distance is farther than the reaching scope of a single step for the character, our system currently is unable to automatically generate multiple steps to achieve the desired step position. This problem generally requires a sophisticated motion planning routine which is beyond the scope of this work.

Motion generation from a system like the one proposed here is useful for directly animating a character. However, we believe it is also potentially valuable for informing a control system when employed in the activation of a physical character. We see this as a promising direction for future work and show our latest system in the next chapter. Moreover, the system we describe is easy to implement and fast to run, and so we hope it is adopted by game developers and animators who need to take steps toward stepping.

# Chapter 4

# Stepping and Push Recovery with Momentum Control

This chapter proposes a technique for animating simulated characters to perform controlled steps. The desired step is controlled by high-level goals, namely step position and step duration. These stepping goals guide desired time-varying values for center of mass and stepping foot which in turn lead to objectives dictating the desired changes in momentum and joint angles over the duration of the step. Our approach employs a multiobjective optimization to solve for joint accelerations from the objectives and uses inverse dynamics to compute joint torques. Our approach can guide a character with purposeful, directable steps for controlling careful navigation of the character's position and orientation. In addition, the same system can be used to create protective steps to prevent falling as a reaction to perturbations. A novel supervisory routine automatically chooses when and where to step based on an analysis of the momentum conditions for the character. We contrast this approach to previous methods for step recovery using the inverted pendulum.

## 4.1 Introduction

Creating controllable responsive characters is a challenging open problem in computer animation and is essential for real-time applications such as electronic games. Physically based simulation holds promise for animating realistic characters in interactive environments. Through simulation, the interplay between characters, objects, and their surroundings can be generated automatically by constraining the motion to follow the dynamic equations of motion. However, developing robust and flexible controllers for simulated characters remains a difficult problem. In this chapter, we present a controller which allows characters to step in arbitrary directions both voluntarily and as a result of perturbations.

Stepping is a fundamental skill involved in common activities such as walking and full-body maneuvering from foot repositioning. As such, stepping is a critical behavior for applications involving virtual human avatars. For example, characters in electronic games must be able to change their stance and facing direction in response to a player's inputs. We propose a controller to conveniently synthesize a wide range of stepping behaviors such as directed stepping for change of stance, reactive step for maintaining balance and continuous steps for walking. The inputs to our controller are generic task goals, namely step position and duration, which allow us to apply the technique to various situations and different character morphology.

Furthermore, we introduce a hierarchical control approach to direct stepping that employs a novel momentum-based analysis in a supervisory stage to determine both when and where to step. Given the supervisor's selection of stepping goals, a parameterized curve generator computes desired trajectories for the center of mass (CM) and the stepping foot. More specifically, we use the trajectories developed in previous chapter as the desired paths. By doing so we benefit from the realistic human motion capture data. Furthermore, we demonstrate that the desired trajectories developed in previous chapter not only are valuable for use in kinematics model, but are also critical for use in physics model. These two values lead to behavior-specific objectives which guide changes in character's linear momentum and joint angles over the duration of the step. Automatic conversion from high-level goals to low-level control signals has been applied to generate procedural gaits and steps using kinematic models [80, 91]. Here we apply similar technique to generate controllable steps using a physical model.

Contrasting our approach to inverted pendulum (IP) techniques [65, 64], we find that considering both linear and angular momenta is especially important for step response to perturbations. This finding is supported by biomechanists whom have shown that humans carefully regulate angular momentum in activities such as walking [26]. Although the IP plus flywheel was also introduced to incorporate angular momentum in [64], our momentum analysis formulation is straightforward and is supported by biomechanics literature [50] as we explain in Section 4.3.

The main contribution of this chapter is our formulation for choosing when and where to step in response to perturbations. Including angular momentum in our for-

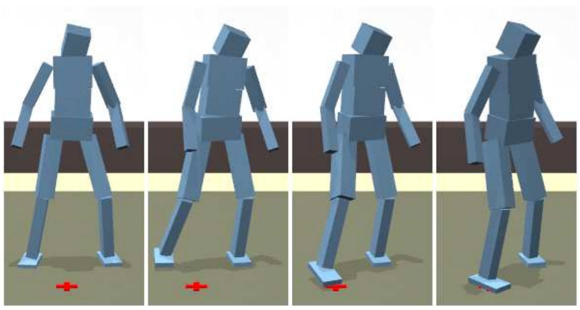

Figure 4.1: A sample output for a directed step

mulation improves a character's robustness under perturbations compared to IP models which only consider linear momentum. A second contribution is our high-level control for the flexible synthesis of goal-directed stepping. We demonstrate the generality of our technique through exploration of three classes of stepping behaviors: directed stepping for navigation, reactive stepping in response to perturbations and continuous stepping for walking. We also show that our stepping controller can be seamlessly applied to different character morphology, such as a character in handstand and a dinosaur character.

## 4.2 Control Structure

We use a three tiered architecture to control a character to step. At the lowest level we employ a control technique described by Macchietto et al. [48]. That is, a multiobjective optimization solver determines desired joint accelerations and an inverse dynamics computes joint torques from the output accelerations to drive a character simulation. In our case, the solver objectives are informed by input signals which are computed once per footstep, automatically, based on the conditions and goals of the specific behavior.

At the core of our controller, the system directs the step through automatic specification of two straightforward "goal" input signals, one for the CM and one for the swing foot. Their desired trajectories are modeled by two parametric curves based on empirical models built to follow similar paths extracted from motion capture data [91]. To convert the input signals to the objectives, we interpret the CM acceleration as linear momentum change and use inverse kinematics (IK) [82] to compute a pose that achieves the desired foot trajectory.

At the highest level, a user or a supervisory routine directs the high-level characteristics of the behavior, namely the position of the swing foot and the duration of a step. For a reactive step, the supervisor guides the choice of when and where to step based on an analysis of the momentum conditions for the character. We highlight details with respect to the supervisor next.

## 4.3 Goal-Directed Stepping

Starting from double stance, the character can take intentional steps by employing our step controller. In the simplest manner, the user can direct the system by specifying a new location for one of the feet. A reasonably large range of foot positions can be controlled. Default timing and stepping height are employed, although these values can also be controlled to change the style of the step. Generating motion in this manner is similar to driving animation with desired footprints as in [83].

### 4.3.1 Reactive Stepping

For reactive stepping, the supervisor automatically determines when and where to step by assessing the character's current momenta. Unlike the linear IP plus a flywheel [64], our formulation does not have the constraint of a constant height of CM and avoids the simplification of the flywheel which has a fixed moment of inertia. Instead, we use the relationship between full-body momenta changes and control over the CM and center of pressure (CP) [48]. This relationship allows us to skip the approximation of the rotational inertia of the body.

