Adaptation of Performed Ballistic Motion

by

Adnan Sulejmanpašić

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY February 2004

© Massachusetts Institute of Technology 2004. All rights reserved.

Author ........

Department of Electrical Engineering and Computer Science
January 30, 2004

Certified by ........

Jovan Popović
Assistant Professor
Thesis Supervisor

Accepted by .............

Arthur C. Smith
Chairman, Department Committee on Graduate Students
Adaptation of Performed Ballistic Motion

by

Adnan Sulejmanpašić

Submitted to the Department of Electrical Engineering and Computer Science on January 30, 2004, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

Abstract

This thesis presents a method for adapting performed ballistic motions of a full human figure with many degrees of freedom by using an optimal trajectory formulation and the dynamics derived from first principles. Computation of the joints, torques, and reaction forces allows the application of a number of optimization criteria that result in creation of natural looking final motion. Alternatively, a reduced-order dynamics constraints can improve solution time by an order of magnitude and still retain the natural quality of the resulting motion in most adaptation scenarios. The adaptation method generates over twenty different adaptations from the original performances of a human jump and run.

Although these results demonstrate the robustness of this method for a full human figure motion adaptation, an automated skeleton simplification is also presented. Applying common model reduction techniques, such as principal and independent component analysis, to the original motion data yields a low-dimensional character representation of a given motion activity. While the reduced character configuration converges faster for some optimization formulations, the high-dimensional character optimization always produces more natural looking motions.

Thesis Supervisor: Jovan Popović
Title: Assistant Professor
Acknowledgments

First and foremost, I would like to thank my advisor, Jovan Popović for his support and encouragement throughout this project. He is at the same time one of the humblest and the smartest people I have ever met; I was lucky enough to have a chance to interact with and learn from him many things about computer graphics and computer animation, but also about life in general. Jovan is also a very good cook who prepares an excellent Balkan dish called “gibanica.”

I believe that the Graphics group in the Computer Science and Artificial Intelligence Laboratory at MIT is one of the most vibrant, intellectually charged, and funniest graduate student groups, which many universities would be more than grateful to have. I thank Daniel, Eugene, Eric, Peter, Fredo, Patrick, Barb, Bob, Rob, Sara, Soonmin, John, Wojciech, Kautz, Matthias, Tom, Tom, Barb, Manish, Ray, Hector, and all the French people that came and visited the lab over the past two years. Their humourous and witty personalities are only surpassed by their superb intellects and affinity for computer graphics.
## Contents

1 Introduction 13
  1.1 Adaptation ........................................ 14
  1.2 Model Reduction .................................. 16

2 Background 17
  2.1 Motion Synthesis .................................. 17
  2.2 Motion Editing .................................... 19
  2.3 Model Reduction .................................. 21

3 Adaptation 23
  3.1 Formulation ........................................ 24
  3.2 Dynamics Constraints ............................... 24
    3.2.1 Standard Formulation ......................... 25
    3.2.2 Reduced-Order Formulation .................... 25
    3.2.3 Discussion ................................... 26
  3.3 Kinematic Constraints .............................. 27
    3.3.1 Character Description ......................... 27
    3.3.2 Point Constraints .............................. 28
    3.3.3 Joint Limits .................................. 29

4 Numerical Solution 31
  4.1 Scaling ........................................... 32
  4.2 Initialization .................................... 33
4.3 Explicit Solution .................................................. 34
4.4 Implicit Solution .................................................. 35

5 Model Reduction ..................................................... 37
  5.1 Projection ...................................................... 37
  5.2 Dynamics Constraints ........................................ 40
  5.3 Kinematic Constraints and Objective Function ............ 40
  5.4 Statistical Analysis ............................................. 41
    5.4.1 Principal Component Analysis .......................... 41
    5.4.2 Independent Component Analysis ....................... 44

6 Results .......................................................... 47
  6.1 Initialization .................................................. 49
  6.2 Jumps .......................................................... 49
  6.3 Runs .......................................................... 50
  6.4 Limitations ..................................................... 51

7 Conclusion ......................................................... 67
List of Figures

3-1 Adaptation of a Human Broad Jump .................................... 23
3-2 Point and Joint Limit Constraint ....................................... 29
3-3 Human Figure Skeleton Configuration ............................... 30

5-1 Motion Synthesis Process ................................................. 38
5-2 Leg Joints Projection ................................................... 39
5-3 Broad Jump PCA Eigenvalues ........................................... 43

6-1 Person in a motion-capture suit ....................................... 48
6-2 Timing Results .............................................................. 49
6-3 Jump Initial Guess ...................................................... 53
6-4 Limp Arm ................................................................. 53
6-5 Briefcase ................................................................. 54
6-6 Briefcase and Limp Ankle ............................................. 54
6-7 Full Twist Left ............................................................ 55
6-8 Full Twist Right .......................................................... 55
6-9 Hop ........................................................................... 56
6-10 Hop and Limp Ankle .................................................. 56
6-11 Half Twist Right ......................................................... 57
6-12 Half Twist Left ........................................................... 57
6-13 Step Landing ............................................................. 58
6-14 Step Takeoff ............................................................. 58
6-15 Diagonal ................................................................. 59
6-16 Wide Landing ........................................................... 59
6-17  Step Takeoff and Landing ........................................ 60
6-18  Bar Grab .............................................................. 60
6-19  Diagonal PCA ......................................................... 61
6-20  Half Twist Right PCA ............................................... 61
6-21  Dunk ................................................................. 62
6-22  Dunk and Short Legs ............................................... 62
6-23  Dunk and Long Torso .............................................. 63
6-24  Cross Step ............................................................ 63
6-25  Wide Step ............................................................. 63
6-26  Run Initial Guess .................................................... 64
6-27  Bouncy Step .......................................................... 64
6-28  Skip Step .............................................................. 65
6-29  Hurdle ................................................................. 65
6-30  Cross Step PCA ....................................................... 66
6-31  Wide Step PCA ....................................................... 66
List of Tables

3.1 Human Body Mass Distribution ........................................ 28

6.1 Optimization Statistics .................................................. 52
Chapter 1

Introduction

The field of computer graphics experienced tremendous growth in the past couple of decades. Computer hardware advances and new graphics algorithms and techniques enabled the generation of virtual reality whose authenticity can fool the human eye. Modelling of complex geometric shapes and simulation of deformable objects, real-time rendering, image-based virtual environment construction, and animation of natural phenomena such as smoke or water are examples of modern computer graphics achievements.

The same techniques can be used to model a human shape; geometric modelling can produce realistic-looking human skin and its deformation. Accurate rendering of skin color can be achieved either by data-driven or complex physics-based lighting models. In addition to producing realistic images of the human shape, computer animation is focused on synthesizing believable human motion. Applications of automated motion generation are abundant; a number of entertainment related products, such as animated movies, special effects, or video games, require realistic looking human motion to produce high quality results.

The focus of this thesis is on generation of human motion. This remains an open problem in computer animation for two fundamental reasons. First, we are accustomed to observing people walk, run, or jump; therefore, we are highly sensitive to any limitations of synthesized motion. Second, the complex structure of a human body hinders our efforts to completely understand the complex biomechanical principles governing human motion. It is difficult to model every muscle and every nerve that affects the motion of various body
parts. By using simpler models we are invariably losing important information about the human figure and its motion.

Some of the standard techniques for generating human motions in computer graphics include kinematic techniques (e.g. key frames, inverse kinematics) and numerical simulation. While these approaches often yield impressive results, they also have significant deficiencies. Key frames require great artistic skill and are often a time consuming and arduous process. Inverse kinematics stumbles on underdetermined problems, which have more than one possible solution, while numerical simulation is very sensitive to careful tuning of the input motion parameters (joint and muscle values).

1.1 Adaptation

Rather than synthesizing motion from scratch, the approach explored in this thesis generates new human motions by adapting the input motion data of a performed ballistic motion. Motion data can come from a number of different sources, such as a motion-capture device, a professional animator’s drawing, or the output of a numerical simulation. The physical correctness of such data is crucial for successful adaptation of the original motion. The problem of motion adaptation is formulated using a method of spacetime constraints, which allows an intuitive description of the motion adaptation problems and works well for ballistic motions.

