Hierarchical Planning and Control for Complex Motor Tasks

Daniel Zimmermann[1,3] Stelian Coros[1,2] Yuting Ye[4] Robert W. Sumner[1,3] Markus Gross[1,3]
[1]Disney Research Zurich [2]Carnegie Mellon University [3]ETH Zurich [4]Industrial Light and Magic

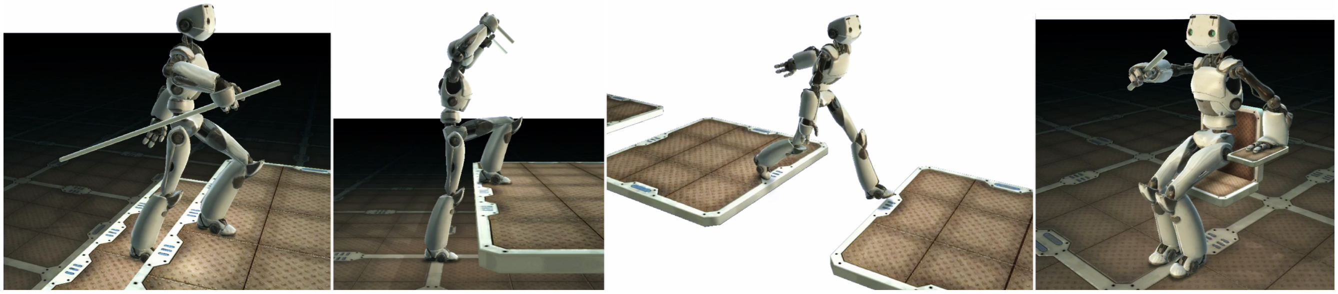


Figure 1: *A variety of motor skills generated by our planning and control framework.*

Abstract

We present a planning and control framework that enables physically simulated characters to perform various types of motor tasks. To create physically-valid motion plans, our method uses a hierarchical set of simplified models. Computational resources are therefore focused where they matter most: motion plans for the immediate future are generated using higher-fidelity models, while coarser models are used to create motion plans with longer time horizons. Our framework can be used for different types of motor skills, including ones where the actions of the arms and legs must be precisely coordinated. We demonstrate controllers for tasks such as getting up from a chair, crawling onto a raised platform, or using a handrail while climbing stairs. All of the motions are simulated using a black-box physics engine from high level user commands, without requiring any motion capture data.

1 Introduction

A long term goal of physics-based character animation is the creation of "directable characters" that are as easy to control as human actors. Such characters should be able to *autonomously* follow high level commands in arbitrary environments. However, even seemingly simple tasks such as standing up from chairs or climbing onto a platform require complex motor skills with coordinated use of hands and feet. Simulated characters must choose when and where to step or grab hold of the environment, and apply appropriate control forces to maintain balance. Despite decades of research in bipedal locomotion, a general, unified strategy for navigation planning in arbitrary environments remains out of reach. The recent DARPA Robotics Challenge [DAR12] highlights the same open research problem in robotics: control policies are needed to enable the use of legged robots in disaster-relief situations such as the failed Fukushima nuclear reactor, where efficient and coordinated movements in unstructured environments is paramount.

As a step towards addressing this problem, we present a planning and control framework that enables physically simulated characters to autonomously perform a wide variety of motor skills, many of which require the arms and legs to be used in unison. The major challenge is to plan for appropriate contacts in a constrained environment, and do so efficiently for interactive applications such as video games. A general solution can be time-consuming to compute [MTP12] because the contact plan has to account for the character's dynamics for the entire duration of the task. Instead of formulating a complex, long-horizon optimization problem to solve for the full body motion trajectory, we employ a hierarchical set of simplified dynamic models that plan at different time horizons.

Our hierarchical model allows computational resources to be focused where they are needed most: motions for the immediate future are modeled and planned to a higher-degree of accuracy, while approximate but still physically valid plans are used on a longer horizon to guide the immediate plan. In particular, for locomotion-based tasks, a long term planner uses linear inverted pendulums to lay out appropriate contact regions in constrained environments. A medium term horizon planner then uses a floating rigid body model (FRB) together with these contact regions to plan the exact timing, locations, and contact force trajectories for all the planned hand-holds and foot-holds. A short-term planner computes optimal ground reaction forces for the full set of contact points available at every simulation time step, with the goal of tracking the motion trajectory output by the medium-term planer. Finally, a full-body motion controller computes joint torques using a combination of Jacobian Transpose control and implicit Proportional Derivative controllers. These joint torques are then fed to a black-box simulator to generate the character's motion. New long- and medium-term plans are optimized while the previous ones are being executed, so that the character can continuously execute lengthy actions while adapting to changing or previously unforeseen scenarios.

The long-term planner is very efficient thanks to the closed-form solution of the inverted pendulum model (IPM). However, it falls short of modeling changes in upper body orientation and arbitrary (or the absence of) environment contacts, and is thus unsuitable for the types of motor tasks we aim to handle. On the other hand, the FRB model incorporates a general contact model and rotation motions around the COM, which are essential for complex tasks. Therefore, the medium-term planner is more accurate, albeit at the expense of solving a more involved numerical optimization problem. To mitigate the added complexity, we initialize the medium-term planner with the result of the long-term planner in order to improve convergence rates. We also employ a new force parameterization to reduce the search space: contact forces are specified in a moving coordinate frame whose vertical axis is aligned with the vector between the contact point and the COM. Because this parameterization effectively decouples the effect of linear and angular forces, we can use linear interpolation on a small number of force samples in the optimization. The resulting contact locations and forces give rise to a desirable COM trajectory that includes purposeful changes in body orientation. When these contact parameters are directly used in simulation however, the full-body character motions inevitably drift away from the planned motion trajectory. As a remedy, the short-term planner re-computes optimal target ground reaction forces for the current set of contact points between the character and the environment at every time step. These forces can lead to control torques that better follow the planned motion. Our three-tier planning hierarchy can produce physically plausible plans for motions featuring a large variety of contact configurations.

The core technical contributions of our work include the hierarchical planning and control framework, the representation of a character using models with varying levels of abstraction, the new parameterization of contact forces used to compute the motion of the floating rigid body, and the demonstration of a new set of physically-based motion controllers that extend the types of motor tasks being shown to date. We validate our system by generating a number of examples showing complex motor skills such as standing up from chairs, crawling onto high platforms, and walking up steep stairs with the aid of handrails. The motions of the character are generated using a black-box physics engine.

