Adaptation of Performed Ballistic Motion

ADNAN SULEJMANPAŠIĆ and JOVAN POPOVIĆ
Massachusetts Institute of Technology

Adaptation of ballistic motion demands a technique that can make required adjustments in anticipation of flight periods when only some physically consistent changes are possible. This article describes a numerical procedure that adjusts a physically consistent motion to fulfill new adaptation requirements expressed in kinematic and dynamic constraints. This iterative procedure refines the original motion with a sequence of minimal adjustments, implicitly favoring motions that are similar to the original performance, and transforming any input motion, including those that are difficult to characterize with an objective function. In total, over twenty adaptations were generated from two recorded performances, a run and a jump, by varying foot placement, restricting muscle use, adding new environment constraints, and changing the length and mass of specific limbs.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; G.1.6 [**Numerical Analysis**]: Optimization—*Constrained Optimization*; G.1.7 [**Numerical Analysis**]: Ordinary Differential Equations—*Boundary value problems*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Animation with constraints, physically based animation, motion capture

1. INTRODUCTION

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. For example, before a twisting jump, a performer bends down, bursts upward to propel himself, and simultaneously spins his body to generate the angular momentum that twists his body in the air. An adaptation technique must emulate this anticipation when adding a twist to a recorded jump performance.

We describe a numerical method that creates such anticipation in ballistic motions of a full human figure with many (in our case, 42) degrees of freedom. This iterative algorithm modifies a physically consistent input motion with a sequence of minimal modifications until all adaptation goals are met. Each adjustment modifies the entire motion to anticipate the lack of control in flight with changes at other times, when the appropriate control is available. As a result, lengthening the jump with kinematic constraints produces a motion with a deeper bend in the knees before the figure leaps from the ground.[1]

[1]This approach is not required for motions without a ballistic period, such as walking and reaching; simpler kinematic techniques are effective in these cases [Bruderlin and Williams 1995; Witkin and Popović 1995].

This research was sponsored by the National Science Foundation and the NTT Corporation.
Authors' address: MIT CSAIL, The Stata Center, 32 Vassar Street, Cambridge, MA 02139; email: {adnans, jovan}@csail.mit.edu.

Existing approaches in the literature employ optimization methods and caution of convergence difficulties, particularly for physically based problems with many degrees of freedom. Our method does not explicitly optimize any objective function. Instead, it searches for a motion that meets the constraints with a bias for motions that are similar to the original performance. The absence of the objective function allows it to transform any physically consistent motion, including biological motions that are typically difficult to characterize with an objective function.

We describe a simple scaling strategy for drastically improving the convergence of our iterative method. Unlike the previous methods, which simplified the state space or the governing dynamics equations, the scaling permits adaptations with restricted muscle use and with general interaction between the character and the environment. The same scaling strategy also improves the optimal trajectory formulations such as the method of spacetime constraints [Witkin and Kass 1988], which allows us to compare such optimizations with our method.

2. BACKGROUND

Motion adaptation has emerged as an important computer graphics problem with the maturation of acquisition technology for archiving high-quality human motion. The first methods broadened the applicability of acquired motion with the introduction of tools that displace (warp) motion to satisfy new kinematic constraints [Bruderlin and Williams 1995; Witkin and Popović 1995]. An animator can displace the foot in a few keyframes and the editing tool propagates the changes to the remaining frames with the interpolated displacements. Although this fast and simple approach is sufficient in many cases, it only satisfies kinematic constraints at the displaced keyframes. If animators want to ensure that the feet do not go below the ground at any point in the animation, they have to introduce displacements at every frame, which defeats the purpose of using the automatic techniques in the first place.

Currently, these problems can be resolved with a technique that alternates between inverse kinematics, which enforces specified constraints, and filtering, which smooths the changes [Lee and Shin 1999]. A more general approach, and the one that is closer to our technique, is an optimal trajectory method that minimizes the difference from the original motion, while enforcing kinematic constraints throughout the motion [Gleicher 1997, 1998]. The motion retargeting techniques described in these papers do not generate physically consistent motions, but suggest applying a sequence of minimal modifications to the input motion. This is also an approach we use to adapt motions in a physically consistent manner.

Optimal trajectory methods were introduced to computer graphics by Witkin and Kass [1988], who developed the spacetime constraints technique for motion synthesis. Dynamics is an integral part of that formulation, which showed that by optimizing power consumption, a numerical procedure could select a natural motion among the many physically valid alternatives. Extending these observations to generate natural motions for a full human figure proved difficult [Cohen 1992; Liu et al. 1994] even as complex simulations of the human musculoskeletal system were verified by experimental data [Pandy and Anderson 2000]. The one exception was a technique for generating short motion transitions [Rose et al. 1996].

It seemed necessary to simplify the state space by reducing the number of degrees of freedom, or to approximate the governing dynamics equations. The first technique for adaptation of ballistic motion projected the original motion onto a simple character, where it made the adjustments using the simplified dynamics, before projecting the changes back onto the full human figure [Popović and Witkin 1999]. However, the final projection step does not produce a physically consistent final motion. Our technique produces physically consistent motions without any simplification.

The governing physical laws could also be simplified with *momentum constraints*, which enforce typical momentum profiles [Liu and Popović 2002; Abe et al. 2004], or with *aggregate-force constraints*, which eliminate internal torques and constrain only the total (aggregate) force [Fang and Pollard 2003].

Fig. 1. The adaptation of a human broad jump (left figure) generates a new physically consistent jump (right figure) with a staggered takeoff and landing. An animator specifies only the new foot placement to evoke this change.