Ballistic motions such as jumps, runs, and other acrobatic maneuvers consist of flight periods during which performers are propelled by the force of gravity and momentum alone. In anticipation of such periods, performers execute specific actions to accomplish the desired outcome despite the limited command over motion in flight. For example, before a twisting jump, a performer bends down, bursts upward to generate the linear momentum that counteracts the force of gravity, and simultaneously spins his body to generate the angular momentum that twists the body in the air. The addition of a twist to a simple broad jump requires similar anticipation because only limited adjustments can be made in flight. Although kinematic techniques can aid in the adjustment process, they are not effective for exploring the confined space of dynamically realistic adaptations.
An optimal trajectory method, which selects an adaptation from physically valid trajectories, resolves these difficulties and enables automatic adaptation of ballistic motion. The method, introduced by Witkin and Kass as a method of spacetime constraints [36], benefits from restrictions placed by the dynamics of ballistic motion—the same restrictions that make manual adjustments particularly tedious. In the confined set of physically valid motions, the efficiency of motion, measured with the expended muscle power, muscle smoothness, or another function of internal torques, can identify a natural motion despite the large number of joints in a human figure. This variational approach, which adapts the entire motion simultaneously, accounts for the lack of control in flight by adjusting the motion at other times, when the appropriate control is available. As a result, a longer jump yields a deeper bend in the knees before the figure leaps from the ground. Although, in published work, the dynamics of a human figure were simplified before computing the optimal trajectories [30], this thesis describes an implementation of the original formulation [36], which successfully adapts motions of a full human figure with many (e.g. 42) degrees of freedom (DOF).

Instead of selecting the most efficient physically valid motion with the optimization of an explicit criterion, an optimal trajectory method can adapt a performed ballistic motion by choosing the physically valid motion that is close to the recorded performance. This proximate criterion retains the advantages of the explicit criterion and can be defined implicitly to enable reduction of the dynamics equations and efficient numerical solution of the adaptation problem. Empirically, the adaptation with the implicit criterion is faster than the adaptation with the explicit criterion by an order of magnitude.

To the best of my knowledge, no one has demonstrated that the dynamics of a human figure with many degrees of freedom can be scaled properly to enable reliable and efficient convergence of optimal trajectory methods. This simple observation enables an implementation of a general adaptation technique, which can enforce kinematic and dynamic constraints on joints, torques, and reaction forces. In addition, a number of efficiency improvements hinted at by previous researchers can be combined in a systematic fashion to

---

1The method becomes inappropriate for motions without any ballistic periods such as walking and reaching because the efficiency alone does not sufficiently distinguish natural motions from other physically valid possibilities. Fortunately, kinematic techniques are effective in these cases [3, 37].
enable rapid adaptations before final refinements and adjustments of the motion are made.

1.2 Model Reduction

Earlier work on spacetime constraints suggests that the complex dynamics of characters with a large number of DOFs (e.g. a human character) has a negative effect on the speed and convergence of the optimization solver [29]. While this thesis describes a successful adaptation of a full human figure motion, I also explored the possibility of performing an automated simplification of a human skeleton using standard statistical tools for model reduction.

Statistical reduction techniques such as principal component analysis (PCA) and independent component analysis (ICA) are applied to the input motion data to produce a low-dimensional manifold for a given motion activity. The low-dimensional space is described in terms of the projection operator that converts the character's variable set, as well as its dynamics, from the full to the reduced formulation. This procedure significantly decreases the number of optimization variables and the nonlinear dynamics constraints, but does not commensurately improve speed and convergence of the optimization. On the contrary, starting with the same initial guess the resulting motions often contain unnatural body poses. While the reduced character configuration converges faster for some optimization formulations, the high-dimensional character optimization always produces motions that appear more natural.
Chapter 2

Background

This chapter presents the background information and the previous research work that is relevant to the adaptation method described in subsequent chapters. The motion synthesis section (2.1) describes the techniques for generating human motion from scratch, i.e. with little or no initial guess about the final motion quality. The motion editing section (2.2) introduces a number of methods that produce new motions starting from existing motion clips. Finally, the model reduction section (2.3) provides information on some of the common procedures for reducing both the dimensionality of complex systems and their dynamics; this section also describes the application of these techniques to solving computer graphics problems.

2.1 Motion Synthesis

Witkin and Kass [36] introduced the concept of spacetime constraints to the graphics community. A highly intuitive motion synthesis approach, spacetime constraints allows an animator to obtain physically valid and realistic looking motions by specifying a set of simple kinematic constraints. The physical realism is guaranteed by dynamics constraints which enforce equations of motion. The task of the objective function is to select the natural from the physically valid motions. The authors generate various jumping motions of a simple linked-rigid body by simultaneously fulfilling kinematic and dynamics constraints and minimizing the power consumption objective function.
Cohen [5] provides an interactive interface for low-dimensional characters to spacetime constraints. The “Spacetime Windows” can select a subset of the character’s degrees of freedom (DOFs) or define a specific time interval within which the solution to the optimization problem is to be found. By solving these partial optimization problems, the animator has more flexibility and a greater degree of control over the original setup that solves for the entire motion sequence [36]. Furthermore, the technique employed uses cubic B-splines instead of the finite differences [36] to represent motion trajectories.

Liu et al [25] address the redundancy arising in the computation of the constraint expressions and the corresponding Jacobians and Hessians within the linked-rigid body hierarchy. A hierarchical encoding of the character’s DOFs uses a combined spline and wavelet representation; this reduces the number of the corresponding control variables by applying the minimum number where the motion is smooth, and adding additional variables only where high detail is needed.

Liu and Popović [23] apply spacetime constraints to the motion synthesis of human characters with as many as 51 degrees of freedom. A simpler dynamics formulation without torques maintains the physical realism. During the unconstrained stage (e.g. flight), physical realism is maintained via angular and linear momentum constraints. The biomechanics data determines the angular momentum profile during the constrained stage (e.g. ground). Motions are synthesized from scratch, with the exception of a few transition poses inferred from motion capture data that enforce specific pose configuration at the instants where constrained and unconstrained stages connect. The lack of torques in the dynamics formulation prevents the use of torque-based objective functions. Instead, joint-based objective functions, such as joint trajectory smoothness and static balance, guarantee the natural look of the final motion.

Fang and Pollard [8] describe a set of efficient physics constraints that allow motion synthesis of complex human characters. These constraints compute aggregate torque and force around a fixed body point and reduce to linear and angular momentum constraints [23]. In addition to limiting the amount of torque around the contact point during the constrained stage, the magnitude and the direction of the friction force between the ground and the character is also controlled. The aggregate force can be computed in linear time.
as opposed to traditional physics constraints, which require a quadratic time computation [25]. The lack of torques specification allows only joint-based objective functions.

Kovar et al [18] synthesize new motions from an existing motion database by finding new ways of traversing the motion data segments. This method computes a motion graph of the input motion data that captures all the plausible transitions between the frames within the data pool. By taking different transitions at given (e.g. user-specified) frames, a rich set of realistic motions is synthesized. The new motions are constrained to have the same qualities of the input motions (i.e. the motion database of human walks can only produce variations of the walking motion); in addition, transition between the motion segments are not guaranteed to be physically valid.

Pandy and Anderson [27] propose a detailed model of the lower human body and use dynamic optimization theory to obtain the necessary muscle forces for execution of a desired motion task. This work derives from the optimal control algorithms introduced by Pandy et al [26], who show the benefit of formulating problems involving complex dynamics systems in terms of a parametrized optimization. The resulting muscle forces and the joint trajectories (vertical jump and walking motions) are close to the joint and torque values obtained from a real person jumping and walking.

