

AN ABSTRACT OF THE THESIS OF

Benjamin J. Hermens for the degree of Master of Science in Computer Science
presented on June 15, 2006.

Title: Anticipating Impacts

Abstract approved: _____

Ronald A. Metoyer

We present an approach for generating a character's response in anticipation of an impending impact. Protective anticipatory movement is built upon several simple actions that have been identified as response mechanisms in monkeys and in humans. These actions are parameterized by a model of the interaction based on the approaching object (the threat) and are defined in procedural rules according to heuristics. We guide the character to perform these actions while obeying physical limits for balance and maintaining characteristics taken from the behavior just prior to the interaction, in our case, a motion capture sequence. We combine our anticipation model with a physically-based dynamic response to produce animations where a character anticipates an impact before collision and reacts to the contact after the collision. We present a variety of examples including threats that vary in approach direction, size and speed.

©Copyright by Benjamin J. Hermens
June 15, 2006
All Rights Reserved

Anticipating Impacts

by
Benjamin J. Hermens

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 15, 2006
Commencement June 2007

Master of Science thesis of Benjamin J. Hermens presented on June 15, 2006.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Benjamin J. Hermens, Author

ACKNOWLEDGEMENTS

I'd like to give special thanks Dr. Ron Metoyer for exceptional advising. It has been a pleasure to work with him. I'd also like to thank my committee, Dr. Eugene Zhang, Dr. Mike Bailey, and Dr. Henri Jansen, for their time and input. I'm grateful to Robert Rose for showing me a starting point, Madhu Srinivasan for taking the time to answer questions and assist in trouble shooting bugs, and to all contributors to the B^E_AT_EX document class that was used to format this work. Finally, I'd like to thank the faculty and staff of Oregon State University and in particular the School of Electrical Engineering and Computer Science for providing this opportunity for education.

CONTRIBUTION OF AUTHORS

This work was completed as a joint project with Dr. Ron Metoyer of Oregon State University and Dr. Victor Zordan, Chun-Chih Wu, and Marc Soriano of University of California, Riverside. Regular phone conferences kept everyone involved in all aspects of the project including the algorithms and the implementation details. The implementation was done in two parts: the creation of the anticipation poses and the generation of the motion. The first part was completed at OSU and the second part at UCR.

A paper was submitted to the Eurographics / ACM SIGGRAPH Symposium on Computer Animation (2006). This thesis is based on that paper but adds many implementation details.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	1
1.1 Overview	3
2 Related Work	6
2.1 Computer Animation	6
2.1.1 Generating Responsive Characters	7
2.1.2 Motion Editing for Balance	8
2.1.3 Creating Poses Procedurally	9
2.2 Anticipation in Psychology	10
3 Computing the Pose	13
3.1 Threat Model	14
3.2 Defining the Rules	15
3.2.1 Rule Control Parameters	17
3.2.2 Lean Away	19
3.2.3 Turn Away	20
3.2.4 Recoil	21
3.2.5 Block	22
3.2.6 Startle	25
3.3 Static Balance	26
3.4 Extensions	31
4 Motion Generation	35
4.1 Motion Sequence	35
4.1.1 Protective Motion	37
4.1.2 Dynamic Response	37
4.1.3 Recovery Motion	38
4.2 Timing	39
4.3 Transitional Blend Mechanism	41
5 Results and Conclusions	44
5.1 Alternative Approaches	47
5.2 Conclusion	48
Bibliography	50

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.1	An example anticipation. The blue poses represent the motion capture, the orange pose represents the anticipation pose, and the grey poses represent blends between them.	2
1.2	The steps of our technique illustrated by snapshots taken at the instant of impact. From left to right: in blue, the pose from the original motion at the time of impact (note that the left foot is in the air); in yellow, the result of executing the rules; in orange, the balanced final anticipation pose; and in red, the collision and dynamic simulation. The front edge of the support polygon and the ground plane projection of the COM are shown for balance comparison.	3
2.1	Eccentricity is the angle, θ , between the facing direction, \mathbf{F} , and the direction to the threat, \mathbf{T}	12
3.1	<i>left:</i> The hierarchy of joints forming the character's skeleton. <i>right:</i> The character's identity pose and the global coordinate frame.	16
3.2	Changes in anticipation as the threat changes. The leftmost image in the figure shows the originating motion capture pose. The three rows to the right represent poses generated by the rules as threat properties change, one per row. The first row shows the threat's position changing from the character's right to left. The second row shows the threat's size varying from small to large. The third row shows the speed varying from slow to fast.	18
3.3	Input/output diagram for the lean rule.	19
3.4	Input/output diagram for the turn rule.	20
3.5	Input/output diagram for the recoil rule.	21
3.6	Input/output diagram for the block rule.	22
3.7	Top-down view of the character reaching out to block the threat. Both arms reach the same distance from their respective shoulder. The threat is on the right side so the right hand is closer to it.	23

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
3.8	The elbow position is set using a swivel angle. The angle is with respect to the shoulder's <i>down</i> direction, illustrated by the dashed line. Positive angles rotate the elbow away from the body.	24
3.9	Two before and after comparisons of static balance. In each image, the yellow pose represents the result of the rules before optimization while the orange pose represents the final sustained pose after optimization. The COM and support polygon are also shown. <i>left</i> : The yellow pose was already in balance and the optimization braced for impact. <i>right</i> : The yellow pose was out of balance and the optimization both balanced and braced for impact.	28
3.10	Three cases of balancing and bracing. The circles represent current and desired COM locations as yellow and orange, respectively. Case 1 occurs when the desired location is also balanced. Case 2 is out of balance to brace for impact. Case 3 was initially too far out of balance to fully correct the problem.	29
3.11	Two views of a threat approaching character's head from behind.	32
3.12	Torso impacts. The blue pose is the original motion capture pose from which the examples are generated. The three orange poses are examples of sustained anticipation responses for varying threat targets. (<i>top-right</i>) Responses to impacts to the head are defined by a set of rules. (<i>bottom-left</i>) Responses to impacts to the pelvis are defined by a similar set of rules. (<i>bottom-right</i>) Responses to impacts to the torso between the head and the pelvis are interpolated.	34
4.1	A timing schematic for our system. Blue lines represent motion capture animation. The red line represents the physical simulation. The orange line represents the sustained pose. Black lines illustrate blends. Grey lines show blends that are not part of the final motion. Dashed lines represent the positioning of the specific computed poses. Each region is labeled with the number of milliseconds it will occupy in the motion.	36

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
5.1	Speed variations. Sustained anticipation poses shown just prior to impact illustrate the effect the threat's speed has on the pose as it increases from left to right.	45
5.2	Three filmstrips show animations generated by our system. Here we demonstrate the result of returning to the original motion (top) or a new motion (middle and bottom) as well as the result of mild (middle) or extreme (bottom) physical threat properties.	46

LIST OF TABLES

<u>Table</u>		<u>Page</u>
3.1	Rule control parameters are initialized by threat model parameters.	17
3.2	Maximum rotation values (degrees). The rules choose angles between 0 and the maximum value to rotate the joints.	19

DEDICATION

To my wife Sheena and daughter Lexi.

I love you both dearly.

Anticipating Impacts

Chapter 1. Introduction

Many techniques exist for creating character animations, such as key framing, motion capture, and physical simulation. Most methods of human motion generation require that motion be computed offline and stored in a motion database. Later, the motion may be extracted from the database and played back as desired. An animation system based on these techniques alone is limited to playing only existing motions or small edits of existing motions. Therefore all necessary motions must be created ahead of time. In general, however, it is impossible to anticipate all the motions that will be needed by an interactive system. Even if all necessary motions were known, recording and storing all of them would be costly.

Researchers have long recognized the need to synthesize motion *on the fly*. Many techniques are available to generate or modify motion that meets certain requirements, including environmental conditions, interactions with other entities, and user-defined constraints such as locomotion paths. One common approach is to generate the new motion as a combination of existing motions, such as blending or interpolating with a weighting function.

Recently, researchers have combined motion capture animation with physical simulation to provide lifelike, physically correct responses to interactions such as collisions. This significantly enhances the visual quality of situations such as a



Figure 1.1: An example anticipation. The blue poses represent the motion capture, the orange pose represents the anticipation pose, and the grey poses represent blends between them.

ball striking a character in the head and makes the animation more believable. However these approaches have only been used to generate motion *after* the interaction has occurred. In most applications, the character appears unaware of the impending collision and does nothing in preparation. This is contrary to reality and detracts from the quality of the motion.

The contribution of this work is a novel solution for generating motion *before* the collision (Figure 1.1), whereby giving the character the humanlike qualities of environmental awareness and anticipation. By combining our solution with the previous work, we can dynamically generate motions for a character to anticipate and prepare for collisions and then respond to them in a realistic manner.

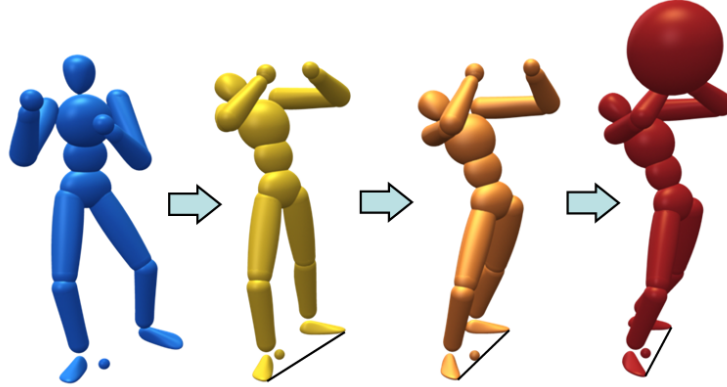


Figure 1.2: The steps of our technique illustrated by snapshots taken at the instant of impact. From left to right: in blue, the pose from the original motion at the time of impact (note that the left foot is in the air); in yellow, the result of executing the rules; in orange, the balanced final anticipation pose; and in red, the collision and dynamic simulation. The front edge of the support polygon and the ground plane projection of the COM are shown for balance comparison.

1.1 Overview

Our proposed solution generates the anticipatory response motion as a blend from the current pose of an animation to a computed anticipation pose (Figure 1.2). After the collision, another blend returns the character to either the original motion or an entirely new motion selected through a search. We used animations produced by motion capture but our algorithm is also applicable to animations produced by any other motion generation technique, such as keyframing or physical simulation.