These simplifications yield efficient adaptation methods at the cost of ignoring the internal muscle use. This implies that an animator is not allowed to restrict muscle usage and not allowed to specify an efficiency criterion that depends on such muscle activation.

Another option is to reparameterize the degrees of freedom with principal components through statistical analysis of variations observed in a specific behavior [Safonova et al. 2004]. The inferred low-dimensional representation facilitates convergence of optimal trajectory methods without simplifying dynamics. This approach complements our technique, but must be used carefully to construct low-dimensional subspaces that are capable of encoding motions that satisfy arbitrary kinematic constraints [Safonova et al. 2004; Sulejmanpašić 2004].

Optimal trajectory methods are ideally suited for the adaptation of ballistic motions that require anticipatory planning. In scenarios in which anticipation plays a less prominent role, dynamic properties such as balancing or physically valid interaction with the environment might be more easily accomplished with kinematic and dynamic filtering [Zordan and Hodgins 1999; Yamane and Nakamura 2000; Tak et al. 2002; Shin et al. 2003]. These approaches are effective if the adjustments are small, and the required look-ahead is localized. When this is not the case, the simultaneous approach of optimal trajectory methods and our trajectory adaptation is a better choice.

3. ADAPTATION

Adaptation reuses a recorded human motion by conforming it to make the required changes. The entire process resembles the standard keyframing techniques, which allow the animator to specify positions of hands, feet, or other end-effectors, with the added capability to enforce laws of physics, limit the use of specific muscles, and otherwise restrict the dynamics with constraints. Figure 1 shows an adaptation of a broad jump: the input motion, in which the performer jumps and lands with parallel footing, is transformed into a physically valid jump with staggered foot placement.

3.1 Formulation

Adaptation computes the motion $\mathbf{q}$, the internal muscle use $\mathbf{f}$, and the external environment-reaction forces determined by the Lagrange multipliers $\boldsymbol{\lambda}$. The solution satisfies the adaptation goals expressed in kinematic, $\mathbf{K}$, and dynamic, $\mathbf{D}$, constraints[2]:

$$\begin{aligned} \mathbf{K}(\mathbf{q}) &= 0 \\ \mathbf{D}(\mathbf{q}, \mathbf{f}, \boldsymbol{\lambda}) &= 0. \end{aligned} \tag{1}$$

[2]Inequality constraints, which are excluded from Equation (1) for simplicity, are also supported with an active-set method, which replaces the active inequality constraints with equalities.

Here, the dynamic constraints contain both the governing physical equations and other restrictions on dynamic quantities such as ground-reaction forces and internal muscle usage.

The constraints alone do not uniquely determine the desired trajectories. Instead, the transformation seeks to identify natural motions within the set of feasible trajectories that satisfy the constraints. Optimal trajectory methods accomplish this goal by maximizing the smoothness of muscle actuation [Popović and Witkin 1999] or the proximity to the input motion [Gleicher 1997]. Our method does not explicitly optimize any such criteria: instead, it modifies input trajectories with a sequence of alterations until it constructs trajectories that meet the constraints. This choice allows us to transform any input motion, including those that are difficult to characterize with an objective function, as is often the case with natural motions. Our method applies minimal alterations throughout the transformation process to keep the resulting motion close to the original input, thereby preserving its natural characteristics. Section 4.1 describes a gradient method for computing these alterations efficiently.

3.2 Dynamic Constraints

Dynamic constraints ensure physical consistency and provide additional controls for animators. For example, animators can restrict the use of a specific muscle with inequality constraints on joint torques. The nonlinear equality constraints that enforce the laws of physics are hardest to fulfill and their formulation has a profound effect on the efficiency of the numerical solution. Their proper scaling improves convergence and enables adaptation of motions with many degrees of freedom. When internal torques are not restricted, this performance can be further improved with a formulation that eliminates the internal torques to reduce the order of constraints [Fang and Pollard 2003].

3.2.1 *Standard Formulation.* The differential algebraic equations that express Newton's laws are derived with the Euler-Lagrange equations from the Lagrangian L of the human figure:

$$\mathbf{D}_T(\mathbf{q}, \mathbf{f}, \boldsymbol{\lambda}) \equiv \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_r}\right) - \frac{\partial L}{\partial q_r} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{P}}{\partial q_r} \\ \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{P}}{\partial q_i} - f_i \\ \mathbf{P}(\mathbf{q}) \end{pmatrix} = 0, \tag{2}$$

where the index r enumerates the degrees of freedom in the unactuated root joint (global position and global orientation) and the index i enumerates the remaining degrees of freedom in joints actuated by muscle forces f_i. For a human figure, the complexity of Euler-Lagrange equations and associated derivatives necessitates systematic evaluation [Liu 1996] and elimination of redundant computations with caching or a recursive formulation [Hollerbach 1980].

The algebraic equations $\mathbf{P}$ and the matching Lagrange multipliers $\boldsymbol{\lambda}$ define the dynamic interaction between the character and the environment.[3] For example, during ground contact these constraints maintain the contact between the feet and the ground. Such holonomic constraints produce algebraic equations that depend only on the kinematic quantities, but they are still a component of the dynamics constraints because of the associated multiplier terms $\boldsymbol{\lambda}^T \frac{\partial \mathbf{P}}{\partial q}$, which describe the exchange of forces between the character and the environment. In contrast, pure kinematic constraints are separated from the dynamic constraints because they do not have the associated multiplier terms.