2.2 Motion Editing

Witkin and Popović [37] describe a motion warping technique for editing input motions based upon a small set of kinematic constraints. This technique is analogous to the standard approach of animation keyframing since the user only need specify desired character poses or keyframes. Deformation and blending of the input motion curves produces the remainder of the motion between the keyframes. While motion warping achieves impressive results for complex human motions such as walking, the lack of dynamics makes any drastic change to the input motion infeasible. A similar approach is described by Bruderlin and Williams [3] who apply various image and signal processing techniques to the motion data. These techniques allow easy modifications of motion duration, blending of simple motion clips, and changes to the quality of motion that add or remove high frequency detail.
Gleicher [13] uses spacetime constraints for interactive motion editing. While numerical optimization is central to this approach, the interactive rates of motion editing are achieved only by removing the physics constraints from the problem formulation. Instead of relying on highly nonlinear and computationally difficult constraints describing equations of motion, a good initial guess and the kinematic constraints guarantee physical realism for non-ballistic motions. The objective function that minimizes the difference between the edited motion and the original aids the optimization convergence and speed.

Popović and Witkin [30] present a spacetime constraints solution to the motion editing problem that includes character dynamics. A manually simplified model of a high-dimensional human character guarantees convergence and reasonable optimization speeds. The simplified model construction requires user input, as it depends on the type of the input motion: the simple character for a broad jump is significantly different than the one used for a walking or running motion.

Lee and Shin [21] propose an interactive motion editing method that relies on a hierarchical representation of motion trajectories and an inverse kinematics solver. A multilevel trajectory formulation allows the editing of select motion sequences. They augment the traditional numerical optimization approach of solving kinematic constraints with additional steps that greatly improve the efficiency of their method. This method does not include character dynamics; therefore, the quality of the results is largely dependent on the artistic skill of the animator.

Brand and Hertzmann [2] use hidden Markov models (HMM) for generating motion clips with new styles based upon the input motion data. The data trains a statistical model for a given motion style; this model then applies the desired style to a different motion sequence. This is a data-driven approach for the editing and stylization of the input motion data that attempts to infer the dynamics from the motion data.

Li et al [22] use a combination of HMMs and linear dynamics systems (LDS) for describing the input motion data. The input motion is sliced into segments, which train the corresponding LDS. HMM graph determines the transitions between the motion segments. The user can also adjust the key poses at the transitions between the motion segments. The authors apply PCA to reduce the number of variables describing input motion trajec-
tories. While the LDS are a powerful mechanism for capturing the input motion dynamics, they only approximate physical principles; consequently, the physical validity of the edited motions is not guaranteed.

The dynamics filter presented by Yamane and Nakamura [38] edits the input motion to ensure physical correctness. The filter scans the input motion sequence one frame at a time, and adjusts each frame so that the equations of motion are satisfied. Thus, the user does not have a global control over the entire motion sequence (for example, extending the landing position of the jump should result in a more explosive takeoff). Nevertheless, the dynamics filter produces physically realistic results for even highly constrained human motions, such as walking.

Spacetime sweeping [33] uses an idea similar to input motion filtering. In addition, kinematic constraints give the animator more flexibility in motion editing. While the input motion is analyzed on a per frame basis, spacetime sweeping allows multiple passes over the same motion sequence; this effectively generalizes to a solution that takes the entire motion into account. This method works well for less-dynamical ballistic as well as constrained motions.

2.3 Model Reduction

Pentland and Williams [28] describe modal analysis technique for reducing the dynamics formulation of non-rigid objects. Modal analysis simplifies the dynamics into the sum of independent vibration modes. This method improves the efficiency of the dynamics simulation, as it breaks a computationally difficult problem into a number of smaller problems that are easier to compute. Modal analysis applies well to a variety of non-rigid body geometric representations. It can also generate some common computer animation effects, such as object squashing and stretching during collision.

James and Pai [17] employ an idea similar to reduction-based modal analysis for the real-time simulation of deformation systems. Precomputed modal vibration models enable physically-based computation of the deformable models' dynamics with minor CPU costs.

James and Fatahalian [16] compute a reduced model representation from simulated
data of deformable objects. By applying PCA to the deformable objects’ motion data, the authors obtain the reduced basis of their dynamics. This procedure, combined with the precomputation of the body’s internal collisions, enables the interactive manipulation of deformable objects (e.g. a table cloth).

Lall et al [19] construct a low-dimensional model of dynamics from empirical or simulated observations. PCA of the systems’ simulation data yields a projection operator that transforms the high-dimensional system representation to a low-dimensional space. This solution preserves the structure of the mechanical systems while reducing the number of parameters that describe them. This improves the dynamics’ simulation speed, and offers greater control over the behavior of mechanical systems.

Full and Koditschek [9] show qualitatively how simple models of a human body and its dynamics can be used to approximate the complex interaction between the body parts during a particular type of motion. They propose models for walking by vaulting, running by bouncing, and running by ricocheting that consist of only a few degrees of freedom. The authors also address the challenge of constructing a proper control mechanism for the simplified models, which can produce the desired motions automatically.

Cao et al [4] apply PCA to human facial motion capture data. They also use ICA to separate the reduced space variables into independent components. The model reduction produces a small set of controls for human expressions, while variable separation makes these controls an easy and intuitive interface for facial animation. This work shows how a small set of parameters can efficiently control complicated biological systems such as a human face.
Chapter 3

Adaptation

Adaptation reuses a recorded human motion by conforming it to new environments that may require different foot placement, modified dynamics, or other changes. The entire process resembles the standard keyframing technique, which enables an animator to adjust the motion by moving the hands, feet, or other end-effectors on the body; but in addition to these kinematic constraints, an animator can also insist on physically valid motions, limit the use of muscle forces, or restrict the motion with other dynamic constraints. Figure 3-1 shows the adaptation of a broad jump to a different foot placement.

Figure 3-1: The adaptation of a human broad jump (left) generates a new physically consistent jump (right) with a step takeoff and a step landing. The four point constraints (shown as red spheres with the matching yellow feet in the left figure) are the only constraints required to effect this change.
3.1 Formulation

The adaptation problem is a restatement of the spacetime technique, which computes the motion trajectories \( q \), the internal torques \( f \), and the Lagrange multipliers \( \lambda \). Lagrange multipliers help define the reaction forces between the environment and the human figure. The optimal motion minimizes the objective function \( E \), which separates natural movement from other physically valid motions that fulfill the adaptation goals indicated by the kinematic \( K \) and the dynamic \( D \) constraints:

\[
\min_{q(t), f(t), \lambda(t)} E(q, f, \lambda) \quad \text{subject to} \quad K(q) = 0 \quad D(q, f, \lambda) = 0.
\]

The choice of the objective function varies with the application. The synthesis applications in the literature optimize power consumption [36], torque output [25], torque smoothness [30], and kinematic smoothness [8], while the adaptation applications minimize joint-angle displacement [13] or mass displacement [30] from the original motion. In our adaptation experiments, the most reliable results were obtained by optimizing the smoothness of internal torques or by minimizing the total change in joint angles, torques, and Lagrange multipliers.

3.2 Dynamics Constraints

The formulation of dynamics constraints that enforce physical laws profoundly affects the efficient solution of the adaptation problem. Scaling the variables, which is discussed in the next section, improves the convergence and efficiency of numerical methods and enables the adaptation of human motions with many degrees of freedom. In some cases the performance can be further improved by eliminating the internal torques to reduce the order of

\[1\text{The same formulation supports inequality constraints, which are excluded from Equation (3.1) for simplicity, but are enforced with a numerical technique that replaces the active inequality constraints with equalities.}\]
the constraints and the number of optimization variables.

3.2.1 Standard Formulation

A standard method in classical mechanics [32] derives the differential algebraic equations that express Newton’s laws with the Euler-Lagrange equations from the Lagrangian $L$ of the human figure:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} - \lambda^T \frac{\partial P}{\partial q_r} = 0, \quad P(q)$$

where the index $r$ enumerates the degrees of freedom in the root joint (global position and global orientation) and the index $i$ enumerates the degrees of freedom in the remaining joints, which are actuated by internal torques $f_i$. The environment constraints $P$ define the interaction between the figure and the environment, such as the contact between the ground and the feet (Section 3.3.2). The complexity of the Euler-Lagrange equations for a human figure demands a systematic evaluation of these quantities [24] and the elimination of redundant computations with a recursive formulation [14] or with caching.