The first step of our algorithm is to compute the anticipation pose. Intuitively, a human’s response to an approaching object depends on some properties of that

object. We call this object the *threat* and model it with a (small) set of controllable parameters. We use these parameters to guide a set of actions that produce a basic anticipation pose. Our intuition regarding these actions is confirmed by a survey of psychological studies of defensive behaviors. We encode the actions as rules that change the character’s posture by adjusting joint rotations. When a threat is detected, the rules are executed to produce the basic anticipation pose (yellow pose in Figure 1.2). The basic pose will be modified in the next step of the algorithm to enforce certain desired properties.

The second step enforces certain physical constraints, mainly balance, that are likely not maintained by the rules. This is framed as an optimization problem where the goal is to bring the character into static balance while maintaining properties from both the rules and the original motion. We apply this optimization to the basic anticipation pose to achieve the final anticipation pose, also called the *sustained pose*, which is braced and ready for impact (orange pose in the image).

Once the sustained anticipation pose is determined, the next step is to generate the anticipation motion by blending from the original motion pose to the sustained pose. Although a simple linear blend sounds straightforward, we found that more natural looking blends are obtained when several non-trivial properties are enforced. Just as we solved for static balance of the anticipation pose, we now solve for dynamic balance of the blended motion. We also found it desirable for the hands to travel along a straight path instead of sweeping an arc (as they do in a linear blend).

The fourth step makes the system complete by including a physical reaction when the threat strikes the character. We control the timing of the blend such that the collision occurs just as the blend is ending. This removes a distracting pause between the end of the blend and the collision.

The final, step returns the character either to the original motion or to an entirely new motion (selected by a search if the animator chooses) using a blend between three inputs. This *double blend* method is discussed in depth in chapter 4. Each of the two blends is similar to the one previously discussed with respect to constraints.

There is also an optional step that immediately follows perception, preceding the sustained anticipation motion. A startle pose is computed in a manner similar to the sustained pose and serves to give the character an initial protective posture. When startle is included, the character has the appearance of suddenly noticing the threat. In some cases we found this abruptness to be distracting and thus allow the inclusion of startle as an option for the animator.

Chapter 2 provides a review of related work. The following chapters presents each of these steps in detail: the computation of the sustained pose in Chapter 3 and the generation of the motion in Chapter 4. Chapter 5 presents results and concludes this work.

Chapter 2. Related Work

Our technique is rooted in a survey of psychological literature related to preparation for collisions. We present this survey following a discussion how our approach relates to previous work in distinct areas of computer animation.

2.1 Computer Animation

Fundamentally, our technique may be categorized as a procedural method for editing or re-using motion capture data. Motion capture editing and procedural motion generation have been hot topics for animation researchers in the last decade, leading to a wide variety of novel approaches. Specifically, we classify our method as a constraint-based motion editing solution applied to a single motion capture sequence. An abundance of other work in character motion synthesis has focused on similar techniques with various goals and results (*eg.* [WP95, BW95, Gle98, LS99, PW99, ALP04, HPP05, LHP05]). See the survey by Gleicher [Gle01] for a more thorough discussion. Within this framework, we describe our technique as employing specialized motion blending between motion capture sequences and procedurally generated poses (and back again) for the dedicated purpose of responding to impending impacts. What makes our technique stand out in this context is both our focus on the creation of a responsive pose and

our motion blending approach for achieving that pose which takes into account physical constraints while blending.

2.1.1 Generating Responsive Characters

Several researchers share our goal of creating responsive characters [FvdPT01, OM01, ZH02, KLK04, Man, AFO05, KHL05, ZMCF05]. The routines described for generating motion in response to an interaction often take into account the physical components related to the impact, either in the form of a simulated reaction [FvdPT01, Man, ZMCF05] or by directly modifying the dynamic parameters of the motion such as the joint velocities [OM01, AFO05] or the momentum [KLK04, KHL05]. Creating these kinds of changes results in character motion that gives the impression of responding *physically* to the impact. Researchers have recently coupled these changes with transitions to new motion sequences following the impact in order to capture more *stylized* and more *sustained* response behaviors [Man, ZMCF05, AFO05, KHL05].

In a similar fashion, after the impact, we include both a simulated reaction and the potential for transitioning to a new motion behavior. However, in comparison to all of these previous examples, our effort is unique in that we emphasize the preparation portion of the response. To the best of our knowledge, ours is the first work which addresses anticipation response in a thorough and explicit manner.

2.1.2 Motion Editing for Balance

In order to generate believable anticipatory movement, the character must obey certain physical laws, mainly balance. We edit the character’s motion leading up to the impact while taking balance into account. Similar to our approach, a small group of researchers have presented method for maintaining or correcting balance by directly modifying the motion data [BMT95, BMT96, TySK00, NF04, KLK04].

The advantage of these techniques over physically based ones is that they do not require a dynamic balance controller, which is generally more complex than a purely kinematic balance approach such as the one we describe. Instead they provide a means for controlling the appearance of balance by ensuring that computed parameters such as the center of mass (COM) and zero moment point (ZMP) remain plausible. The balance problem is then generally posed as a constraint optimization problem which can be solved using an inverse kinematic-like approach that takes into account the mass displaced while adjusting joint angles [BMT95, BMT96, NF04] or by making adjustments to the entire body to correct balance while minimizing changes to the original motion [TySK00, KLK04]. Coincidentally or not, these two types of solutions also reveal a clean division between controlling balance with the COM or the ZMP.

We found through experiments that the ZMP, as defined in [TySK00], and the COM produce very similar values across the examples in our database, including standing responses to impacts. Therefore, for ease of implementation and to

maximize our control over the process, we devise a technique somewhat similar to the first group, and use the COM to adjust balance.

2.1.3 Creating Poses Procedurally

One final related area in computer animation is highlighted in the expressive stance research of Neff and Fiume [NF04]. This work is similar to our own in that it focuses on the procedural creation of a pose (they call a stance) that is derived from combinations of rules (they call shape sets). For example, Neff and Fiume combine a reach rule with a parameter that ranges from interest to repulsion and that is encapsulated in the stance of the character. The power of creating complex character poses from simple components is appealing because many animations can be produced once the rule set is in place. We note that developing the primitive components of the rules does require a (somewhat) skilled animator but once these rules are in place, the many ways in which they can be combined make the benefit worth the overhead.

In comparison to Neff and Fiume’s work, there are a few specific details that make our work different. First, we focus on a multilayered system for response. Even though this is a more focused domain, we have taken it further by using it to build a complete response system. Also, we augment the generation process by building a model of the threat and parameterizing the rules using the model. Finally, where they investigated stance from the perspective of theater and dance performance, we focus on reflexive and low-level cognitive behaviors surrounding

anticipatory response, and thus draw from psychological studies associated with such activities. We describe our findings from these studies in the following section.

2.2 Anticipation in Psychology

Researchers in psychology have studied defensive movements for many years. Psychologists are generally interested in understanding the brain control mechanisms associated with such motions. However, many of the studies involving both monkeys and humans also describe typical movements associated with defensive behavior. For example, in recent work, Cooke et al. [CG03] performed experiments on monkeys to compare the defensive reactions caused by electrical stimulation to cortical areas with those reactions caused by an external air puff. The 'threat' was simulated as a puff or burst of air directed at the monkey. The authors qualitatively describe six movements that were evoked by the air puff. Among these six were three that are of particular interest when preparing the entire body for impact: shoulder shrug, retraction of the head from the direction of the impact, and arm movements which depend on the location of the puff. These movements included bringing the hand into the upper area near the head when the puff was directed at the head and bringing the elbow to the torso when the puff was directed at the side of the torso.

Further, in the same study, the researchers observed that responses included an initial spike of muscle activity and then a sustained muscle response. The ini-

tial spike was attributed to a startle reflex, commonly seen in infants. Other researchers have also described the startle response as being relatively insensitive to stimulus type and generally bilaterally symmetric [LH39, YSF02]. The sustained response is comprised of a set of spatially directed movements and can involve ducking, withdrawing from the direction of the stimulus, navigational veering during locomotion, and blocking the object with a body part (such as the forearm) to protect another part (such as the ribcage) [KC92, KDRD92, LH39, Sch65].

In our work, we build upon these findings as the basis for a set of rules for creating movements in anticipation of impacts. Based upon our survey, our implementation involves rules for recoil (shoulder shrug and arm retraction), blocking, leaning, and turning away. In addition, we include two distinct phases, one for the startle response and one for the sustained response.

A study by Li and Laurent [LL01] reported findings which involved the response of individuals to impending collisions to the head from a ball with various velocities and eccentricities. Of particular interest in this work are the findings on how the velocity of the threat affects both the time when the participants began to react and the speed of their movement. They found that the direction and the speed of the velocity each affect the reaction differently. The participants acted sooner when the angle of eccentricity (Figure 2.1) was larger. Large angles of eccentricity also caused the participants to overcompensate and react more strongly than they did from the same threat at smaller angles. Defensive movement speed of the participants did not vary with angle, however, it did vary with the speed of the ball. The speed affected both the maximum movement

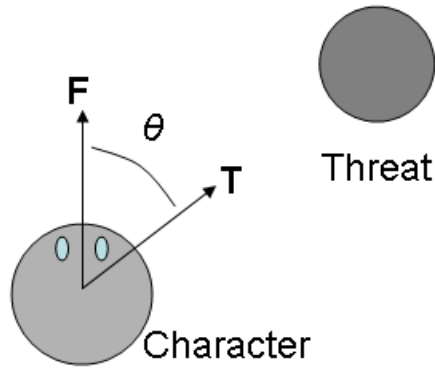


Figure 2.1: Eccentricity is the angle, θ , between the facing direction, **F**, and the direction to the threat, **T**.

speed as well as how quickly the participants reached that maximum speed. We build upon these findings to parameterize our protective responses based on the speed of the approaching threat.