3.2.2 *Low-Order Formulation.* When internal torques f_i are not bound by the objective function (to select a natural motion) or by the constraints (to restrict the use of a muscle), the torques are free

[3]Another option is to select an independent set of reduced coordinates for which the environment constraints are satisfied by construction. Section 3.2.3 discusses the two design choices in more detail.

variables, and can be eliminated from the optimization problem:

$$\mathbf{D}_G(\mathbf{q}, \boldsymbol{\lambda}) \equiv \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_r}\right) - \frac{\partial L}{\partial q_r} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{P}}{\partial q_r} \\ \mathbf{P}(\mathbf{q}) \end{pmatrix} = 0. \tag{3}$$

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

Eliminating optimization variables and lowering the order of dynamic constraints improves the efficiency of numerical solutions—often by an order of magnitude—but might have an adverse effect on the smoothness 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 (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 low-order constraints of Equation (3) enables rapid prototyping before the final refinements and smoothing are made.

3.2.3 *Discussion.* The low-order dynamics constraints are equivalent to the aggregate-force constraints introduced by Fang and Pollard [2003]. Fang and Pollard use reduced coordinates to enforce environment constraints explicitly. In contrast, our formulation adds Lagrange multipliers to determine the forces needed to enforce the environment constraints. Baraff [1996] summarizes the differences between the reduced and multiplier formulations for general applications in computer graphics. For the adaptation problem, we prefer the multiplier formulation for its explicit modeling of reaction forces and for its systematic treatment of all environment constraints $\mathbf{P}$, especially 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 [Liu and Popović 2002]. In flight, they 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. These momentum patterns emerge naturally from our aggregate-force formulation. In addition to this generality, our formulation exposes the reaction forces and enables their use in objective functions (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 [Fang and Pollard 2003]).

The two expressions for dynamics, with torques $\mathbf{D}_T$ and without torques $\mathbf{D}_G$, 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.

4. NUMERICAL SOLUTION

Our numerical method approximates trajectories with the cubic B-spline expansion [Cohen 1992] and computes the expansion coefficients with collocation by fulfilling kinematic and dynamic constraints at

prescribed time points. Because of the large number of unknowns, we employ a gradient method that computes the adjustments by solving a sequence of quadratic subproblems with linearized constraints. Proper scaling of constraint equations ensures efficient convergence of this method.

4.1 Gradient Method

Our adaptation technique selects a natural motion with a sequence of minimal modifications that iteratively modify the original motion until all adaptation goals are met. The gradient method alternates between computing the direction vector $\mathbf{d}_i$ of smallest magnitude, and computing the step size $\alpha_i \in (0, 1]$ for the modification in this direction. The small size of the individual direction vectors ensures that the cumulative change is small, even though the proximity to the original motion is never enforced explicitly.

At each iteration, the direction for the next modification $\mathbf{d}_i$ is a minimum-norm solution of an underdetermined linear system produced by a linearization of the kinematic and dynamic constraints $\mathbf{C} = (\mathbf{K}, \mathbf{D})$:

$$\begin{aligned} \mathbf{d}_i = \arg\min_{\mathbf{d}} \quad & \|\mathbf{d}\|^2 \\ \text{subject to} \quad & \mathbf{C}(\mathbf{x}_i) + \frac{\partial \mathbf{C}}{\partial \mathbf{x}}(\mathbf{x}_i)\, \mathbf{d} = 0. \end{aligned} \tag{4}$$

The current iterate $\mathbf{x}_i$ and the adjustment $\mathbf{d}_i$ contain the coefficients for joint angles, torques, and multipliers. In effect, the solution to this quadratic problem computes the smallest change in joint angles, reaction forces, and muscle actuations.

The dimensions of the Jacobian matrix $\partial \mathbf{C}/\partial \mathbf{x}$ 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 [Demmel and Higham 1993]. In our implementation, we chose the quadratic programming technique SQOPT [Gill et al. 1997] to support inequality constraints with sparse LU factorization of the Jacobian matrix.

The minimum-norm solution defines the direction for the line search, which computes the step size α_i for the modification that minimizes the distance of the next iterate $\mathbf{x}_{i+1}$ from the adaptation constraints[4]:

$$\begin{aligned} \alpha_i &= \arg\min_{\alpha} \quad \|\mathbf{C}(\mathbf{x}_i + \alpha \mathbf{d}_i)\|^2, \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha_i \mathbf{d}_i. \end{aligned} \tag{5}$$

The iteration stops once the constraints are satisfied with the desired accuracy. Although the magnitude of the total adjustment could be improved by additional steps along the constraints surface, these improvements have negligible effect on the visual quality of the resulting motion. A similar effect was also observed in adaptation problems without dynamics [Gleicher 2004].

4.2 Scaling

Under restrictive theoretical assumptions, some gradient methods can be shown to produce the same sequence of iterates regardless of the scaling. In practice, this scale invariance cannot be achieved and proper scaling is essential to resolve the difficulties in the conditioning of difficult nonlinear problems [Gill et al. 1989]. In our experiments, proper scaling of joint angles, torques, and multipliers improved efficiency and prevented divergence.

[4]The same approach would accommodate a weighted norm, but our scaling made such norm unnecessary for our experiments.