CP can be expressed as a function of the linear momentum change, $\dot{L}$, angular momentum change, $\dot{H}$, and the CM, $c$,

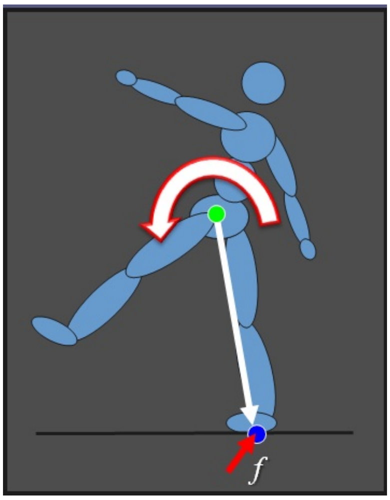


Figure 4.2: Assuming no additional forces, change of angular momentum around the CM (green circle) can be expressed as a cross product of the ground reaction force and the torque arm which is the distance from the CM to the CP (blue circle).

$$p_x = c_x - \frac{\dot{L}_x}{f_z} c_z - \frac{\dot{H}_y}{f_z} \tag{4.1}$$

$$p_y = c_y - \frac{\dot{L}_y}{f_z} c_z + \frac{\dot{H}_x}{f_z} \tag{4.2}$$

where $f_z = \dot{L}_z + mg$ is the vertical ground reaction force, $m$ is the total mass of the character and $g$ is the positive gravitational acceleration. The above equations are the expansion of $\dot{H} = (p - c) \times (\dot{L} + mg)$ from a static analysis of momenta (See Figure 4.2). More specifically, assuming no additional forces, change in a character's linear momentum is equivalent to the difference between the ground reaction force and force due to the gravity, i.e. $\dot{L} = f - mg$. Next, since torque is equivalent to change in angular momentum. If we draw a line from a character's CM to its CP, the character's change in angular momentum can be expressed as a cross product of the ground reaction force and the torque arm which is the distance between CM and CP, i.e. $\dot{H} = (p - c) \times f$. These two equations can be combined into one due to the same variable, $f$. By combining these two equations we have the final equation $\dot{H} = (p - c) \times (\dot{L} + mg)$. The same

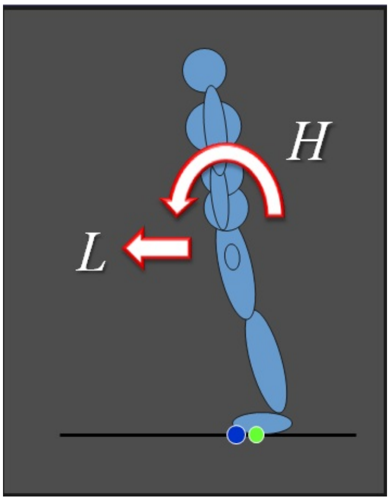


Figure 4.3: The goal of a reactive step is to arrest the induced linear and angular momenta denoted by $L$ and $H$ respectively.

equations 4.1 and 4.2 are referred to as the predicted zero moment point (ZMP) in robotics literature [60]. CP and ZMP are equivalent [23] if the predicted ZMP is within the support and the character is only in contact with flat ground. The value at the edge of the support means the support is or will rotate and is a powerful indicator that the character should take a step.

According to the fall recovery mechanisms reviewed in biomechanical literature, linear and angular momenta induced by a push are neutralized during the impact phase of swing foot contact [50]. We infer that a reactive step arrests the induced momenta by a push through proper foot placement. Assuming current linear and angular momenta of the character are $L$ and $H$, simple but effective desired momenta changes can be specified as

$$\dot{L}_{des} = -d_l \cdot L \tag{4.3}$$

$$\dot{H}_{des} = -d_h \cdot H \tag{4.4}$$

where $d_l$ and $d_h$ are damping variables. Substituting these desired momenta changes into Equations 4.1 and 4.2 gives us a new *desired* CP which accounts for the desired momentum changes:

$$p_{x_{des}} = c_x + \frac{d_l \cdot L_x}{f_z} c_z + \frac{d_h \cdot H_y}{f_z} \tag{4.5}$$

$$p_{y_{des}} = c_y + \frac{d_l \cdot L_y}{f_z} c_z - \frac{d_h \cdot H_x}{f_z} \ . \tag{4.6}$$

We use Equations 4.5 and 4.6 to determine whether the character should step or not. We use the condition that the desired CP is outside of the current support as an indicator for when the character needs to step. When this occurrence is indicated, we anticipate that the support foot will soon rotate and therefore the character should take a step to prevent it. In contrast, Equations 4.1 and 4.2 do not indicate a step until the CP (ZMP) reaches the actual edge. We point out that Equation 4.5 and 4.6 depends only on two momentum damping terms $d_l$ and $d_h$ to decide when to step. The rest terms are the character's current states, such as CM position, linear and angular momenta, which can be computed.

CM position and velocity have been used for the prediction of step initiation based on the IP model [58, 65]. Ours is different in that we also consider the angular momentum around the CM (Equations 4.5 and 4.6). The combination of linear and angular momenta provides better prediction of a character's stability than the IP. We also note that the damping values $d_l$ and $d_h$ affect the character's tendency to step. Higher damping values imply the character is more conservative and is more l to take protective steps. Equations 4.1 and 4.2 give us no equivalent control over this tendency. In all of our results, we set $d_l = 4$ and $d_h = 6$.

### 4.3.2 Where to Step

To step, we select the foot which is closest to the new desired CP as the swing foot. Default timing and stepping height are employed unless otherwise specified by the user. Next, we employ Equations 4.5 and 4.6 with increased gain values ($d_l = 9$ and $d_h = 18$) to compute a conservative position which is farther out for where to step. The reason to compute a conservative step position is because the desired step position is just outside of support polygon when a step is indicated. Instead of taking a small step or being liberal, we opt to take a more conservative step to improve a character's stability. We choose to place the character's foot at a (conservative) estimate for the desired CP value in order to enable the ability to push from that point on the ground plane. In practice, this simplification works well because the stepping foot location provides the most promising vantage point from which to push through the desired CP. This is especially clear in situations when multiple steps are required to maintain balance because the old support is lifted quickly following the stepping foot's touchdown.

### 4.3.3 Comparison to Capture Point

Before going on to step synthesis, we perform a brief analysis of our method in comparison to capture point control, which is an alternative technique used to choose where to step in robotics. Capture point [65, 64] is based on an IP model with a constant height of CM and can be shown to be the same as the position of CM plus a velocity-scaled term or

$$x_{foot} = c_x + \sqrt{\frac{c_z}{g}}\dot{c}_x \ . \tag{4.7}$$

(We focus on the $x$ axis for brevity). While this result is derived from an energy analysis, upon observation we see that our approach adds an extension to the capture point with a term that accounts for the change in full-body angular momentum induced by external perturbations.

With careful inspection, we can reduce the differences between the capture point and our method. First, since capture point does not consider angular momentum, we could ignore angular momentum by zeroing out $d_h$ in Equation 4.5. Second, capture point's constant CM height assumption implies $f_z = mg$ and by definition $L_x = m\dot{c_x}$. Finally, capture point is a model of single support; therefore the desired CP coincides with the foot. Applying these differences to Equation 4.5 gives us the following expression:

$$x_{foot} = c_x + \frac{d_l \cdot c_z}{g} \dot{c_x} \ . \tag{4.8}$$

Comparing this simplified version of our system to capture point, we see that if we choose $d_l = \sqrt{\frac{g}{c_z}}$, Equation 4.8 is exactly the same as capture point. Assuming an average human CM height of 1 meter, $\sqrt{\frac{g}{c_z}} \approx 3.1$ is not far from our choice of $d_l = 4$. Further comparison appears in our animation results.

## 4.4 Parameterized Stepping Model

Based on the stepping goals specified by the supervisor, our system automatically plans the desired positions of the CM and the swing foot over the duration of the step. The desired trajectories are idealized by two parametric curves based on empirical evidence extracted from motion capture data.