Chapter 3. Computing the Pose

We derive our anticipation model from a survey of psychological studies of monkeys and humans [CG03, KDRD92] that observed and described reaction behaviors among their subjects. The particular findings of each study was discussed in the previous chapter. The reported behaviors define anticipation action primitives which we encode as rules that manipulate the character’s posture by leaning, turning, recoiling, and blocking. The inputs to the rules are the character’s current pose, the threat’s properties, and a set of rule control parameters.

We defined the rules primarily for impacts to the head where the threat’s approach is generally in front of the character. We have also extended the rules to cover impacts when the threat approaches the torso from the front and when the threat approaches the head from the rear. The implementation of these two extensions involves only slight modifications to the original rules. By changing only a few details, new or specialized behavior can be added. We first present the details of the primary case, including the rules and the optimization step. We conclude the chapter with a discussion of the extensions.

3.1 Threat Model

In order to generate reactions to an impending collision, we must define a model of the threat. We define a model with very few parameters which allows an animator to easily generate the desired response. The parameters of a threat are:

- the angle of approach (or normalized approach vector)
- the speed
- the size.

In our application, these parameters are selected by sliders. We use speed values in the range of 22.8 to 29.7 $\frac{m}{sec}$ which corresponds to the speeds of spiked volleyballs [FCC*05]. The size is an abstract quality that refers to either the volume, the mass, or both. We adjust the interpretation of size as necessary to achieve the desired artistic results for both the sustained pose and the dynamic response.

To visualize the threat's approach over time we also need to know the position and velocity when it is perceived by the character. To allow control of the position and velocity when the collision occurs, we use a reverse physical simulation to compute the starting properties. We initialize the position and velocity with the desired collision-time values: a location on character's body in global space (the head in most examples) for position and the combination of the speed and approach angle for velocity. We also select an amount of time until the collision will occur. Starting at the point of impact, we step backwards in time,

recomputing the position and velocity, until the perception time is reached. The alternative approach would be to initialize the threat’s position and velocity with the perception-time values. This would require guessing the values that would cause a collision but could be useful if the application doesn’t require every threat to collide with the character.

3.2 Defining the Rules

Given the threat model parameters and a current pose from some motion, the rules compute a basic version of the anticipation pose. An optimization is applied to this basic pose to enforce certain desired properties. This section discusses the motivation of each rule as well as the implementation details. We present the rules in the order they are to be applied, which is necessary to achieve the intended results. *Lean away* must be used before *turn away* due to the order of Euler angle rotations with a moving coordinate system. *Recoil* is flexible in that it could be applied anytime before block irrespective of lean away and turn away. *Block* must be applied last because if the character blocked before turning away, for instance, the turn would move the block to the wrong location.

Our implementation uses a character with a coordinate frame where the Z-axis is up, the X-axis is the facing direction, and the Y-axis extends to the character’s left side. A hierarchal skeleton is rooted at the pelvis and is shown in Figure 3.1. The character’s arms hang vertically in the unrotated identity pose. The elbows and knee are hinge joints but all other joints have three degrees of freedom.

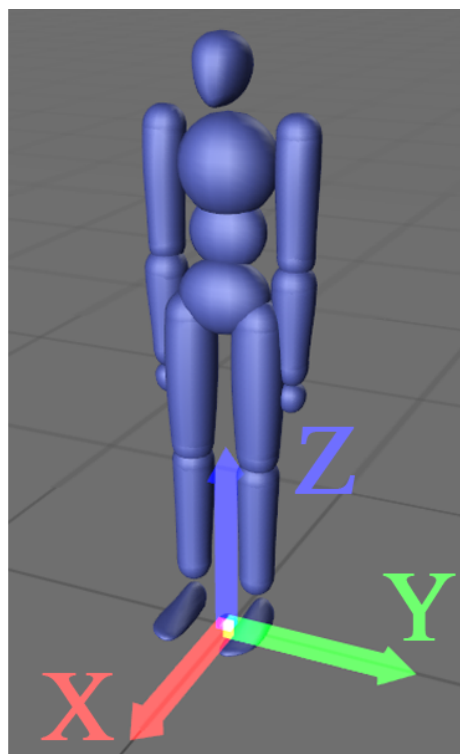


Figure 3.1: *left*: The hierarchy of joints forming the character’s skeleton. *right*: The character’s identity pose and the global coordinate frame.

<i>Parameter Name</i>	<i>Used By Rule</i>	<i>Initialized By</i>
Lean	Lean Away	Threat Speed
Turn	Turn Away	Threat Size
Recoil	Recoil	
Reach Left/Right	Block	
Swivel Left/Right	Block	

Table 3.1: Rule control parameters are initialized by threat model parameters.

3.2.1 Rule Control Parameters

Rule control parameters in the range of 0 to 1 are used to control the influence of each rule. For instance, a large turn away parameter and a small lean away parameter will cause the character to turn a lot but lean only a little. The properties of the threat determine the default control parameter values. This initialization step occurs when a threat is detected, before the rules are run. To compute each control's value, we simply normalize the threat property that is associated with that control. For example, if a system used a threat size in the range of 10 to 20 $\frac{m}{sec}$ and the character perceived the threat at 15 $\frac{m}{sec}$, the Turn, Recoil, Reach, and Swivel controls would be initialized to $0.5 = \frac{(15-10)}{(20-10)}$. The Lean control is computed similarly but is based on the threat's speed. Table 3.1 lists the controls and the threat properties used to initialize each one.

Figure 3.2 shows the effects of each of the threat properties on the rules. Note that this really means the control parameters are initialized differently and in turn vary the responses of the rules. The application of the controls and their effects on the rules is covered in the following sections.

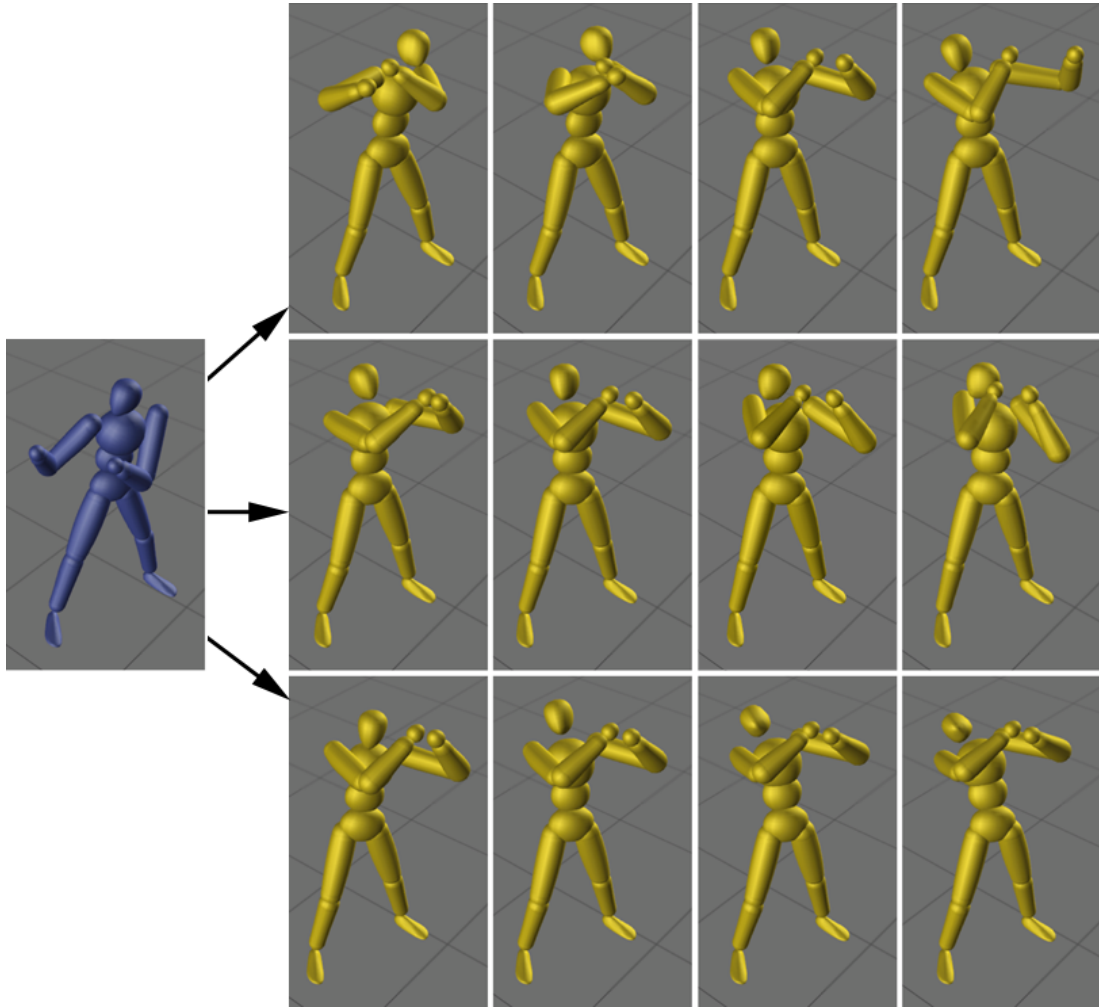


Figure 3.2: Changes in anticipation as the threat changes. The leftmost image in the figure shows the originating motion capture pose. The three rows to the right represent poses generated by the rules as threat properties change, one per row. The first row shows the threat's position changing from the character's right to left. The second row shows the threat's size varying from small to large. The third row shows the speed varying from slow to fast.

Lean and Turn Away			
<i>Joint</i>	<i>X</i>	<i>Y</i>	<i>Z</i>
neck	50	50	65
back	25	25	8
waist	10	10	12
pelvis	15	15	20

Recoil and Startle			
<i>Joint</i>	<i>X</i>	<i>Y</i>	<i>Z</i>
clavicle	35	25	0
shoulder	30	0	0
elbow	0	150	0

Table 3.2: Maximum rotation values (degrees). The rules choose angles between 0 and the maximum value to rotate the joints.

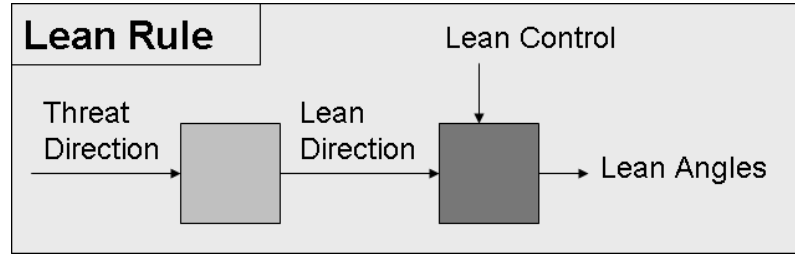


Figure 3.3: Input/output diagram for the lean rule.