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 with collocation and gradient methods. Without scaling, the computation of the pendulum trajectory requires 129 iterations and 11.6 seconds of 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 parameters of the gradient procedure are identical: the pendulum starts in a configuration perpendicular to the direction of gravity and swings for 0.6 s, ending just before the back swing. On adaptation problems, the effect is even more drastic: improper scaling 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 affects the scaling of the Jacobian matrices. The simplest solution is to scale the mass density of each limb by a uniform constant factor s, and to solve for the new joint angles $\mathbf{q}'$, torques $\mathbf{f}'$, and Lagrange multipliers $\boldsymbol{\lambda}'$. This scaling changes the Lagrangian $L' = sL$ and the expression of the dynamics in terms of the unscaled Lagrangian L:

$$\mathbf{D}'_T(\mathbf{q}', \mathbf{f}', \boldsymbol{\lambda}') \equiv \begin{pmatrix} s\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}'_r}\right) - s\frac{\partial L}{\partial q'_r} - \boldsymbol{\lambda}'^T \frac{\partial \mathbf{P}}{\partial q'_r} \\ s\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}'_i}\right) - s\frac{\partial L}{\partial q'_i} - \boldsymbol{\lambda}'^T \frac{\partial \mathbf{P}}{\partial q'_i} - f'_i \\ \mathbf{P}(\mathbf{q}') \end{pmatrix}. \tag{6}$$

As division by the factor s reveals, this scales the values of torques and Lagrange multipliers without changing the joint-angle trajectories:

$$\mathbf{D}'_T(\mathbf{q}', \mathbf{f}', \boldsymbol{\lambda}') = \mathbf{D}_T\left(\mathbf{q}, \frac{\mathbf{f}}{s}, \frac{\boldsymbol{\lambda}}{s}\right). \tag{7}$$

Loss of accuracy, which could prevent accurate specification of kinematic constraints, is one possible drawback of this simple scaling transformation. We avoid this problem in our experiments by empirically choosing the scale factor s = 0.001. We found that larger values of 0.1 and 0.01 degraded convergence, and smaller values of 0.0001 and less interfered with our internal constant, which determines the feasibility of each constraint. The same scaling factor worked in all of our experiments, but should the loss of accuracy become problematic with this simple strategy, proper scaling could also be established by determining the range of torque and multiplier forces precisely [Gill et al. 1989].

Scaling is not as critical for alternative formulations with momentum constraints [Liu and Popović 2002] and constrained aggregate forces [Fang and Pollard 2003]. Momentum constraints do not compute Lagrange multipliers as they do not model ground-reaction forces. The aggregate-force constraints use reduced coordinates, which do not employ Lagrange multipliers. This leaves only joint-angle trajectories for unknowns, as internal torques are also not computed in either formulation. Because joint angles have identical units, the problem is scaled automatically, but this convenience prevents these alternatives from solving adaptation problems that must restrict muscle forces.

4.3 Initialization

The initial values $\mathbf{q}$, $\mathbf{f}$, and $\boldsymbol{\lambda}$ are estimated by fitting the joint-angle trajectories $\bar{\mathbf{q}}$ in the original performance. This initialization solves an optimal trajectory problem with the least squares objective

function:

$$\begin{aligned} \arg \min_{\mathbf{q}(t),\mathbf{f}(t),\boldsymbol{\lambda}(t)} \quad & \int \|\mathbf{q}(t) - \bar{\mathbf{q}}(t)\|^2 \mathrm{d}t \\ \text{subject to} \quad & \mathbf{K}(\mathbf{q}) = 0 \\ & \mathbf{D}(\mathbf{q}, \mathbf{f}, \boldsymbol{\lambda}) = 0. \end{aligned} \tag{8}$$

A solution to a similar optimization problem was previously used to estimate the parameters of a simplified character from the original performance [Popović and Witkin 1999]. The key difference is that our estimation computes joint angles, torques, and multipliers for the entire human figure using the full dynamics of Equation (2), without any simplification.

For this optimization, the torques and multipliers are initially set to zero, and the joint angles are computed with cubic B-spline interpolation of joint angles in the original performance. Initialization is performed only once for each recorded performance. Subsequently, the estimated trajectories are stored with the motion and used for every adaptation problem.

The 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 article, a different initialization procedure might eliminate this requirement by consulting a stored database of transition poses [Liu and Popović 2002], or by restricting the ground-reaction forces [Fang and Pollard 2003].

5. RESULTS

In our adaptation experiments, we varied foot placement, restricted the use of specific muscles, introduced new environment constraints, added additional weights, and changed the skeleton dimensions to generate over twenty adaptations from two performances: a run and a jump. The distinctive attributes of the original motion were visibly preserved in all examples.

In these experiments, we needed only a few constraints to indicate, the desired adjustments. In contrast, kinematic techniques would require many manually specified constraints to ensure realism of resulting motions. We relied primarily on point constraints to indicate placement of feet and hands. For aesthetic reasons, cone constraints were used to enforce natural joint-angle limits and to prevent interpenetration of limbs. Full pose constraints were used sparingly because they were not necessary to generate high-quality adaptations, and because they are, in general, difficult to specify. On occasion, it was easier to prevent limb intersections with a full pose constraint than with a cone constraint for each joint. The figures include a visual illustration of almost all kinematic constraints required to generate the motion shown. The only exceptions are inequality constraints (feet above the ground, feet clearing the hurdle, or hand above the rim) and orientation constraints, which specified starting and final body orientation for the twist jumps.

5.1 Discretization

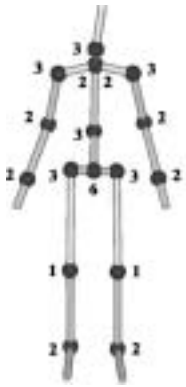
Our implementation uses a collocation method to enforce the constraints at designated discrete time points, which are also called collocation points. During takeoff, landing, and other periods with contact between the character and the environment, the environment constraints maintain the contact and the Lagrange multipliers determine the associated reaction forces. These multipliers and environment constraints are not part of the dynamic constraints evaluated on collocation points in periods of free flight.