To fit within the multiobjective controller described in [48], our parameterized step model should provide both tracking and momenta objective values. The tracking objective requires joint accelerations which follow a given reference trajectory. In our case, we use a default pose and modify it using IK to follow a synthesized foot path. For momentum, we control the CM trajectory and convert this trivially to desired changes in linear momentum. We also control angular momentum change, but only about the vertical axis since angular momentum about the horizontal axes is controlled by the step position.

### 4.4.1 Swing Foot Control

We found the appearance of the overall behavior particularly sensitive to the chosen stepping path. After some experimentation, we model the desired motion of the swing foot as if it is performing a point-to-point reach - that is, considering the foot as if it is the end effector and treating the step as if it is a reaching task [91]. There has been in-depth investigation performed on hand point-to-point movement [3, 21] and in this body of work the hand traverses an approximately straight line path with a idealized bell-shaped speed profile. We adopt a similar estimate for the foot trajectory (See Figure 2). We use a synthetic Gaussian function to serve as our speed profile with a width set to 0.08, as in [91]. We automatically tune the Gaussian by scaling its amplitude such that the traversing distance matches the desired step displacement.

For tracking, we compute the desired acceleration for the joint angles:

$$\ddot{\theta}_{des} = k_T(\theta_r - \theta) + d_T(\dot{\theta}_r - \dot{\theta}) + \ddot{\theta}_r \tag{4.9}$$

where $\theta_r$, $\dot{\theta}_r$, $\ddot{\theta}_r$ are the reference joint angle, velocity and acceleration computed at runtime based on the swing foot trajectory. The reference joint angle is resolved at runtime using IK from the simulated CM. Ideal positions of the CM and the swing foot are treated as the desired root and end-effector for solving ideal motion and reference

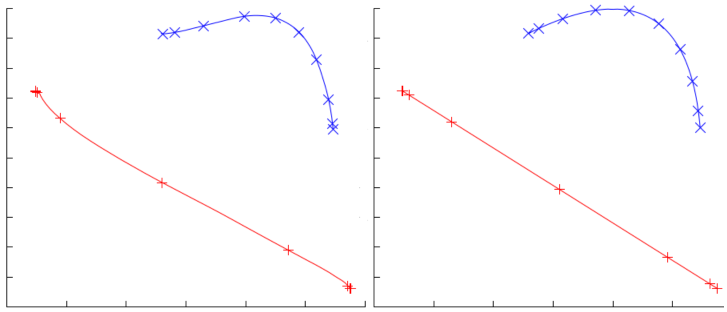

Figure 4.4: Cartesian plots for the paths of swing foot (line) and CM (curve) for a motion capture example (left) and our model (right).

joint velocity and acceleration are approximated using finite differencing from this ideal motion. In practice, we also found it necessary to lift the foot slightly to avoid unwanted contact between the foot and the floor - a second 0.08-width Gaussian function with controllable maximum height served for this requirement.

### 4.4.2 Center of Mass Control

Empirically, we have found that the path of CM observed in the motion capture data could be reasonably mapped using a quadratic curve (See Figure 4.4). Specifically, we employ a quadratic Bézier with the position of the current CM, the support foot (pivot), and the midpoint between the pivot and the desired step position as the successive control points [91]. Although this CM trajectory seems overly simplified, while in single support the CM does follow closely to a quadratic curve, since the body is in a controlled fall. This simplification is consistent with the IP model commonly used for prediction in robotics [36, 64].

We compute the desired CM acceleration using the following equation:

$$\ddot{c}_{des} = k_L(c_r - c) + d_L(\dot{c}_r - \dot{c}) + \ddot{c}_r \tag{4.10}$$

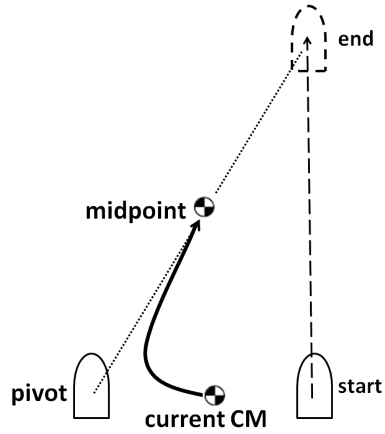


Figure 4.5: The desired CM trajectory can be automatically computed by using current CM, the pivot and the midpoint as control points.

where $c_r$, $\dot{c}_r$, $\ddot{c}_r$ are the reference CM position, velocity and acceleration respectively. The values $k_L$ and $d_L$ are manually selected and kept as constants in all our results. We sample the entire curve using an ease-in/ease-out function to determine the reference CM positions. Reference CM velocity and acceleration are approximated numerically from the sampled CM positions over time. Equation 4.10 is then transformed to linear momentum change by multiplying the character's mass.

Finally, we control the angular momentum in the character's vertical axis. More specifically, damping the angular momentum around the character's vertical axis creates the swing of the arms. This result is supported by work in both biomechanics [26] and robotics [35] fields. We accomplished this by adding a simple damper in the angular momentum objective:

$$\dot{H}_z = -d_z \cdot H_z \ . \qquad (4.11)$$

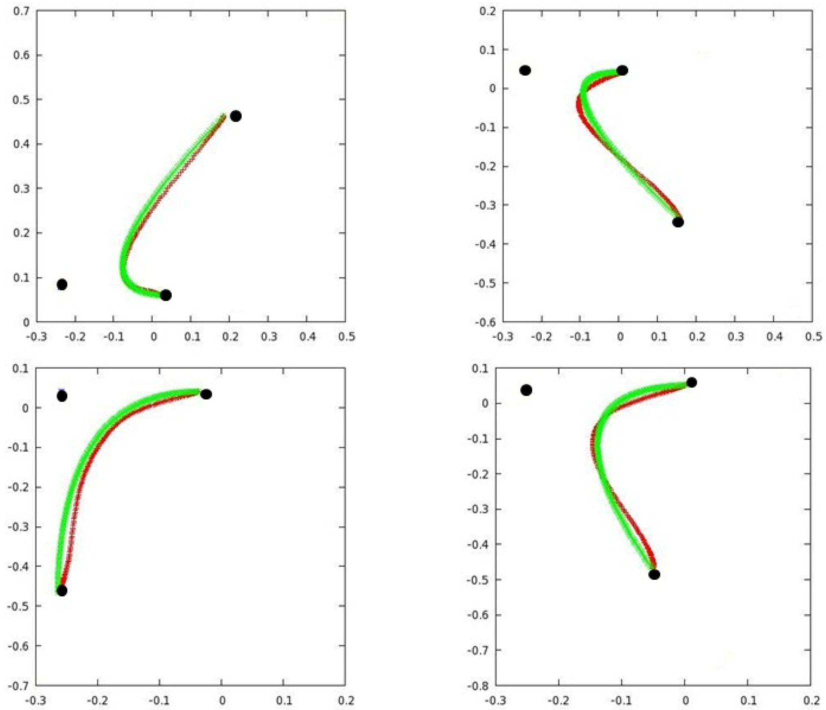


Figure 4.6: More examples on the approximation of CM trajectories (red) using quadratic Bezier curves (green)

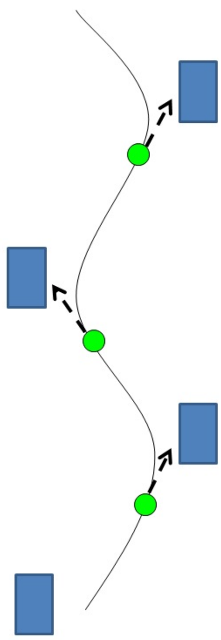

Figure 4.7: CM follows a sinusoidal path during walking and the required step duration is automatically determined by capture point which is CM position plus a scaled CM velocity term. We know exactly when to put the foot down because CM position and CM velocity are explicitly controlled by our controller as in Equation 4.10.