2 Related Work

Locomotion control is a core topic in animation research. Early work developed specialized controllers for walking [RH91; LvdPF96], running [HWBO95], jumping and diving [Woo98]. Among the various skills, walking controllers have seen tremendous improvements in robustness [YLvdP07; WFH10] and generality [CBvdP10]. To a large extent, the focus so far has been on generating controllers for legged locomotion in relatively simple environments. However, our ability to skillfully navigate arbitrary, unstructured environments requires a rich variety of motor tasks that extend well beyond bipedal walking. Such motor skills typically involve the coordinated use of arms and legs, and provide the inspiration for our work.

Perhaps the most general approach to motion synthesis is trajectory optimization or optimal control from high level task goals. Continuous optimization has been applied to create motion trajectories from scratch [WK88; SC92], adapt motion styles to different tasks [LHP05], or for interactive control of an autonomous character [JYL09]. On the other hand, sampling-based approaches are better suited for gait pattern discovery. The pioneering work of Sims [Sim94] inspired many others to use genetic algorithms to mimic evolutionary processes. More recently, the Covariance Matrix Adaptation (CMA) [Han06] method was introduced to the animation community where it was used to discover locomotion gait patterns for a variety of character types [WP09; WPP13; GvdPvdS13] including virtual humans [WHDK12]. Though effective, these methods are very demanding computationally due to the expensive evaluation of each sample. We therefore use CMA in conjunction with a simplified representation of the character.

Simplified models are commonly used to approximate the dynamics of a high dimensional system in motion planning. A general point-mass model is most favorable due to it simplicity [vdP97]. For bipedal walking, the inverted pendulum model is perhaps the most widely used [KKK*01; SCCH09; CBvdP10]. Mordatch et al. [MdLH10] use the Spring Loaded Inverted Pendulum (SLIP) model to effectively plan footsteps and the COM trajectory. Their work is most similar to ours except that the SLIP model doesn't capture angular motions and arbitrary contact configurations. Instead of modeling rotation directly, an enhanced point-mass model with angular momentum is usually sufficient for motion analysis [SL11], editing [PBM00; LP02], and tracking [YL10]. Inspired by this body of work, we employ a model that explicitly captures body rotation while allowing for arbitrary contact plans that involve both hands and feet. To mitigate the added complexity of this model, we propose a novel force parameterization that largely decouples linear and angular components, thus drastically reducing the interpolation resolution of force samples while retaining the flexibility needed to generate controllers for a variety of motor skills.

Since contact phases are paramount to the success of most control tasks, much research has been aimed at developing techniques for handling discrete contact dynamics. In addition to sampling in a meaningful space for the task at hand [TJ07; LYvdP*10; YL12], a smooth approximation of the contact dynamics was shown to be successful. One can use soft contact geometry [JL11], or formulate a smooth complimentary problem with slack variables [Tod11]. Mordatch et al. [MTP12] use the latter approach to synthesize a variety of tasks that require complex contact planning and coordination. They later show that the same approach can be combined with detailed muscle models to create realistic human walking motions [MWTK13]. While our work shares the same goal, the hierarchical set of simplified models we use improves performance significantly, and the results we obtain are generated using a standard, black-box physics simulator.

In some of the most recent work in this area, Han and his colleagues [HNJS14] describe a control framework similar to that presented by Ye and Liu [YL10]. A motion graph is used to drive the motion of the simulated character, and a floating body model is used to compute feedback policies. However, the exact timing of the contact phases are explicitly provided by the input data and cannot be adapted. In contrast, the timing of foot/hand holds can be optimized by our planner, so they do not have to be fixed a priori, and new contact phases can easily be added on the fly.

3 Motion Planning

Model Predictive Control (MPC) is a well-established control paradigm, where simplified models are used to predict and plan the dynamic behavior of complex systems. In controlling high-dimensional articulated figures, MPC has proven to be effective for both robotics and computer animation applications. However, choosing a simplified model that provides an optimal trade-off between predictive power and computational efficiency is still an open problem. To address this, our framework employs a hierarchical set of simplified models, as illustrated in Figure 2, which allows computational resources to be focused on creating detailed motion plans for the near future, while ensuring that coarser, yet physically-plausible plans with a longer planning horizon are generated as well. In particular, a linear inverted pendulum model (IPM) [KKK*01] is used to compute an initial footstep plan for walking tasks, using a long planning horizon. In scenarios where constraints in the environment restrict the locations of the footholds, the IPM is used to determine the discrete stepping regions the character should aim for with every step. A floating rigid body (FRB) model [PBM00; YL10; MTP12; HNJS14] is used to compute motion plans for medium-term planning horizons of 1-2s. The FRB model predicts the effect of arbitrary contact configurations involving both hands and feet, and it accurately captures the coupling between linear and angular body motions. A novel parameterization of the contact phases used by the FRB model leads to motion plans that can be efficiently generated. A short-term motion plan-

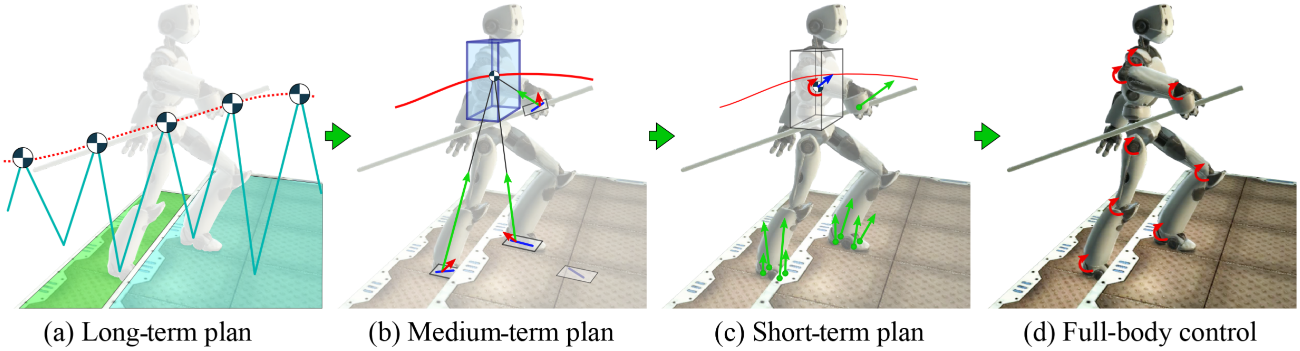


Figure 2: *Overview of our planning and control framework. Simplified models of increasing fidelity are used hierarchically to generate optimized motion plans: a) a long-term planner uses inverted pendulums to compute an initial footstep plan for walking tasks; b) the medium-term planner employs a floating rigid body model to refine the motion plans by considering the effect of arbitrary contact configurations involving the hands and the feet; c) the short-term planner further increases the fidelity of the simplified model — the exact contact configuration at every simulation step is used to compute optimal ground reaction forces to track the planned trajectories; d) the full-body controller computes joint torques that reproduce the desired ground reaction forces output by the short-term planner.*

ner further increases the fidelity of the simplified model used to approximate the dynamics of the character. At this stage, the character is still represented as a floating body, but the exact contact configuration of the character at every simulation time step is used to compute the optimal set of ground reaction forces (GRF) needed to track the FRB motion plan.