The unknowns, which include the joint-angles, torques, and multipliers, are also discretized. The Lagrange multipliers $\boldsymbol{\lambda}$ are sampled at the collocation points, and the joint-angle $\mathbf{q}$ and torque trajectories $\mathbf{f}$ are parameterized with cubic B-spline curves. Because the number of multiplier samples and

Table I.

Motion	**q** c. p.	**f** c. p.	λ samples	unknowns		dynamic constraints	
				reduced	full	reduced	full
Jump	24	30	20	1128	2388	288	1296
Run	30	38	20	1320	2916	276	1572

Our implementation approximates each joint-angle **q** and torque trajectory **f** with a cubic B-spline curve. The Lagrange multipliers λ are sampled at prescribed points that enforce the environment constraints and are not approximated by a cubic B-spline curve. Our adaptation can use the full dynamics of Equation (2), or the reduced formulation of Equation (3). If the internal torques are not smoothed or restricted, then the reduced formulation enables faster convergence.



Body Part	% Mass	Body Part	% Mass
Pelvis	15.3	Upper arm	3.3
Thorax	22	Forearm	1.9
Clavicle	4	Hand	0.6
Head	7.1	Thigh	10.5
Shank	6	Foot	1.5

Fig. 2. Motion adaptation adjusts motions of high-dimensional human characters. A skeleton configuration with 42 degrees of freedom was used in our adaptation experiments. The mass of each limb was assigned as a percentage of the total body mass.

B-spline control points determines the total number of unknowns,[5] the appropriate setting is chosen through trial-and-error to minimize the total number of unknowns. In general, we begin by matching the control points, and collocation points, and then reduce the number of control points until we reach the minimum necessary to avoid an overconstrained formulation. Table I lists the number of samples and control points used in our experiments.

5.2 Initialization

The original performances were captured in a motion-capture studio. Standard commercial tools [Vicon 2003] were used to compute the limb lengths for the 42 degree-of-freedom character shown in Figure 2. The mass of each limb is computed as a percentage of the total weight with the ratios shown in Figure 2. These ratios were manually adjusted for the limbs in our skeleton from the weight ratios reported in a standard biomechanical reference [Winter 1990].

Given the inferred physical parameters and the original motion, the initialization (Section 4.3) estimates the joint-angle trajectories that are physically consistent with internal muscle use and the environmental reaction forces. For this process, the environmental constraints and the duration of each ground-contact and free-flight stage were identified manually by selecting the appropriate frames in the original performance.

[5]In a jump, for example, the unknowns for the full dynamics of Equation (2) include $24 \times 42 = 1008$ joint-angle control points (24 for each of 42 degrees of freedom), $30 \times 42 = 1260$ torque control points, and $20 \times 6 = 120$, lagrange multiplier samples (6 multipliers for location of both feet in each of 20 time samples). The reduced dynamics of Equation (3) have fewer unknowns because they do not include torques.