## 4.5 Continuous Stepping

By recursively applying the supervisor after each step, we are able to generate multiple reactive steps to regain a character's balance. Furthermore, with the power and versatility of our step controller, the construction of a walking behavior required only slight modifications to allow for alternating steps.

Similar to directed and reactive stepping, the inputs to our walking controller are also step position and step duration. We use a series of equal length steps to generate a steady walk for our walking example. According to Herr et al [26], the angular momentum is highly regulated and the value is close zero during human walking. Furthermore, as our comparison with capture point in section 4.3.3, if the angular momentum is negligible, the desired CP in Equations 4.5 and 4.6 can be simplified to capture point. Given the equal length step positions, we use capture point to automatically determine the step duration. As mentioned in section 4.3.3, capture point is simply the position of CM plus a term with the scaled CM velocity. Since we dictate the desired CM position and CM velocity in our CM control (section 4.4.2), we can easily compute the position of capture point at any time. In order to make the character walk continuously, the step duration is automatically determined such that the swing foot aligns with capture point when it touches the ground. In Figure 4.7, CM is represented as a green circle and the arrows indicate the positions of capture point which are the desired step positions in order for the character to walk continuously.

## 4.6 Implementation and Results

To demonstrate the power of our approach we present a series of animation results that highlight unique aspects of our system. All simulations were performed in real-time on a 2.4 GHz processor. The multiobjective optimization was solved at a frequency of 60 Hz and the inverse dynamics computed joint torques at the simulation rate 2000 Hz.

**Directed stepping.** To show the basic operation of the tool, we input a series of footsteps for the character to follow. Each footstep is shown as a red indicator in the animation (as in Figure 4.1). We demonstrate that we can reorient and position the character by taking a small number of directed steps. Further, because only high-level goals are controlled and no character specific parameters are set, we can change the character's configuration. We showcase the value of this aspect of our system by making the character take steps in a handstand.

**Reactive stepping.** Responsivity is an important feature of the controller. In the related animations, we show that the character can sustain multiple impulses by taking steps in various directions. Each impact is 170 N applied for 0.1 sec. The resulting action is both complex and believable, especially considering no motion capture data was used (Figure 4.8). In addition, we show that the supervisor can opt not to take a step and instead use standing balance control [48] to respond to the impact. As mentioned in Section 4.3.1, $d_l$ and $d_h$ can be used to control the character's tendency to step.

Our stepping mode is considered between statically stable steps, i.e. zero initial velocity and zero target velocity. However, the impact force created by the swing foot contact might cause non-zero target velocity after each step. This effect is not a problem as our system can recursively apply the supervisor after each step to automatically determine if another step is needed.

**Comparison to IP.** We contrast our momentum supervisor to IP by follow-

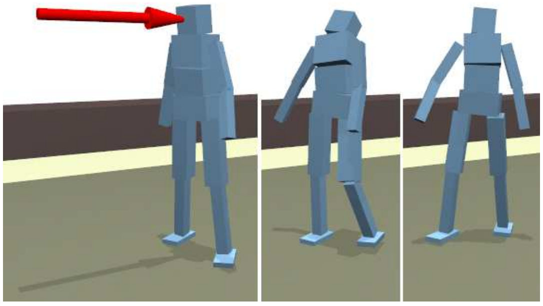

Figure 4.8: A reactive step generated automatically in response to a disturbance.

ing the descriptions in Section 4.3.3. In this comparison, the IP does not include the angular momentum terms in Equations 4.5 and 4.6. The result shows the momentum supervisor initiates the recovery earlier than the IP under the same impact. Under small disturbance, both IP and momentum supervisor are able to maintain character's balance by stepping. However, being able to quickly initiate a step is especially important under large disturbance. We show that our supervisor can still keep the character in balance while the IP fails when the force is increased to 250 N for 0.1 sec. For clarity we note that this is not exactly the capture point since our domain is double support while capture point is single support.

## 4.7 Conclusions

In this chapter, we present a goal-directed controller for simulated characters to perform directed and reactive steps by guiding the CM, and the swing foot. The character is able to follow the desired step positions (footprints) specified by the user. The same controller works for different character morphology. To react to a disturbance, the character can take protective steps computed automatically by our momentum-based

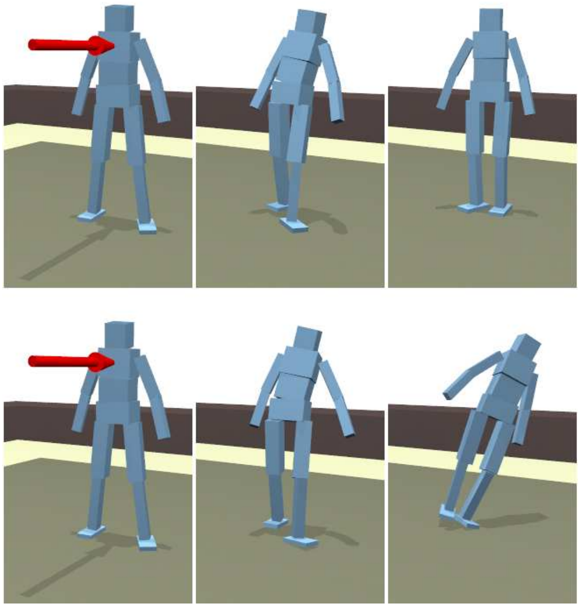

Figure 4.9: The character is able to take multiple steps under a large perturbation when we apply the supervisor after each step (top). IP does not consider angular momentum; therefore, the character is less robust than the one using our supervisor and fails to maintain the character's balance under a large perturbation (down).

supervisor. Considering both linear and angular momenta in the supervisor improves character's robustness to disturbances. Lastly, the required number of steps is automatically determined by applying the supervisor after each step.

# Chapter 5

# Anticipation from Example

Automatically generated anticipation is a largely overlooked component of response in character motion for computer animation. In this chapter, we present an approach for generating anticipation to unexpected interactions with examples taken from human motion capture data. Our system generates animation by quickly selecting an anticipatory action using a Support Vector Machine (SVM) which is trained offline to distinguish the characteristics of a given scenario according to a metric that assesses predicted damage and energy expenditure for the character. We show our results for a character that can anticipate by blocking or dodging a threat coming from a variety of locations and targeting any part of the body, from head to toe. Since this work is a collaboration with my other colleagues, this chapter will focus on my contribution, motion interpolation synthesis, i.e. creating physical plausible motion transition from current to selected anticipatory motion from the database. Please refer to [102] for details of other aspects of the system such as assessing damage and supervised learning.

