Modal-Space Control for Articulated Characters

SUMIT JAIN and C. KAREN LIU
Georgia Institute of Technology

We present a novel control algorithm for simulating an articulated character performing a given reference motion and its variations. The unique feature of our controller is its ability to make a long-horizon plan at every time step. Our algorithm overcomes the computational hurdle by applying modal analysis on a time-varying linear dynamic system. We exploit the properties of modal coordinates in two ways. First, we design separate control strategies for dynamically decoupled modes. Second, our controller only applies long-horizon planning on a subset of modes, largely reducing the size of the control problem. With this decoupled and reduced control system, the character is able to execute the reference motion while reacting to unexpected perturbations and anticipating changes in the environment. We demonstrate our results by simulating a variety of reference motions, such as walking, squatting, jumping, and swinging.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Character animation, motion capture, modal analysis

ACM Reference Format:

Jain, S. and Liu, C. K. 2011. Modal-Space control for articulated characters. ACM Trans. Graph. 30, 5, Article 118 (October 2011), 12 pages.
DOI = 10.1145/2019627.2019637
http://doi.acm.org/10.1145/2019627.2019637

1. INTRODUCTION

The ability to respond to the changes in the environment and predict the consequences of our own action is fundamental for everyday motor tasks. Physically simulating a virtual character who exhibits both reactive and anticipatory behaviors presents immense challenges in many facets. First, the human motor system is both under-actuated and redundant. The former leads to complex issues with balance while the latter results in a high-dimensional and underconstrained problem. Second, the interaction with the environment via contacts is discrete in nature. The discontinuity introduced by the change of contact states further complicates both simulation and control problems.

One possible approach to achieving both a reactive and anticipatory virtual character is to formulate a long-term planning problem, such as spacetime optimization, and update the plan at every time step according to the current state of the character and of the environment. Motion produced by long-term planning usually appears more compliant because the character is not always in the urgency of matching the immediate goal. In addition, the frequent replanning allows the character to respond to unexpected perturbations in a timely manner. Though straightforward, this problem is extremely difficult and prohibitively expensive to solve in practice. Long-term planning on a full human dynamic system requires us to resolve all the aforementioned challenges. To date, offline solutions to optimal trajectory problems are very sensitive to parameters and initial conditions of the problem. We certainly cannot apply such brittle solutions at every time step in an online fashion.

This article tackles a more feasible problem: designing a control system capable of long-term planning and frequent replanning for simulating *a specific motion sequence*. We introduce a new control system that tracks the reference motion while reacting to unexpected perturbations and adapting to anticipated changes in the environment. Our key insight is that the long-term planning can be largely simplified by approximating the dynamic system using *modal analysis*. In our formulation, we do not solve one long-term planning problem in the generalize coordinates, rather, we formulate a set of control strategies in a reduced and dynamically decoupled modal coordinates. Modal analysis offers two advantages to our problem.

—*Independent control*. In the modal space, each mode is governed by an independent equation of motion. This reduces a N-dimensional optimal control problem to N independent one-dimensional problems.

—*Model reduction*. Modal analysis organizes modes by the natural frequencies of the dynamic system. Typically a few modes are sufficient to capture the dynamic behaviors of the system. This property potentially reduces the dimension of the control variables.

In spite of these great advantages, modal analysis is only suited for linear dynamic systems. We circumvent the issue by linearizing the nonlinear dynamic equations around the current state at each time step, resulting in a time-varying linear dynamic model.

We propose a new control system that makes long-term plans based on the reference motion and revises the plan at every time step in response to perturbations in the environment. We present our results by simulating a few drastically different human motions, including walking, squatting, jumping, and swinging. The virtual character can passively respond to external forces and actively replan for new tasks. For example, we demonstrate an online modification of a normal walk motion to walk on slopes, with different step sizes and different timing of steps. We also show that

This work was supported by the NSF grant CCF-CISE 0742303 and the Alfred P. Sloan Research Fellowship.
Author's addresses: S. Jain (corresponding author), C. K. Liu, Georgia Institute of Technology, Atlanta, GA 30332; email: sumit@cc.gatech.edu.

DOI 10.1145/2019627.2019637
http://doi.acm.org/10.1145/2019627.2019637

anticipated changes can be achieved by modifying the reference motion on-the-fly, such as modification of a squatting action to pick up a heavy box, or transition from a broad jump to swing.

2. RELATED WORK

Synthesizing natural motion for virtual characters has been a long-standing challenge in computer animation. For offline applications, physics-based trajectory optimization is able to create human-like motion that exhibits anticipatory behaviors [Witkin and Kass 1988; Cohen 1992; Liu et al. 1994; Popović and Witkin 1999]. With some variations in formulations, these methods essentially solve for a motion sequence while minimizing a chosen objective function under physical constraints, such as the equations of motion, joint limits, or contacts. Due to high dimensionality and nonlinearity in constraints or objective function, these methods are limited to simulating simple characters [Cohen 1992; Liu et al. 1994; Popović and Witkin 1999] or simplified dynamics [Liu and Popović 2002; Fang and Pollard 2003]. Researchers have utilized motion capture data to reduce the dimensionality of the motion [Safonova et al. 2004; Sulejmanpašić and Popović 2005], but the problem remains highly nonconvex and prone to local minima. Our method takes a different approach to simplifying the long-horizon planning problem using an approximate dynamic system. We leverage the advantages of modal coordinates such that the optimization only involves a *subset* of *decoupled* dynamic equations.

Departing from the offline approach, physics-based simulation methods coupled with active control are capable of synthesizing responsive motion in an interactive setting. Much research has focused on designing controllers for performing specific tasks such as standing balance [Raibert 1986; van de Panne and Lamouret 1995; Sharon and van de Panne 2005; Abe et al. 2007; Kudoh et al. 2006; Macchietto et al. 2009], locomotion [Hodgins et al. 1995; Laszlo et al. 1996; Yin et al. 2007; Shiratori et al. 2009; Wang et al. 2009], or other complex human movements [Hodgins et al. 1995; Wooten 1998; Faloutsos et al. 2001]. These controllers generate impressive results, but they usually depend on highly customized control parameters or prior knowledge of the motion. Consequently, they do not generalize well to other types of activities. Our method makes no assumption of the underlying reference motion, leading to a generic control algorithm suitable for a wide variety of motions.

Recent research work in physics-based locomotion controllers [Coros et al. 2010; de Lasa et al. 2010; Mordatch et al. 2010; Wang et al. 2010; Wu and Popović 2010] improves upon the previous work in terms of robustness to changes in environment, character topology, and physical properties. Intuitive interfaces to author controllers for responsive and robust locomotion tasks are presented [Coros et al. 2010; de Lasa et al. 2010]. Several methods specialize in adapting walking characters on uneven or unknown terrains [Mordatch et al. 2010; Wang et al. 2010; Wu and Popović 2010]. In contrast to all these methods that employ strategies specific for locomotion, our method is generic to execute different tasks. The ability to replan and change the reference trajectories online makes our method suitable to adapt to different situations.

Incorporating motion capture data with dynamic controllers has a potential to create more natural and human-like motion. A number of control algorithms have been proposed to directly track the reference mocap motion using Proportional-Derivative (PD) servos and their variations [Zordan and Hodgins 2002; Yin et al. 2003; Abe and Popović 2006; Sok et al. 2007; Allen et al. 2007]. With proper physical parameters for the controllers, these methods can effectively generate passive responses to external perturbations. However, many of these methods require fine tuning of physical parameters or expensive precomputation specific to the target motion and the skeletal model. The former limits the range of the variations generated by the controller, while the latter eliminates the possibility to modify the reference motion on-the-fly. Our control system does not require manual tuning of physical parameters or any precomputation, making it suitable to track reference trajectories that can be modified online. In a recent work, Lee et al. [2010] demonstrated a locomotion controller that modifies the reference motion by synchronizing it with the online simulation based on the contact changes. This results in improved robustness of the tracking controller while retaining the quality of motion capture data. However, their method is based on SIMBICON-style control [Yin et al. 2007] for locomotion making it hard to generalize to completely new motions.

To generalize control methods for different activities, many researchers suggested exploiting optimization techniques to compute control forces based on the current state of the character. Quadratic programming is applied to regulate body center of mass [Abe et al. 2007] and momenta [Macchietto et al. 2009]. Multiple quadratic objectives can also be organized by prioritized optimization control [de Lasa and Hertzmann 2009]. Similarly, nonconvex optimization can be used to directly control kinematic goals [Jain et al. 2009]. The control forces computed from an optimization process usually result in more robust motion than simple PD tracking, but these methods still rely on short-horizon optimization, lending themselves poorly for activities that require long-term planning. In contrast, our method optimizes a window of the control forces towards a future goal, rather than meeting the immediate goal.

To circumvent the issues of short-horizon planning, some researchers explored optimal feedback control techniques which consider the entire motion trajectory for computing the control forces. da Silva et al. [2008] applied a Linear Quadratic Regulator (LQR) to control a simplified model for maintaining dynamic balance in locomotion. The optimal control policy derived from the LQR framework minimizes the cost throughout the entire trajectory, taking into account the future. Their controller tracks a reference motion and allows some variations due to perturbations. Muico et al. [2009] modified the time-varying LQR to account for the dynamic constraints violations. They developed a look-ahead control policy by constructing the ground contact force predictions. Both methods are capable of responding to small perturbations in a passive manner. However, the control forces are driven by the deviation between the current state and the reference trajectory, rather than the anticipation of the changes in the environment. To actively replan for a new task, the underlying reference motion must be modified accordingly. Unfortunately, these methods require an offline process (Ricatti equations) to compute control parameters for each new reference motion. In contrast, our method allows online editing of the reference motion and employs a completely online process to replan a look-ahead control policy at each time step. As a result, we are able to create motion drastically different from the reference motion.