3.2.2 Lean Away

The lean away rule (Figure 3.3) causes the character to lean in the direction the threat is traveling by rotating about the horizontal axes (X and Y). It therefore increases the distance the threat must travel before collision. We start by finding the direction, \vec{D} , with respect to the character's root, from the threat's perceived position to its target. After normalizing \vec{D} , \vec{D}_x is used to scale the maximum Y-axis rotation angle and $-\vec{D}_y$ is used to scale the maximum X-axis rotation angle

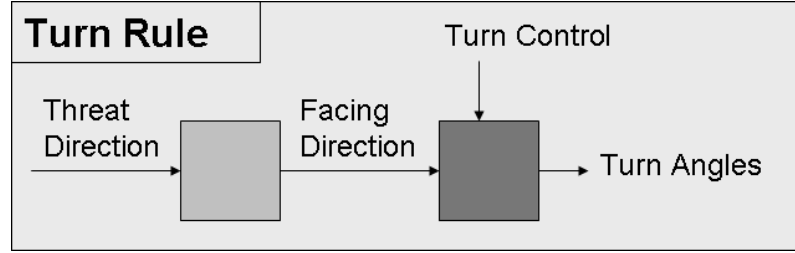


Figure 3.4: Input/output diagram for the turn rule.

(see Table 3.2). With a normalized direction, \vec{D}_x and \vec{D}_y are, of course, in the range of -1 to 1 so the two lean angles are each between 0 and the corresponding maximum value from the table, in the positive or negative direction. We finally use the lean control parameter (initialized by the threat's speed) to scale down these angles so that a slow moving threat will yield a smaller reaction than a fast moving threat.

To clarify with examples, a threat traveling in the negative X direction (approaching from the front) will cause the character to lean backward (rotation about the Y-axis). Similarly, a threat traveling in the negative Y direction (approaching from the left side) will cause the character to lean to the right (rotation about the X-axis).

3.2.3 Turn Away

After leaning, we employ the turn away rule (Figure 3.4) to cause the character to rotate about the vertical axis (Z) with the goal of orienting the back of its head towards the threat. This is motivated by the idea that it is less damaging to be

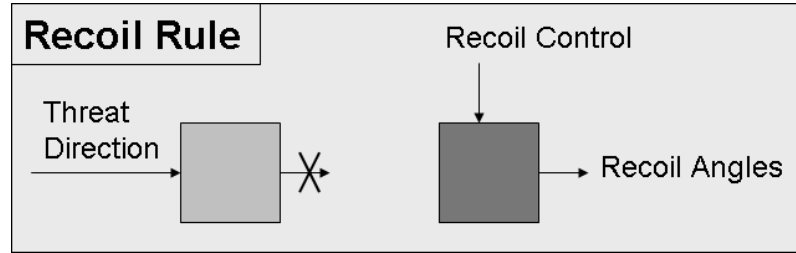


Figure 3.5: Input/output diagram for the recoil rule.

struck in the back of the head than in the face. King et al. [KDRD92] describe the behavior as generally ranging from a small contralateral turn of the head to a large contralateral pivoted movement involving both the head and upper body that could create as much as a 45 degree pivot.

We begin by finding the angle necessary to meet the goal of facing the back of the head towards the threat. We then rotate the neck up to its maximum angle (Table 3.2) and rotate the torso joints according to the remaining angle. The torso rotation, if any, is distributed between the back, waist, and pelvis at 50%, 30%, and 20% respectively, up the the maximum values. We scale all angles by the turn control parameter (initialized by the threat’s size) to scale down the rotation for small threats. The direction of the rotation depends on the threat’s approach; if the threat approaches from the left, the character turns right.

3.2.4 Recoil

The recoil rule (Figure 3.5) “buries” the head into the shoulders by rotating the clavicle joints up. This helps to protect the neck, a vital and sensitive area.

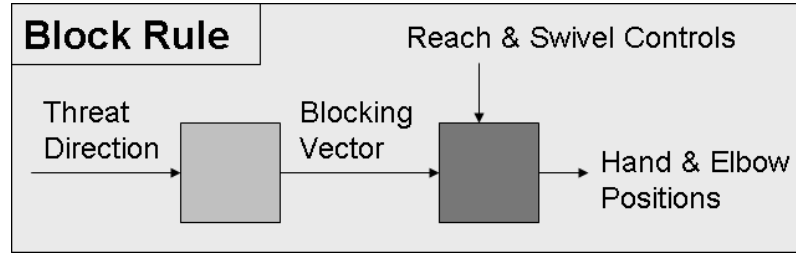


Figure 3.6: Input/output diagram for the block rule.

The recoil control parameter selects angles for the clavicle joints between 0 and the maximums from Table 3.2. Therefore the character recoils more from large threats than from small ones.

The recoil rule can also be used to make the character “smaller” by retracting the arms to the front of the chest. However, in our implementation, recoiling the shoulder and elbow joints has no effect on the final pose because those joints are controlled by the block rule.

For a more in-depth discussion of recoil behaviors, we refer the reader to the expressive stance work by Neff and Fiume [NF04].

3.2.5 Block

The psychological studies show that after the startle reflex, the arms generally move quickly to block the impending threat [LH39, Sch65]. The fourth and final rule accomplishes the blocking action (Figure 3.6). Cooke et al. [CG03] describe the movement to be primarily toward a position which guarded or blocked the threat; a puff of air directed at a monkey in their study.

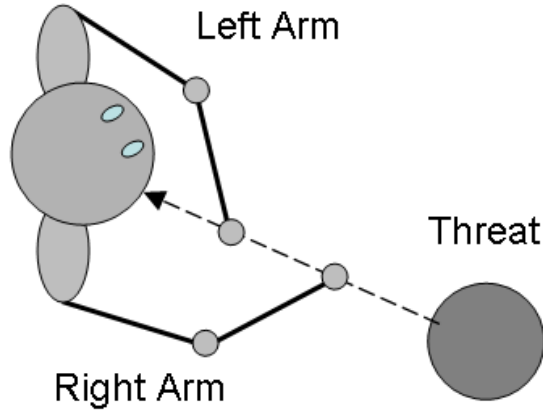


Figure 3.7: Top-down view of the character reaching out to block the threat. Both arms reach the same distance from their respective shoulder. The threat is on the right side so the right hand is closer to it.

We define a blocking posture for the arms as one in which the hands are positioned on the approach vector the threat, between the threat and the body. We then define a range of reach distances (0.3 to 0.9 *meters*) such that the character will reach further for small threats than for large threats. This has the effect making the pose “smaller” when the threat is large, thereby incorporating recoil properties in the block rule. The reach control parameters are initialized by the threat’s size and control the reach distance. The same initial parameter value is used for each arm but the distances are measured from the respective shoulder, so when the threat is to the character’s right side, the right hand’s blocking position is naturally closer than the left hand’s and vice versa, as illustrated in Figure 3.7.

The approach vector and reach distances define the positions for each hand. Two separate inverse kinematic problems are solved to achieve the blocking pos-

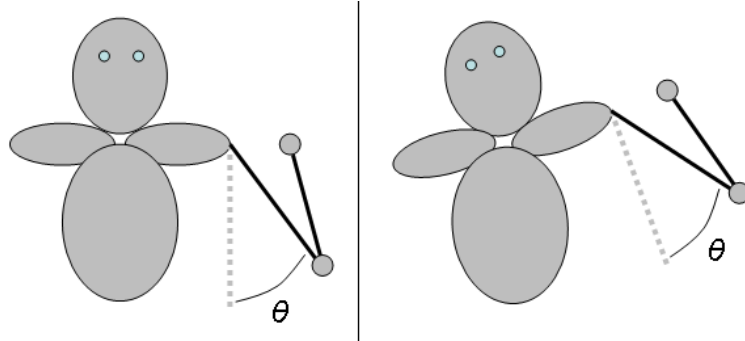


Figure 3.8: The elbow position is set using a swivel angle. The angle is with respect to the shoulder’s *down* direction, illustrated by the dashed line. Positive angles rotate the elbow away from the body.

tures of the arms. We use the analytical solver, IKAN [TGB00], which is applicable to simple joint structures such as an arm or a leg which have three joints: a hinge joint in the middle and ball-and-socket joints on the ends. An analytical solver uses a closed form solution and is therefore faster than numerical methods.

With the shoulder and wrist at fixed locations, the elbow is free to sweep out a circle on a plane with normal vector equal to the vector from the shoulder to the wrist. IKAN allows this “swivel” angle to be specified and so we introduce the swivel control parameter (initialized by the threat’s size). This control selects an angle in the range of 10 to 70 degrees that rotates the elbow away from the “down” direction (defined relative to the shoulder). The elbows rotate so that they swing away from the body as in Figure 3.8.

Coupling the two blocking control parameters produces a range of responses based on the threat. For a large threat, the wrists and elbows to stay close to the body in a tightly protective pose. For a small threat, the wrists reach out to

knock it away and the elbows rotate outwards in what appears to be a natural, comfortable manner.

3.2.6 Startle

The rules as described define the basic properties of a protective posture referred to as the sustained anticipation pose. Several researchers [CG03, LH39, YSF02] have also reported a startle response that occurs before the sustained response. It is described as a short latency, bilaterally symmetric response to intense stimuli that is thought to put the body in an initial protective pose. Cooke et al. report that the startle response consisted of a shoulder shrug and a turn of the head from the direction of stimuli. Retraction of the arms is also generally reported.

Our startle rule incorporates these described behaviors by defining a maximum recoiled pose using the angles from the lower portion of Table 3.2. Then we compute the actual startle pose as a spherical linear interpolation between the fully recoiled pose and a pose from the original animation with a slight look ahead from the current time and the using weights of 0.2 and 0.8 respectively. Refer to the section on timing (Section 4.2) for a discussion of the look ahead. In addition to the shoulder and arm reaction, we turn the head 40% of the way *towards* the threat to give the appearance of perception. In this manner, the character initially “flinches” when the threat is perceived and then proceeds with the more cerebral sustained defensive action.