3.1 Long-term planning

For walking tasks, the ability to make decisions by considering the longer-term impact plays an important role in the way virtual characters should move. For instance, before stepping over an incoming gap, a character needs to predict how much momentum it needs to make it over, and whether it is better to step before or after the gap. The discrete aspects of such planning problems require an efficient method that can be used to quickly evaluate the feasibility of different options. To this end, we develop a simple and efficient long-term planner based on linear inverted pendulum models (IPM), similar to Zimmermann [Zim13].

Inputs to the long-term planner consist of environment constraints, user commands (e.g. desired walking speed, step duration, final position etc.), and the initial COM state of the character. Based on the type of environment the character is in, the ground is decomposed into distinct regions where footsteps are considered to be safe. For instance, regions within 3*cm* of the start of stairs are deemed unsafe to step on, as are gaps in the terrain. The goal of this planner is to efficiently choose a sequence of safe stepping regions and determine one or more ways in which the character can traverse the environment while satisfying user commands.

We begin by laying out a sequence of IP plans onto the environment, as illustrated in Fig. 3. The character's current COM position and velocity are used as initial conditions for the first IP phase. The initial configuration of each subsequent IP phase is set to the final predicted state of the previous phase. Given as input the desired speed at the end of each step, as well as the step duration, appropriate locations for the COP of each IP phase are computed efficiently using a closed-form solution [KKK*01; MdLH10]. When the computed COP location of an IP phase falls within an unsafe region (Fig. 3 (a)), we project it to all nearby reachable stepping regions (Fig. 3 (b)) to obtain different strategies for negotiating the environment. However, this operation can significantly affect the computed COM trajectory of the current IP phase, and therefore future phases. To reduce the introduced disturbance, we recompute the target COP position of the previous IP phase, again using the closed-form solution of the linear IPM, but this time concurrently considering the coupled dynamics of both the previous and current IP phases (Fig. 3 (c)).

We use this simple algorithm to create a variety of plausible motion plans with a planning horizon of 5 steps. As a means of exploring multiple plans in parallel, we keep a queue of the most promising plans encountered, sorted by how well the plan satisfies user objectives (Sec. 3.2.2). The n most promising motion plans are selected to further expand on. We used $n = 5$ in all our experiments. After the desired number of steps have been planned, the long-term planner returns the best plan that was found, which is then used as input for the medium-term planner.

3.2 Medium-term planning

While the long-term planner can be used to quickly create plausible motion plans using a simple model, the IPM cannot capture significant aspects of the character's dynamics. The long-term planner therefore cannot be used to plan body orientations, and motions that involve flight phases or concurrent contacts involving hands and feet. To address these limitations, the medium-term planner uses a higher fidelity model to predict the dynamics of the character.

3.2.1 Floating Rigid Body

We approximate the character's body dynamics as a floating rigid body (FRB) moving under the influence of gravity and contact forces. The state of the simplified model is defined as

$$\mathbf{x} = (\mathbf{c}_3, \mathbf{v}_3, \mathbf{R}_{3\times 3}, \mathbf{w}_3),$$

where the individual state variables represent the position, velocity, orientation, and angular velocity of the body respectively. Modeling rotations explicitly allows us to plan motions that involve significant body rotation as compared to the models used by Ye and Liu [YL10] and Han et al. [HNJS14].

Similar to Mordatch et al. [MdLH10], we plan individual contact phases for each limb that is considered by the planner. The *contact phase* Ω of a limb is defined by the starting time t_s, end time t_f, a trajectory of the COP $\mathbf{p}(t)$, and a trajectory of the aggregated contact force $\mathbf{f}(t)$ acting on $\mathbf{p}(t)$, as illustrated in Figure 2(b). The trajectories $\mathbf{p}(t)$ and $\mathbf{f}(t)$ are parameterized as piece-wise linear curves with h and n control points respectively. The control points for the COP are parameterized by the environment they are on (e.g. along a line for handrail, or on a 2D plane for the ground). Depending on the type of environment, we only need one or two scalar parameters to define a control point $\mathbf{p}_i$. The set of control parameters we optimize for (Sec 3.2.2) therefore contains all N active contact phases $\mathbf{u} = (\Omega^0, \ldots, \Omega^N)$, where $\Omega = (t_s, t_f, \mathbf{p}_0, \ldots, \mathbf{p}_h, \mathbf{f}_0, \ldots, \mathbf{f}_n)$.

Force parameterization To plan efficiently, we want to use as few control parameters as possible. Linear COP trajectories are often sufficient for walking tasks [MdLH10], but we found that our model requires a large number of control points for contact forces due to the coupling between linear and angular motions. Consider, for example, the case when only two control points are used to define the force trajectory over a relatively large time window (on the order of 0.5s-1.0s). The force applied at any moment in time needs to be large enough to support the weight of the body. At the same time, while this force is linearly interpolated, its moment arm, which is related to the vector between the COP and the COM, changes non-linearly. Therefore, even if the contact forces are chosen so that they do not induce changes in the angular momentum at the start and end of the contact phases, unwanted torques can be induced on the body nevertheless. This problem can be alleviated by increasing the number of control points to modulate contact forces at a much finer granularity, but with significant computational overhead because of the enlarged parameter space that must be explored.

We propose to instead parameterize the contact forces using two distinctive sets of parameters: $\mathbf{f}(f_l, \mathbf{f}_r)$. The first parameter f_l is a scalar representing the magnitude of the force acting along the vector $\mathbf{c} - \mathbf{p}$, from the COP of the contact phase to the COM of the body frame (Figure 2(b), green). Because they go through the COM, these forces affect only the linear motion of the body, and are typically used to support the body weight while walking. The second parameter $\mathbf{f}_r$ is a force vector acting along any arbitrary direction, thus influencing both the linear and angular momentum (Figure 2(b), red). In practice, $\mathbf{f}_r$ is usually quite small because f_l already accounts for a majority of the body weight. Therefore, this parameterization effectively reduces the coupling between the linear and angular motion, allowing our system to plan a variety of motions with as few as two control points for each contact force trajectory. With this parametrization, we can formulate the equation of motion at time t as follows, using m to denote mass and $\mathbf{g}$ to denote gravity.

$$\begin{aligned} m\ddot{\mathbf{c}}(t) &= m\mathbf{g} + \sum_i (\mathbf{f}_r^i(t) + (\mathbf{c}(t) - \mathbf{p}^i(t)) f_l^i(t)), \\ \dot{\mathbf{L}}(t) &= \sum_i (\mathbf{p}^i(t) - \mathbf{c}(t)) \times \mathbf{f}_r^i(t). \end{aligned} \tag{1}$$

The force and COP trajectories in a contact phase $[t_s^i, t_f^i]$ are evaluated using piece-wise linear interpolation of control points $\mathbf{f}(t_s^i)$, $\mathbf{f}(t_f^i)$, $\mathbf{p}(t_s^i)$, and $\mathbf{p}(t_f^i)$. Starting from an initial state $\mathbf{x}(0)$, we compute $\mathbf{x}(t)$ by a numeric simulation of Eqn (1). The simulation time step is fixed to $1/30s$ for all our experiments.