Modal analysis has been previously applied to deformable models [Faloutsos et al. 1997; Hauser et al. 2003; James and Pai 2002; Barbič et al. 2009], and character animation [Kry et al. 2009]. We draw inspiration from Kry et al. [2009] and develop a control algorithm that fully takes advantage of the reduced and decoupled dynamic system in modal coordinates. Kry et al. [2009] select a few modes based on heuristics and manually create motions for each selected mode. They are able to create dynamic motions for simple characters but can only synthesize kinematic motions for complex characters like dog and human since they do not use any control algorithm that can handle balance issues. We demonstrate that

time-varying linearized dynamic system can be a good approximation to the dynamics of a full-body, articulated character. In our method, we control the low-frequency modes by formulating a long-term control problem at every time step, resulting in a robust control algorithm. The reference trajectory for each mode is derived from the given reference motion sequence rather than being manually defined.

Much previous work has explored a variety of dimension reduction techniques for character motion synthesis. Some techniques parametrize a subspace for control based on the motion data [Safonova et al. 2004; Chai and Hodgins 2007; Ye and Liu 2008], while others define an abstract model based on domain knowledge or heuristics about the motion and the character model [Ye and Liu 2010; Mordatch et al. 2010]. Safonova et al. [2004] employed PCA to a small set of similar example motions and selected a reduced bases to synthesize physically plausible motion in a spacetime setting. However, their algorithm still needs to solve a coupled dynamic system which requires offline computation unsuitable for interactive applications. In addition, the synthesized motion is restricted to a linear subspace of example motions. Our algorithm employs modal analysis to decouple the equations of motion. Decoupling gives a significant speedup allowing us to interactively replan for a window of time at every time step. In addition, we control a subset of modes that depend on the physical properties of the character rather than example motions, allowing us to synthesize interactions and variations in the motion that are not present in the given reference motion. Ye and Liu [2010] solved an optimal control problem based on a simplified model that abstracts the DOFs of the character into a small number of parameters. This simplification greatly reduces the complexity of the control problem rendering it suitable to be solved for the entire motion trajectory. The computation for the optimal feedback parameters is done offline, hence the reference trajectory cannot be altered online. In other concurrent work, Mordatch et al. [2010] used a spring loaded inverted pendulum model to solve for optimal control in an online fashion. The control algorithm is designed for walking or running and does not use any motion trajectory. In our method, modal analysis allows for fast computation suitable for online replanning. In addition, tracking a reference trajectory gives us benefits of constructing a generic control strategy that is independent of the performed task and synthesizing more natural motion.

3. MODAL ANALYSIS: A REVIEW

Modal analysis is used to transform a multi-degree of freedom (DOF) system into decoupled single-DOF systems. We review modal analysis for linear systems [Shabana 1997] and discuss approximations to apply modal analysis for nonlinear systems.

3.1 Linear Multi-DOF Systems

For a system with N DOFs $\mathbf{q} = (q_1, q_2, \ldots, q_N)^T$ and linear dynamics, the general equations of motion are given by

$$M\ddot{\mathbf{q}} + D\dot{\mathbf{q}} + K\mathbf{q} = \mathbf{b} + \mathbf{f}(t), \tag{1}$$

where M, D, and K are constants that denote the mass, damping, and stiffness matrices, respectively. $\mathbf{b}$ is some constant vector and $\mathbf{f}(t)$ denotes a time-dependent generalized force being applied to the system. It is convenient to choose a proportional damping model for D, that is, $D = \alpha M + \beta K$ for some damping parameters α and β.

Now, we define a modal transformation matrix Φ whose columns, $\boldsymbol{\phi}_i$'s, are the eigenvectors of the generalized eigenvalue problem $K\boldsymbol{\phi} = \omega^2 M\boldsymbol{\phi}$, that is, $M^{-1}K = \Phi\Omega\Phi^{-1}$, where Ω is a diagonal matrix of eigenvalues or squared natural frequencies $\Omega = \text{diag}(\omega_1^2, \omega_2^2, \ldots, \omega_N^2)$. The columns of matrix Φ are both M and K-orthogonal, that is, we can write $\Phi^T M\Phi = M_d = \text{diag}(m_1, m_2, \ldots, m_N)$ and $\Phi^T K\Phi = K_d = \text{diag}(k_1, k_2, \ldots, k_N)$. Note that $\omega_i^2 = \frac{k_i}{m_i}$.

Using the modal transformation matrix Φ, we transform the generalized coordinates $\mathbf{q}$ to a set of *modal coordinates* $\mathbf{p}$ as: $\mathbf{q} = \Phi\mathbf{p}$. Premultiplying Eq. (1) by Φ^T, we arrive at a new set of equations of motion that govern the modal coordinates.

$$M_d\ddot{\mathbf{p}} + (\alpha M_d + \beta K_d)\dot{\mathbf{p}} + K_d\mathbf{p} = \Phi^T\mathbf{b} + \Phi^T\mathbf{f}(t) \tag{2}$$

Because M_d and K_d are diagonal matrices, Eq. (2) can be decoupled into N independent one-dimensional equations, each of which is written as

$$m_i\ddot{p}_i + d_i p_i + k_i p_i = b_i + f_i(t), \tag{3}$$

where $d_i = \alpha m_i + \beta k_i$ and b_i and f_i are the i^{th} elements of vectors $\Phi^T\mathbf{b}$ and $\Phi^T\mathbf{f}(t)$ respectively. For those modes with $k_i \neq 0$, called *deformation modes*, the unforced solution ($f_i = 0$) for an underdamped system ($\xi_i^2 < 1$, where damping ratio $\xi_i = \frac{d_i}{2\sqrt{k_i m_i}}$) can be written as

$$\tilde{p}_i(t) = \frac{b_i}{k_i} + S_i e^{-\xi_i\omega_i t} sin(\omega_{d,i}t + \psi_i), \tag{4}$$

where $\omega_{d,i} = \omega_i\sqrt{1-\xi_i^2}$, and S_i and ψ_i are amplitude and phase, respectively, determined from the initial conditions $p_{0,i}$ and $\dot{p}_{0,i}$ by solving $p_{0,i} = S_i sin(\psi_i)$ and $\dot{p}_{0,i} = S_i(\omega_{d,i}cos(\psi_i) - \xi_i\omega_i sin(\psi_i))$. The initial conditions for all the modes, $(\mathbf{p}_0, \dot{\mathbf{p}}_0)$, can be transformed from the generalized space as $\mathbf{p}_0 = \Phi^{-1}\mathbf{q}_0$ and $\dot{\mathbf{p}}_0 = \Phi^{-1}\dot{\mathbf{q}}_0$.

If $\mathbf{I}_f$ is an impulse applied at time t_f (i.e., $\mathbf{I}_f = \int_{t_f}^{t_f+\epsilon}\mathbf{f}(t)dt$), the impulse response of the system in Eq. (3) is given by

$$\check{p}_i(t) = I_i\left(\frac{e^{-\xi_i\omega_i(t-t_f)}}{m_i\omega_{d,i}} sin(\omega_{d,i}(t-t_f))\right), \tag{5}$$

where $I_i = \int_{t_f}^{t_f+\epsilon} f_i(t)dt$ is the i^{th} element of vector $\Phi^T\mathbf{I}_f$. Adding the unforced solution (Eq. (4)) to the impulse response (Eq. (5)) gives the closed-form solution for the modal state $p_i(t) = \tilde{p}_i(t) + \check{p}_i(t)$ when the impulse $\mathbf{I}_f$ is applied at time t_f.

Modes with $k_i = 0$ are called *rigid body modes*. In this case, Eq. (3) reduces to $m_i\ddot{p}_i + d_i\dot{p}_i = b_i + f_i(t)$. Assuming there is no damping in rigid body modes ($d_i = 0$), the unforced solution is given by

$$\tilde{p}_i(t) = p_{0,i} + \dot{p}_{0,i}t + \frac{b_i}{2m_i}t^2 \tag{6}$$

and the impulse response by

$$\check{p}_i(t) = I_i\left(\frac{t-t_f}{m_i}\right). \tag{7}$$

Consolidating different modes in a common equation, we collect the coefficients of the impulse from Eq. (5) and Eq. (7) into a time-dependent diagonal matrix $A(t-t_f)$. The modal state at any time $t > t_f$, with response to impulse $\mathbf{I}_f$ can be expressed as

$$\mathbf{p}(t) = \tilde{\mathbf{p}}(t) + A(t-t_f)\mathbf{I}_f, \tag{8}$$

$$\dot{\mathbf{p}}(t) = \dot{\tilde{\mathbf{p}}}(t) + \dot{A}(t-t_f)\mathbf{I}_f. \tag{9}$$

Finally, the solution in the original space can be recovered by $\mathbf{q}(t) = \Phi\mathbf{p}(t)$ and $\dot{\mathbf{q}}(t) = \Phi\dot{\mathbf{p}}(t)$.