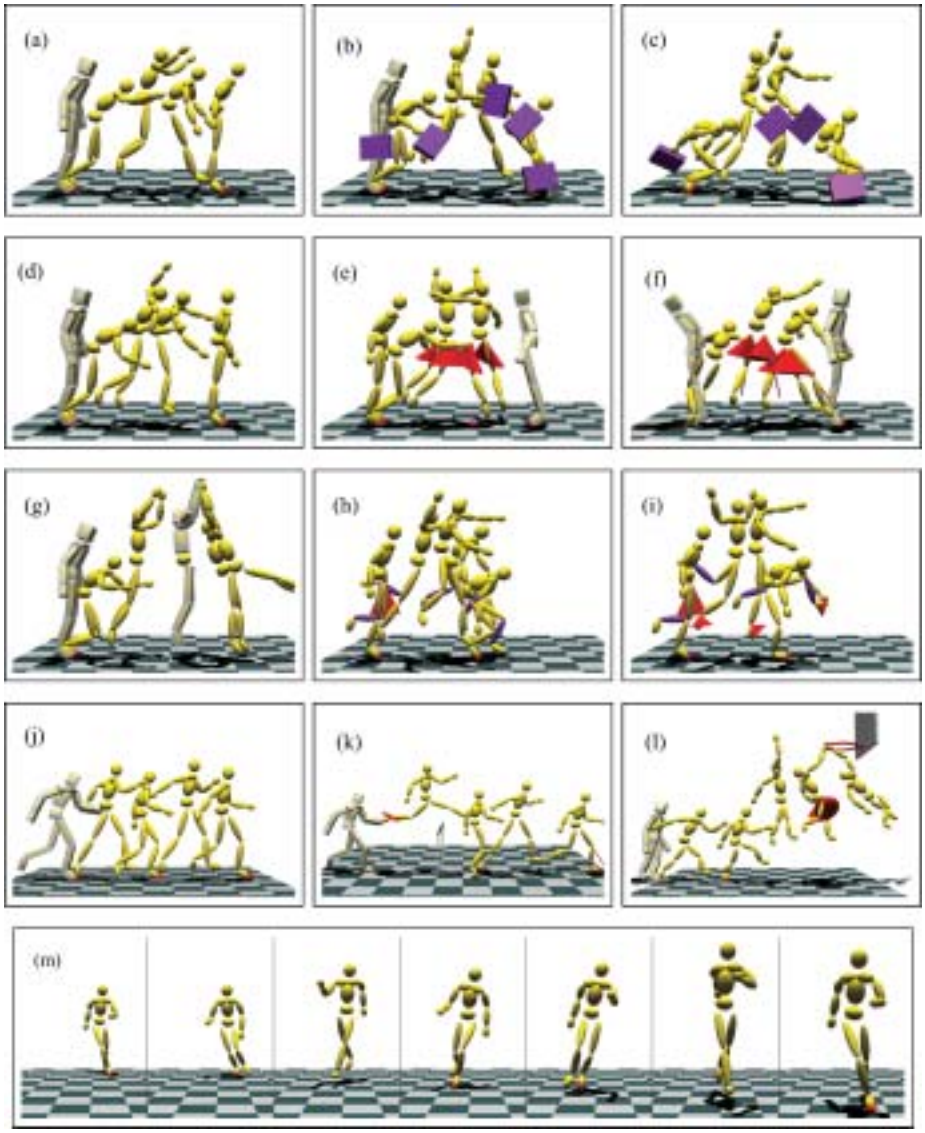


Fig. 3. All motions were generated by adapting two performances: a human jump (a) and a run (j). The figures visualize the constraints required for each adaptation. Red spheres represent point constraints, red cones the joint-limit constraints visualized by the limits drawn around the corresponding limb, and the transparent gray boxes identify pose constraints, which constrain the values of all joints in the skeleton. The shin is colored purple when the knee is constrained to a fixed value. A line extending from a knee shows the maximum allowed range for a knee joint. Some inequality and orientation constraints are not shown.

5.3 Jumps

The original jump, after initialization, is shown in Figure 3(a). Our adaptation process transforms this jump into new motions that meet the desired constraints. In Figure 1, for example, the constraints specify new foot placement, and the adaptation produces a jump with staggered takeoff and landing. In Figure 3(b), the constraints maintain the foot placement in the original jump and add a briefcase

to increase the weight of the right arm. The adaptation lowers the motion of the heavy right arm and compensates by raising the lighter left arm higher. This motion is then further refined in Figure 3(c), with a constraint that limits the torque applied around the right ankle joint. This upper bound simulates the effect of an injured right ankle. The healthy left leg carries more weight throughout the takeoff and trails the injured right leg to generate momentum needed for the liftoff. In the air, the same leg hurries forward to absorb most of the landing impact. In Figure 3(d), a similar upper bound constrains the torque applied at the right shoulder joint. As a result, the limp right arm dangles throughout the motion and the upper body compensates to swing it forward. As with a briefcase jump, the healthy arm is lifted higher to increase the vertical momentum.

In Figures 3(e) and 3(f), the constraints enforce the pelvis orientation, along with the foot placement. This produces twisting jumps: first in 3(e) with the original takeoff and modified landing, and then in 3(f) with the modified takeoff and the original landing. In the second example, we needed to adjust the original jump to rotate the takeoff stage. This manual adjustment is a limitation that's revisited in Section 5.6. Still, such adjustments are simple to perform because our adaptation process resolves any discontinuities such as the gap between the takeoff stage and the remaining motion.

In Figure 3(g), a new environment constraint replaces the ground with a high bar. In the new "landing" stage, the feet are unconstrained and the constraints maintain the contact between the hands and the high bar. As a result, the character jumps, reaches for the bar, and swings while holding onto it. To initialize the optimization, we only use the first half of the original jump, from the start to the flight apex. The last frame of this sequence was extended, as a still frame, for the duration of the swinging stage. The adaptation adjusts the entire motion to produce a physically consistent motion that satisfies new environment constraints.

Both one-legged hops shown in Figure 3(h) and 3(i) were produced with adaptations that maintain a fixed angle in the right knee. The first adaptation mimics the broad jump to create a character that absorbs most of the impact with torques applied at the hip and knee joints. The second adaptation produces a more typical one-legged landing by enforcing additional constraints that restrict the torques applied at these two joints. As a result, the character avoids excessive strain by landing with an extended leg.

5.4 Runs

The original run is shown in Figure 3(j). Our adaptation produces a hurdle jump shown in Figure 3(k) by meeting the specified constraints: increased distance between the ground contacts and height required to clear the hurdle. New distances are enforced by displacing all footprints. This generates bouncier runs with longer strides. The hurdle constraint specifies a lower bound on the height attained by the left and right toe. The dunk motion in Figure 3(l) is similar to the high-bar example. A new environment constraint enforces the contact with the rim. Just as in the high-bar example, the last frame of the run cycle was extended, as a still frame, for the duration of the swing stage. We've also modified the skeleton by shortening its legs. The adaptation process compensates by producing a more explosive motion. Cone constraints were added to prevent interpenetration of limbs. In Figure 3(m), the adaptation produces a cross-step run with the proper step-to-step lean by satisfying the displaced footprints.

5.5 Comparison

We compared the efficiency of our adaptation method with the optimal trajectory method that employed sequential quadratic programming (SQP) [Gill et al. 2002]. SQP is an iterative descent technique that computes the optimal trajectory by minimizing the merit function along search directions given by solutions to quadratic programming subproblems. The merit function balances the competing goals of improving the objective function and remaining on the nonlinear constraint surface, while the quadratic