Due to the short duration and reflexive nature of the startle response, we

choose not to consider balance in its computation. In some cases, startle is inappropriate and even distracting so we present it as an option.

3.3 Static Balance

The rules define the protective properties of the startle and sustained responses but may leave the character in a pose that is statically unbalanced. We say a pose is statically unbalanced when the ground plane projection of the center of mass (COM) falls outside the support polygon. The support polygon is the convex hull of the contact points between the body and the ground (generally the feet). The COM is computed as a weighted average of the COMs of the individual body parts

$$COM = \sum_{i=1}^n \left(\frac{w_i}{\sum_{j=1}^n w_j} COM_i \right) \quad (3.1)$$

where each weight, w_i , is the mass of body part i and COM_i is the position of that part's COM. The computed COM position is then projected to the ground plane.

Besides requiring static balance we also want the character to brace for the impact. Motivated by the need to remain balanced after the collision, we adjust the COM towards the threat, expecting the collision to push the character back. The first component of a braced, ready stance is for both feet to be planted on the ground. When only one foot is on the ground, the support polygon is very small and the character could easily be knocked over.

At the time of perception, if only one foot is planted, we look ahead some

small time in the animation and move the pelvis and unplanted foot accordingly. Then the unplanted foot is placed at the nearest reachable ground location using IK. If the foot cannot reach the ground because the hip is too high, we skip the IK step. Note that moving the pelvis also requires solving an IK problem for the *planted* foot to prevent it from moving.

Once both feet are planted, we run the rules to create the basic anticipation pose. The final step is a quasi-Newton optimization that moves the character's COM to the desired position, thereby balancing and bracing for impact. The effect of the optimization is clearly shown by the change in pose and COM in each image of Figure 3.9.

The primary goal of the optimization is to move the current COM, COM_{cur} , to the desired position, COM_{des} . COM_{cur} is initialized by Equation 3.1 after the rules are executed. COM_{des} is computed as some distance (10 cm) from COM_{cur} towards the threat. The exact distance could depend on some property of the threat or the character's motion for a more dynamic response. Given these two 2D locations, the optimizer preforms a search to minimize the balance error:

$$err_{bal} = \sqrt{(COM_{des_x} - COM_{cur_x})^2 + (COM_{des_y} - COM_{cur_y})^2}. \quad (3.2)$$

Our system is unique in that it treats the legs as two IK chains rooted at the hips and runs the search to determine the optimal pelvis position while preventing the feet from moving.

It should be noted that the COM_{des} will not necessarily be within the support

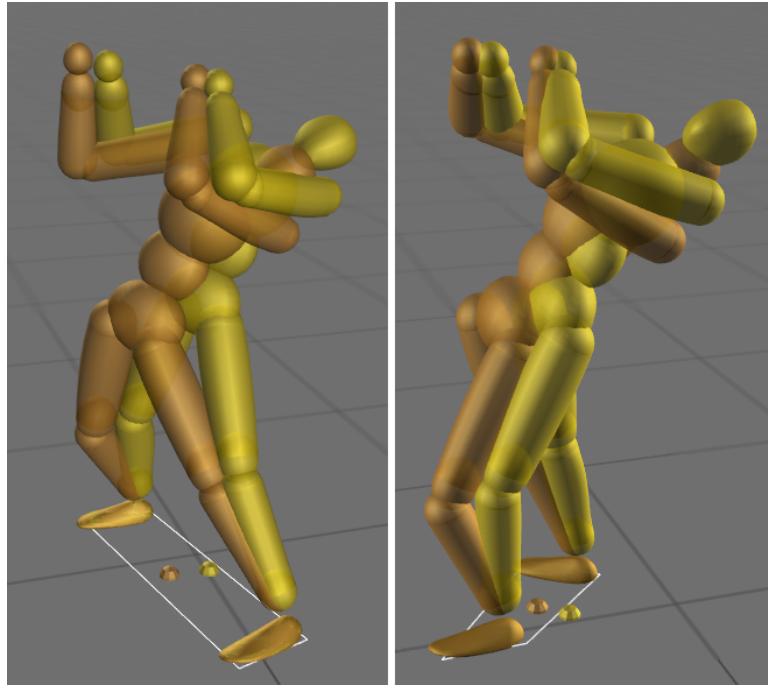


Figure 3.9: Two before and after comparisons of static balance. In each image, the yellow pose represents the result of the rules before optimization while the orange pose represents the final sustained pose after optimization. The COM and support polygon are also shown. *left:* The yellow pose was already in balance and the optimization braced for impact. *right:* The yellow pose was out of balance and the optimization both balanced and braced for impact.

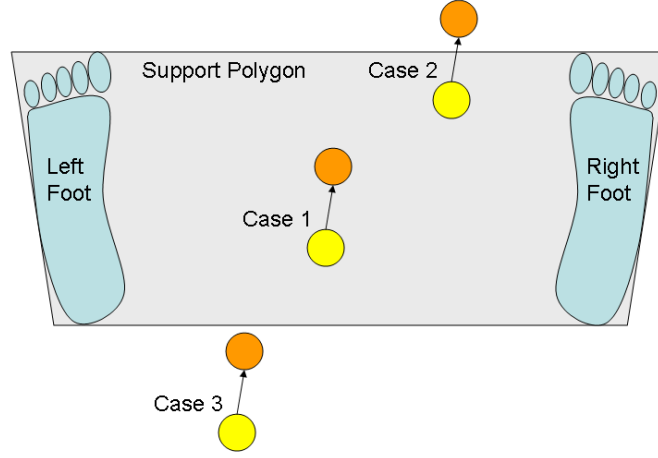


Figure 3.10: Three cases of balancing and bracing. The circles represent current and desired COM locations as yellow and orange, respectively. Case 1 occurs when the desired location is also balanced. Case 2 is out of balance to brace for impact. Case 3 was initially too far out of balance to fully correct the problem.

polygon and therefore the sustained anticipation pose will not necessarily be balanced. Figure 3.10 illustrates three possible cases with yellow and orange circles representing COM_{cur} and COM_{des} , respectively. Case 1 occurs when the character was initially in balance and remains in balance after bracing for the impact. Case 2 may happen if the character goes out of balance when bracing, which is acceptable because the threat is expected to push the character back into balance again. Case 3 could occur if the character was very far out of balance initially and bracing doesn't fix the problem. It would be possible to avoid this undesirable case by choosing a new COM_{des} location that would lead to a balanced pose. Our reason for not doing so is two-fold. Firstly, we do not want the defensive motion to differ greatly from the original motion the character was following. If that motion was far out of balance (which it probably

was in this case) a large modification would be required to achieve balance. Large changes in *COM* would not reflect the character’s momentum and would appear awkward. Secondly, this case never arose in practice with our input motions and the proposed rules.

It is important for the balance technique to integrate well with the rules so that protective properties of the pose are not lost during optimization. We therefore introduce additional terms to measure the error between the current optimized pose and the unbalanced sustained pose. These terms measure properties of the rules: hand positions, orientation of torso body parts, and pelvis height. Foot skate may occur if the pelvis is moved too far from the position of a planted foot so the error includes a measurement of the differences in foot positions between the two poses. Each term is computed using either the Euclidean distance or the quaternion distance, as appropriate, and the results are summed, yielding the total error:

$$err_{total} = err_{bal} + err_{handPos} + err_{torsoOrient} + err_{pelvisHgt} + err_{footPos}. \quad (3.3)$$

The optimization minimizes the error by manipulating a limited set of ten degrees of freedom: the pelvis position and the seven control parameters. We use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, as implemented by Numerical Recipes [PTVF94], which we found to generally converge on a solution in only a few iterations. This algorithm requires computing derivatives of the DOF vector. We compute finite differences with the midpoint rule using an

offset of 0.01 in each direction, recomputing the error after each individual adjustment. Every time a DOF is changed, either from an iteration of the optimizer or during the derivative computation, we start over with the original animation pose, move the pelvis, plant the feet, and execute the rules again using the modified control parameters. Allowing the control parameters to change is a tradeoff between balancing and protecting. The optimal solution may require a change in the protective posture. The error measurements prevent this change from being drastic.

3.4 Extensions

The rules and optimization as presented above are applicable to cases where the threat is approaching the character’s head from a position generally in front of the character. While this alone allows a wide range of dynamic interactions, we provide simple extensions which greatly expand the application of our algorithm.

The first extension covers the case where the threat approaches the character’s head from the rear. This extension is necessary because the original blocking rule may not produce realistic arm postures when the desired hand positions are behind the shoulders. We implicitly enforce joint limits by only requesting IK solutions that are feasible. After the turn rule is applied, we test to see if the threat lies in front of or behind the shoulders, using the dot product of the direction to the threat and the local X-axis of the unrotated shoulder. If the threat is in front of the shoulders, we apply the original blocking rule as presented.

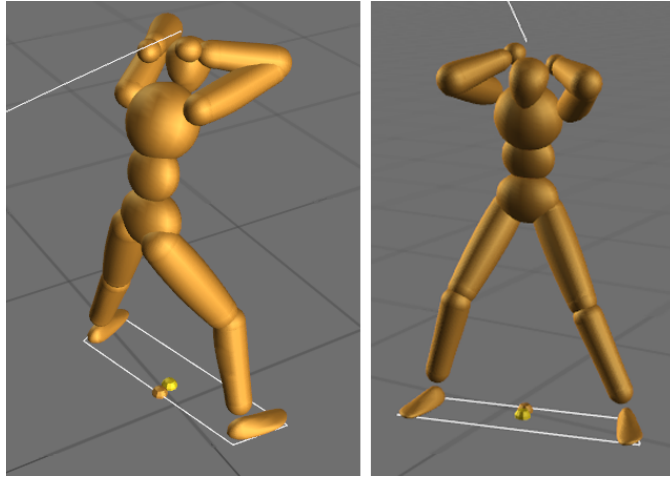


Figure 3.11: Two views of a threat approaching character's head from behind.