3.2 Articulated Characters

The motion of an articulated character is governed by nonlinear dynamic equations, to which modal analysis does not directly apply. These nonlinear equations of motion for articulated characters can be written as

$$M(\mathbf{q})\ddot{\mathbf{q}} + (C(\mathbf{q}, \dot{\mathbf{q}}) + D(\mathbf{q}))\dot{\mathbf{q}} + G(\mathbf{q}) + K\mathbf{q} + \mathbf{k}_0 = \mathbf{f}(t), \quad (10)$$

where M is the mass matrix, C is the matrix for Coriolis and centrifugal forces, G represents gravity forces, and $\mathbf{f}(t)$ are the generalized forces. To model passive forces, each joint is equipped with a spring and a damper. Matrices K and D then represent the stiffness and damping coefficients of the coupled system and $\mathbf{k}_0$ is a constant vector. $\mathbf{q}$ represents the character's root position, root orientation, and joint DOFs in generalized coordinates.

Time-varying linear dynamics. At any time t_0, we linearize these equations around the pose $\mathbf{q}_0$ of the character at time t_0 and zero velocity. For any deviations in the position and velocity around the state $(\mathbf{q}_0, \mathbf{0})$, $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q}_0$ and $\Delta\dot{\mathbf{q}} = \dot{\mathbf{q}} - \mathbf{0}$, we approximate the equations of motion as

$$M(\mathbf{q}_0)\Delta\ddot{\mathbf{q}} + (C(\mathbf{q}_0, \mathbf{0}) + D(\mathbf{q}_0))\Delta\dot{\mathbf{q}} + G(\mathbf{q}_0) + K(\mathbf{q}_0 + \Delta\mathbf{q}) + \mathbf{k}_0 = \mathbf{f}(t).$$

Note that C vanishes at zero velocity. Denoting $-(G(\mathbf{q}_0)+K\mathbf{q}_0+\mathbf{k}_0)$ by a constant $\mathbf{b}$, we rewrite the equation as

$$M\Delta\ddot{\mathbf{q}} + D\Delta\dot{\mathbf{q}} + K\Delta\mathbf{q} = \mathbf{b} + \mathbf{f}(t). \quad (11)$$

This equation represents an approximate linear dynamics model in $\Delta\mathbf{q}$ and is similar to Eq. (1). Therefore, modal analysis discussed in Section 3.1 can be applied to this equation as well. The continuous force function $\mathbf{f}(t)$ is broken down into a series of impulses. If we assume that the force $\mathbf{f}(t)$ remains constant over a small time step Δt, we can approximate the impulse at current time t_0 as $\mathbf{I}_0 = \mathbf{f}(t_0)\Delta t$. Based on Eq. (8), the position in modal space after time step Δt is then given by

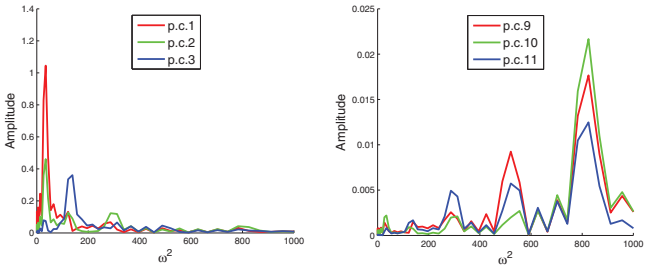
$$\mathbf{p}(t_0 + \Delta t) = \tilde{\mathbf{p}}(t_0 + \Delta t) + A(\Delta t)\mathbf{I}_0.$$

Using the modal transformation, we recover the pose at the next time step: $\mathbf{q}(t_0 + \Delta t) = \Phi\mathbf{p}(t_0 + \Delta t)$. Advancing the time by Δt, we linearize the dynamic system around the new pose and repeat the same process to compute the next modal state.

4. CONTROL METHODOLOGY

Given a reference motion, we seek to design control strategies that track the reference motion while responding realistically to both unexpected perturbations and anticipated changes in the environment. Our approach to controlling an articulated character leverages the advantages offered by the modal coordinates. We apply modal analysis to the linearized N-DOF dynamic system (Eq. (11)), resulting in N decoupled one-dimensional equations. This transformation allows us to compute the control forces for each decoupled mode independently. We now classify these modes into three categories and develop separate control strategies for each type (Section 5).

(1) *Rigid body modes*: Because the global DOFs are not equipped with springs, the stiffness matrix K in Eq. (11) is always singular. These underactuated DOFs result in six eigenvectors in Φ that correspond to zero eigenvalues. These six eigenvectors, called the rigid body modes, are only affected by external forces.



(a) (b)

Fig. 1. We apply principal component analysis on the poses of a captured walking sequence. Left: The first three principal components have dominant low frequencies. Right: The last three principal components have dispersed frequency content biased towards high-frequency range. Note that the scales of the vertical axes in the plots are different for better illustration.

(2) *Low-frequency modes*: Except for the rigid body modes, all the remaining modes of the articulated system are actuated. For those modes with corresponding eigenvalue less than a chosen threshold, we classify them as low-frequency modes. The choice of this threshold depends on the application and is discussed in Section 7. The motivation to focus on low-frequency modes is due to the observation that visually more significant movements in human motion typically correspond to low-frequency motion. In our experiments, we applied frequency analysis on recorded human motion projected on its principal components. The results show that the first few principal components, that capture most of the variations in the motion [Safonova et al. 2004], have dominant lower frequencies (Figure 1(a)) while the last few have dispersed frequency components biased toward the high-frequency range (Figure 1(b)). This implies that controlling low-frequency modes can effectively change visually significant movements of the character.

(3) *High-frequency modes*: All the remaining modes above the threshold belong to the high-frequency modes. Comparing to the low-frequency modes, the motion in the high-frequency modes has a little visual effect to the appearance of the motion (Figure 1(b)). Because of the high stiffness, controlling the motion in these modes usually requires large forces.

Once we classify the modes for the linearized dynamic system around the current state of the character, we apply long-horizon planning to the rigid body and the low-frequency modes and short-horizon planning to the high-frequency modes. The choice of horizon for different modes is due to the following two reasons. First, because the long-term planning is more computationally costly, we only apply it on the modes that can make significant visual differences, namely, rigid body and low-frequency modes. Second, because the long-term planning allows for temporary deviation from the reference trajectory, applying it to high-frequency modes can cause large corrective forces, leading to instability and unnatural oscillations.

5. CONTROL FORMULATION

We now discuss our formulation of control strategies for each class of modes. Our goal is to compute the required contact forces and joint actuation within a window of time, such that the character can reach the corresponding reference state at the end of the window. We choose a window size of $n = 1$ for the short-horizon planning and a larger number for the long-horizon planning (Section 7).

Notation. Our simulation discretizes the time domain with a fixed time step Δt. t_0 indicates the current time while $t_k = t_0 + k\Delta t$ is the time at k time steps in the future, where $k \in \mathbb{Z}^+$. The horizon for each planning problem is determined by the window size of n frames or $n\Delta t$ seconds. We represent the state of the character at time t_k as $(\mathbf{q}_k, \dot{\mathbf{q}}_k)$. Similarly, the corresponding state of the reference motion is represented as $(\bar{\mathbf{q}}_k, \dot{\bar{\mathbf{q}}}_k)$.

Applying the modal transformation to Eq. (11), we express the equation in the modal coordinates $\mathbf{p}$, with the linear invertible relation $\Delta\mathbf{q} = \Phi\mathbf{p}$. The modal transformation matrix Φ is updated at each time step based on the current state of the character, $\mathbf{q}_0$: $M^{-1}(\mathbf{q}_0)K = \Phi\Omega\Phi^{-1}$. The desired state at the end of the current window, $(\bar{\mathbf{q}}_n, \dot{\bar{\mathbf{q}}}_n)$, can be transformed into modal coordinates as $(\bar{\mathbf{p}}_n, \dot{\bar{\mathbf{p}}}_n) = (\Phi^{-1}(\bar{\mathbf{q}}_n - \mathbf{q}_0), \Phi^{-1}\dot{\bar{\mathbf{q}}}_n)$.

5.1 Estimate Contact Forces: Rigid Body Modes

Because the rigid body modes are not equipped with actuators, they are directly controlled by the contact forces. Our goal is to compute the contact forces such that, at the end of the planning horizon t_n, the states of rigid body modes are as close as possible to the desired states. To express the contact forces in the modal coordinates, first we must determine the number of contact points at each time step within the planning window $[t_0, t_n]$. The contacts at the current frame t_0 are determined by a collision detection routine. For any other time step in the window, $t_k, k = 1, \ldots, n$, the contact information is directly taken from the reference motion.

We denote the contact forces $\mathbf{f}_{k,i}, i = 1 \ldots n_k$ at a given time t_k, where n_k denotes the number of contact points at time t_k. The sum of these contact forces in the modal coordinates can be expressed as $\Phi^T \sum_i J_i^T \mathbf{f}_{k,i}$, where $J_i = \frac{\partial \mathbf{x}_i}{\partial \mathbf{q}_0}$. J_i is the Jacobian evaluated at the point of application $\mathbf{x}_i$. If we assume that the contact forces hold fixed over a small interval of time Δt, we can approximate the effect of the contact forces by an impulse $\mathbf{I}_k^c = Z_k\mathbf{f}_k$, where $Z_k = \Delta t\Phi^T\mathbf{J}_k^T$, $\mathbf{J}_k = [J_1^T, \ldots, J_{n_k}^T]^T$ and $\mathbf{f}_k = (\mathbf{f}_{k,1}^T, \ldots, \mathbf{f}_{k,n_k}^T)^T$.