## 5.1 Introduction

Anticipation behavior has been largely overlooked in computer-generated characters, especially in interactive settings such as virtual environments and electronic games where such motions must be computed automatically. While responding *after* an interaction has received considerable attention and is necessary to uphold the physical realism of contact resulting from an interaction, anticipatory response *before* an interaction is an important component for making characters appear alert to their environment and conscious about themselves. In this chapter, we introduce a novel technique for generating anticipation that selects from a database of possible motion capture examples of anticipation based on the specific conditions of an impending interaction.

As our testbed we focus on making a character anticipate and block a threat coming from a range of directions, heights, and speeds. We focus our domain on a character which starts from an idle, standing state (rather than anticipating interactions starting from any arbitrary state.) However, we do not make any assumptions about the activities that the character can do in regards to anticipation. Instead, we add a large variety of anticipation clips (*examples*) to a database, including actions such as taking protective steps, ducking, or lifting a leg off the ground to protect against a threat. In contrast to previous research [51] which builds a model of anticipation drawn from psychology, we rely on human performance in the form of anticipation examples to produce lifelike anticipatory actions and focus our effort on the motion interpolation systhesis technique which creates seamless transition between motions.

## 5.2 Related Work

Several researchers have introduced techniques that generate responses for motion capture-driven characters reacting to unforeseen influences [56, 101, 95, 70, 49, 103, 6, 38]. In general, previous methods for responding to an interaction take into account the physical components related to the impact, either in the form of a simulated collision, as in [19, 101, 49, 103], or by modifying dynamic parameters of the character motion, such as joint velocities [56, 6] or momentum [37, 38]. The result of creating these kinds of changes is character motion that gives the impression of responding physically following the impact.

To our knowledge, our previous research effort [51] is the only one reported on anticipation for character response. In that paper, we use insights drawn from psychological literature to infer the proper behavior mechanisms to employ for anticipatory action and develop heuristic rules that are consistent with psychological findings. Our previous results included anticipating impacts to the head and upper body. In contrast, in this work we employ examples of anticipation taken from motion capture which are more natural-looking.

## 5.3 Overview

Our system combines a selection routine, which decides the anticipation action to employ from a library of examples, with a motion interpolation synthesis step which blends from the current motion to the anticipation motion, taking into account balance. Following the system flow diagram in Figure 5.1, we assume the character starts in idle standing balance (upper left.) When an interaction (or threat) is recognized, the

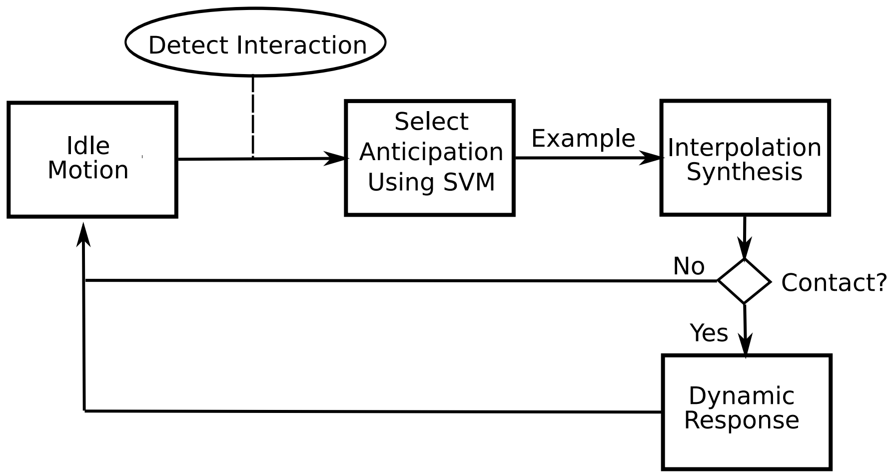


Figure 5.1: System flow diagram

system selects the anticipatory clip to employ using the SVM. Next, in the interpolation synthesis step, a transition to the example motion is generated, making adjustments to balance by controlling the center of mass across the transition. After the anticipation is complete, we compute a dynamic response [103] if there is contact and finally return to the idling motion with a final blend.

## 5.4 Balance Adjustment

To generate animation given the anticipation action selected from trained SVM, we propose a simple but effective motion interpolation synthesis technique that accounts for balance. The problem here is to concatenate the motion the character is currently doing, for example standing or idling, and to follow it with the selected anticipation motion from example. To be successful, the transition should not introduce any unwanted artifacts such as food slide. The most straightforward solution is to align the anticipation motion globally to the character's current position and facing direction and then to blend the root position and orientation as well as the joint angles. However, this naive approach would likely introduce unnatural foot sliding since there is no guarantee that the feet are in the same configuration before and after the transition.

To overcome this issue, we apply a balance adjustment step for the motion interpolation synthesis. Our specific routine is somewhat related to previous approaches for balance filtering [81, 51] but is unique because we approximate a purposeful weight shift during transition from one motion to another. That is, the system shifts the weight toward one foot, based on the foot in the anticipation example that is carrying more of the weight. Then, we use the newly selected "support foot" as the fixed root for blending and use the inverse kinematics (IK) routine described by Metoyer et al. citemetoyer:2007 on the other leg to keep the foot on the ground. The goal is to make the character lean in the direction of one foot and slide the other foot into place - the effect is a quick adjustment of weight that is largely imperceptible and, we believe, quite natural for situations where double support is followed by a sudden anticipatory movement.

To accomplish balance adjustment, our algorithm moves the center of mass smoothly toward the support foot, running an optimization step using BFGS to place the pelvis, and using IK to reposition the legs while keeping the feet fixed. This step is similar to center of mass control employed in Chapter 3 but without the quadratic Bezier trajectory. With the center of mass over the support foot, the system performs the blend to the selected example by aligning the support foot in the example motion with the current motion. During blending, the center of mass is returned smoothly to the unmodified position for the anticipation example. While we do not move the non-support foot explicitly in the balance adjustment process, the weight is shifted to the support foot which is used as the (fixed) root for the interpolation blend, and the non-support foot is therefore allowed to move as the blend takes place. We found it important for visual quality to use IK on the non-support leg and keep the foot on the

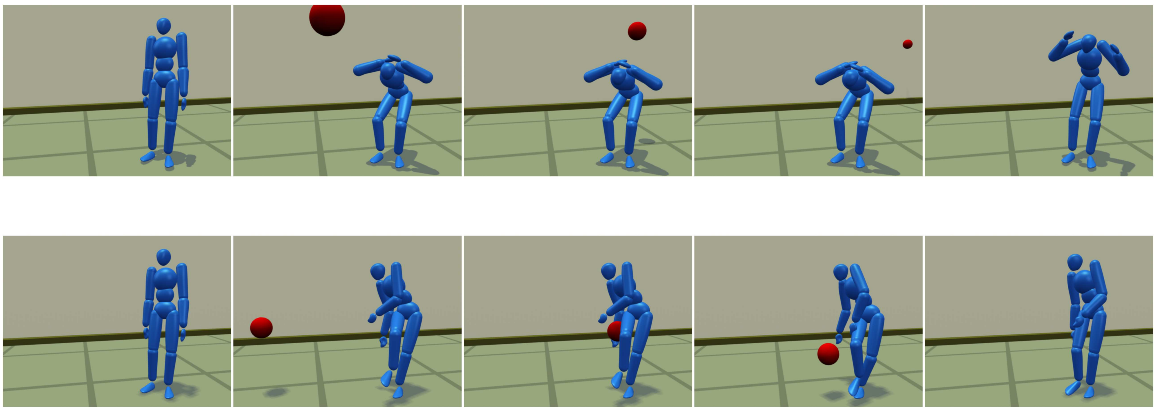

Figure 5.2: Two examples of anticipation from our system (view left to right).

ground, both avoiding lifting the foot and from potentially passing through the ground.