However, if it is behind the shoulders, we choose hand positions behind the head and elbow swivel angles that cause the arms to wrap around the sides of the head, as illustrated by Figure 3.11. We do not take the threat's approach vector into account when choosing the hand positions. This reflects the character's lack of precise knowledge about a threat that cannot be seen. With this extension in place, we can generate protective poses in response to threats approaching the character's head from anywhere on a full 360° surrounding circle.

The second extension includes the character's pelvis as a possible target area for threats approaching from the front. Two modifications are applied to the original implementation. We use shorter reach distances (0.4 to 0.5 *meters*) to keep the arms closer to the torso. We also change the lean rule by using only the pelvis lean and negating the angles. Thus the upper body leans towards the threat instead of away from it. The idea is that moving the upper body towards

the threat will cause the pelvis to shift backwards during the balancing step. Figure 3.12 illustrates the difference in anticipation posture when the threat is approaching the pelvis instead of the head. This extension is less applicable to threats from the rear because the blocking component still protects behind the head as computed by the first extension. However, this could be considered the desired behavior.

With reactions defined for threats approaching the head or pelvis, we can interpolate to prepare for threats anywhere in between. The Euler distances from the threat's target point to the head, D_{head} , and to the pelvis, D_{pelvis} , give weights for the defined response sets, W_i , computed as

$$W_{head} = \frac{D_{total} - D_{head}}{D_{total}} \quad (3.4)$$

for the head and similarly for W_{pelvis} , where D_{total} is the sum of all D_i , in this case, $D_{total} = D_{head} + D_{pelvis}$. We then use these weights to interpolate the joint angles for the lean rule, and the reach distances and swivel angles for the blocking rule. Figure 3.12 also shows an example of an interpolated anticipation response.

These are merely two extensions that we have developed. It would be possible to generate many more, such as for impacts to the legs. Some systems may also require specialized rules to achieve particular preparation behaviors. The rule structure of our solution is flexible and easily allows customized implementations to be combined with our rules, or to replace them completely.

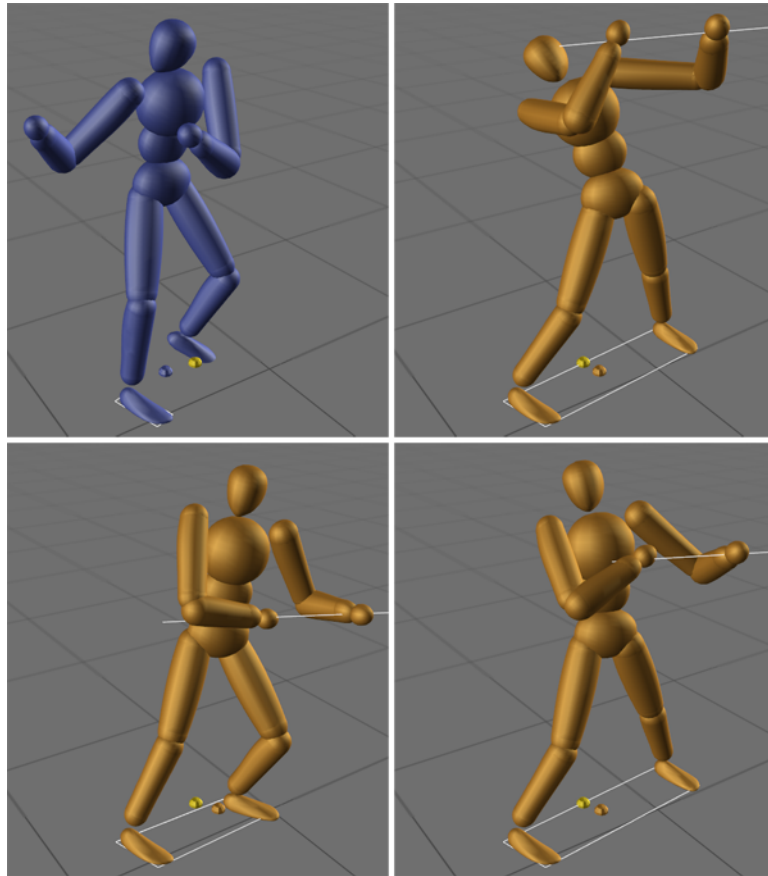


Figure 3.12: Torso impacts. The blue pose is the original motion capture pose from which the examples are generated. The three orange poses are examples of sustained anticipation responses for varying threat targets. (*top-right*) Responses to impacts to the head are defined by a set of rules. (*bottom-left*) Responses to impacts to the pelvis are defined by a similar set of rules. (*bottom-right*) Responses to impacts to the torso between the head and the pelvis are interpolated.

Chapter 4. Motion Generation

After computing the sustained pose and (optionally) the startle pose, the anticipatory motion can be generated. This requires careful consideration of the timing and balance for the motion to appear realistic. We also found it important to control the trajectories of the hands and the head. A basic implementation involves only simple transitional blends and is rather straightforward. However, our actual blending mechanism is rather complex in order to account for these requirements.

The next two sections discuss the various portions of the generated motion sequence and all the timing details of our system. Figure 4.1 is a schematic illustration of these topics. The chapter ends with a detailed description of our complex blending mechanism.

4.1 Motion Sequence

The generated motion consists of three distinct parts: the protective motion, the dynamic response, and the recovery motion. The concatenation of these parts yields a compelling animation of the character preparing for a collision, physically responding to it, and then returning to a meaningful action.

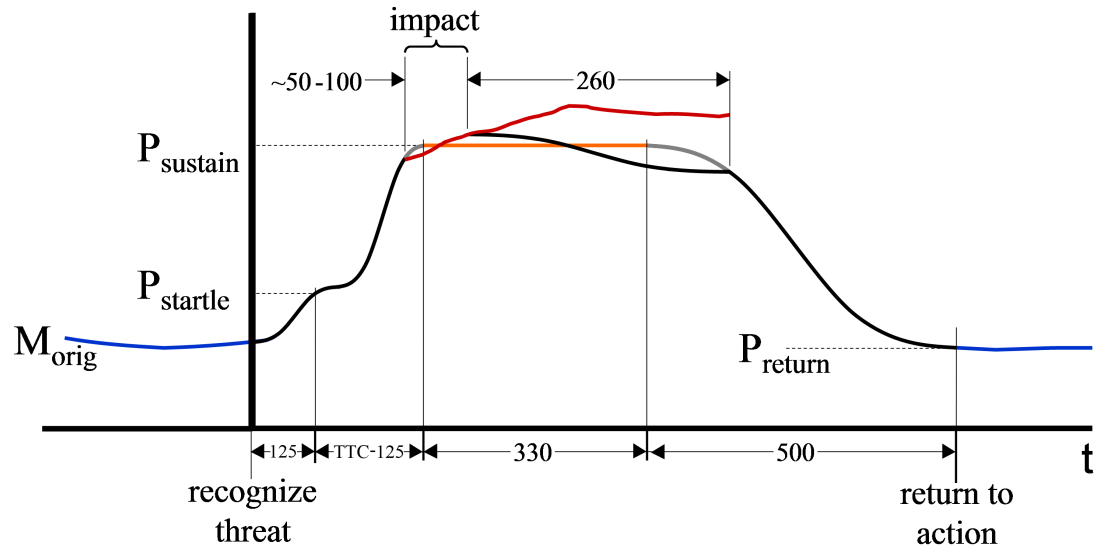


Figure 4.1: A timing schematic for our system. Blue lines represent motion capture animation. The red line represents the physical simulation. The orange line represents the sustained pose. Black lines illustrate blends. Grey lines show blends that are not part of the final motion. Dashed lines represent the positioning of the specific computed poses. Each region is labeled with the number of milliseconds it will occupy in the motion.

4.1.1 Protective Motion

The protective motion readies the character for impact by transitioning from the current pose of the animation to the sustained anticipation pose. This blend is illustrated in the schematic (Figure 4.1) as the first black line, between the blue line and the red line. The artist may choose to include the startle pose, in which case one blend transitions from the current pose to the startle pose and a second blend transitions from the startle pose to the sustained pose. The sustained pose and the protective motion are the primary contributions of this work. The other portions are included to make the solution complete.

4.1.2 Dynamic Response

When the collision occurs, we switch to a physical simulation to generate a realistic reaction. To accomplish this, we employ a version of Zordan et al.’s [ZMCF05] dynamic response technique which detects the collision between the geometric objects of the threat and body parts, integrates the forces, and manipulates the objects according to physical laws. This allows responses that are not pregenerated and is able to cover a very wide range of collisions. We allow the simulation to run for a short time before beginning the recovery motion.

4.1.3 Recovery Motion

The last portion of the motion occurs as the character regains control and returns to some meaningful motion. There are two possible choices at this stage. The character either returns to the animation that was active when the threat was perceived or enters a totally new animation that may be selected through a search (as done by [ZMCF05]) based on the simulated response and a database of example response motions. We left this choice to the animator but it could be easily made algorithmically based on the threat properties or on the degree of the physical reaction. In either case, the point where the character (re)enters the motion is labeled the “return pose,” or P_{return} in the schematic (Figure 4.1).

The generated motion for this phase is a two part transition from the dynamic response to the return pose. Part one is the only blend in our system that doesn’t transition between two static poses, but rather between two motion sequences. The initial sequence is the output of the physical simulation (red line in the schematic), which is initiated at the collision and continues to run through the end of this blend. The destination sequence is obtained by holding the sustained pose for $\frac{1}{3}$ sec (orange line) and then blending 25% of the way to the return pose. The second part of the recovery motion is the remaining 75% of the blend between the sustained and return poses. Concatenating the two parts yields the complete recovery motion.

4.2 Timing

Timing plays a critical role in anticipation and thus must be precisely controlled to generate realistic looking motion. Our implementation requires special consideration of timing because we have no perception model for detecting approaching threats and because we use a kinematic algorithm instead of a physical simulation to generate transition movement. To manually simulate perception, we choose the collision time properties for the threat and simulate backward to find the perception time properties, as discussed previously (Section 3.1).