Based on Eq. (8) and Eq. (9) , the states of rigid body modes at time t_n under the influence of contact forces can be expressed as

$$\mathbf{p}_n^r = \tilde{\mathbf{p}}_n^r + \sum_{k=0}^{n-1} A^r(t_n - t_k)\mathbf{I}_k^c, \tag{12}$$

$$\dot{\mathbf{p}}_n^r = \dot{\tilde{\mathbf{p}}}_n^r + \sum_{k=0}^{n-1} \dot{A}^r(t_n - t_k)\mathbf{I}_k^c. \tag{13}$$

These modal state equations are linear in $\mathbf{f}_k$'s. Now, we formulate an optimization problem to solve for the optimal contact forces $\mathbf{f}^*$ and minimize the state deviation and the contact force magnitude. We have

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{p}_n^r - \bar{\mathbf{p}}_n^r\|_{W_1}^2 + \|\dot{\mathbf{p}}_n^r - \dot{\bar{\mathbf{p}}}_n^r\|_{W_2}^2 + \|\mathbf{f}\|_{W_3}^2, \tag{14}$$

where $\mathbf{f}$ represents all the contact forces $(\mathbf{f}_0^T, \ldots, \mathbf{f}_{n-1}^T)^T$ and W_1, W_2, and W_3 are positive diagonal weighting matrices (see details in Section 7). $\|\mathbf{v}\|_W$ denotes the norm $(\mathbf{v}^T W\mathbf{v})^{\frac{1}{2}}$ for any vector $\mathbf{v}$. Eq. (14) is an unconstrained convex Quadratic Programming (QP) problem and can be solved efficiently. This formulation is valid for bilateral contacts such as a hand grasp.

Coulomb friction. We apply unilateral forces for the ground contacts. The contact forces are constrained within a cone defined by friction coefficient μ. Assuming static contact, we define a contact force $\mathbf{f}$ by its component f along the contact normal $\hat{\mathbf{n}}$ and tangential component $\mathbf{f}_T$ such that $\|\mathbf{f}_T\| \le \mu f$. We approximate $\mathbf{f}_T$ by a set of unit vectors $\hat{\mathbf{d}}_j$s.These unit vectors span the tangential plane as described in Stewart and Trinkle [1996] and Anitescu and Potra [1997]. We arrange these vectors as columns of matrix D. The coefficients corresponding to the unit vectors $\hat{\mathbf{d}}$'s are represented by $\boldsymbol{\beta}$ (with $\boldsymbol{\beta} \ge \mathbf{0}$). The legal contact force can be approximated by a polyhedral friction cone.

$$\mathbf{f} = f\hat{\mathbf{n}} + D\boldsymbol{\beta} \quad \text{with} \sum_j \beta_j \le \mu f \quad \text{and} \quad f, \boldsymbol{\beta} \ge \mathbf{0} \tag{15}$$

Now, substituting Eq. (15) in Eq. (14), and adding the boundary conditions, we get a constrained convex QP problem in $f, \boldsymbol{\beta}$

$$\begin{aligned} \min_{f,\boldsymbol{\beta}} \quad & \|\mathbf{p}_n^r - \bar{\mathbf{p}}_n^r\|_{W_1}^2 + \|\dot{\mathbf{p}}_n^r - \dot{\bar{\mathbf{p}}}_n^r\|_{W_2}^2 + \|\mathbf{f}\|_{W_3}^2 \\ \text{subject to} \quad & \mu f - \mathbf{e}^T\boldsymbol{\beta} \ge 0 \quad \text{with } f, \boldsymbol{\beta} \ge \mathbf{0}, \end{aligned} \tag{16}$$

where $\mathbf{e} = [1, \ldots, 1]^T$.

5.2 Estimate Joint Actuation: Low-Frequency Modes

The low-frequency modes can be controlled by the contact forces, as well as their own actuators. Given the contact forces $\mathbf{f}^*$ optimized for the rigid body modes, the control strategy for the low-frequency modes is to optimize the actuation such that, at the end of the planning horizon t_n, the states of the low-frequency modes match the desired reference states.

The state of these modes at time t_n under the influence of actuation impulses $\mathbf{I}^a$ and optimized contact forces $\mathbf{f}^*$ is given by

$$\mathbf{p}_n^l = \tilde{\mathbf{p}}_n^l + \sum_{k=0}^{n-1} A^l(t_n - t_k)Z_k\mathbf{f}_k^* + \sum_{k=0}^{n-1} A^l(t_n - t_k)\mathbf{I}_k^a, \tag{17}$$

$$\dot{\mathbf{p}}_n^l = \dot{\tilde{\mathbf{p}}}_n^l + \sum_{k=0}^{n-1} \dot{A}^l(t_n - t_k)Z_k\mathbf{f}_k^* + \sum_{k=0}^{n-1} \dot{A}^l(t_n - t_k)\mathbf{I}_k^a. \tag{18}$$

Again, these equations are linear in $\mathbf{I}^a$. In addition, because matrix $A^l(t)$ is diagonal, these equations can also be decoupled. We rewrite the equations for m^{th} low-frequency mode as

$$p_{n,m}^l = c_{n,m}^l + \sum_{k=0}^{n-1} a_m^l(t_n - t_k)I_{k,m}^a, \tag{19}$$

$$\dot{p}_{n,m}^l = \dot{c}_{n,m}^l + \sum_{k=0}^{n-1} \dot{a}_m^l(t_n - t_k)I_{k,m}^a, \tag{20}$$

where a_m^l and $\dot{a}_m^l$ are the diagonal elements of A^l and $\dot{A}^l$ respectively, corresponding to mode m. $\mathbf{c}_n^l$ and $\dot{\mathbf{c}}_n^l$ indicate the first two terms of Eq. (17) and Eq. (18) respectively.

Now, we formulate an optimization to solve for actuation $\mathbf{I}_m^a = (I_{0,m}^a, \ldots, I_{n-1,m}^a)^T$ for each mode m independently. Our goal is use the least amount of actuation to track a future state of the reference motion. We have

$$\mathbf{I}_m^{a*} = \underset{\mathbf{I}_m^a}{\operatorname{argmin}}\, w_4\big(p_{n,m}^l - \bar{p}_{n,m}^l\big)^2 + w_5(\dot{p}_{n,m}^l - \dot{\bar{p}}_{n,m}^l)^2 + \|\mathbf{I}_m^a\|_{W_6}^2, \tag{21}$$

where w_4 and w_5 are scalar weights for position and velocity matching, respectively, and W_6 is a positive diagonal matrix. This is again an unconstrained convex QP problem. Since these problems are uncoupled for all the modes, we can solve them independently and efficiently.

5.3 Track High-Frequency Modes

We use a short-horizon to plan the control forces for the high-frequency modes, namely, $n = 1$. Similar to Eq. (19), the state of each high-frequency mode at the next state can be expressed as $p^h_{1,m} = c^h_{1,m} + a^h_m(t_1 - t_0)I^a_{0,m}$. Since there is only one control variable, $I^a_{0,m}$, we can solve it analytically.

$$I^{a*}_{0,m} = \frac{\bar{p}^h_{1,m} - c^h_{1,m}}{a^h_m(\Delta t)} \tag{22}$$

5.4 Enforce Physical Contact Model

If we directly apply the optimal contact forces $\mathbf{f}^*_0$ and actuation impulses $\mathbf{I}^{a*}_0$ at the current time step, we can analytically compute the modal state at the next time step t_1 by virtue of Eq. (8).

$$\mathbf{p}^*_1 = \tilde{\mathbf{p}}_1 + A(\Delta t)(Z_0\mathbf{f}^*_0 + \mathbf{I}^{a*}_0) \tag{23}$$

However, the estimated state $\mathbf{p}^*_1$ might cause artifacts at contact points, such as slipping, penetration, and breakage of contacts. We formulate a Linear Complementarity Problem (LCP) to solve this issue. Specifically, we adjust the contact forces by $\Delta\mathbf{f} = (\Delta\mathbf{f}^T_1, \ldots, \Delta\mathbf{f}^T_{n_0})^T$ and the joint actuation by $\Delta\mathbf{I}^a$, such that the movement of each contact point, $\Delta\mathbf{x}_i \approx J_i\Delta\mathbf{q} = J_i\Phi\mathbf{p}_1$, and the corrective forces, $\Delta\mathbf{f}$ and $\Delta\mathbf{I}^a$, satisfy the Coulomb friction model. The modal state at the next time step, $\mathbf{p}_1$, is then computed as

$$\mathbf{p}_1 = \mathbf{p}^*_1 + A(\Delta t)(Z_0\Delta\mathbf{f} + \Delta\mathbf{I}^a). \tag{24}$$

Control preference. Because there can be many solutions for the corrective contact forces $\Delta\mathbf{f}$ and the corrective joint actuation $\Delta\mathbf{I}^a$ that satisfy the Coulomb friction model, we can formulate an optimization with an objective function, $E(\Delta\mathbf{f}, \Delta\mathbf{I}^a)$, that minimizes a certain desired criterion.