3.2.2 Reduced-Order Formulation

In some adaptation problems, the internal torques $f_i$ need not be bound by the objective function (to select a natural motion) or by the constraints (to restrict the use of a muscle). In the most favorable case, all torques are free variables, which can be eliminated from the optimization problem in Equation (3.1) and the dynamics constraints in Equation (3.2):

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} - \lambda^T \frac{\partial P}{\partial q_i} = 0. \quad (3.3)$$

On the ground, the total force on the body is given by the ground reaction force $\lambda^T \frac{\partial P}{\partial q_r}$.

This force is a function of the root position and the angle values of all other joints between
the root and the contact point. In flight, there are no environment constraints and the total force is zero.

The reduction in the number of optimization variables and in the order of the dynamics constraints improves the efficiency of numerical solutions—often by an order of magnitude—but might have an adverse effect on the appeal of the resulting motion, as freely varying torques may cause the body limbs to jerk undesirably. In many cases, minimizing the total change made to joint angles and multipliers corrects this problem. Additionally, the torque constraints from Equation (3.2) can be reinserted and their smoothness controlled by the number of control points in the spline parameterization. Even when the visual quality of the final motion requires optimizing every joint-angle, torque, and multiplier value, the efficiency of adaptation with the reduced constraint in Equation (3.3) enables rapid prototyping before the final refinements and smoothing are made.

3.2.3 Discussion

The reduced expression for the dynamics constraints is the Lagrange multiplier formulation of the aggregate-force constraints, which were introduced by Fang and Pollard in reduced coordinate form [8]. Baraff summarizes the differences between the reduced and multiplier formulations for general applications in computer graphics [1]. Both the standard and the reduced-order dynamics use the Lagrange multiplier formulation because of its systematic treatment of all environment constraints $P$ in the presence of cyclic dependencies (e.g. both feet constrained to be on the ground simultaneously).

The aggregate-force constraints are similar to the momentum constraints [23]. In flight, the constraints are identical and equivalently state that the total angular momentum remains constant while the center of mass follows a parabolic trajectory. On the ground, or whenever the human figure interacts with the environment, the momentum constraints employ characteristic momentum patterns, without modeling the reaction forces. The appropriate momentum patterns emerge naturally from the aggregate-force constraints in either formulation. In addition to their generality, the aggregate force constraints expose the reaction forces and enable their use in the objective function (e.g. to match the impact forces in the
original motion or to reduce them for a "softer" run) and constraints (e.g. to keep the impact forces within a friction cone [8]).

The two expressions for dynamic constraints, with torques $D_T$ and without torques $D_G$, define the two extremes in a range of possibilities. Intermediate formulations, which include some but not all torques, can exploit the benefits of the reduced formulation, which generates results rapidly but has limited applicability, and the benefits of the full formulation, which generates the best results and applies to all adaptation problems. For example, if the adaptation requires a jump with an injured ankle, the values of the ankle torques can be restricted and added to the optimization along with the Euler-Lagrange equations for the corresponding degrees of freedom. Or, if some of the limbs jerk undesirably, their torques can be included and their change minimized along with the modifications to joint angles and multipliers.

3.3 Kinematic Constraints

Kinematic constraints are high-level controls\textsuperscript{2} that an animator can use to guide the motion generation process. By constraining the specific body point (e.g. right foot toe) the character is placed at a desired location in space. Pose constraints determine the entire body pose at a desired time. To prevent the limbs from assuming an unnatural pose, the motion range of joints and limbs can be constrained via joint and cone constraints. The following sections describe the character configuration and provide the details on kinematic constraints.

3.3.1 Character Description

A human character is represented as a collection of rigid bodies (i.e. limbs) connected via joints in a tree-like hierarchy. Each joint consists of one, two or three degrees of freedom (DOFs). The root of the hierarchy (located in pelvis) has three additional translation DOFs that allow the character to move in space. The human character used in my experiments\textsuperscript{2}Some dynamics related constraints, such as the restriction in the torque magnitude, can also be thought of as high-level user controls.\textsuperscript{2}Some dynamics related constraints, such as the restriction in the torque magnitude, can also be thought of as high-level user controls.\textsuperscript{2}Some dynamics related constraints, such as the restriction in the torque magnitude, can also be thought of as high-level user controls.\textsuperscript{2}Some dynamics related constraints, such as the restriction in the torque magnitude, can also be thought of as high-level user controls.
has a joint configuration with 42 DOFs, as shown in Figure 3-3.

Limb $i$ has a corresponding local transformation matrix $R_i$ and a global transformation matrix $W_i$. The local matrix $R_i$ expresses the position and the orientation of the limb with respect to the coordinate frame attached to the parent limb and the global matrix $W_i$ with respect to the global coordinate frame. Both matrices are functions of character’s DOFs. The global coordinate frame is assumed to be the parent of the root joint, which implies that the local matrix $R_0$ defines the global position and orientation of the root joint. The global matrix $W_i$ for any limb is defined with a chain of local matrices:

$$W_i = R_0R_1...R_{i-1}R_i.$$ 

Geometrically, each limb is defined as a parallelepiped with a uniform mass distribution. Limb masses are derived from the performer’s total weight, based on the relative mass percentages obtained from a biomechanics reference [35] and shown in Table 3.1.

<table>
<thead>
<tr>
<th>Body Part</th>
<th>% Total Mass</th>
<th>Body Part</th>
<th>% Total Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pelvis</td>
<td>15.3</td>
<td>Upperarm</td>
<td>3.3</td>
</tr>
<tr>
<td>Thorax</td>
<td>22</td>
<td>Forearm</td>
<td>1.9</td>
</tr>
<tr>
<td>Clavicle</td>
<td>4</td>
<td>Hand</td>
<td>0.6</td>
</tr>
<tr>
<td>Head</td>
<td>7.1</td>
<td>Thigh</td>
<td>10.5</td>
</tr>
<tr>
<td>Shank</td>
<td>6</td>
<td>Foot</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 3.1: Mass of each limb as a percentage of the total body mass.

### 3.3.2 Point Constraints

A point constraint is defined in terms of an end-effector $p$ on the limb $i$ and the desired position $d$ in the world coordinate frame:

$$K_{\text{pose}} = W_ip - d,$$

where the transformation matrix $W_i$ changes with time along with the configuration of joints. Left side of figure 3-2 shows the position of the pose constraint on a foot of the character’s leg. The toe is fixed at the appropriate location.
The same point constraint can be used to specify both kinematic and environment constraints. The difference is that the environment constraint also introduces forces onto the character via Lagrange multipliers (Equation (3.2)), while the kinematic constraint does not.

In most cases a point constraint is enforced without bounds (i.e. the only location of the end-effector that satisfies the constraint is at the desired position). Sometimes, however, the bounds are relaxed to allow the whole range of points to satisfy the desired end-effector position. For example, the toe point constraint with the lower bound of zero and the relaxed upper bound on its vertical direction makes any point in the vertical plane above the ground a valid toe location.

### 3.3.3 Joint Limits

The joint limit constraints prevent the character from assuming unnatural poses during the motion (e.g. driving its arm through the torso). For pin joints the limits are simple upper and lower bounds of the joint angle value. For two and three DOF joints, a cone constraint defines the range of motion of the entire limb (right side of figure 3-2). The cone constraint is defined in terms of the cone axis and the minimum value of the dot product between the limb orientation vector and the cone axis. Given the cone axis \( \mathbf{a} \) and the direction vector \( \mathbf{l} \) of the limb \( i \) the cone constraint becomes:

\[
K_{cone} = \mathbf{a} \cdot SR_{i} \mathbf{l}
\]
Cone axis is defined with respect to the limb parent’s coordinate frame. Vector $I$ is a limb direction vector in the limb’s coordinate frame. Local transform $R_i$ computes $I$’s orientation in the parent’s coordinate frame. Matrix $S$ is used to “shape” the cone (i.e. to squash it or stretch along the desired axis). $S$ is a diagonal matrix whose diagonal entries define the scale of the cone deformation along the x, y, and z axes.