The amount of time between perception of the threat and collision is called the time to contact (TTC). We compute the TTC based on Li and Laurent’s [LL01] study which reported that the time varies with the threat’s speed and approach angle. We use the threat’s speed to select a time in the range from 250 to 1000 *msec* using an inverse relationship so that the character responds earlier to faster threats and the TTC is larger. We then find the angle between the facing direction and the direction to the threat and use it to select a scale factor in the range from 1.0 to 1.2 such that for a constant speed, the TTC is higher when the threat approaches from the side. The TTC is therefore computed as:

$$TTC = [250 + 750 * (1 - \text{norm}(\text{speed}))] * [1.0 + 0.2 * (1 - \vec{facing} \bullet \vec{threat})] \quad (4.1)$$

where $\text{norm}(\text{speed})$ is the normalized speed of the threat in the range of 0 to 1 and \vec{facing} and \vec{threat} are the facing direction and direction from the head to the threat, respectively.

The TTC is the amount of time the character has to respond before the collision occurs. We use the TTC as the duration of the protective motion transitions, thereby imposing the assumption that the character reach the sustained pose “just in time.” If the blend were preformed faster, the character would reach the pose early and freeze awkwardly in that pose waiting for the collision. The collision detection is computed based on the geometry of the threat and the character and thus will not generally occur exactly at the estimated time. To ensure there is no pause, we introduce a slight amount of slack by extending the duration of the blend just a little. Therefore the character may not quite reach the sustained pose before the collision but will get close enough that the protective properties will be demonstrated.

Note that our “just in time” assumption is a contradiction to the reported findings of Li and Laurent. As Section 2.2 mentioned, the defensive movement speed of the character should not depend on the approach angle, even though the TTC does. Our implementation directly relates the speed of a movement to the duration of the blend producing that movement. This means that when the threat approaches from the side, the character should start acting sooner but still blend over the same amount of time as when the threat approaches from the front. Doing so would lead to the undesirable pause discussed in the previous paragraph. An alternative to our “just in time” solution for the pausing problem is a controller that would simulate motion to fill the pause, such as overshooting the pose and oscillating around it. Developing such a controller is left to future work.

When the threat is perceived and the rules are executed, we look ahead in time and move the pelvis and unplanted foot accordingly. This has the effect of maintaining the momentum of the body during the protective motion. There is a look ahead for the startle pose and another for the sustained pose. We found that $\frac{1}{12}$ sec for startle and $\frac{1}{6}$ sec for sustained worked well, with both of these being offsets from the time of perception. Following the dynamic response, if control returns to the original motion, the look ahead of $\frac{1}{6}$ msec gives the re-entry point. The times of $\frac{1}{12}$ and $\frac{1}{6}$ msec correspond to 10 and 20 samples from motion capture data sampled at a rate of 120 Hz.

The last set of times to consider are the durations of each portion of the generated motion. As discussed, the protective blend takes slightly longer than the TTC. The duration of the startle portion (if included) is fixed at $\frac{15}{120}$ sec with the remaining time used for the sustained portion. We allowed the dynamic response to run for 50 to 100 msec with the exact time determined by the animator. The two concatenated portions of the recovery motion cover 260 and 375 msec, respectively. If no collision occurs, the sustained pose is held for $\frac{1}{3}$ sec and the following recovery skips the physical response and associated blend.

4.3 Transitional Blend Mechanism

Transitioning between two poses or, similarly, between two motion sequences, with a linear interpolation (lerp) of the root's position and spherical linear interpolations (slerp) of the root's orientation and each joint's rotation is a straightfor-

ward process but introduces several undesirable artifacts. The two most obvious artifacts appear if the character becomes unbalanced or if the non-support foot moves while it is planted on the ground. We fix both of these problems through rebalancing, possibly introducing a wobbling motion to the head. Two possible artifacts involving the arms aren't as obvious and yet were found to distract from the realistic quality of the motion. A slerp blend of joint angles causes the hands to swing along an arc but a straight-line trajectory appears more directed and therefore is preferred. Also, the swivel angle of the elbows may change unpredictably, resulting in the elbows swinging in and out in an unnatural manner.

We propose a novel blending mechanism which accounts for all of these artifacts. To begin, we select the foot closest to the projected COM as the primary support foot. This foot becomes the root of a kinematic chain that branches out to the rest of the body. A blend (lerp/slerp) with an ease-in ease-out (EIEO) weighting for each blend parameter is computed as a first pass and balancing artifacts are removed later, in a second pass. We remove the artifacts related to arm movement, during the first pass. Instead of slerping the arm joint rotations, we treat each arm as an IK chain and, at each frame, solve for the position that moves the hand along a straight line between its positions in the starting and ending frames. Similarly, we compute the swivel angle of the elbow in the starting and ending frames and use interpolated values as the input to the IK solver to control the elbow positions.

A second pass ensures the character remains balanced throughout the blend. Similarly to (Section 3.3), at each frame of the blended motion we need to move

COM_{cur} to COM_{des} . In this case however, COM_{des} is an interpolated position along the straight line from the COMs of the starting and ending frames. We run an optimization, as before, to find the transformation that moves the pelvis and minimizes err_{bal} , which is computed by Equation 3.2 as the Euclidean distance between the current and target COM locations.

To compute the optimized pose, we treat the legs as two IK chains, rooted at the feet and ending at the hips. We use the IK solver, IKAN [TGB00], as before, to solve the two IK problems and move the pelvis according to the optimal transformation. A swivel angle is found to minimize the distance the knees traveled from their positions in the previous frame. Since the pelvis is higher than the upper body in the kinematic chain, moving the pelvis moves the entire upper body and thus may cause the head to wobble. To fix this problem, we include a third IK chain rooted at the waist and ending at the neck. Since IKAN is only applicable to arm and leg style chains (where the middle joint is a hinge), we temporarily reduce the back from a 3DOF joint to a hinge joint. The goal for this IK problem, at each frame, is to position the neck joint along a straight line between the starting and ending frame neck positions. The orientations of the feet and head are computed using a slerp at each frame.

This novel blend mechanism produces motion which remains balanced and transitions smoothly between the two poses or sequences.

Chapter 5. Results and Conclusions

Using our proposed solution, we were able to generate anticipatory motion for a wide variety of examples. We explore the range of capabilities of our system by varying several inputs:

- the motion sequence the character follows before collision,
- the frame within that motion when the threat is perceived,
- the option to return to the same motion or to search for a new motion after the collision,
- the option of including the startle actions,
- the threat’s parameters (direction/approach angle, size, and speed),
- the physical interpretations of the size and speed, and
- the duration of the dynamic response phase.

Figures 5.2 and 5.1 demonstrate results varying three of these options, including a case where a large impact caused dramatic motion. Figure 3.2 illustrated the change in sustained pose as the threat changed. The accompanying video demonstrates examples for many sets of varied inputs.

The space of all possible interactions between two actors in a dynamic environment is very large. This work focuses on the subspace where one actor is a human character and the other actor may be considered a threat and is on a collision course. Furthermore, we ignore the possible cases where the character

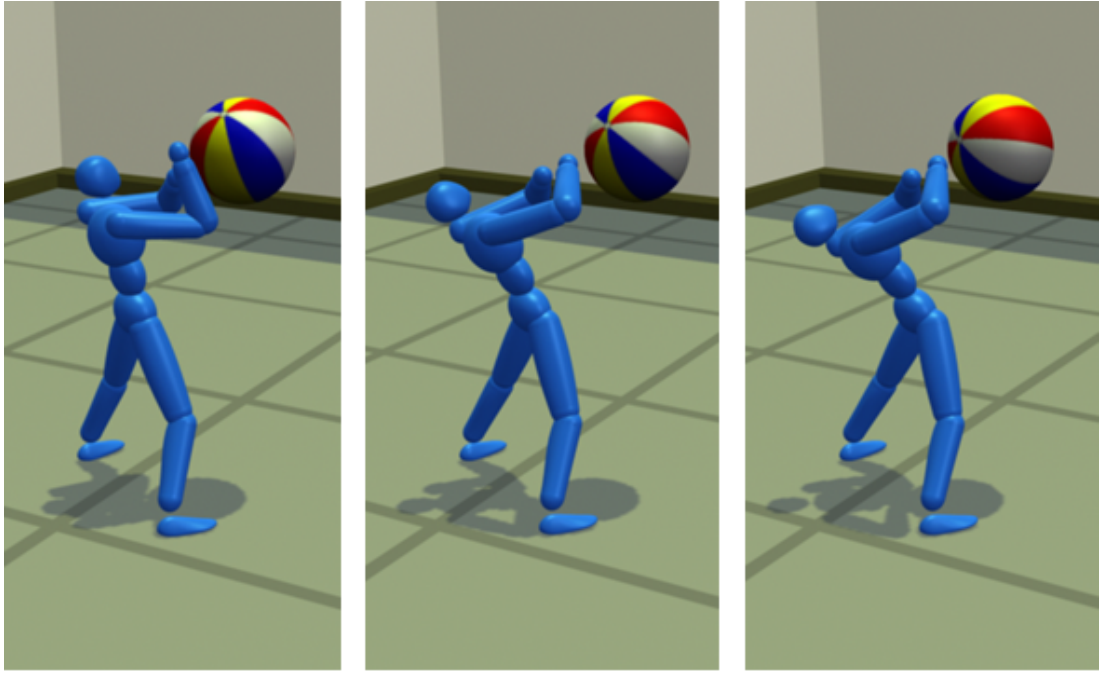


Figure 5.1: Speed variations. Sustained anticipation poses shown just prior to impact illustrate the effect the threat’s speed has on the pose as it increases from left to right.

may be able to dodge the threat, completely avoiding any collision. This suggests that the character was unaware of the threat until shortly before the collision, allowing just enough time for protection.