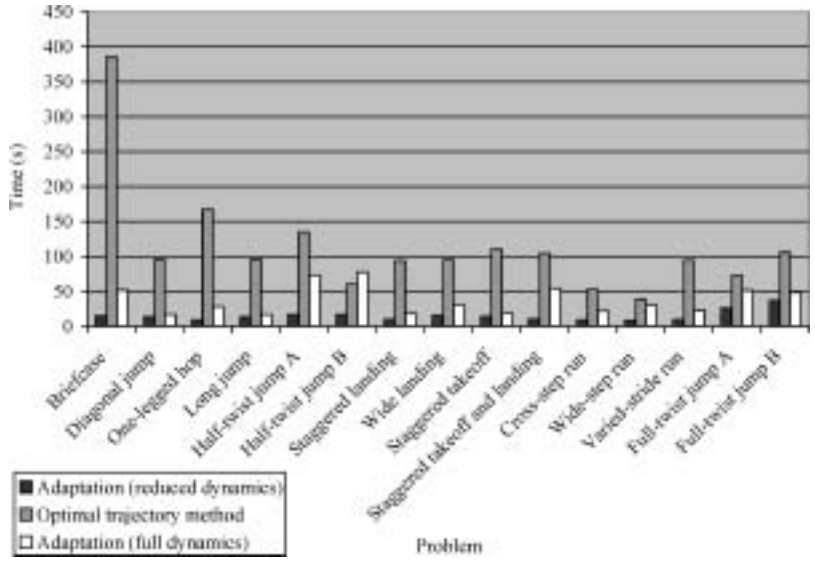


Fig. 4. The timing chart compares the efficiency of our adaptation method—with and without reduced dynamics—against the best performing optimal trajectory method that maximized the smoothness of muscle actuation. The corpus of problems includes all adaptations on which all three formulations were applicable. The adaptations were timed on a single-processor 1.7 GHz Pentium III workstation with 1 GB of main memory. In general, the visual quality of each adaptation was similar, but on occasion, the adaptation with reduced dynamics produced jerky motions.

subproblems linearize constraints and approximate the merit function with a quadratic expansion around the current iterate. Just as with our gradient method, proper scaling and initialization ensured that the SQP method solves adaptation problems for a full human figure.

The SQP method uses an objective function to identify a unique motion that satisfies specified constraints. Prior literature proposed several objective functions with the aim of selecting motions that appear most natural. The synthesis applications optimized power consumption ($\int \|\mathbf{f}\cdot\dot{\boldsymbol{\theta}}\|^2 \mathrm{d}t$) [Witkin and Kass 1988], torque output ($\int \|\mathbf{f}\|^2 \mathrm{d}t$) [Liu et al. 1994], torque smoothness ($\int \|\ddot{\mathbf{f}}\|^2 \mathrm{d}t$) [Popović and Witkin 1999], and kinematic smoothness ($\int \|\ddot{\boldsymbol{\theta}}\|^2 \mathrm{d}t$) [Fang and Pollard 2003], while the adaptation applications minimized either joint-angle displacement [Gleicher 1997], or mass displacement [Popović and Witkin 1999] from the original motion.

In our experiments with motions from Figures 4 and 3, optimization of torque smoothness consistently generated better results than any other objective function. The power-consumption objective, which selects the most efficient motion, produced natural motions on some adaptation problems, but performed poorly in many cases, especially during adaptation of running motions. The most efficient motions frequently lead to interpenetration of limbs (between the arms and the torso or the two legs), which could not be resolved with joint-limit constraints. Optimization of torque output demonstrated identical problems and also produced jerky motions unless used in combination with the kinematic-smoothness objective. Similarly, the mass displacement objective, a kinematic analogue of the power consumption, had to be applied in conjunction with kinematic smoothness to prevent the human figure from jumping with a still body, and using only its ankles. We opted not to combine multiple objective functions because the quality of the resulting motion became highly dependent on weights assigned to each component in the objective function. Optimization of kinematic smoothness alone produced motions that are smooth, but not natural. For example, during the jump takeoff, the body curled up into an

unnatural position from which it would smoothly, but unnaturally, unwind to gain the appropriate momentum.

Figure 4 compares our adaptation technique (full and reduced dynamics) with the best performing optimal formulation that maximizes the smoothness of internal torques. Note that proper scaling of optimization variables permits efficient convergence of the optimal formulation even for the skeleton with many degrees of freedom. With reduced dynamics, our adaptation technique converges faster than the optimal formulation, by an order of magnitude on almost all problems. This improvement is smaller with the full dynamics, which suggests that adaptation with reduced and full dynamics is complementary. Once the desired constraints are met in rapid prototyping with the reduced dynamics, the full dynamics can be used to generate smoother motions.

5.6 Limitations

The extent of any adaptation has its limits. If the difference between the initial and the final motion is very large, motion adaptation will not be able to converge to a natural motion. For example, although we could produce a jump with a half-twist directly from the original motion, the jump with a full-twist required that we manually rotate the takeoff and the landing stages. Our technique, however, eliminates the most tedious work. It removes the discontinuities between the flight and the transformed ground stages, and adjusts them to add the required anticipation. The success of this simple approach suggests that a combination of sequencing methods (e.g., Kovar et al. [2002]) and motion adaptation could further extend the value of the original motion.

Optimization is sensitive to discretization of joint-angle and torque trajectories. If too few control points are chosen for the cubic B-spline approximation, the optimization might not converge because it cannot satisfy the dynamic constraints. On the other hand, with too many control points, the solution will satisfy the dynamics at prescribed time points, but will not look physically correct in between. Often, trial-and-error is the best way to determine the settings. Table I shows the settings used for the adaptation of jumps and runs.

6. CONCLUSION

Motion adaptation with dynamics simplifies the generation of very complex human motions. An animator need specify only a sparse set of keyframes to define the goals for the motion, and the adaptation will generate a new natural motion automatically. This method, like previously published optimal trajectory techniques, benefits from restrictions imposed by the dynamics of ballistic motion—the same restrictions that make manual adjustments particularly tedious. Although adaptation in the previously published work relies on the simplification of dynamics constraints, this article demonstrates that such simplifications are not necessary. This generalization allows for the specification of torque-based objective functions and torque-based constraints and broadens the set of motions that can be generated with an adaptation approach.