Figure 3-3: The motion adaptation method is applied to high dimensional human characters. A skeleton configuration with 42 degrees of freedom was used for all of the adaptations.
A discretization of the adaptation problem produces an optimization with many unknowns and many nonlinear constraints. A direct collocation solution of such problem computes the trajectories by solving for the coefficients in an expansion with finite differences [36], cubic B-splines [5], or wavelets [25] to fulfill the values of kinematic and dynamics constraints at prescribed time points. The number of nonlinear constraints increases with every time point by at least the dimension of the dynamics constraint $D$, while the number of unknowns increases with each approximation coefficient for every state and control trajectory. In these circumstances, a numerical technique that exploits sparsity in the constraint Jacobian is essential for an efficient solution of the adaptation problem. Although general-purpose nonlinear programming packages such as LANCELOT [6] and SNOPT [11] capably address this requirement, their success is contingent on the quality of the initial guess and on the proper scaling of the optimization variables.

The explicit and implicit solutions of the adaptation problem select the natural motion from the set of physically valid motions based upon different criteria. In the explicit case, the optimization minimizes an energy function such as muscle smoothness, which is an explicit function of torque accelerations. In the implicit case, the optimization modifies the original motion with a sequence of minimal adjustments, which implicitly favors motions that are closer to the original performance. Both solutions enhance the adaptation process: the implicit solution results in an efficient numerical solution, which permits rapid adaptation of the performed motion, and the explicit solution enables further refinement and
4.1 Scaling

To achieve the results presented in this thesis, the proper scaling of joint angles, torques, and multipliers is required. Under restrictive theoretical assumptions, numerical optimization methods can be shown to produce the same sequence of iterates regardless of the scaling. In practice, however, this scale invariance cannot be achieved, and proper scaling is essential to resolve the difficulties in the conditioning of difficult optimization problems [12].

A simple physical pendulum weighing 70 kg (a typical human weight) and 1.70 m long (a typical human height) demonstrates the effect of scaling on the computation of physically valid trajectories. Without scaling, the computation of the pendulum trajectory requires 129 iterations and 11.6 seconds of the computation time. With the simple scaling procedure described in this section, the computation of identical trajectories requires 24 iterations and 1.96 seconds. In both instances, the values of every joint optimization variable are identical. The effect of improper scaling is even more drastic on the adaptation problem, as it prevents convergence without excessively small steps, which can extend the computation of several seconds to as long as several hours.

The problem can be traced back to the discrepancy in the range of the state variables, the torques, and the Lagrange multipliers, which in turn influences the scaling of the Jacobian and Hessian matrices. The simplest solution is to scale the mass density of each limb by a uniform constant factor $s$ and solve for the new joint angles $q'$, torques $f'$ and Lagrange multipliers $\lambda'$. This scaling changes the Lagrangian $L' = sL$ and the expression of the dynamics constraints in terms of the unscaled Lagrangian $L$:

$$
D'_{\tau}(q', f', \lambda') = \begin{pmatrix}
    s \frac{d}{dq'} \left( \frac{\partial L}{\partial q'} \right) - s \frac{\partial L}{\partial q'} - \lambda'^T \frac{\partial P}{\partial q'} \\
    \frac{d}{dq'} \left( \frac{\partial L}{\partial q} \right) - \frac{\partial L}{\partial q} - \lambda^T \frac{\partial P}{\partial q} - f_i' \\
    P(q')
\end{pmatrix}.
$$ (4.1)

As division by the factor $s$ reveals, this scales the values of torques and the Lagrange mul-

smoothing of the adaptation.
tipliers without changing the joint-angle trajectories:

$$D'_T(q', f', \lambda') = D_T(q, \frac{f}{s}, \frac{\lambda}{s}). \quad (4.2)$$

One possible drawback of this simple scaling transformation is a reduction in accuracy. All of the experiments, however, used the scale factor $s = 0.001$ and did not exhibit such problem. Should the loss of accuracy become problematic, proper scaling could also be established by determining the range of torque and multiplier values precisely [12].

Scaling is not as critical for the motion synthesis techniques with the momentum constraints [23] and constrained aggregate forces [8]. Neither technique computes the internal torques, and the momentum constraints do not model the ground reaction forces, while, in the reduced-coordinate formulation, the aggregate-force constraints do not require the Lagrange multipliers. These choices allow the techniques to solve for joint angles, which have the same units and thus proper scaling, but prevent them from solving adaptation problems that must restrict or smooth muscle forces.

### 4.2 Initialization

The values of joint angles $q$, torques $f$, and the Lagrange multipliers $\lambda$ are initialized to the joint-angle trajectories $\bar{q}$ in the original performance. The estimated trajectories are the solution of the optimal trajectory problem (Equation (3.1)) with the least squares objective function:

$$E(q) = \int ||q(t) - \bar{q}(t)||^2 dt. \quad (4.3)$$

This optimization is initialized with torques and multipliers set to zero, and joint angles estimated with a cubic B-spline interpolation of angles in the original performance. The entire process needs to be executed only once for each original performance. The estimated trajectories are stored with the motion and used for every adaptation problem. A solution to a similar optimization problem was previously used to estimate the parameters of a simplified character from the original performance [30]. The initialization procedure described here, however, computes the joint angles, torques, and multipliers for the entire
human figure without any simplification.

This initialization assumes that the original performance is a physically valid motion. When it is not, the initialization might not successfully compute the torques and the multipliers required by our adaptation technique. Although not explored in this thesis, a different initialization procedure might eliminate this requirement by consulting a stored database of transition poses [23], or by restricting the ground reaction forces [8].

4.3 Explicit Solution

With proper scaling and initialization, sequential quadratic programming [11] can solve the adaptation problem for a full human figure. This iterative descent technique computes the optimal trajectory by minimizing the merit function along a search direction given by the solution of a quadratic programming subproblem. The merit function balances the competing goals of improving the objective function and remaining on the nonlinear constraint surface, while the quadratic subproblems linearize constraints and approximate the merit function with a quadratic expansion around the current iterate.

An alternative technique could minimize the merit function by solving a sequence of bound constrained nonlinear subproblems as implemented by the LANCELOT optimization package [6] and demonstrated by Fang and Pollard [8] on an optimal trajectory method with constrained aggregate forces. Because solutions to these nonlinear subproblems might require multiple evaluations of the objective and constraint functions, this approach should be used only if the evaluation of these functions and their gradients is cheap. The sequential quadratic programming, on the other hand, economizes by solving the subproblems which do not require additional evaluation of functions and their gradients.

The explicit solution allows comparison of several objective functions on otherwise identical adaptation problems. The power-consumption objective, which selects the most efficient motion, produces a natural motion on some adaptation problems, but in many cases it minimizes power consumption with unnatural movement or interpenetration of limbs, neither of which can be easily resolved with cone or joint-limit constraints. The mass displacement objective, a kinematic analogue of the power consumption, needs to be
applied in conjunction with the joint-angle displacement to prevent the human figure from jumping with a still body and using only its ankles. On its own, the joint-angle displacement produces jerky motions, which can be resolved in some cases by simultaneously optimizing the smoothness of joint angles.

For the adaptation problem, the optimization of muscle smoothness,

\[
E(f) = \int \|\dot{\dot{f}}(t)\|^2 \, dt,
\]

consistently generates results better than the optimization of power consumption or power output. However, slow convergence on some adaptation problems suggests that it should be used primarily as a post-processing step to clean up and refine motions. The next section describe an implicit solution method, which is more suitable for prototyping and experimentation.

4.4 Implicit Solution

The implicit solution selects a natural motion with a sequence of minimal modifications, which iteratively modify the original motion until all adaptation goals are met. The iterative algorithm alternates between computing the direction \(d_i\) for the next modification and computing the step size \(\alpha_i \in (0, 1]\) for the modification in this direction. The cumulative changes are small, even though the proximity to the original motion is never enforced explicitly. The proximity criterion emerges from the requirement that the magnitude of the search direction be as small as possible.