## 5.5 Implementation and Results

To realize our approach, we had to make several engineering and design decisions including how to construct our database and how to implement the various components. For our anticipation library, we include a set of examples which we capture methodically as blocks and anticipations protecting from several directions, targeting various areas of the body (legs, pelvis, trunk, head) and with two varying degrees, mild and exaggerated. In the capture, the subject was asked to imagine a threat approaching from each of eight directions. To ensure that the directions were consistent, a second person stood outside of the capture region and acted out a throwing motion to cue the subject. After each direction, the subject reset to a *home* position and the 'thrower' moved to the next location in preparation for the next direction. In total, we include sixty examples of anticipation which are segmented and mirrored (left-to-right) totaling 120 in the final repository.

To create the animations shown in the video, our system interpolates to the anticipation example by 'slerp'-ing quaternions, with a simple ease-in/ease-out (EIEO) time-based weighting across the transition. With the blended motion to the anticipation in place, the interaction itself is finally computed. If the interaction results in a contact (e.g. the SVM did not choose an anticipation that dodges a ball completely,) we incorporate a version of Zordan et al.'s [103] technique for responding to unpredicted impacts. This subsystem utilizes the ODE simulation to react physically to collision forces and then generates a smooth transition by interpolating between the anticipation and simulated motion. After a short duration, the simulation is blended back to the idle behavior or, at the animator's discretion, to a reaction example as described in the original implementation.

**Results.** We show in the accompanying video a variety of examples (See Figure 5.2) where the character is able to successfully dodge an incoming ball as well as believably anticipate and physically react to threats which the character is unable to avoid completely. In addition we show a set of animations where we make other modifications to test the limits of the system, namely: we modify the starting state to begin from a fighting motion taken from a different source (actor); and we add an example where the dynamic response returns to a new reaction example. The dynamic response is too slow for real-time currently, but without it, the system runs interactively using an AMD Athlon64 CPU with 2 Gigabytes of memory (with hardware rendering.)

## 5.6 Conclusions

In this chapter, we present an approach for generating anticipation using human motion capture examples. We employ supervised learning to select the example based on the given scenario and train our learner on observations where damage and energy are factors in determining the most suitable anticipation for the conditions of the scenario. To limit our scope, we focus our attention on a specific testbed where a standing character responds to a threat approaching from a variety of trajectories.

As described, there are a number of limitations with our current approach. Foremost, we assume that the character is starting in standing balance and while we do allow the character's state to vary, the range of starting states for most of our animations is fairly narrow. This issue is due to the fact that the anticipation we generate is computed using interpolation synthesis and to generalize to a larger set of starting states we would need to improve this component of the system in order to uphold the quality of the motion generated. Lastly, in some animations, the character appears omnipotent or "super-human." Adding delays, noise, and/or failed attempts at anticipation would fix this problem and fit nicely within our existing framework.

Even with these limitations, this work represents a large step forward in the state of the art for automatically generating anticipation action for characters and we look forward to further advances in this exciting topic in the near future.

# Chapter 6

# Conclusions

Through this work, I have demonstrated that controlling high-level goals such as full-body linear momentum, angular momentum and end-effector is an effective way to produce realistic coordinated movements of stepping. Controlling these high-level goals only requires low-dimensional input signals despite a character's high degrees of freedom which usually range from 30 to 60. The design of these desired input signals is inspired by principles and knowledge from biomechanics, robotics and neuroscience.

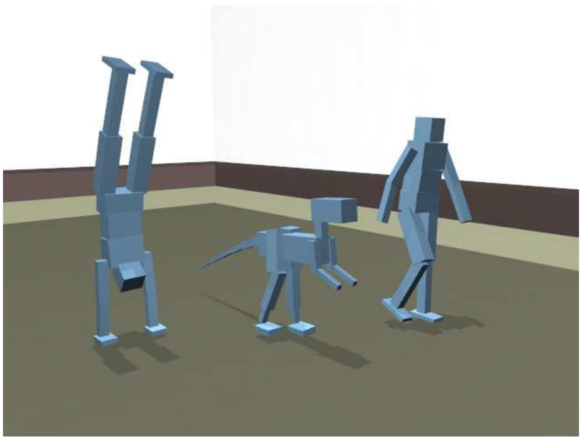

Figure 6.1: Since only high-level goals are controlled, we can apply the algorithm to different character morphology.

Controlling a character's linear momentum change is equivalent to the control of CM acceleration. CM behaves like an IP pivoting about the support foot during the swing phase of stepping and walking [89]. During this phase, humans shift their weights toward the support foot to avoid falling laterally while moving forward. I have shown that the desired CM trajectory during this phase can be reasonably mapped out by a simple quadratic Bezier curve with the beginning of CM, the pivot and the midpoint between the pivot and end position of swing foot as control points.

Several biomechanists and roboticists have investigated the underlying mechanics of arm swing during human walking. Herr and Popovic [26] found that the whole body angular momentum is small, despite substantial segmental momenta, indicating large segment-to-segment cancellations during walking. In particular, the movement of arm swing cancels out the leg angular momentum in the vertical axis. Normal arm swing also reduces the vertical ground reaction moment acting on the foot [11]. Based on these findings, I have shown that by continuously damping out the angular momentum of a character in the vertical axis results in natural arm swings during stepping and walking.

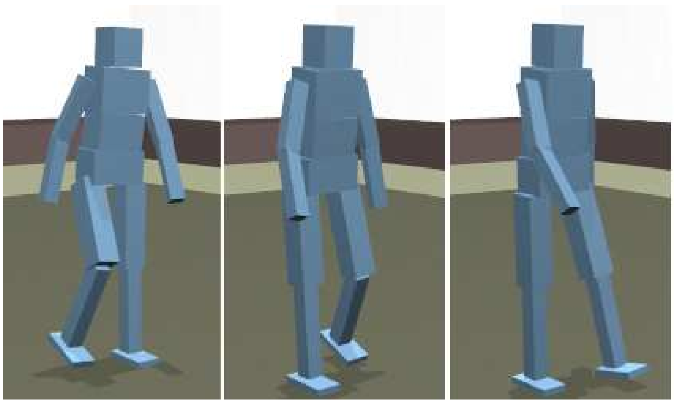

Figure 6.2: Natural arm swings automatically emerge during walking when we control the character's angular momentum around the vertical axis to be zero.

For the control of swing foot, I modeled the motion as if it is a point-to-point reach. There has been in-depth investigation performed on hand point-to-point movement for reaching tasks [3, 21] in neuroscience. In this body of work, it is commonly accepted that the hand traverses an approximately straight line path with a bell-shaped speed profile. I adopted a similar estimate for the foot trajectory by controlling it to follow the line segment formed by the starting and ending foot positions. I used a normal Gaussian function with automatically tuned width and amplitude to serve as the bell-shaped speed profile.