However, we cannot directly include this objective function in the LCP formulation (LCP is not an optimization problem). If we choose a quadratic form for E, the LCP solution can be biased towards the minimum of E, by deriving a linear relation between $\Delta\mathbf{I}^a$ and $\Delta\mathbf{f}$ using the optimality condition. We can then express $\Delta\mathbf{I}^a$ in terms of $\Delta\mathbf{f}$ in the LCP formulation. In our implementation, we choose to minimize the impact of corrective forces in the state of rigid body modes and the low-frequency modes.

$$E(\Delta\mathbf{f}, \Delta\mathbf{I}^a) = \|A^r(\Delta t)Z_0\Delta\mathbf{f}\|^2_{W_r} + \|A^l(\Delta t)(Z_0\Delta\mathbf{f} + \Delta\mathbf{I}^a)\|^2_{W_l} \tag{25}$$

Based on the optimality condition, the gradients of E vanish at the minimum. We obtain a linear relation between $\Delta\mathbf{I}^a$ and $\Delta\mathbf{f}$.

$$P\Delta\mathbf{I}^a + Q\Delta\mathbf{f} = 0$$

The coefficient matrix of $\Delta\mathbf{I}^a$ may not be full rank and invertible, but we can solve this issue by reformulating the objective function

$$\min_{\Delta\mathbf{I}^a} \|P\Delta\mathbf{I}^a + Q\Delta\mathbf{f}\|^2 + \|\Delta\mathbf{I}^a\|^2_{W_a}, \tag{26}$$

where $\boldsymbol{W}_a$ is a positive definite weighting matrix. We now solve for the corrective actuation $\Delta\mathbf{I}^a$ that minimizes the error in optimality condition of E and the magnitude of $\Delta\mathbf{I}^a$. The optimality condition of the new objective function is given by a linear relation.

$$\begin{aligned} \Delta\mathbf{I}^a &= X_f\Delta\mathbf{f} \\ \text{where } X_f &= -(P^TP + \boldsymbol{W}_a)^{-1}P^TQ \end{aligned} \tag{27}$$

Note that X_f is well-defined since $P^TP + W_a$ is always positive definite. This is a similar treatment as in Muico et al. [2009], in that the relation was derived based on matching the acceleration of certain chosen features.

For the high-frequency modes, we would like to maintain the planned actuation computed in Section 5.3. We define the components corresponding to the high-frequency modes in X_f such that for m^{th} high-frequency mode, $\Delta I^a_m = -(Z_0\Delta\mathbf{f})_m$ resulting in $p_{1,m} = p^*_{1,m}$.

We can now rewrite the modal state $\mathbf{p}_1$ in Eq. (24) as

$$\mathbf{p}_1 = \mathbf{p}^*_1 + A(\Delta t)(Z_0 + X_f)\Delta\mathbf{f}. \tag{28}$$

Note the only explicit variables are the corrective contact forces, $\Delta\mathbf{f}$. We now can formulate a LCP based on the following constraints derived from Coulomb friction model.

Unilateral contact force. Breaking down each force in normal and tangential components (as in Eq. (15)), we have

$$\mathbf{f}^*_{0,i} + \Delta\mathbf{f}_i = f_i\hat{\mathbf{n}}_i + D_i\boldsymbol{\beta}_i, \quad \text{with} \quad f_i, \boldsymbol{\beta}_i \geq \mathbf{0}. \tag{29}$$

Normal constraints. In the normal direction, the movement of the contact point $\mathbf{x}_i$ is given by $\hat{\mathbf{n}}^T_i\Delta\mathbf{x}_i$. The following complementarity conditions must be satisfied for each contact.

$$(\hat{\mathbf{n}}^T_i\Delta\mathbf{x}_i)f_i = 0 \quad \text{with} \quad f_i \geq 0, \quad \hat{\mathbf{n}}^T_i\Delta\mathbf{x}_i \geq 0 \tag{30}$$

Tangential constraints. In the tangential direction, we need to ensure that if the point moves, the contact force lies on the boundary of the cone and in the direction opposite to the movement. Using the formulation similar to Stewart and Trinkle [1996] and Anitescu and Potra [1997], we introduce a parameter λ_i that represents the relative tangential movement of the contact point. Therefore, the constraints are written as

$$\left(\lambda_i\mathbf{e} + D^T_i\Delta\mathbf{x}_i\right)^T\boldsymbol{\beta}_i = 0 \quad \text{with} \quad \lambda_i\mathbf{e} + D^T_i\Delta\mathbf{x}_i \geq \mathbf{0}, \boldsymbol{\beta}_i \geq \mathbf{0}, \tag{31}$$

$$(\mu f_i - \mathbf{e}^T\boldsymbol{\beta}_i)\lambda_i = 0 \quad \text{with} \quad \mu f_i - \mathbf{e}^T\boldsymbol{\beta}_i \geq 0, \lambda_i \geq 0. \tag{32}$$

Colloquially, Eq. (31) enforces that the direction of movement is opposite to the friction force and Eq. (32) enforces that the friction force is on the boundary if the point of contact moves.

Linear complementarity problem (LCP). We define $\mathbf{z} = (\mathbf{z}^T_1, \ldots, \mathbf{z}^T_{n_0})^T$ where $\mathbf{z}_i = (f_i, \boldsymbol{\beta}^T_i, \lambda_i)^T$ and rewrite Eq. (29) as $\Delta\mathbf{f}_i = N_i\mathbf{z}_i - \mathbf{f}^*_{0,i}$, where $N_i = [\hat{\mathbf{n}}_i, D_i, 0]$. Stacking all the points together, we write

$$\Delta\mathbf{f} = N\mathbf{z} - \mathbf{f}^*_0, \tag{33}$$

where N is a block diagonal consisting of N_i's. Recalling $\Delta\mathbf{x}_i \approx J_i\Delta\mathbf{q} = J_i\Phi\mathbf{p}_1$, we substitute Eq. (28) and Eq. (33) in Eq. (30), Eq. (31), and Eq. (32). As a result, a standard LCP of the form can be derived

$$\mathbf{w} = C\mathbf{z} + \mathbf{h} \text{ and } \mathbf{w}^T\mathbf{z} = 0 \text{ with } \mathbf{w}, \quad \mathbf{z} \geq \mathbf{0}, \tag{34}$$

where C and $\mathbf{h}$ are derived in Appendix A.

Once the solution $\mathbf{z}$ is obtained, we compute the required forces using Eq. (33) and Eq. (27). Finally, we compute the modal position at the next time step $\mathbf{p}_1$ using Eq. (24) and the next pose $\mathbf{q}_1$ as $\Phi\mathbf{p}_1$.

5.5 Summary

We summarize the control procedure that advances the character from the current time step to the next. Starting out with a new current state t_0, we solve the control forces for a time window $[t_0, t_n]$ by following steps:

—Solve for the ideal contact forces $\mathbf{f}^*$ such that the components of the state of the rigid body modes at t_n match the corresponding reference state.

—Given $\mathbf{f}^*$, solve for the ideal actuation for the low-frequency modes $\mathbf{I}^{a*}$ such that the state of the low-frequency components at t_n matches the corresponding reference state.

—Compute actuation for the high-frequency modes such that the state of the high-frequency modes at t_1 exactly matches the next reference state.

—Compute the corrective forces $\Delta\mathbf{f}$ and $\Delta\mathbf{I}^a$ to satisfy the Coulomb friction model.

Once the corrective forces are added to the ideal forces, we get the state at the next time step $\mathbf{q}_1$. We then advance the time step and repeat the same procedure. The ideal forces computed for the rest of the window have to be recomputed again at the next time step for two reasons. First, the environment state can change at any moment and the character must adapt her plan accordingly. Second, the ideal forces are computed based on a locally linearized dynamic system. Replanning ensures that more accurate control forces are applied to the character.

The control and simulation are performed with respect to linearized dynamics and no numerical integration is needed since the next state can be computed analytically using Eq. (24).

6. INTERACTION AND MOTION EDITING

To create variations from the original reference motion, our system allows the user to apply external forces to the character, as well as directly modify the reference trajectory on-the-fly.

Perturbations. We assume that the character reacts to unexpected perturbations with 200 ms latency to simulate the muscle activation delay in the sensory feedback [Miall et al. 1985; Georgopoulos et al. 1981]. We denote the response time as t_r and perturbation time as t_0. For a simulation time t_k in the time interval $[t_0, t_0 + t_r]$, we replace the current state of the character, $(\mathbf{q}_k, \dot{\mathbf{q}}_k)$, by the corresponding state in reference motion, $(\bar{\mathbf{q}}_k, \dot{\bar{\mathbf{q}}}_k)$ for the purpose of computing control forces in the rigid body and the low-frequency modes. During this response interval, the control system makes a long-term plan as if the character has not sensed the external force. Once the control forces for the rigid body and the low-frequency modes are computed, we restore the current state $(\mathbf{q}_k, \dot{\mathbf{q}}_k)$ and compute the forces for the high-frequency modes and the corrective forces for the contacts. After the response interval, we compute the control forces in the usual manner to recover back to the reference motion.

Motion editing. When the character responds to larger perturbations or changes behaviors volitionally, tracking the original reference trajectory becomes a poor control strategy. Since our control algorithm does not require any offline computation based on the reference trajectory, we are free to edit the reference motion on-the-fly. In theory, any online trajectory editing technique can be applied, we implemented following three generic methods.