At each iteration, the direction for the next modification \(d_i\) is a minimum-norm solution of an underdetermined linear system produced by a linearization of the kinematic and dynamic constraints \(C^T = (K^T, D^T)\):

\[
\min_d \|d\|^2 \\
\text{subject to} \quad C(x) + \frac{\partial C}{\partial x}(x_i) \, d = 0.
\]

The current iteration \(x_i\) contains the coefficients for joint angles, torques, and multipliers.
The dimension of the Jacobian matrix $\partial C/\partial x$ restricts the choice of numerical solutions to techniques that can exploit its sparsity. The Conjugate-Gradient algorithm converges slowly because the corresponding normal equations are poorly conditioned and do not improve with diagonal preconditioning. A faster solution can be derived by explicit construction of the null-space basis with Q-less QR factorization of the sparse Jacobian matrix [7]. The quadratic programming technique SQOPT [10] supports more easily the inequality constraints, and maintains the null-space basis with the sparse LU factorization of the Jacobian matrix.

The minimum-norm solution defines the direction for the line search, which computes the step size $\alpha_i$ for the modification that minimizes the distance of the next iterate $x_{i+1}$ from the adaptation constraints:

$$\alpha_i = \arg\min_\alpha \|C(x_i + \alpha d_i)\|^2, \quad (4.6)$$

$$x_{i+1} = x_i + \alpha_i d_i.$$

The iteration stops once the constraints are satisfied with the desired accuracy. Although the distance from the original values could be further improved by stepping along the constraint surface, the additional iterations do not significantly improve the visual quality of the resulting motion. It is more effective, in general, to stop the iteration and, if necessary, proceed to refinement with torque smoothness. The unscaled norm works well because the variables are scaled properly. For more control, the norm could be scaled to weigh differently the modifications made to each variable.
Chapter 5

Model Reduction

Earlier work on spacetime constraints showed how to simplify the skeletal structure of complex characters to combat the computational difficulties associated with highly non-linear dynamics constraints [30]. This chapter presents an automated procedure for model simplification that relies on common statistical tools, such as PCA, to infer the reduced character representation and the corresponding dynamics.

Figure 5-1 shows the main steps of the motion synthesis using the reduced model. Analysis of the input motion data infers the projection operator, which is used to project the input data and the character dynamics to reduced state space. The motion adaptation problem and the relevant constraints are formulate in the reduced space. Projection of the optimal solution in the low dimensional space to the original high dimensional space generates the final motion.

5.1 Projection

Let $P$ be a projection matrix, which maps a joint configuration $q \in \mathbb{R}^n$ in the high-dimensional space to the new configuration $z \in \mathbb{R}^d$ in the reduced space

$$z = Pq$$

Conversely, the inverse of the projection matrix maps the configuration in the reduced
captured motion

motion generated by simple model

project data and dynamics

reduced state space

Figure 5-1: Motion Synthesis Process

\[ q = P^{-1}z \] (5.1)

Projection matrix achieves an effect similar to that of joint removal described in [30]. This approach is illustrated on the character performing a broad jump. Given a two-legged creature with six pin joints, the identity matrix is a trivial projection matrix which makes the low-dimensional configuration equal to the high dimensional. Since corresponding joints on the two legs (e.g. left and the right knee) follow almost identical trajectories during the broad jump, a denser projection matrix reduces the number of DOFs from six to three:

\[
P = \begin{bmatrix}
    1 & 0 & 0 & 1 & 0 & 0 \\
    0 & 1 & 0 & 0 & 1 & 0 \\
    0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\]

Columns in the matrix \( P \) above correspond to the creature’s joints in the following order: left hip, left knee, left ankle, right hip, right knee, right ankle. \( P \) constrains the

---

1The root of the character’s hierarchy always remains in its high-dimensional six DOF configuration.
corresponding joints on the legs to follow the same motion curves. It is possible, however, to further reduce the structure of the two legs to a single DOF by observing that all three joints on a leg move in unison during the broad jump. Labeling their respective angles as $\alpha$, $\beta$, and $\gamma$ (Figure 5-2), the following equations reflect the correlations between these joints:

\[
\begin{align*}
\alpha &= \frac{\pi}{2} - \frac{\beta}{2} \\
\gamma &= \frac{\beta}{2}
\end{align*}
\] (5.2) (5.3)

The right hand side of Figure 5-2 shows a hopper character [30] with a single prismatic joint connecting the upper and the lower part of the character's leg. The projection matrix corresponding to the joint angle equalities above achieves the similar level of simplification:

\[
P = \begin{bmatrix}
-1 & 2 & 1 & -1 & 2 & 1
\end{bmatrix}
\]

Thus, a joint angle $\gamma$ is a substitution variable for all joints on the legs. Variable substitution is a key word here. Even though the character representation is reduced from six to one DOF, the structure of the character has not changed. $\gamma$ still describes rotation around each joint; in contrast, the single DOF hopper's prismatic joint determines the amount of translation between the lower and the upper part of its leg [30].

Figure 5-2: Leg Joints Projection

\footnote{Note that one should add $\frac{\pi}{2}$ to the value of $\alpha$ obtained after computing $P^{-1}z$}
5.2 Dynamics Constraints

Projection matrix enables the reduction of the number of DOFs describing a human character. It also reduces its dynamics by expressing the equations of motions in terms of the low-dimensional variable set [19]. The Euler-Lagrange equations of the simplified model are derived from the Lagrangian of the high-dimensional character after variable substitution (Equation (5.1)):

\[
D_R(q, f, \lambda) = \left( \frac{d}{dt} \left( \frac{\partial L}{\partial q} \right) p^{-1} - \frac{\partial L}{\partial q} p^{-1} - \lambda^T \frac{\partial p}{\partial q} p^{-1} \right) = 0 \tag{5.4}
\]

The standard formulation of the high-dimensional character’s physics (Equation (3.2)) shows that the number of dynamics constraints equals the number of the character’s DOFs. Equation (5.4) above shows that the same is true for the simplified character as well; the projection matrix reduces the number of equations from the size of the high-dimensional to the size of the low-dimensional character’s state space.

5.3 Kinematic Constraints and Objective Function

While the projection matrix can drastically reduce the number of dynamics constraints, kinematic constraints retain the same dimensionality regardless of the character’s complexity. Point constraints, for example, are defined in terms of the points on the character’s body; their dimension equals the dimension of the world coordinate space. Similarly, objective functions give a single scalar value, and model reduction does not impact their dimensionality either.

Character simplification, however, can still improve the convergence speed with respect to the kinematic constraints and the objective functions, since the number of variables on which these functions depend on is significantly reduced. In addition, some of the objective functions, such as a joint and torque smoothness are expressed in terms of the reduced space variables; this can further improve the efficiency of the optimization process.
5.4 Statistical Analysis

Previous sections show how a manually constructed projection matrix can simplify a character's representation and its dynamics. The statistical analysis tools perform this reduction process automatically by inferring the projection matrix from the input motion data. A number of statistical techniques exist for data reduction; PCA and ICA are the most convenient, as they produce the linear operators that can replace the projection matrix.

Statistical model reduction has several advantages over manual skeletal simplification. First, the reduction is automatic, which makes it an ideal candidate for methods that transform motion clips in a physically appropriate manner. If the motion clips are also clustered into motion primitives [22, 20] then statistical analysis can explore correlations between these similar motion clips. Second, statistical analysis can reveal hidden correlations between multiple joints in the human body to further reduce the dimensionality of the optimization problems. In walking, for example, the motion of arms and legs is highly correlated; yet, it is not immediately obvious how to hand-design projection matrix to account for this symmetry. Statistical analysis, on the other hand, produces such a projection matrix.

5.4.1 Principal Component Analysis

PCA has been extensively used for dimensionality reduction in the computer graphics community ([22, 2]). This section presents a brief overview of PCA, which is fully described in a book by Edward Jackson [15].