Parameter filtering Not all parameters of the COP and contact forces are physically valid for the simulation of the FRB model. Before each forward simulation step, we filter these parameters by projecting them to the closest point within their respective constraints, so that the simulated motions are plausible.

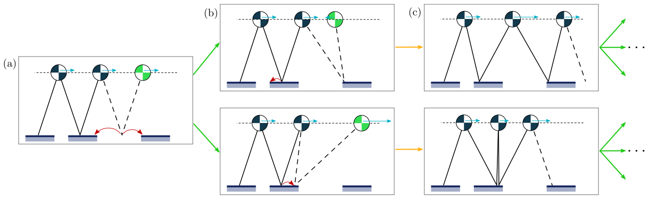


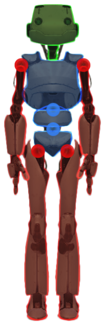
Figure 3: *A schematic view of the long-term planning stage.* (a) *Prediction: Footstep location is predicted using a linear inverted pendulum model. Predicted footsteps are projected to nearby safe regions (blue boxes).* (b) *For each reachable stepping region, a separate plan is generated (green arrows). Moving the footstep locations within the stepping region yields potentially unwanted COM velocities (blue arrows).* (c) *Re-planning the center of pressure of the previous IP phase improves the prediction for the next step.*

Each COP trajectory is constrained to lie within the support polygon of the limb that the contact phase is planned for. Given a COP trajectory from the planner, we project its control points to a bounding rectangle whose dimensions match those of the corresponding limb. If this limb is already in contact, then the position of the bounding rectangle is set to the world coordinates of the contact area. If the COP trajectory is for a future contact phase, we first position the bounding rectangle at the center of the trajectory $\mathbf{p}_c$, then project each one of the control points onto it. This allows the global location of the contact phase (i.e. foot or hand placements) to be modulated, while ensuring that the entire COP trajectory can be covered by the support polygon of the limbs.

Similarly, we need to make sure the contact forces are always valid, and subject to the type of physical constraints that apply to the simulated character. First, for contact phases corresponding to unilateral constraints (e.g. feet on the ground), the net force $\mathbf{f} = \mathbf{f}_r + (\mathbf{c} - \mathbf{p}) f_l$ is restricted to have a positive component along the contact normal, and projected to lie within the friction cone. For contact phases corresponding to bilateral constraints (e.g. hands on handrail), we only enforce the friction cone requirement. We also ensure that the simulated motion is feasible by applying zero contact forces when the COP is out of reach for a limb (i.e. the distance between the COP and the origin of the limb, such as the hip or the shoulder, is longer than the length of the limb.)

3.2.2 Objectives and Optimization

The FRB model is used to predict the motion of the character over moderately long time periods (1-2s). The underlying assumption is that if the character is able to impart onto the environment the same set of contact forces used when simulating the simplified model, then it will naturally follow similar motion trajectories. The responsibility of the motion planner is therefore to find a set of contact phases and the parameters that define them, such that the motion of the simplified model has a desirable outcome.



When the motion planner is invoked, we begin by setting the initial state of the FRB according to the state of the character. To this end, we conceptually decompose the rigid bodies of the character into two groups: the *body frame* (blue highlights), consisting of the pelvis, middle-back, torso, and optionally the head; and the *limbs* (red highlights), as illustrated in the inset figure. The mass and moment of inertia of the FRB are computed, respectively, as the sum of the masses and composite moment of inertia of the individual body parts of the *body frame*. The state of the FRB (position, velocity, orientation, angular velocity) is computed by averaging the states of the same set of rigid bodies.

The first task of the motion planner is to determine which limbs to plan contact phases for. For walking motions, we use the IPM plan to initialize the contact phases corresponding to the foot steps. We further use a simple heuristic to add contact phases for the hands: the environment is scanned for handrails or platforms that are reachable by the character given its world location as predicted by the long term planner. For every potential handhold that is found, a contact phase is added to the motion plan. The force parameters associated with these contact phases are initialized to zero. With a rough schedule of contact phases, the next task of the medium-term motion planner is to optimize their timing, COP trajectory, and contact force trajectories such that desired motion goals are reached. More formally, we concatenate the individual parameters for each contact phase into a vector $\mathbf{u}$ (Sec 3.2.1), and minimize E, a weighted sum of simple objectives supplied by users, as is commonly done for control problems.

The following objectives are used in all the examples we demonstrate:

- **Target state**: Users can specify full or partial desired state information for specific moments in time. For instance, a user can provide target velocities, body height or orientation for the end of the motion plan. Alternatively, IPM states computed by the long-term planner can be used as waypoints. This objective measures the distance between the target state $\tilde{\mathbf{x}}$ and the state of the simplified model at the specified time instances: $E_{targetState}(t) = \|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)\|^2$.
- **Step width**: Users can specify a desired stepping width $\tilde{w}$. We measure the width by the coronal distance $cor()$ between the COM of the body and the location of a foot placement at the end of the planned contact phase (i.e. end of stance): $E_{stepWidth} = \|cor(\mathbf{c}(t_f) - \mathbf{p}(t_f)) - \tilde{w}\|^2$.
- **Parameter filtering**: We discourage the optimization method from exploring infeasible regions of the parameter space. This objective adds a cost proportional to how far away a parameter (e.g. a control point for the COP or contact force) is from its projection $proj()$ in the feasible region (Sec 3.2.1): $E_{parameterFiltering} = \|\mathbf{f} - proj(\mathbf{f})\|^2 + \|\mathbf{p} - proj(\mathbf{p})\|^2$.
- **Smoothness**: We penalize the net force and torque acting on the floating body as a way of promoting smooth motions: $E_{smoothness} = \|m\dot{\mathbf{v}}\|^2 + \|\dot{\mathbf{L}}\|^2$.