—Forward and inverse kinematics (FK/IK): The user can directly change the joint angles of the reference motion via FK or modify the position of a body point via IK.

—Time warping: The user can select parts of the motion and change their speed.

—Motion transition: The user can select a new motion sequence and blend the current state of the character into the new motion over a time interval.

7. RESULTS

We now present our results and report all the parameters for controlling a 3D human articulated character under perturbations and changes of the environment. Full animations can be seen in the supplemental video available online in the ACM Digital Library.

Character description. The skeleton for our articulated character consists of 18 rigid links and 42 DOFs. We define the stiffness for each DOF as the total mass of the subtree rooted at the joint where the DOF resides. We use zero damping in all our examples. The threshold for classifying the modes (Section 4) is chosen as $100\ (rad/s)^2$. For our articulated character, this choice of threshold results in 14 low-frequency modes and 22 high-frequency modes apart from 6 rigid body modes, that is, we use around half of the total number of modes to formulate the long-term planning problem and track the motion for the rest of the modes using a short-term plan.

Global constants. We use time step $\Delta t = 1.0/120$ s for all our examples. The friction cone in Eq. (15) is approximated using six basis vectors defining a hexagonal boundary for the cone. The friction coefficient μ is chosen to be 1.0 for all the simulations.

Control parameters. We use the same set of weights in the control algorithm for synthesizing all the motions sequences. For Eq. (16), we define W_1 as a diagonal matrix with first three components corresponding to global positions as 0.1 and the remaining three corresponding to global rotations as 0.5. The matrix W_2 is an identity matrix, I, and $W_3 = 0.1I$. For Eq. (21), we choose $w_4 = 0.5$, $w_5 = 2.0$, and $W_6 = 5 \times 10^{-3} I$. In our experience, penalizing velocity terms more than the positions gives a more robust solution for control. Finally, we choose the weighting matrix in Eq. (26) as $W_a = 10^{-3} I$.

We use MOSEK (`mosek.com`) to solve the convex QPs formulated in Eq. (16) and Eq. (21). We solve the LCP in Eq. (34) through a C++ interface to Matlab's "$lcprog$" solver. All the results are synthesized on a single core of Intel Core 2 Duo 2.8 GHz processor.

7.1 Tracking a Reference Motion

With all the global constants and control parameters defined, the only parameter specific to each sequence is the window size for long-term planning.

We choose a window size of 12 frames to synthesize a biped character walking on a flat surface while tracking a mocap reference motion. In our experience, window sizes ranging from 10–15 work well with walking motion. We synthesize walking motion at 4–10 frames per second (fps) or 3–10% real-time speed. The bottle neck in control computation is solving the QP for the rigid modes (Eq. (16)) due to a large number of contact force parameters, and takes more than 80% of the computation time at every time step. The rest of the computation involves eigenvalue decomposition for modal analysis, QP solution for low-frequency modes actuation (Eq. (21)), and LCP for contact correction.

We use a window size of 16 frames for tracking a squatting motion and 24 frames for a swinging motion. For these motions, we simplified the QP problem for the rigid modes by choosing a single force for a contact point that lasts for a few frames in the window, thereby largely reducing the number of unknowns. As a result, the control algorithm can reach 30–40 fps or 25–30% real-time speed.

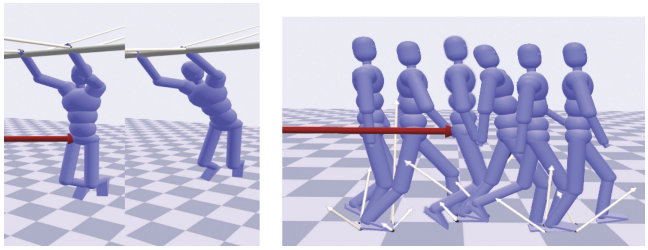
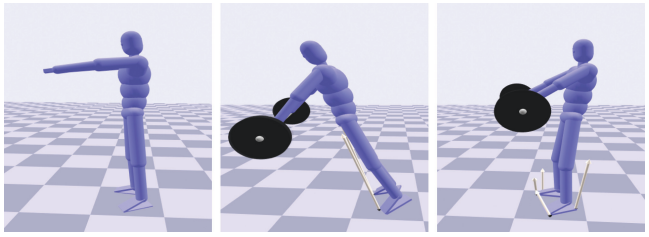


Fig. 2. Perturbation response while performing different tasks.



(a) original (b) shorter window (c) longer window

Fig. 3. Weight-lifting simulation with different window sizes.

7.2 Responding to External Perturbations

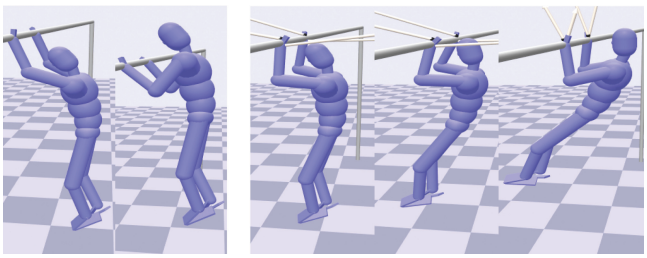
To demonstrate that the character is able to passively respond to arbitrary external perturbations, we synthesize responses of the character to user-applied forces while performing different tasks, such as walking, squatting, and swinging. When pushed at different body parts with different direction and magnitude of forces, the character reacts passively and eventually recovers (Figure 2). We apply forces varying between 100–175 N on walking motions for a period of 10 frames, 75–250 N on squatting for 20 frames, and 75–175 N on swinging for 20 frames.

7.3 Editing the Reference Motion

Our control algorithm also allows the reference motion to be modified online. We edit the reference motion using following three editing techniques.

FK/IK. We use a very simple online editing technique to modify the original walking motion to walking on a ramp. The editing only involves changing the vertical root translation to align with the slope and using IK to match the feet to the surface of the ramp. We synthesize motions walking up the slope of up to 3 degrees and walking down the slope up to 5 degrees. Similarly, we can modify the length of the strides in walking motion by changing the horizontal root translation and using IK to maintain the original duration of contacts. With these simple edits, we can increase the stride lengths by 10 cm and reduce it by 15 cm for every step of the reference motion.

To demonstrate more drastic changes on the reference motion, we modify the squatting motion to a weight-lifting motion by applying IK to the hands so they come into contact with a 30 kg barbell. The edited motion looks very unrealistic because the modification does not consider the changes in dynamics. However, the motion simulated by our system appears more natural as the character struggles to balance when she lifts the bell. In addition, we compare the results with two long-term window sizes of 12 and 36 frames (0.1 s and 0.3 s respectively, Figure 3). Simulating with 12 frame window, the character picks up the barbell but topples forward eventually.



(a) Key frames (b) simulation

Fig. 4. Simulation of a chin-up exercise from two key frames.

Increasing the window size to 36 frames allows the character to plan control forces further ahead of time. As a result, the character chooses to lean backward to balance the weight and remains steady after picking up the weight.

In addition to editing existing mocap sequences, we synthesize a completely new motion by interpolating a few edited keyframes. Starting out from a single pose of character hanging from the bar, we create a very rough chin-up motion by changing the vertical root position and applying IK to maintain the hand positions (Figure 4(a)). The character in this hand-crafted reference motion appears stiff and unnaturally strong. In contrast, the output motion exhibits realistic dynamics and responses to the external perturbations (Figure 4(b)).

Time warping. In addition to spatially editing the motion trajectories, we also edit the timing to speed up or slow down parts of a motion as desired. To demonstrate this, we speed up the walking motion by 10% and simultaneously increase the stride length by 15 cm per step. The character is able to walk steadily while tracking this edited motion. Time warping can also be used to aid in better recovery from unexpected perturbations. We give a strong backward push to a walking character. Losing forward momentum, the character fails to walk soon after the push because she is not able to keep up with the timing of the original motion. We repeat the experiment with time warping immediately after the push to let the character recover back to original motion. We slow down the reference motion by 25% for an interval of 1 second after the push. Now, the character is able to recover and walk steadily with a different phase as a result of warping.

Motion transition. Finally, we demonstrate tracking two different reference motions in sequence. We edit a jumping motion of the character by raising her arms. When the character is airborne, we add constraints on the hands to simulate bilateral contacts with a high-bar. The character passively swings on the bar while still tracking the jumping motion. We then transition into a swinging motion by blending into a swing reference motion over a time interval of 0.5 s. Just before the transition, the pose of the character differs from the reference motion including the contact points on the hand. We seamlessly synthesize the transition and the character starts tracking the swinging motion.

8. ANALYSIS

To demonstrate the effectiveness of our modal analysis approach, we conducted a few quantitative experiments and produced qualitative results to support the following two claims. First, we argue that dimension reduction is critical for online (re)planning and the modal analysis approach is very efficient due to modal decoupling. Second,

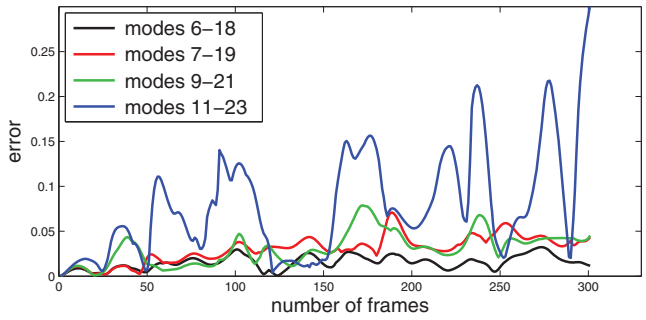


Fig. 5. Error comparison for control of different set of modes.

we argue that using a natural frequency to select controlled modes is appropriate for human motion.