The input motion data consists of samples of the character's DOFs:

$$D = \begin{bmatrix} q_0^0 & q_0^1 & \cdots & q_0^n \\ \vdots & \vdots & \ddots & \vdots \\ q_n^0 & q_n^1 & \cdots & q_n^n \end{bmatrix}$$

Each column in the matrix $D$ is a vector of DOFs $(q_0 \ldots q_n)$ sampled at a particular time $t$. Relative ordering of these samples does not affect the PCA's results. The covariance matrix $S$ is a symmetric matrix whose eigenvalue decomposition [31] produces eigenvectors
and eigenvalues:

\[ QLQ^T = S \]

Eigenvalues are stored in columns of \( Q \), while the diagonal of \( L \) contains the corresponding eigenvalues. \( Q, L, \) and \( Q^T \) are square matrices with the rank \( n \). The projection matrix from a high-dimensional state space \( R^n \) to a lower-dimensional space \( R^d \) is constructed from the subset of the most significant eigenvectors (\( d \) top rows of \( Q^T \)):

\[
P = \begin{bmatrix}
  e_n \\
  \vdots \\
  e_{n-d}
\end{bmatrix}
\]

Thus, the lower-dimensional space is spanned by the axes corresponding to the eigenvectors of the data covariance matrix. The ordering of eigenvectors in \( Q^T \) (and the projection matrix) is not accidental; the eigenvector in the first row of \( Q^T \) corresponds to the highest eigenvalue, and it accounts for the most variance in the original data set. The second eigenvector’s eigenvalue is the second highest, and so on.

There are several methods for selecting the dimensionality of the reduced space: visual inspection of the quality of reprojection \( P^{-1}Pq \), selection of an appropriate threshold for some norm \( \|q - P^{-1}Pq\| \), or visual inspection of the eigenvalue curve. Figure 5-3 shows the eigenvalues obtained from the data of a single broad jump. Since the first six eigenvalues are much larger than the remaining 22, the graph suggests that six eigenvectors corresponding to these eigenvalues should be used as the basis of the reduced space.

PCA attempts to find correlations among the high dimensional data variables and capitalize on those by producing a smaller set of uncorrelated variables. While the reduced space variables are uncorrelated, they are not independent. The variables in \( R^n \) dimensional space are a linear combination of more than one variable in \( R^d \) (conversely, the value of the variable in \( R^d \) is a function of more than one variable in \( R^n \)). Put in the spacetime formulation context, this implies the denser structure of the constraint functions’ Jacobian. A denser Jacobian has a negative effect on the convergence and the speed of the optimization.
process.

This problem becomes more obvious when one considers the projection matrix for $R^d$ where $d$ equals the original data dimension $n$. An $n \times n$ identity matrix would suffice as a projection matrix for this trivial example. PCA, however, builds a projection matrix that for a typical data set will have no zero entries. A basis vector rotation can remedy this problem by changing the relative axes’ alignment of the eigenvectors and effectively making the variables in $R^d$ more independent.

Rotations are defined as an optimization process that maximizes the desired properties of the projection matrix [15]. For example, the number of zeros in every eigenvector should be maximized, while the occurrence of two nonzero entries in the same projection matrix’ column (eigenvectors are distributed across the rows) should be minimized. These two goals make the variables in $R^d$ as independent as possible.

Rotations are either orthogonal or oblique. Orthogonal rotations (orthomax, quartimax, varimax) keep the eigenvectors perpendicular to each other. Oblique rotations do not keep this property of basis vectors; relaxation of perpendicularity constraint makes oblique rota-
tions more suitable for achieving the largest possible independence among reduced space variables. For the trivial example above both types of rotations produce the identity matrix as a final result.

5.4.2 Independent Component Analysis

ICA is another statistical technique that is becoming increasingly popular in the field of computer graphics [4]. Unlike PCA, however, ICA is not a dimensionality reduction technique. Instead, ICA is typically used for separation of variables generated by independent signal sources. A classic example involves a setup where two microphones are placed in opposite corners of one room. Sound recorded in the room is a mixture of the signals coming from both microphones. ICA can separate the recorded signal into two statistically independent sources.

PCA produces a reduced character representation that can be thought of as a mixture of independent sources. All the reduced space variables control all the variables in the high-dimensional space. Alternatively, a subset of the reduced space variables could correspond to a cluster of DOFs of the full human figure that move synchronously (e.g. legs during the jump). Application of ICA to the data samples in the low-dimensional space achieves this goal of reduced variable separation.

The projection matrix $P$ generated by PCA transforms the original data samples $D$ to the set of the samples in the low-dimensional space:

$$Z = PD$$

ICA analyzes $Z$ to generate a linear transformation between the set of the original reduced space variables $z$ and a new set of the independent variables $i$:

$$z = Ai$$

Equation (5.1) can now be expressed in terms of a projection matrix $P$ and matrix $A$:

$$q = P^{-1}Ai$$
Since ICA output conforms to the requirements of the model simplification described in the previous section, ICA-based model reduction simplifies the character and the corresponding dynamics following a similar recipe. ICA achieves an additional effect similar to that of PCA's basis vector rotation: the reduced space variables are as independent with respect to each other as possible.
Chapter 6

Results

The adaptation experiments explore a wide range of motion edits with constraints that vary the foot placement, restrict the use of specific muscles, and introduce new environment constraints. The addition of weights to the character’s skeleton, and changes to the skeletal dimensions further illustrate the flexibility of the adaptation method. These modifications generate over twenty adaptations from two performances: a run and a jump. The performances are captured in a motion-capture studio by tracking the motion of small markers attached to the body of the performer. (Figure 6-1). Standard commercial tools [34] estimate the joint-angle trajectories and the length of each limb.

To facilitate evaluation of each adaptation, as few constraints as possible are used to specify the adaptation goals. Point constraints (visualized by red spheres), restrict the placement of feet and hands. Cone (red cones drawn around the corresponding limbs) and joint limit constraints (a line extending from a knee showing the maximum allowed range for a knee joint) improve the aesthetic quality of the results by preventing interpenetration of limbs, or enforcing natural joint-angle limits. Full pose constraints (transparent white boxes) are not necessary to generate high-quality adaptations; on occasion, however, they are the easiest tool for preventing limb interpenetration. The figures visualize every kinematic constraint used to obtain the motion shown. The only exceptions are the inequality constraints (which ensure that the feet remain above the ground and that the hand reaches above the rim in the dunking motion) and the orientation constraints (used to constrain the orientation of the pelvis joint for the twist jumps).
For a full human character with 42 DOFs, PCA generates a reduced character representation with as few as 16 DOFs. Using fewer DOFs makes it more difficult to find a solution which satisfies all the constraints; this jeopardizes the quality of the resulting motion, as the optimizer spends more time "searching" the motion space and is likely to produce an unnatural looking motion. Using more than 16 DOFs does not improve the results, but it slows down the speed of the optimization process. The reduced character adaptation is always slower for an implicit solution with and without torques. While the explicit solution with PCA outperforms the explicit solution without PCA, the difference in timing is negligible compared to the quality of the two motions; the high-dimensional character optimization always produces more realistic-looking results than the low-dimensional one.

The timing results for adaptation experiments are shown in Figure 6-2. The implicit solution with only $q$ and $\lambda$ always takes the least amount of time compared to the implicit solution with $q$, $f$, and $\lambda$ and compared to the explicit solution, which, in almost all cases, is the slowest.
6.1 Initialization

The initialization step uses the dynamics of the human performer to estimate the torques and the Lagrange multipliers for both performances. The environment constraints specify the ground contacts in each performance. The ground contacts also define the duration of every ground or flight stage. The resulting trajectories approximate the joint trajectories and define physically consistent torques and ground reaction forces. Figures 6-3 and 6-26 show the results of the initialization for the jump and run motion, respectively. Both of these motions serve as an initial guess for subsequent motion adaptations.

6.2 Jumps

Translating the original feet positions produces the diagonal and jumps with the step-like takeoff and landing (Figure 6-15, Figure 6-19, Figure 6-14, Figure 6-13, Figure 6-17).
Rotating the landing position of the feet by 90 degrees generates a half twist to the right (Figure 6-11, Figure 6-20). Similarly, rotation of the takeoff position by 90 degrees yields a half twist to the left (Figure 6-12). Figure 6-9 shows the hop jump obtained by constraining the left knee to a fixed angle (visualized by the purple color of the left shank). Adding the additional weight (about 20% of the body’s total mass) in the character’s right hand produces a motion where the left arm extends far in the air to account for a change in the body’s mass distribution (Figure 6-5).