Lastly, maintaining a character's balance is also a low-dimensional problem. From Equations 4.5 and 4.6 we know the important features are CM position, linear momentum L and angular momentum H relative to the support polygon. I have shown that considering both linear and angular momenta enhances a character's robustness under perturbations compared to IP which only considers linear momentum. The proposed supervisor automatically computes when and where to step under perturbations with a straightforward goal: to remove all linear and angular momenta induced by a push. This control law is supported by biomechanics literature of fall recovery [50].

# Bibliography


[1] M. Abdallah and A. Goswami. A biomechanically motivated two-phase strategy for biped upright balance control. In *IEEE Int. Conf. Robotics and Automation*, 2005.

[2] Y. Abe, M. DaSilva, and J. Popović. Multiobjective control with frictional contacts. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007.

[3] W. Abend, E. Bizzi, and P. Morasso. Human arm trajectory formation. *Brain*, 105(2):331–348, 1982.

[4] O. Arikan and D. Forsyth. Interactive motion generation from examples. *ACM Trans. Graph.*, 21:483–490, July 2002.

[5] O. Arikan, D. Forsyth, and J. O'Brien. Motion synthesis from annotations. *ACM Trans. Graph.*, 22:402–408, July 2003.

[6] O. Arikan, D. Forsyth, and J. O'Brien. Pushing people around. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005.

[7] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152. ACM Press*, Shanghai, China, 2002.

[8] R. Boulic, R. Mas, and D. Thalmann. Position control of the center of mass for articulated figures in multiple support. *Proc. 6th Eurographics Workshop on Animation and Simulation*, pages 130–143, 1995.

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

[10] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[11] S. H. Collins, P. G. Adamczyk, and A. D. Kuo. Dynamic arm swinging in human walking. In *Proceedings of The Royal Society, Biological Sciences*, July 2009.

[12] S. Coros, P. Beaudoin, and M. van de Panne. Generalized biped walking control. *ACM Transactions on Graphics*, 29(3), 2010.

[13] S. Coros, P. Beaudoin, KK Yin, and M. van de Panne. Synthesis of constrained walking skills. *ACM Transactions on Graphics*, 27(5):113:1–113:9, December 2008.

[14] Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Robust task-based control policies for physics-based characters. *ACM Transactions on Graphics*, 28(5), December 2009.

[15] M. DaSilva, Y. Abe, and J. Popović. Interactive simulation of stylized human locomotion. *ACM Transactions on Graphics*, 27(3), 2008.

[16] M. DaSilva, Y. Abe, and J. Popovicć. Simulation of human motion data using short-horizon model-predictive control. *Computer Graphics Forum*, 27(2), 2008.

[17] M. de Lasa, I. Mordatch, and A. Hertzmann. Feature-based locomotion controllers. *ACM Transactions on Graphics*, 29(3), 2010.

[18] A. Egges and B. van Basten. One step at a time: animating virtual characters based on foot placement. *The Visual Computer*, 26:497–503.

[19] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable controllers for physics-based character animation. In *SIGGRAPH*, pages 251–260, August 2001.

[20] M. Farrell and H. Herr. Angular momentum primitives for human turning: Control implications for biped robots. In *IEEE-RAS International Conference on Humanoid Robots*, 2008.

[21] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985.

[22] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen. Snap-together motion: assembling run-time animations. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, I3D '03, pages 181–188, New York, NY, USA, 2003. ACM.

[23] A. Goswami. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *The International Journal of Robotics Research*, 18(6), 1999.

[24] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. *IEEE Int. Conf. Robotics and Automation*, 2004.

[25] N. Hansen. The cma evolution strategy: a comparing review. In *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, pages 75–102, 2006.

[26] H. Herr and M. Popović. Angular momentum in human walking. *Journal of Experimental Biology*, 211(4):467–481, February 2008.

[27] J. Hodgins and N. Pollard. Adapting simulated behaviors for new characters. acm siggraph. pages 153–162, August 1997.

[28] J. Hodgins, W. Wooten, D. Brogan, and J. O'Brien. Animating human athletics. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, pages 71–78, August 1995.

[29] A. Hofmann, S. Massaquoi, M. Popovic, and H. Herr. A sliding controller for bipedal balancing using integrated movement of contact and non-contact limbs. In *Intelligent Robots and Systems*, 2004.

[30] A. Hofmann, M. Popovic, and H. Herr. Exploiting angular momentum to enhance bipedal center-of-mass control. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, pages 2483–2489, Piscataway, NJ, USA, 2009. IEEE Press.

[31] L. Ikemoto, O. Arikan, and D. Forsyth. Knowing when to put your foot down. *Interactive 3D graphics and games*, pages 49–53, 2006.

[32] L. Ikemoto, O. Arikan, and D. Forsyth. Quick transitions with cached multi-way blends. In *In ACM Symposium on Interactive 3D Graphics*, pages 145–151, 2007.

[33] S. Jain, Y. Ye, and C. K. Liu. Optimization-based interactive motion synthesis. *ACM Transaction on Graphics*, 28(1):1–10, 2009.

[34] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, and K. Yokoi. Biped walking pattern generation by using preview control of zero-moment point. In *in Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1620–1626, 2003.

[35] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. *Intelligent Robots and Systems*, 2003.

[36] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3D Linear Inverted Pendulum Mode: a simple modeling for a biped walking pattern generation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 239–246, 2001.

[37] T. Komura, H. Leung, and J.J. Kuffner. Animating reactive motions for biped locomotion. In *Proc. ACM Symp. on Virtual Reality Software and Technology (VRST)*, 2004.

[38] Taku Komura, Edmond S.L. Ho, and Rynson W.H. Lau. Animating reactive motion using momentum-based inverse kinematics. *Compute Animation and Virtual Worlds*, 1(16), 2005.

[39] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics*, 23(3):559–568, August 2004.

[40] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Trans. Graph.*, 21:473–482, July 2002.

[41] L. Kovar, J. Schreiner, and M. Gleicher. Footskate cleanup for motion capture editing. *Symposium on Computer animation*, pages 97–104, 2002.

[42] S. Kudoh, T. Komura, and K. Ikeuchi. The dynamic postural adjustment with the quadratic programming method. In *Intelligent Robots and Systems*, 2002.

[43] S. Kudoh, T. Komura, and K. Ikeuchi. Stepping motion for a humanlike character to maintain balance against large perturbations. In *IEEE Int. Conf. Robotics and Automation*, 2006.

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

[45] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21:491–500, July 2002.

[46] Y. Lee, S. Kim, and J. Lee. Data-driven biped control. *ACM Transactions on Graphics*, 29(3).

[47] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture: a two-level statistical model for character motion synthesis. *ACM Trans. Graph.*, 21:465–472, July 2002.

[48] A. Macchietto, V. Zordan, and C. Shelton. Momentum control for balance. *ACM Transactions on Graphics*, 28(3), 2009.

[49] Michael Mandel. Versatile and interactive virtual humans: Hybrid use of data-driven and dynamics-based motion synthesis, 2004. *Master's Thesis, Carnegie Mellon University.*

[50] W. Mathiyakom and J. McNitt-Gray. Regulation of angular impulse during fall recovery. *Journal of Rehabilitation Research & Development*, 45(8):1237–1248, 2008.

[51] R. Metoyer, V. Zordan, B. Hermens, C.-C. Wu, and M. Soriano. Psychologically inspired anticipation and dynamic response for impacts to the head and upper body. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2007. to appear.

[52] I. Mordatch, M. de Lasa, and A. Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics*, 29(3), 2010.