Optimal control for a long horizon is computationally expensive and typically prohibitive for online applications. A standard method for long-term optimal control in computer animation is to formulate a spacetime optimization problem. For our problem, a 12 frame window of spacetime results in more than 700 unknowns and 70 constraints. This highly nonconvex optimization requires computation of several Jacobians and Hessians for every frame, leading to an excruciatingly slow motion synthesis method at a rate of several minutes per frame. On the other hand, modal analysis reduces computational time by transforming the optimal control problem into a low-dimensional space. Furthermore, modal analysis exploits the fact that the equations of motion can be decoupled, which significantly improves the performance in addition to the speed gain from dimension reduction. The details of performance can be found in Section 7.1.

For online long-term optimal control, the necessity of dimension reduction is evident, but the selection of the low-dimensional space is also critical. Similar to PCA and other spectral embedding methods, modal analysis reduces the domain to a subspace spanned by a subset of modes, which can be organized in a specific order. For modal analysis, the modes are ordered based on the natural frequencies of the physical model. We conduct two experiments to show that mode selection based on natural frequency can effectively control human motion.

Our algorithm applies long-term control only in low-frequency modes. To justify the choice of this control scheme, we compare the motions generated by selecting different frequency ranges. In the first experiment, our algorithm selects the lowest-frequency modes (mode 6 to 18) to simulate a walking sequence. This baseline is compared against motion sequences generated with control modes numbered from $6 + k$ to $18 + k$, where $k \in \mathbb{Z}^+$. The error metric used for comparison is defined as the norm of the difference in the global orientation in the simulated and the input motion. Figure 5 shows sequences controlled by the modes in different frequency ranges. We notice that the error increases and the system becomes more unstable as the value of k is increased. We also simulate a few motion sequences with randomly selected modes. The simulation becomes unstable quickly after a few frames. These results suggest that controlling the low-frequency modes leads to the most stable control algorithm.

As we demonstrated, dimension reduction based on natural frequency is a viable approach for long-term, online optimal motion control, but what is the cutoff threshold that defines "low" frequency? In the second experiment, we analyze our choice of the threshold for classifying the modes as low frequency. Our chosen threshold works for all of the demonstrated examples. In our experience, the control algorithm works successfully within a variance of 3–4 modes from the chosen threshold. If the chosen threshold is too small, most of the modes are categorized as the high-frequency modes, resulting in very kinematic, less responsive motion. On the other hand, if the threshold is too large the resulting motion becomes very jerky and unstable. This is because the objective function in our control policy in Eq. (21) penalizes the actuation in the modes below the threshold. However, modes with high frequency require large forces to execute the desired motion; hence our control policy for low-frequency modes is clearly not suited for these modes.

The recent work of Ye and Liu [2010] shares some common goals of our method of simulating a given reference motion and its online variations using a generic control method. They perform an offline control computation based on a given reference motion that restricts their method to handle small changes during online simulation. In contrast, our completely online control computation handles larger changes in the reference motion. To compare the performance of these two methods, we analyze two examples: first, walking motion with perturbations and second, squatting motion while lifting heavy weights. In the first example, it is not necessary to change the reference motion online since the perturbations are relatively small. Therefore, the computation time for the method in Ye and Liu [2010] consists of one time offline computation of 1–2 minutes and online feedback computation that runs at an average of 20 fps. Our online control computation runs at 4–10 fps. For the second example of squatting while lifting weights, it is required for the method in Ye and Liu [2010] to change the reference trajectory to successfully perform the motion. The control has to be reevaluated frequently that would make their computation prohibitively expensive during online simulation. In our method, the control computation runs at 30–40 fps thereby outperforming the other method.

9. DISCUSSION

We present an algorithm to interactively simulate and control an underactuated articulated character using a time-varying linear dynamic system. We apply modal analysis to transform the space of DOFs to the modal space based on the natural frequencies. The design of the control scheme exploits modal analysis to reduce the control space and decouple the equations of motion. This approach offers two key advantages.

(1) Robust control. Long-horizon optimization produces look-ahead control that allows the underactuated system to operate robustly when tracking the reference motion and recovering from small perturbations.

(2) Online editing of reference trajectory. Because the look-ahead control is computed online at every time step, our algorithm does not rely on any offline computation for the control policy. This allows us to edit the reference trajectory online in response to large changes in the environment.

We demonstrate the ability to synthesize anticipatory motion and passive responses to external perturbations. The control policy is generic as it does not depend on the performed task. In addition, chosen weights and parameters work with a wide variety of motions and scenarios.

However, our approach suffers a few limitations. Our control algorithm works better for physically correct reference motions. More challenging activities, such as walking, require a higher-quality reference motion than other activities, such as squatting. To use motion capture data as reference, we must ensure two conditions. First, we

compute the contact forces required to achieve the global trajectories (six underactuated degrees of freedom) from the reference motion. The computed contact forces need not satisfy the dynamic and friction constraints for the entire reference motion, but the longest interval that violates these constraints must be less than 10–15 frames (at 120 fps). Second, the contact points should remain in contact with the environment for their expected duration. For the first condition, we simply use a motion capture data that looks physically plausible without glaring artifacts. For the second condition, we preprocess the motion by applying inverse kinematics. To utilize keyframe animation or physically inconsistent mocap data (violating the afore-said two conditions), we could employ the spacetime optimization process as used in Muico et al. [2009] as a preprocess to produce higher-quality reference motion. Online editing of the reference motion should also satisfy these two conditions. Our current editing method modifies the root position in accordance with the changes in the contact positions. This simple kinematic editing might result in motions that occasionally violate the equations of motion, but as long as the physically inconsistent frames do not stretch longer than 10–15 frames, our controller can successfully handle the edited motion.

Although our system can synthesize large variations from the reference motion by editing the reference motion online, the modification is artificially done by user intervention, rather than caused by a physical response. Since the modification of the reference motion does not incorporate high-level strategies based on human postural responses to physical perturbations, the character can only handle relatively small pushes. To recover from large perturbations using drastically different control strategies, it is inevitable to modify the reference trajectory or use an entirely different trajectory. Our method suffers the inherent drawback of the tracking controller in that deviations have to be confined in a small neighborhood of the reference trajectory. However, since our controller does not rely on any offline computation based on the reference trajectory, we can directly modify the trajectory online, resulting motion largely different from the original reference trajectory (e.g., from squatting to weight lifting).

The contact information and the timing is obtained from the reference motion. In the event of perturbation, we continue to use the same contact information for planning, but the actual contacts might be different from those in the reference motion, for example, earlier or later heel strike for double support. To increase the robustness in recovering from larger perturbations and environment changes, we would like to design dynamic policies to estimate the contact position and the timing.

We use a simple rigid body to model the foot with its four corners as the contact points. We suspect that the forces generated in our system have significant discrepancy from those measured by the force sensing platforms due to this simplification in modeling of foot geometry and contacts. However, though the individual forces computed at the contact points may not be realistic, for example, the forces at the corners face opposite directions, the aggregated force strictly satisfies the following two physical constraints. First, the center of pressure lies within the convex hull formed by the contact points on the foot. Second, the net ground reaction force produced by the foot obeys the Coulomb friction law. These constraints are enforced by solving a LCP (Eq. (34)), which corrects the difference in expected contact forces by the character and the actual physical forces.

Our control algorithm does not run in real time. The bottleneck is the estimation of contact forces for controlling the rigid body modes, especially when the number of contacts is large. One possible solution is to replan less frequently for control forces.

Size of the planning window plays an important role in the control policy. Despite the apparent advantages of having larger window sizes for farther look ahead, approximation of dynamics for the entire window introduces errors that limits us from choosing arbitrarily large windows. For example, choosing window size of 1 frame is similar to having a per-frame PD control. In this case, the character is extremely stiff and fails to track underactuated tasks. Increasing the window size significantly improves the stability and ability to handle perturbations. However, the quality of control starts to degrade if the window size is too large as large errors accumulate due to the approximation of the dynamic equations (i.e., linearization of dynamics around the current frame). In the current implementation, the window size is chosen manually for every motion. We would like to explore ways of choosing the optimal window size automatically and adaptively based on the reference motion and the simulated state.

Approximation of the dynamics equation in Eq. (11) works well for demonstrated motions such as walking. This linearization around zero velocity may introduce larger errors in dynamics for motions involving high joint velocities. We would like to explore higher-order approximations to the dynamic equations that are more suited to high-velocity motions.

We demonstrated our examples based on an underdamped system, specifically with zero damping. However, our control design is generic to include overdamped systems as well. We choose an underdamped system because it better captures the natural dynamics that exploits the passive elements of a character.

Currently, we manually choose the constant stiffness parameters for the articulated character irrespective of the reference motion. Inspired by biomechanics studies on how stiffness of the passive elements varies by different activities [Farley and Morgenroth 1999], we would like to automatically design the stiffness parameters of the character based on the reference motion. Our design goal is to solve an inverse problem of modal analysis such that the desired motion resembles the motion of the biomechanical system vibrating at its natural frequencies.