By constraining the torques of the right hip and a knee during the hop jump landing, the right leg is prevented from bending upon impact with the ground (Figure 6-10). Similarly, the limp right arm (constrained right shoulder torques) causes the character to extend its left arm in flight much higher than during the regular jump (Figure 6-4). A jump with the additional weight in the right hand, and the constrained right ankle’s torque (Figure 6-6) results in the character shifting the weight as far to the left as possible; in this way, it allows the left ankle to take the additional weight and propel the character into flight.

For the bar grabbing motion (Figure 6-18), only the first half of the jump motion is used as the initial guess. For the motion segment where the character is hanging on the bar the flight apex frame is used as the initial guess. Since this single frame initial guess does not have valid torque or Lagrange multiplier values, it is impossible to compute the explicit solution with the torque smoothness objective for the bar grabbing and the dunking motion.

### 6.3 Runs

Runs with cross steps (Figure 6-24, Figure 6-30), wide steps (Figure 6-25, Figure 6-31) and different length steps (Figure 6-28) are adapted from the original run by changing the foot placement constraints. The run with the bouncy steps (Figure 6-27) retains the original feet positions, but the character has more time to spend between contacts with the ground. The length of both steps in the hurdle run is extended by moving the feet landing positions (Figure 6-29). The flight time of both steps is also longer. The inequality point constraints imposed on the location of the character’s left and right toes (i.e. minimum toe height) make her clear the hurdle. Since the hurdle run is substantially different from the original
run motion sequence, the cone and joint limit constraints are required to eliminate unnatural limb orientations of the right foot and the shank.

The "explosive" nature of the hurdle run puts a strain on the explicit solution with the smoothness objective (Equation 4.4) optimization. On one side the optimizer has to satisfy the imposed kinematic and dynamics constraints of a highly dynamic motion; at the same time it is trying to maintain the smoothness of torque curves by minimizing the specified objective function. These two sets of competing goals have a negative effect on the speed and the convergence of the explicit formulation optimization.

For the dunking motion (Figure 6-21) the first step computes a spacetime problem to obtain the pose in which the character reaches for the hoop. This pose then initializes the hoop hanging segment of the dunking motion. Similar to the bar grab motion, the lack of valid initial guess for the hanging segment prevents computation of the explicit solution.

Figures 6-22 and 6-23 show the results of adapting a dunking motion to a character with different limb sizes and proportions. The resulting motions maintain the physical validity and realism, even though the initial guess comes from the character with the original character configuration.

6.4 Limitations

Sometimes, the difference between the initial and the adapted motion is so large that it is not possible to attain reasonable results by using the original jump or a run as an initial guess. Instead, additional steps have to be applied before the final optimization run. For the full twist left and right motions shown in Figures 6-7 and 6-8 respectively (where the character orientation changes by 180 degrees between the takeoff and the landing), the input motion is sliced in three stages (takeoff, flight, landing). The character is then rotated in all the frames of the appropriate stage (i.e. takeoff on the full twist right, and the landing on the full twist left motion). Splicing the stages back together produces a better initial guess than the original performance. Successful application of these techniques suggests that there might be some benefit in combining the spacetime constraints with the sequencing methods (e.g [18]).
Table 6.1: The adaptation method’s implementation approximates each joint-angle $q$ and torque trajectory $f$ with a cubic B-spline curve. The Lagrange multipliers $\lambda$ are sampled at prescribed points and are not approximated by a cubic B-spline curve. Listed above are the number of control points used in jump and run adaptations along with the range for the total number of optimization unknowns and constraints.

Besides the sensitivity to the quality of the initial motion, the least robust aspects of the motion adaptation are the discretization settings for the kinematic and dynamic constraints and the motion curves. The decision on the number of b-spline basis functions (for motion curves’ representation), and the number of time slices at which to enforce the constraints is empirical and requires careful tuning. If too few constraints or too many b-spline control points are specified the resulting motion will not look physical. Similarly, too many time slices for the constraints or too few control points will make the adaptation problem over-constrained and infeasible. Table 6.1 shows the settings used for the adaptation of jumps and runs. The number of constraints for the implicit formulations without the torques is the same for both the reduced and full human character, because model reduction only applies to joints with the torques (i.e. all but the root joints). The reduction in the number of constraints is obvious for the implicit formulation that includes the torques.
Figure 6-3: Jump Initial Guess

Figure 6-4: Limp Arm
Figure 6-5: Briefcase

Figure 6-6: Briefcase and Limp Ankle
Figure 6-7: Full Twist Left

Figure 6-8: Full Twist Right
Figure 6-9: Hop

Figure 6-10: Hop and Limp Ankle
Figure 6-11: Half Twist Right

Figure 6-12: Half Twist Left
Figure 6-13: Step Landing

Figure 6-14: Step Takeoff
Figure 6-15: Diagonal

Figure 6-16: Wide Landing
Figure 6-17: Step Takeoff and Landing

Figure 6-18: Bar Grab
Figure 6-19: Diagonal PCA

Figure 6-20: Half Twist Right PCA
Figure 6-21: Dunk

Figure 6-22: Dunk and Short Legs
Figure 6-23: Dunk and Long Torso

Figure 6-24: Cross Step

Figure 6-25: Wide Step
Figure 6-26: Run Initial Guess

Figure 6-27: Bouncy Step
Figure 6-28: Skip Step

Figure 6-29: Hurdle
Figure 6-30: Cross Step PCA

Figure 6-31: Wide Step PCA
Chapter 7

Conclusion

The method for adaptation of ballistic performances is inspired by the method of spacetime constraints introduced to computer graphics literature more than a decade ago. Spacetime constraints offer a powerful and highly intuitive setting for a generation of physically valid and realistic computer animations. By specifying high-level motion constraints such as the feet placement or the motion duration, the user can quickly synthesize new motions.

The subsequent work in this domain seemed to indicate that the spacetime constraints method became prohibitively slow for complex characters such as a human figure with many DOFs. Several researchers tackle this problem by simplifying either computer character configuration or its dynamics. This thesis shows that the spacetime constraints can be an effective way of adapting input human motions for characters with as many as 42 DOFs.

A full dynamics formulation derived from the first principles models joints, torques, and the reaction forces and enables the use of torque and joint-based optimization functions, which ensure the natural and smooth-looking final motions. An alternative, reduced-order formulation reduces the order of dynamics constraints and allows rapid motion prototyping. A tailored numerical procedure can exploit this formulation to select among the number of possible physical solutions the one that is closer to the input motion. Empirically, this approach produces motions that retain physical validity while improving solution speeds by an order of magnitude.

The adaptation method requires that the input motions be a recorded human performance or come from some other physically realistic motion source. Optimal trajectory
methods are also highly dependent on the proper selection of the motion curves' parameters, as well as the number of time slices at which dynamics and kinematics constraints are enforced. The user needs to tweak the number and placement of control points guiding the joint and torque trajectories. This work would greatly benefit from a principled selection method for time slices of the parameters and constraints. Some of the results mentioned in the previous section also suggest the possibility of combining the adaptation with the previous motion sequencing methods to produce a tool that is more robust to significant differences between the input and the final motion.

While the adaptation method works well on high dimensional human characters, I also explored the possibility of performing an automatic human character simplification. Common model reduction techniques, such as PCA, analyze the original data to produce low-dimensional character representation for a given motion activity. This method reduces the number of optimization variables by at least the factor of two, and it halves the number of constraints for optimizations that include the torques. Nevertheless, the results obtained with PCA-based model reductions are worse than the ones without the model reduction. While for some optimization formulations the reduced character configuration converges faster than the one with the full human figure, the high-dimensional character optimization always produces motions that appear more natural. The empirical results suggest that the linear projection of otherwise independent body joints (e.g., left and right hip) onto the same set of reduced variables creates a more difficult optimization problem, since the density of Jacobian of the constraints actually increases. This hinders the optimization process in its search for a motion that satisfies constraints, and consequently has a negative effect on the quality of the final motion.

Despite the failure of the model simplification to further improve the adaptation results, the number and versatility of the adapted motions demonstrate the effectiveness of the adaptation method when applied to full human characters. With a few improvements (e.g., automatic constraint allocation) to its otherwise highly intuitive interface and an already near-interactive computation speeds, the method described in this thesis could become an important addition to the toolbox of future computer animation tools.
Bibliography