We minimize E using CMA [Han06], leveraging the fast computation times of running forward simulations with the FRB model and the trivial parallelization of roll-outs. The planning horizon is set to 2 steps for locomotion-based tasks and 1-2s for other motor behaviors. To initialize the first motion plan, we run two hundred iterations of the CMA solver to get an initial solution. Subsequently, as the full-body motion of the character start to simulate and track the plan, we continue to run CMA in parallel in order to further fine tune the motion trajectories for the remainder of the planning horizon. To promote temporal coherence between consecutive plans and to improve computation times, the character tracks only the first half of each motion plan. The planning process is then restarted: the state of the FRB is re-synchronized with that of the character, and the control parameters for the second half of the old plan are used to initialize the control parameters for the first half of the new motion plan. In this manner, the first half of each motion plan has already been optimized once and can therefore start being tracked by the character without requiring it to be optimized from scratch.

3.3 Short-term motion planning

As in real-life, physically-simulated characters move by modulating the ground reaction forces (GRF) at the points of contact with the environment. The short-term planner therefore increases the fidelity of the simplified model by considering the exact contact configuration (i.e. multiple contact points per limb) at every time step of the simulation, rather than the aggregated COP used in IPM and FRB. This step is essential because the contact forces computed by the medium-term planner do not account for the discrepancy between simulations of the simplified model and the full character model. The goal of the short-term planner is therefore to compute desired GRFs at every moment in time such as to best track the COM motion trajectory output by the medium-term planner.

Given the planned FRB state and the state of the character at current time t, we first compute the desired force and torque that can track the FRB state using a standard PD law:

$$\begin{aligned} \mathbf{f_D} &= k_{p,f}(\mathbf{x}_D - \mathbf{x}_c) + k_{d,f}(\dot{\mathbf{x}}_D - \dot{\mathbf{x}}_c) + m(\ddot{\mathbf{x}}_D - \mathbf{g}), \\ \tau_D &= k_{p,\tau}\phi(\alpha_D, \alpha_c) + k_{d,\tau}(\dot{\alpha}_D - \dot{\alpha}_c), \end{aligned} \tag{2}$$

where $\mathbf{x}_D$, $\dot{\mathbf{x}}_D$, $\ddot{\mathbf{x}}_D$, α_D and $\dot{\alpha}_D$ represent the desired linear and angular trajectories and their derivatives from the FRB model; $\mathbf{x}_c$, $\dot{\mathbf{x}}_c$, α_c and $\dot{\alpha}_c$ are the character's current COM position, velocity, orientation and angular velocity; the function $\phi(\theta_1, \theta_2)$ returns an angle-axis representation of the relative orientation between θ_1 and θ_2; $k_{p,f}$, $k_{d,f}$, $k_{p,\tau}$ and $k_{d,\tau}$ are tracking gains that are set to $4000N/m$, $1000Ns/m$, $500Nm$ and $100Nms$, respectively, across all our experiments.

Since the COM is not directly actuated, these desired forces can only be realized by applying joint torques to the limbs that are in contact with the environment. In previous work, $\mathbf{f}_D$ is divided (often equally) between the stance legs to control the COM [CBvdP10], and τ_D is directly applied to the stance hips to control body orientation [YLvdP07]. In our experiments (see accompanying video), and as noted by Jain and Liu [JL11], these strategies often lead to undesirable artifacts, such as foot slipping or ankle rolling. This problem can be ameliorated by increasing the size of the feet [GPvdS12], by filtering the ankle torques [CBvdP10], or by modeling the feet using soft, compliant tissues [JL11]. We instead choose to optimally distribute $\mathbf{f}_D$ and τ_D to the set of active contact points at every time step, considering both hands and feet as applicable. As illustrated in Figure 2(c), the relation between a set of N contact forces f and the resulting net force and torque on the COM is

$$\begin{pmatrix} \mathbf{f}_{net} \\ \tau_{net} \end{pmatrix} = \underbrace{\begin{pmatrix} I & \cdots & I \\ [\mathbf{r}_1]_\times & \cdots & [\mathbf{r}_N]_\times \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_N \end{pmatrix}}_{\mathbf{f}} = \mathbf{A}\mathbf{f}, \tag{3}$$

where $\mathbf{r}_i$ is the vector from contact point i to the COM, and $[\mathbf{r}_i]_\times$ denotes a skew symmetric matrix. The desired contact forces are ones that result in a body force $\mathbf{f}_{net}$ and torque τ_{net} as close as possible to $\mathbf{f}_D$ and τ_D, while maintaining static contact. This amounts to solving the following quadratic program:

$$\begin{aligned} \min_{\mathbf{f}} \; & \|\mathbf{A}\mathbf{f} - \mathbf{b}\|_S^2 + \|\mathbf{f}\|_W^2 \\ s.t. \quad & \mathbf{f}_i \in \mathcal{K}_i, \forall i, \end{aligned} \tag{4}$$

where S and W are diagonal matrices, and $\mathbf{b}$ is defined as $(\mathbf{f}_D, \tau_D)^T$. We use a small regularizer for the diagonal entries of W to discourage the use of excessively large forces, and the entries of S are set to

1 for the linear part of the objective, and 3 for the rotational objective. We also constrain the contact force $\mathbf{f}_i$ to be within a linearized friction cone $\mathcal{K}_i$ [AdSP07]. This constrained optimization problem is solved using *OOQP* [GW01]. The optimal solution $\mathbf{f}_i^*$ corresponds to the GRF the character should exert at contact point i in order to best follow the motion plan.

4 Full-body Motion Control

The output of the motion planner consists of an optimal set of GRFs on the current contact points, as well as target locations for upcoming foot and hand placements. The goal of the full-body controller is to compute joint torques that achieve these planned targets.

The GRFs are treated as virtual forces by the full-body controller. The Jacobian transpose method is used to compute torques along the joints of the limbs that are in contact with the environment as $\tau_v = -\sum_i \mathbf{J}_i^T \mathbf{f}_i^*$, where i iterates through the contact points associated with the limb. Virtual forces are also applied to the end effectors of the swing limbs in order to guide them towards their target locations. We compute a smooth trajectory from the swing limbs' current position to the planned contact location as in Coros et al. [CBvdP10], then use virtual force control to track this trajectory. Finally, gravity compensation for each body part is also realized through virtual forces.

The overall posture of the character is controlled using PD controllers. To handle the kinematically redundant characteristic of human limbs (i.e. the same point in space can be reached using different configurations), we use inverse kinematics to compute desired poses for the swing limbs, including the hands, fingers, and ankles, in order to match the desired positions along the swing trajectories and the configuration of the planned contact regions. The pose for the spine, neck and head can be directly controlled by the user to affect the style of the motion.