The adaptation problem provides a common ground for evaluation and comparison of optimal trajectory methods. In this article, we define a numerical solution that adapts a motion with a biased search for trajectories that satisfy the adaptation goals. The search favors the motions that are close to the original performance to preserve the distinctive attributes of the original performance. Our method is compared to an explicit optimization of muscle actuations on an identical set of adaptation problems. The results show that our method efficiently produces visually indistinguishable solutions. In the future, we would like to improve the robustness with automated selection of collocation points, and to explore techniques that use sequencing methods (e.g., Kovar et al. [2002]) to construct a better initial guess for each adaptation problem.

REFERENCES

ABE, Y., LIU, C. K., AND POPOVIĆ, Z. 2004. Momentum-based parameterization of dynamic character motion. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. At press.

BARAFF, D. 1996. Linear-time dynamics using lagrange multipliers. In *Proceedings of SIGGRAPH 96*. Computer Graphics Proceedings, Annual Conference Series. 137–146.

BRUDERLIN, A. AND WILLIAMS, L. 1995. Motion signal processing. In *Proceedings of SIGGRAPH 95*. Computer Graphics Proceedings, Annual Conference Series. 97–104.

COHEN, M. F. 1992. Interactive spacetime control for animation. In *Comput. Graph. (Proceedings of SIGGRAPH 92)*. Annual Conference Series. ACM, 293–302.

DEMMEL, J. W. AND HIGHAM, N. J. 1993. Improved error bounds for underdetermined system solvers. *SIAM J. Matrix Anal. Applicat. 14*, 1–14.

FANG, A. C. AND POLLARD, N. S. 2003. Efficient synthesis of physically valid human motion. *ACM Trans. Graph. 22*, 3 (July), 417–426.

GILL, P. E., MURRAY, W., AND SAUNDERS, M. A. 1997. User's guide for SQOPT 5.3: A fortran package for large-scale linear and quadratic programming. Tech. Rep. NA 97-4, University of California, San Diego.

GILL, P. E., MURRAY, W., AND SAUNDERS, M. A. 2002. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM J. Optimiz. 12*, 4, 979–1006.

GILL, P. E., MURRAY, W., AND WRIGHT, M. H. 1989. *Practical Optimization*. Academic Press, London.

GLEICHER, M. 1997. Motion editing with spacetime constraints. In *Symposium on Interactive 3D Graphics*. 139–148.

GLEICHER, M. 1998. Retargeting motion to new characters. In *Proceedings of SIGGRAPH 98*. Computer Graphics Proceedings, Annual Conference Series. 33–42.

GLEICHER, M. 2004. Personal communication.

HOLLERBACH, J. M. 1980. A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Syst., Man, Cybernet. 10*, 11 (Nov.), 730–736.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. In *Comput. Graph. (Proceedings of SIGGRAPH 2002)*. Annual Conference Series. ACM SIGGRAPH, 473–482.

LEE, J. AND SHIN, S. Y. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH 99*. Computer Graphics Proceedings, Annual Conference Series. 39–48.

LIU, C. K. AND POPOVIĆ, Z. 2002. Synthesis of complex dynamic character motion from simple animations. *ACM Trans. Graph. 21*, 3 (July), 408–416.

LIU, Z. 1996. Efficient animation techniques balancing both user control and physical realism. Ph.D. thesis, Princeton University.

LIU, Z., GORTLER, S. J., AND COHEN, M. F. 1994. Hierarchical spacetime control. In *Proceedings of SIGGRAPH, 94*. Computer Graphics Proceedings, Annual Conference Series. 35–42.

PANDY, M. G. AND ANDERSON, F. C. 2000. Dynamic simulation of human movement using large-scale models of the body. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 676–681.

POPOVIĆ, Z. AND WITKIN, A. P. 1999. Physically based motion transformation. In *Proceedings of SIGGRAPH 99*. Computer Graphics Proceedings, Annual Conference Series. 11–20.

ROSE, C. F., GUENTER, B., BODENHEIMER, B., AND COHEN, M. F. 1996. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of SIGGRAPH 96, Computer Graphics*. Annual Conference Series. 147–154.

SAFONOVA, A., HODGINS, J., AND POLLARD, N. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3 (Aug.), 514–521.

SHIN, H. J., KOVAR, L., AND GLEICHER, M. 2003. Physical touchup of human motions. In *Proceedings 11th Pacific Conference on Computer Graphics and Applications*. 194–203.

SULEJMANPAŠIĆ, A. 2004. Adaptation of performed ballistic motion. M.S. thesis, Massachusetts Institution of Technology.

TAK, S., YOUNG SONG, O., AND KO, H.-S. 2002. Spacetime sweeping: An interactive dynamic constraints solver. In *Proceedings of the Computer Animation (CA 2002)*. IEEE Computer Society, 261–270.

VICON. 2003. *Vicon iQ Reference Manual*. Vicon Motion Systems Inc., Lake Forest, CA.

WINTER, D. A. 1990. *Biomechanics and Motor Control of Human Movement*, 2nd Ed. John Wiley and Sons, Inc., New York.

WITKIN, A. AND KASS, M. 1988. Spacetime constraints. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, Vol. 22. 159–168.

WITKIN, A. AND POPOVIĆ, Z. 1995. Motion warping. In *Proceedings of SIGGRAPH 95*. Computer Graphics Proceedings, Annual Conference Series. 105–108.

YAMANE, K. AND NAKAMURA, Y. 2000. Dynamics filter—concept and implementation of on-line motion generator for human figures. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 688–694.

ZORDAN, V. B. AND HODGINS, J. K. 1999. Tracking and modifying upper-body human motion data with dynamic simulation. In *Computer Animation and Simulation '99*.

Recieved March 2004; revised September 2004; accepted September 2004