[53] U. Muico, Y. Lee, J. Popović, and Zoran Popović. Contact-aware nonlinear control of dynamic characters. *ACM Transactions on Graphics*, 28(3):81:1–81:9, July 2009.

[54] T. Mukai and S. Kuriyama. Geostatistical motion interpolation. *ACM Trans. Graph.*, 24:1062–1070, July 2005.

[55] Naturalmotion. Endorphin and euphoria, 2010. www.naturalmotion.com.

[56] M. Oshita and A. Makinouchi. A dynamic motion control technique for human-like articulated figures. *Computer Graphics Forum*, 20(3), 2001.

[57] A. Egges P. Peeters and B. van Basten. The step space: example-based footprint-driven motion synthesis. *Computer Animation and Virtual World*, 21:433–441.

[58] YC Pai and J. Patton. Center of mass velocity-position predictions for balance control. *Journal of Biomechanics*, 30(4), 1997.

[59] S. I. Park, H. J. Shin, and S. Y. Shin. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 105–111, New York, NY, USA, 2002. ACM.

[60] M. Popović, A. Goswami, and H. Herr. Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *International Journal of Robotics Research*, 24(12), 2005.

[61] M. Popovic, A. Hofmann, and H. Herr. Angular momentum regulation during human walking: biomechanics and control. *IEEE Int. Conf. Robotics and Automation*, 2004.

[62] M. Popovic, A. Hofmann, and H. Herr. Zero spin angular momentum control: definition and applicability. *IEEE/RAS International Conference on Humanoid Robots*, 2004.

[63] Zoran Popović and Andrew Witkin. Physically based motion transformation. In *Proceedings of ACM SIGGRAPH 1999*, pages 11–20, 1999.

[64] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture Point: a step toward humanoid push recovery. *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 2006.

[65] J. Pratt and R. Tedrake. Velocity-based stability margins for fast bipedal walking. *Fast Motions in Robotics and Biomechanics Optimization and Feedback Control*, 2005.

[66] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, New York, 1994.

[67] C. Rose, B. Bodenheimer, and M. Cohen. Verbs and adverbs: multidimensional motion interpolation using radial basis functions. *IEEE Computer Graphics and Applications*, 18:32–40, 1998.

[68] C. Rose, B. Guenter, B. Bodenheimer, and M. Cohen. Efficient generation of motion transitions using spacetime constraints. In *ACM Siggraph*, pages 147–154, 1996.

[69] J.G. Cheng G. S.-H. Hyon, Hale. Full-body compliant humanhumanoid interaction: balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, 23:884–898, 2007.

[70] A. Shapiro, F. Pighin, and P. Faloutsos. Hybrid control for interactive character animation. *Pacific Graphics*, 2003.

[71] H Shin, J Lee, S Y Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20:67–94, April 2001.

[72] H. J. Shin, L. Kovar, and M. Gleicher. Physical touch-up of human motions. *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 194, 2003.

[73] T. Shiratori, B. Coley, R. Cham, and J. Hodgins. Simulating balance recovery responses to trips based on biomechanical principles. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2009.

[74] K. W. Sok, M. Kim, and J. Lee. Simulating biped behaviors from human motion data. *ACM Transactions on Graphics*, 26(3), 2007.

[75] B. Stephens. Humanoid push recovery. In *IEEE-RAS International Conference on Humanoid Robots*, 2007.

[76] B. Stephens and C. Atkeson. Multiple balance strategies from one optimization criterion. *The IEEE-RAS International Conference on Humanoid Robots*, 2007.

[77] B. Stephens and C. Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *IEEE-RAS International Conference on Humanoid Robots*, 2010.

[78] J. Stewart and J. Cremer. Animation of 3d human locomotion: climbing stairs and descending stairs. In *In Eurographics Workshop on Animation and Simulation*, 1992.

[79] J. Stewart and J. Cremer. Beyond keyframing: an algorithmic approach to animation. In *Proceedings of Graphics Interface '92*, pages 273–281, 1992.

[80] H. Sun and D. Metaxas. Automating gait generation. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 261–270, August 2001.

[81] S.Y. Tak, O.Y. Song, and H.S. Ko. Motion balance filtering. *Computer Graphics Forum*, 19(3):437–446, August 2000.

[82] D. Tolani, A. Goswami, and N. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graph. Models Image Process.*, 62(5):353–388, 2000.

[83] M. van de Panne. From footprints to animation. In *Computer Graphics Forum*, volume 16, pages 211–223, 1997.

[84] J. Wang and B. Bodenheimer. An evaluation of a cost metric for selecting transitions between motion segments. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 232–238, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[85] J. Wang and B. Bodenheimer. Computing the duration of motion transitions: an empirical approach. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pages 335–344, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[86] J. Wang and B. Bodenheimer. Synthesis and evaluation of linear motion transitions. *ACM Trans. Graph.*, 27:1:1–1:15, March 2008.

[87] J. Wang, D. J. Fleet, and A. Hertzmann. Optimizing walking controllers. *ACM Transactions on Graphics*, 28(5), 2009.

[88] D. Wiley and J. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Comput. Graph. Appl.*, 17:39–45, November 1997.

[89] D. A. Winter. *Biomechanics and motor control of human movement*. Wiley, 2009.

[90] W. Wooten and J. Hodgins. Animation of human diving. *Computer Graphics Forum*, pages 3–13, 1996.

[91] C.-C. Wu, J. Medina, and V. Zordan. Simple steps for simply stepping. *International Symposium on Visual Computing*, 2008.

[92] C.-C. Wu and V. Zordan. Goal-directed stepping with momentum control. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2010.

[93] J.-C. Wu and Z. Popović. Terrain-adaptive bipedal locomotion control. *ACM Transactions on Graphics*, 29(3).

[94] Y. Ye and C.K. Liu. Optimal feedback control for character animation using an abstract model. *ACM Transactions on Graphics*, 29(3), 2010.

[95] K.K. Yin, MB Cline, and DK Pai. Motion perturbation based on simple neuromotor control models. *Pacific Graphics*, 2003.

[96] K.K. Yin, S. Coros, P. Beaudoin, and M. van de Panne. Continuation methods for adapting simulated skills. *ACM Transactions on Graphics 27(3)*, 2008.

[97] KK Yin, K. Loken, and M. van de Panne. Simbicon: simple biped locomotion control. *ACM Transactions on Graphics*, 26(3), 2007.

[98] KK Yin, D. Pai, and M. van de Panne. Data-driven interactive balancing behaviors. *Pacific Graphics*, pages 118–121, 2005.

[99] KK Yin and M. van de Panne. Omnidirectional humanoid balance control: multiple strategies for reacting to a push. In *Univ. of British Columbia, CS Technical Report*, 2006.

[100] V. Zordan. Angular momentum control in coordinated behaviors. In *Third Annual International Conference on Motion in Games*, 2010.

[101] V. Zordan and J. Hodgins. Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2002.

[102] V. Zordan, A. Macchietto, J. Medina, M. Soriano, C.-C. Wu, R. Metoyer, and R. Rose. Ancitipation from example. In *ACM Virtual Reality Software and Technology (VRST)*, 2007.

[103] V. Zordan, A. Majkowska, B. Chiu, and M. Fast. Dynamic response for motion capture animation. *ACM Transactions on Graphics*, 24(3), 2005.