We use an implicit form of PD control to generate torques that track this desired pose. We prefer an implicit formulation over a typical explicit PD controller because the latter is unstable when relatively large time steps are used, or when the rigid bodies involved have small masses and moments of inertia. Our approach is related to the implicit PD control method described by Tan et al. [TLT11]. However, to avoid solving a large coupled system, we compute the joint torques individually. In addition, we noticed that numerical instabilities are caused almost entirely by the damping terms. We therefore implement PD controllers that are explicit in positions, but implicit in velocities, as we found this presents a favorable trade-off between tracking accuracy and numerical stability. More specifically, we express the PD torque τ at time t for each joint as:

$$\tau_t = k_p \phi(\theta_t, \theta^d) + k_d(\omega_{t+1} - \omega^d) \tag{5}$$

where, as before, ϕ returns an angle-axis representation of the relative difference between two orientations, and θ, θ^d, ω and ω^d represent the current and desired orientations and angular velocities for the joint. The subscript denotes the time index for each quantity. The relative angular velocity of the joint is given by the difference between the angular velocity of the child and parent rigid bodies that it connects: $\omega = \omega_c - \omega_p$. Applying a torque τ at a joint means that equal and opposite torques are applied to the two connected rigid bodies. The relative angular velocity of the joint at time step $t+1$ is therefore given by:

$$\begin{aligned} \omega_{t+1} &= \omega_{c,t} + \Delta t I_c^{-1} \tau_t - (\omega_{p,t} - \Delta t I_c^{-1} \tau_t) \\ &= \omega_t - \Delta t (I_c^{-1} + I_p^{-1}) \tau_t \end{aligned} \tag{6}$$

where I_c and I_p are the world coordinates inertia tensors for the child and parent links, evaluated about the position of the joint. Substituting (6) into (5) gives the means to compute the PD torques:

$$\tau_t = (\iota + \frac{\Delta t}{k_d}(I_c^{-1} + I_p^{-1}))^{-1}(k_p \phi(\theta_t, \theta^d) + k_d(\omega_t - \omega^d)) \tag{7}$$

All quantities needed to compute τ are readily available, and ι is a 3x3 identity matrix. As the time step decreases, this method becomes equivalent to the explicit formulation of PD control. For larger time steps, however, we found it to be much more stable: we can stably execute the walking controllers at a frequency of $200Hz$ using the implicit PD controllers, but using explicit PD controllers with the same gains requires a control frequency of at least $2000Hz$. We note that the implicit PD controllers we describe are related to the constraint formulation employed by some classes of physics engine, such as ODE [Smi00]. However, they are described in terms of the familiar stiffness and damping gains, rather than the Constraint Force Mixing (CFM) and Error Reduction Parameter (ERP) values, and they can be implemented independently of the physics engine used for simulation.

5 Experiments and Results

5.1 Locomotion-based tasks

We applied our hierarchical control framework to three locomotion-based tasks of increasing difficulty. For each of these tasks, the horizon used by the long-term planner corresponded to a sequence of 3 strides (i.e. 6 steps), the medium-term planner considered a horizon of one stride, and the horizon of the short-term planner was set to the simulation time step ($1/500s$). The long-term and medium-term planners are re-invoked after each step is taken. The COP location and contact forces computed by the long-term planner are used to initialize the second half of the medium-term plan, while the first half is initialized as described in Sec 3.2.2. Users can control high level features of the motion by specifying desired properties such as the walking speed or body height. This is accomplished through the $E_{targetState}$ objective which sets target states for the moments in time corresponding to the end of each step.

Walking The first experiment we performed is aimed at validating our framework on typical walking tasks. As demonstrated in the accompanying video, the motion plans generated by our planners allow the character to quickly transition between walking speeds ranging from $-1m/s$ to $1.5m/s$. Two control points were used to define the COP and force trajectories for each planned contact phase. We further used this task to validate the force parameterization we employ when computing the motion of the FRB. Without separating the forces and their moments by specifying them in a moving coordinate frame, we needed about 4 times as many control points to get motion plans of comparable quality. This leads to larger parameter spaces that need to be explored, and increases the overhead of the CMA optimization routine. In addition, we have noticed that convergence rates decrease quite significantly, and undesirable local minima that lead to poor-quality motions (i.e. do not satisfy well the user-provided objectives) are encountered relatively often. This finding suggests that our force parameterization results in a parameter space that is more easily and efficiently explored by the optimizer.

Stepping stones To increase the level of difficulty for the walking experiments, we set up an environment where gaps in the terrain restrict the set of available stepping targets. The high-level goal provided by the user consists of the desired walking speed at the end

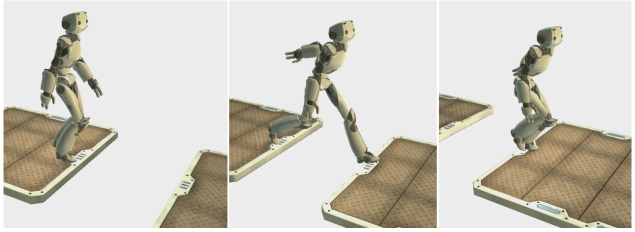


Figure 4: *Stepping over a $1.1m$ gap.*

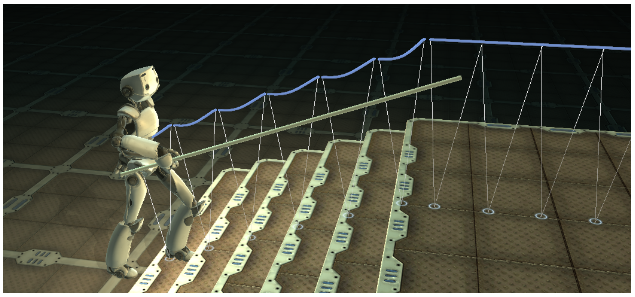


Figure 5: *Walking up $15cm$ stairs with the aid of a handrail. The long-term plan (visualized) informs the medium-term planner which stairs should be stepped on.*

of each step taken by the character. The benefit of the hierarchical planning model is evident here. The medium-term planner does not have to consider the combinatorial set of options that are available in terms of stepping regions. Rather, the much more efficient long-term planner is used to make discrete decisions regarding which stepping region to aim for at each step (i.e. before the incoming gap or after). These stepping regions are then treated as hard constraints by the medium-term planner.