APPENDIX

A. LINEAR COMPLEMENTARITY PROBLEM

Using Eq. (28) and Eq. (33), we can write $\mathbf{p}_1$ as a function of $\mathbf{z}$.

$$\mathbf{p}_1 = \left(\mathbf{p}_1^* - A(\Delta t)(Z_0 + X_f)\mathbf{f}_0^*\right) + \left(A(\Delta t)(Z_0 + X_f)N\right)\mathbf{z} \quad (35)$$

Using Eq. (30), Eq. (31) and Eq. (32), we write the complementarity conditions for each contact point $\mathbf{x}_i$ as $\mathbf{w}_i = F_i\mathbf{p}_1 + G_i\mathbf{z}_i$ and $\mathbf{w}_i^T\mathbf{z}_i = 0$ with $\mathbf{w}_i, \mathbf{z}_i \geq \mathbf{0}$, where

$$F_i = \begin{pmatrix} \hat{\mathbf{n}}_i J_i \Phi \\ D_i^T J_i \Phi \\ 0 \end{pmatrix} \quad \text{and} \quad G_i = \begin{pmatrix} 0 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{e} \\ \mu & -\mathbf{e}^T & 0 \end{pmatrix}.$$

Stacking the equations for all the contact points together, we get $\mathbf{w} = F\mathbf{p}_1 + G\mathbf{z}$ and $\mathbf{w}^T\mathbf{z} = 0$ with $\mathbf{w}, \mathbf{z} \geq \mathbf{0}$, where matrices $F = (F_1^T, \ldots, F_{n_0}^T)^T$ and $G = \text{blockdiag}(G_1, \ldots, G_{n_0})$. Substituting Eq. (35) in the preceding we get the following LCP problem.

$$\mathbf{w} = C\mathbf{z} + \mathbf{h} \quad \text{and} \quad \mathbf{w}^T\mathbf{z} = 0 \text{ with } \mathbf{w}, \mathbf{z} \geq \mathbf{0}$$

where

$$\begin{aligned} C &= FA(\Delta t)(Z_0 + X_f)N + G \\ \mathbf{h} &= F\left(\mathbf{p}_1^* - A(\Delta t)(Z_0 + X_f)\mathbf{f}_0^*\right) \end{aligned}$$

ACKNOWLEDGMENTS

We thank Professor Jun Ueda for technical discussions in the early phase of the project.

REFERENCES

ABE, Y., DA SILVA, M., AND POPOVIĆ, J. 2007. Multiobjective control with frictional contacts. In *Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation*. 249–258.

ABE, Y. AND POPOVIĆ, J. 2006. Interactive animation of dynamic manipulation. In *Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation*. 195–204.

ALLEN, B., CHU, D., SHAPIRO, A., AND FALOUTSOS, P. 2007. On the beat!: Timing and tension for dynamic characters. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 239–247.

ANITESCU, M. AND POTRA, F. A. 1997. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlin. Dynam. 14*, 231–247.

BARBIČ, J., DA SILVA, M., AND POPOVIĆ, J. 2009. Deformable object animation using reduced optimal control. *ACM Trans. Graph. 28,* 3, 1–9.

CHAI, J. AND HODGINS, J. K. 2007. Constraint-Based motion optimization using a statistical dynamic model. *ACM Trans. Graph. 26,* 3, 8.

COHEN, M. F. 1992. Interactive spacetime control for animation. In *Proceedings of SIGGRAPH*. Vol. 26. 293–302.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2010. Generalized biped walking control. *ACM Trans. Graph. 29,* 4, 1–9.

DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Interactive simulation of stylized human locomotion. *ACM Trans. Graph. 27,* 3, 1–10.

DE LASA, M. AND HERTZMANN, A. 2009. Prioritized optimization for task-space control. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-based locomotion controllers. *ACM Trans. Graph. 29,* 4, 1–10.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 1997. Dynamic free-form deformations for animation synthesis. *IEEE Trans. Vis. Comput. Graph. 3,* 3, 201–214.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *Proceedings of SIGGRAPH*. 251–260.

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

FARLEY, C. AND MORGENROTH, D. 1999. Leg stiffness primarily depends on ankle stiffness during human hopping. *J. Biomech. 32*, 267–273.

GEORGOPOULOS, A., KALASKA, J., AND MASSEY, J. 1981. Spatial trajectories and reaction times of aimed movements: Effects of practice, uncertainty and change in target location. *J. Neurophy. 46*, 725–743.

HAUSER, K. K., SHEN, C., AND O'BRIEN, J. F. 2003. Interactive deformation using modal analysis with constraints. In *Proceedings of the Graphics Interface Conference*. 247–256.

HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In Proceedings of the *SIGGRAPH Conference*. 71–78.

JAIN, S., YE, Y., AND LIU, C. K. 2009. Optimization-based interactive motion synthesis. *ACM Trans. Graph. 28,* 1, 1–10.

JAMES, D. L. AND PAI, D. K. 2002. Dyrt: Dynamic response textures for real time deformation simulation with graphics hardware. In *Proceedings of the SIGGRAPH Conference*. 582–585.

KRY, P. G., REVERET, L., FAURE, F., AND CANI, M.-P. 2009. Modal locomotion: Animating virtual characters with natural vibrations. *Comput. Graph. Forum 28,* 2.

KUDOH, S., KOMURA, T., AND IKEUCHI, K. 2006. Stepping motion for a human-like character to maintain balance against large perturbations. In *Proceedings of the ICRA Conference*. 2661– 2666.

LASZLO, J., VAN DE PANNE, M., AND FIUME, E. 1996. Limit cycle control and its application to the animation of balancing and walking. In *Proceedings of the SIGGRAPH Conference*. 155–162.

LEE, Y., KIM, S., AND LEE, J. 2010. Data-driven biped control. *ACM Trans. Graph. 29,* 4, 1–8.

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

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

MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. R. 2009. Momentum control for balance. *ACM Trans. Graph. 28,* 3, 1–8.

MIALL, R. C., WEIR, D. J., AND STEIN, J. F. 1985. Visuomotor tracking with delayed visual feedback. *Neurosci. 16,* 3, 511–520.

MORDATCH, I., DE LASA, M., AND HERTZMANN, A. 2010. Robust physics-based locomotion using low-dimensional planning. *ACM Trans. Graph. 29,* 4, 1–8.

MUICO, U., LEE, Y., POPOVIĆ, J., AND POPOVIĆ, Z. 2009. Contact-Aware nonlinear control of dynamic characters. *ACM Trans. Graph.* 1–9.

POPOVIĆ, Z. AND WITKIN, A. 1999. Physically based motion transformation. In *Proceedings of the SIGGRAPH Conference*. 11–20.

RAIBERT, M. H. 1986. *Legged Robots That Balance*. Massachusetts Institute of Technology, Cambridge, MA.

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

SHABANA, A. A. 1997. *Vibration of Discrete and Continuous Systems*. Springer.

SHARON, D. AND VAN DE PANNE, M. 2005. Synthesis of controllers for stylized planar bipedal walking. In *Proceedings of the ICRA Conference*.

SHIRATORI, T., COLEY, B., CHAM, R., AND HODGINS, J. K. 2009. Simulating balance recovery responses to trips based on biomechanical principles. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 37–46.

SOK, K. W., KIM, M., AND LEE, J. 2007. Simulating biped behaviors from human motion data. *ACM Trans. Graph. 26,* 3, 107.

STEWART, D. E. AND TRINKLE, J. C. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Int. J. Numer. Methods Engin. 39,* 15, 2673–2691.

SULEJMANPAŠIĆ, A. AND POPOVIĆ, J. 2005. Adaptation of performed ballistic motion. *ACM Trans. Graph. 24,* 1.

VAN DE PANNE, M. AND LAMOURET, A. 1995. Guided optimization for balanced locomotion. In *Proceedings of the Computer Animation and Simulation Conference*. 165–177.

WANG, J. M., FLEET, D. J., AND HERTZMANN, A. 2009. Optimizing walking controllers. *ACM Trans. Graph. 28,* 5, 1–8.

WANG, J. M., FLEET, D. J., AND HERTZMANN, A. 2010. Optimizing walking controllers for uncertain inputs and environments. *ACM Trans. Graph. 29,* 4, 1–8.

WITKIN, A. AND KASS, M. 1988. Spacetime constraints. In *Proceedings of the SIGGRAPH Conference*. Vol. 22. 159–168.

WOOTEN, W. L. 1998. Simulation of leaping, tumbling, landing, and balancing humans. Ph.D. thesis, Georgia Institute of Technology.

WU, J.-C. AND POPOVIĆ, Z. 2010. Terrain-Adaptive bipedal locomotion control. *ACM Trans. Graph. 29,* 4, 1–10.

YE, Y. AND LIU, C. K. 2008. Animating responsive characters with dynamic constraints in near-unactuated coordinates. *ACM Trans. Graph. 27,* 5, 1–5.

YE, Y. AND LIU, C. K. 2010. Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph. 29,* 4, 1–9.

YIN, K., CLINE, M. B., AND PAI, D. K. 2003. Motion perturbation based on simple neuromotor control models. In *Proceedings of the Pacific Graphics Conference*. 445.

YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. Simbicon: Simple biped locomotion control. *ACM Trans. Graph. 26,* 3, 105.

ZORDAN, V. B. AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation*. 89–96.

Received October 2010; revised February 2011; accepted May 2011