We have developed rules that are applicable when the threat will strike the character’s head (from any direction) or torso (from the front). We believe that similar rules could be created to handle any type of collision. For example, the character may prepare for a collision to the upper legs by shifting its weight to the leg that is further away and raising the closer leg. It may turn so that the threat will strike from the side instead of the front or back. The arms may be

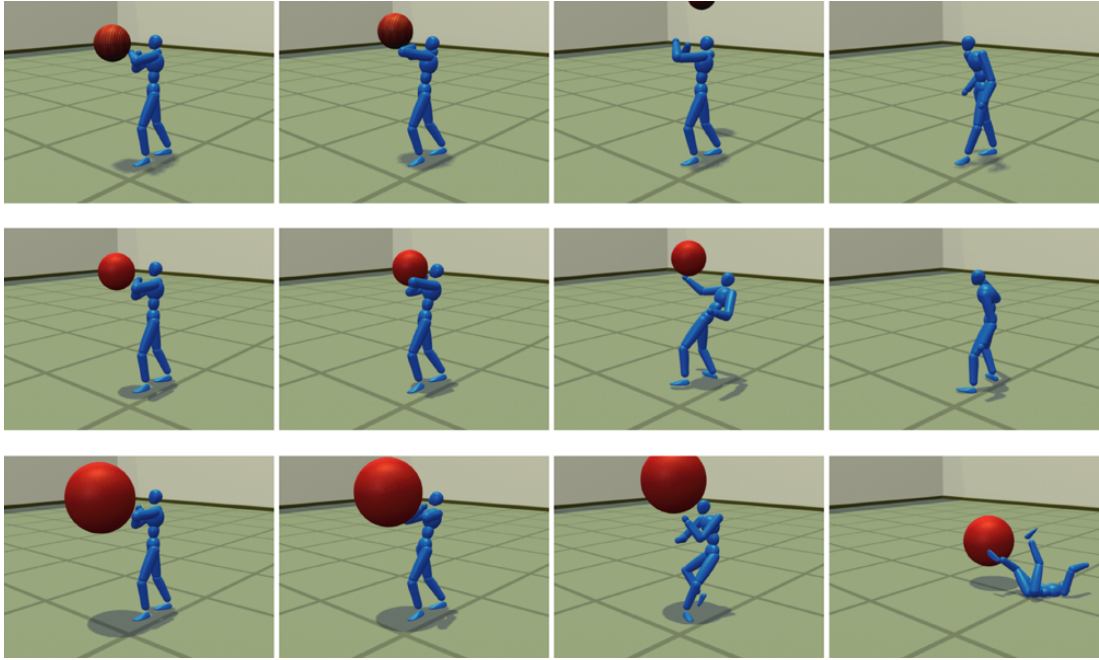


Figure 5.2: Three filmstrips show animations generated by our system. Here we demonstrate the result of returning to the original motion (top) or a new motion (middle and bottom) as well as the result of mild (middle) or extreme (bottom) physical threat properties.

used solely for balancing or they may be moved to block the threat if it deflects up towards the upper body. Although we strove for realism in all aspects of our implementation, this is not necessarily a requirement. Rules could be created to generate motion with a less-realistic, more-artistic feel.

This work could also be extended to account for characteristics of the character, such as age, physical fitness, and personality. Besides modifying the rules to account for these factors, it may also be important to replace our blending mechanism with a more physically simulated technique. Unlike in our solution, where the character always (nearly) reaches the protective pose, a character with

low mobility may only be able to partially achieve the pose before the collision occurs.

We have only considered medium sized, spherical threats (*eg.* volleyballs). We suspect that the responses to different types of threats, such as sticks, splashes of water, human or animal attacks, and even smaller spherical threats (*eg.* baseballs), may need to vary from our results to remain realistic. Further, we have only considered cases where the character was performing an upright motion (walking, fighting, etc.) with relatively low momentum before the impact. Such motions allow the freedom of modifications like leaning and planting the feet. If the character was initially sitting or running, realistic responses would likely vary.

5.1 Alternative Approaches

We considered several different solutions to the problem of anticipating impacts. One of these was a learning approach where the character was bombarded by a threat repeatedly and the “pain” was computed each time. Machine learning algorithms such as neural networks were used to learn the pose that would minimize the pain. This resulted in some acceptable solutions but many that didn’t appear humanlike even though the pain was minimized. Deriving a good measure of a pose suggested by the algorithm is not a simple matter, making this approach neither easily extensible nor flexible.

We also considered a data-driven approach where motion data would be collected, using a motion capture system, for several examples of anticipatory re-

actions. The example motions would be stored in a database along with the environmental parameters that they represented. Interpolations could be generated from the samples in the database using a nearest neighbor search on the parameters. We also considered using a similar technique to verify our results by capturing real motions and measuring the difference.

Collecting motion data for anticipatory actions is difficult because the action is tightly coupled with the threat. This implies that realistic actions cannot necessarily be obtained unless the threat actually collides with the capture subject. For substantially sized threat, bombarding a person repeatedly, or even once, could cause injury and is clearly not possible! To avoid this limitation, an actor could imagine the threat and “act out” the preparation but this gives no guarantee of realism. Another option is to throw a less dangerous object, such as a styrofoam ball, which the subject would think was actually dangerous. The first capture may be good but the subject’s new knowledge of the object would compromise further captures. It is impractical to capture a large set of motions when each subject can only produce one.

Due to these limitations, we chose to follow the procedural approach built upon the psychological literature.

5.2 Conclusion

In conclusion, we have presented an approach for generating reactionary motions including anticipation of and response from collisions with an approaching object.

Our approach is based on the definition of protective poses from actions studied and reported in the field of psychology. We produce and achieve these poses using a hybrid kinematic and dynamic technique that maintains physical constraints while remaining close to the underlying motion sequence.

Bibliography

- [AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Pushing people around. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 59–66.
- [ALP04] ABE Y., LIU C. K., POPOVIĆ Z.: Momentum-based parameterization of dynamic character motion. In *2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (jul 2004), pp. 173–182.
- [BMT95] BOULIC R., MAS R., THALMANN D.: Position control of the center of mass for articulated figures in multiple support. *Proc. 6th Eurographics Workshop on Animation and Simulation* (1995), 130–143.
- [BMT96] BOULIC R., MAS R., THALMANN D.: A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics* 20, 5 (1996), 693–701.
- [BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *Proceedings of ACM SIGGRAPH 95* (1995), Cook R., (Ed.), Computer Graphics Proceedings, Annual Conference Series, Addison Wesley, pp. 97–104.
- [CG03] COOKE D., GRAZIANO M.: Defensive movements evoked by air puff in monkeys. *Journal of Neurophysiology* 90, 1 (2003), 3317–3329.
- [FCC*05] FORTHOMME B., CROISIER J.-L., CICCARONE G., CRIELAARD J.-M., CLOES M.: Factors correlated with volleyball spike velocity. *American Journal of Sports Medicine* 33, 10 (2005), 1513–1519.
- [FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Com-*

- puter graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 251–260.
- [Gle98] GLEICHER M.: Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 33–42.
- [Gle01] GLEICHER M.: Comparing constraint-based motion editing methods. *Graph. Models* 63, 2 (2001), 107–134.
- [HPP05] HSU E., PULLI K., POPOVIĆ J.: Style translation for human motion. *ACM Trans. Graph.* 24, 3 (2005), 1082–1089.
- [KC92] KING S., COWEY A.: Defensive responses to looming visual stimuli in monkeys with unilateral striate cortex ablation. *Neuropsychologia* 30, 1 (1992), 1017–1024.
- [KDRD92] KING S., DYKEMAN C., REDGRAVE P., DEAN P.: Use of a distracting task to obtain defensive head movements to looming visual stimuli by human adults in a laboratory setting. *Perception* 21, 2 (1992), 245–259.
- [KHL05] KOMURA T., HO E. S., LAU R. W.: Animating reactive motion using momentum-based inverse kinematics. *Compute Animation and Virtual Worlds* 1, 16 (2005), 213–223.
- [KLK04] KOMURA T., LEUNG H., KUFFNER J.: Animating reactive motions for biped locomotion. In *Proc. ACM Symp. on Virtual Reality Software and Technology (VRST '04)* (2004).
- [LH39] LANDIS C., HUNT W.: *The Startle Pattern*. Farrar and Rinehart, 1939.
- [LHP05] LIU C. K., HERTZMANN A., POPOVIĆ Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics* 24, 3 (aug 2005), 1071–1081.
- [LL01] LI F.-X., LAURENT M.: Dodging a ball approaching on a collision path: Effects of eccentricity and velocity. *Ecological Psychology* 13, 1 (2001), 31–47.

- [LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of ACM SIGGRAPH 1999* (Los Angeles, 1999), Rockwood A., (Ed.), Addison Wesley Longman, pp. 39–48.
- [Man] MANDEL M.: Versatile and interactive virtual humans: Hybrid use of data-driven and dynamics-based motion synthesis. *Master's Thesis, Carnegie Mellon University*.
- [NF04] NEFF M., FIUME E.: Methods for exploring expressive stance. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2004), ACM Press, pp. 49–58.
- [OM01] OSHITA M., MAKINOCHI A.: A dynamic motion control technique for human-like articulated figures. *Computer Graphics Forum (Eurographics 2001)* 20, 3 (2001), 192–202.
- [PTVF94] PRESS W., TEUKOLSKY S., VETTERLING W., FLANNERY B.: *Numerical Recipes in C*. Cambridge University Press, New York, 1994.
- [PW99] POPOVIĆ Z., WITKIN A.: Physically based motion transformation. In *Proceedings of ACM SIGGRAPH 1999* (1999), pp. 11–20.
- [Sch65] SCHIFF W.: Perception of impending collision. *Psychological Monographs: General and Applied* 79, 11 (1965), 1–26.
- [TGB00] TOLANI D., GOSWAMI A., BADLER N. I.: Real-time inverse kinematics techniques for anthropomorphic limbs. *Graph. Models Image Process.* 62, 5 (2000), 353–388.
- [TySK00] TAK S., YOUNG SONG O., KO H.-S.: Motion balance filtering. *Computer Graphics Forum* 19, 3 (2000), 437–446.
- [WP95] WITKIN A., POPOVIĆ Z.: Motion warping. In *Proceedings of ACM SIGGRAPH 95* (1995), Cook R., (Ed.), Computer Graphics Proceedings, Annual Conference Series, Addison Wesley, pp. 105–108.

- [YSF02] YEOMANS J., SCOTT B., FRANKLAND P.: Tactile, acoustic, and vestibular systems sum to elicit the startle reflex. *Neuroscience Biobehavioral Reviews* 26, 1 (2002), 1–11.
- [ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (jul 2002), pp. 89–96.
- [ZMCF05] ZORDAN V. B., MAJKOWSKA A., CHIU B., FAST M.: Dynamic response for motion capture animation. *ACM Trans. Graph.* 24, 3 (2005), 697–701.