We note that the timing of the contact phases output by the long-term planner are further optimized by the FRB model, so double stance or flight phases can emerge naturally as needed. The additional flexibility of the FRB (i.e. modeling body rotations) also results in motions that are more natural than if the IPM trajectory was tracked directly. As shown in Fig. 4, the simulated character can hop over gaps up to $1.1m$ in length, showing that both our planner and the full-body controller can handle motions with flight phases. This experiment also illustrates interesting dynamic behaviors emerging from the interaction between the long term and medium term motion planners: while the long term planner decides to take small steps before a gap, the medium term planner chooses to lower the COM and generate enough momentum to hop over.

Stair climbing For the next experiment, we used the long-term planner to determine which stair to aim for with each step as the character's goal is to move forward at a prescribed walking speed. In addition, we also tested the ability of the character to use handrails to aid with this motor control task. We add hand contact phases Ω to the FRB, as described in Sec 3.2.2, whenever the handrail comes within reach based on the planned motion trajectory from the long-term planner. The timing parameters of each new contact phase are initialized by assuming the hand will move in unison with the opposite leg (i.e. they will make contact with the environment at the same time). The COP parameters are set to the projection of the predicted COM trajectory on the handrail at the time when the contact phase starts, and all contact force parameters are set to 0. Initially, these additional contact phases do not affect the motion plan because they use zero forces, and the character is capable of climbing moderately tall stairs (in the order of $25cm$) without any help from hand contacts. However, once the contact phases for the hands are optimized, they are put to good use—the average force applied through the hands in the stair climbing sequence shown in the video has a magnitude of about $280N$. Using both arms for support, the maximum stair height that the character can climb on is $50cm$, as opposed to $25cm$ when the hands are not used. Figure 5 shows the character using the hand rail while starting to climb stairs. The output of the long-term planner is also visualized.

5.2 Additional motor tasks

In addition to the handrail-assisted stair climbing controller described above, we designed two more motor tasks that require coordinated actions of both the hands and feet. Since these tasks are not walking, we cannot employ the IPM-based long-term planner. Instead, we decomposed each task into several stages using a typical finite state machine (FSM). For each stage of the motion, and as a function of the environment, the FSM outputs an adequate sequence of contact phases, and optionally the final desired states, to guide the style of the motion.

Standing up Armrests and handrails are often used casually for a variety of motor tasks. To investigate the ability of our framework to generate controllers for motor skills that utilize hand-holds, we set up an experiment where the task of the character was to stand up from a chair. This task was broken down into two phases: reaching for hand-holds and getting up. For the first part of the motion, the environment is automatically scanned for reachable hand supports. These can be either planar regions that the character's hands can push on, or bars that can be grabbed onto. Once possible handholds are found, the FSM adds corresponding contact phases to the FRB motion plan. We note that all force parameters are initially set to 0. Consequently, the medium-term planner is aware that they are available for use, but they do not initially contribute to the dynamics of the motion plan. For the second phase of the motion, the FSM specifies that the character should no longer be making use of its hands (i.e. contact phases are only added to the medium-term planner for the feet). The FSM also provides a target state for the end of the motion to ensure that the character ends up in an upright position. As the initial set of parameters for the contact phases are far away from an optimal solution, CMA requires 1000-2000 iterations to converge. Once a successful medium-term motion plan is found, it is executed using the short-term planner and the full-body controller.

Although the structure of this task is pre-specified, our framework is able to create motions that generalize to a large variety of related scenarios. We demonstrate the robustness of this task by placing the hand holds at arbitrary positions in the world and re-running the task. Various combinations of reachable handrails and armrest can all lead to successful motions, as illustrated in Fig 6, without requiring any changes to the input or the structure of the task.

Crawling on platform For this final task, we tested the ability of the character to climb onto obstacles using its hands for support, as illustrated in Fig. 7. This task is broken down into three stages: placing hands on platform, stepping onto platform, and standing upright. For each of these stages, the FSM specifies which limbs should be in contact with the environment. Two target states specify that the character should lean over during the second motion stage, and assume an upright position at the end of the motion. We note that while the contact phases corresponding to the limbs are scripted through the FSM, their exact timing, placement, and acting ground reaction forces are automatically optimized by the FRB

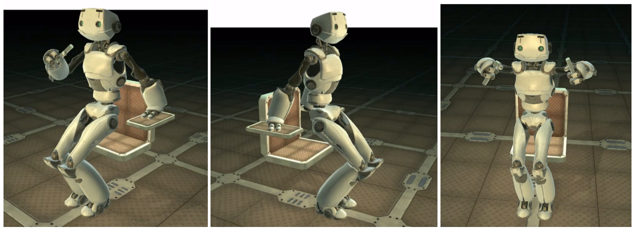


Figure 6: *Getting up from a chair using a mixture of hand bars and arm-rests.*

motion planner, which is invoked before the start of each motion stage. In addition to the cooperative actions of the hands and feet, this experiment also shows that the character is able to get up from a crawling position to a bipedal stance, demonstrating the ability of our framework to concurrently plan and control both the linear and angular motion of the character.

5.3 Implementation details

We use the Open Dynamics Engine [Smi00] as a black-box physics simulator to generate the motions of the character. The time step for both the physics simulator and the controller is fixed to $500Hz$. To simulate the motion fo the FRB, we use a $1/30s$ time step. The computational bottleneck of our framework is the medium-term motion planner. Currently, for simpler motor tasks such as walking, our framework runs in real-time. For more complex motions involving a large number of contact phases, especially when we cannot use the long-term planner to initialize parameters of the medium-term plan, our framework runs at interactive rates (5-10 FPS, or 15%-30% of real time).

6 Conclusion and Limitations

We presented a planning and control framework that enables physically simulated characters to perform a variety of motor tasks. In particular, we are beginning to investigate a relatively unexplored class of motor skills that require precise cooperation between the hands and the feet. The key to our method is the use of a hierarchical set of simplified models that represent the character's dynamics at different levels of abstraction; this allows motion plans for complex motor tasks to be efficiently generated. We demonstrate the flexibility of our method by generating controllers for tasks ranging from walking, to climbing stairs with the aid of handrails, then to standing up from chairs. The motions of the character are generated using a black-box physics simulator, with high level user guidance and without requiring any motion capture data.

Our work takes a step further towards the goal of creating autonomous, skilled virtual humans. However, many obstacles remain yet to be overcome. In addition to improving computation times, there are many opportunities to improve motion quality. For instance, for agile motions, the predictive power of the FRB model begins to deteriorate, as the effect of fast-moving limbs is not captured. Consequently, the character is not able to closely track motion plans for motor skills of increased agility. We also found that better models need to be developed for foot and hand placements. The knees of the simulated character, for instance, occasionally penetrate the body or the environment, and the way in which handholds are reached is not always natural-looking. We would also like to extend our model to types of motions that exhibit less structured contact phases—climbing onto tables using knees, or pushing against the back of a chair using the shoulders. The motion structure can be loosened by relaxing two restrictions in our algorithm: state-machine-like contact phases, and planning of only hands and feet as valid contacting end effectors. Lastly, we would like to explore more intuitive ways for a user to control not only what the character should do, but also how it should perform a task.

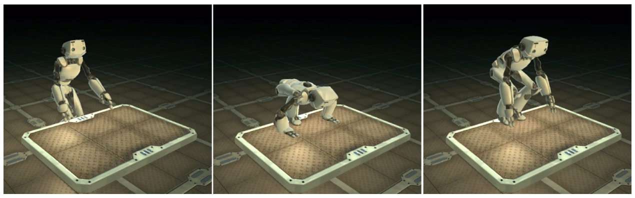


Figure 7: *Crawling up on a* $0.5m$ *platform*

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and suggestions. Many thanks also to Maurizio Nitti for creating the *DeeZee* character model.

References

ABE Y., DA SILVA M., POPOVIĆ J.: Multiobjective control with frictional contacts. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 249–258.

COROS S., BEAUDOIN P., VAN DE PANNE M.: Generalized biped walking control. *ACM Transctions on Graphics 29*, 4 (2010), Article 130.

DARPA: DARPA Robotics Challenge. Tactical Technology Office, April 2012. Solicitation Number DARPA-BAA-12-39.

GEIJTENBEEK T., PRONOST N., VAN DER STAPPEN A. F.: Simple data-driven control for simulated bipeds. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), pp. 211–219.

GEIJTENBEEK T., VAN DE PANNE M., VAN DER STAPPEN A. F.: Flexible muscle-based locomotion for bipedal creatures. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 206:1–206:11.

GERTZ E. M., WRIGHT S. J.: Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software 29* (2001), 58–81.

HANSEN N.: The cma evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, vol. 192 of *Studies in Fuzziness and Soft Computing*. 2006, pp. 75–102.

HAN D., NOH J., JIN X., SHIN J.: On-line real-time physics-based predictive motion control with balance recovery. *Computer Graphics Forum (Eurographics 2010) 33* (2014).

HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *ACM Trans. on Graphics (SIGGRAPH)* (Aug. 1995), pp. 71–78.

JAIN S., LIU C. K.: Controlling physics-based characters using soft contacts. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 163:1–163:10.

JAIN S., YE Y., LIU C. K.: Optimization-based interactive motion synthesis. *ACM Trans. on Graphics 28*, 1 (2009).

KAJITA S., KANEHIRO F., KANEKO K., YOKOI K., HIRUKAWA H.: The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on* (2001), vol. 1, pp. 239–246 vol.1.

LIU C. K., HERTZMANN A., POPOVIĆ Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. on Graphics (SIGGRAPH) 24*, 3 (July 2005), 1071–1081.

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

LASZLO J., VAN DE PANNE M., FIUME E.: Limit cycle control and its application to the animation of balancing and walking. In *ACM Trans. on Graphics (SIGGRAPH)* (1996).

LIU L., YIN K., VAN DE PANNE M., SHAO T., XU W.: Sampling-based contact-rich motion control. *ACM Trans. Graph. 29*, 4 (July 2010), 128:1–128:10.

MORDATCH I., DE LASA M., HERTZMANN A.: Robust physics-based locomotion using low-dimensional planning. *ACM Trans. Graph. 29*, 4 (July 2010), 71:1–71:8.

MORDATCH I., TODOROV E., POPOVIĆ Z.: Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph. 31*, 4 (July 2012), 43:1–43:8.

MORDATCH I., WANG J. M., TODOROV E., KOLTUN V.: Animating human lower limbs using contact-invariant optimization. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 203:1–203:8.

POLLARD N. S., BEHMARAM-MOSAVAT F.: Force-based motion editing for locomotion tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation* (April 2000).

RAIBERT M. H., HODGINS J. K.: Animation of dynamic legged locomotion. In *ACM Trans. on Graphics (SIGGRAPH)* (July 1991), vol. 25, pp. 349–358.

STEWART A. J., CREMER J. F.: Beyond keyframing: An algorithmic approach to animation. In *Proceedings of Graphics Interface* (1992).

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

SIMS K.: Evolving virtual creatures. In *ACM Trans. on Graphics (SIGGRAPH)* (July 1994).

SHAPIRO A., LEE S.: Practical character physics for animators. *IEEE Comput. Graph. Appl.* (August 2011).

SMITH R.: Open dynamics engine, 2000. URL: `http://www.ode.org`.

TWIGG C. D., JAMES D. L.: Many-worlds browsing for control of multibody dynamics. *ACM Trans. Graph. 26*, 3 (July 2007).

TAN J., LIU K., TURK G.: Stable proportional-derivative controllers. *IEEE Comput. Graph. Appl. 31*, 4 (July 2011), 34–44.

TODOROV E.: A convex, smooth and invertible contact model for trajectory optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (2011), pp. 1071–1076.

VAN DE PANNE M.: From footprints to animation. *Computer Graphics Forum 16*, 4 (October 1997).

WANG J. M., FLEET D. J., HERTZMANN A.: Optimizing walking controllers for uncertain inputs and environments. *ACM Trans. Graph. 29*, 4 (July 2010), 73:1–73:8.

WANG J. M., HAMNER S. R., DELP S. L., KOLTUN V.: Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans. Graph. 31*, 4 (July 2012), 25:1–25:11.

WITKIN A., KASS M.: Spacetime constraints. *SIGGRAPH Comput. Graph. 22*, 4 (June 1988), 159–168.

WOOTEN W. L.: *Simulation of Leaping, Tumbling, Landing, and Balancing Humans*. PhD thesis, Georgia Institute of Technology, 1998.

WAMPLER K., POPOVIĆ Z.: Optimal gait and form for animal locomotion. *ACM Trans. on Graphics 28*, 3 (2009), 1–8.

WAMPLER K., POPOVIC J., POPOVIC Z.: Animal locomotion controllers from scratch. *Comput. Graph. Forum 32*, 2 (2013), 153–162.

YE Y., LIU C. K.: Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph. (SIGGRAPH 2010)* (2010), 1–9.

YE Y., LIU C. K.: Synthesis of detailed hand manipulations using contact sampling. *ACM Trans. Graph. (SIGGRAPH 2012) 31*, 4 (2012), 1–10.

YIN K., LOKEN K., VAN DE PANNE M.: Simbicon: simple biped locomotion control. *ACM Trans. on Graphics (SIGGRAPH) 26*, 3 (2007), 105.

ZIMMERMANN D.: Complex locomotion tasks for physically-simulated virtual humans. MSc Thesis, December 2013. URL: `http://tinyurl.com/nsgjl7s`.