Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Task-driven Posture Optimization for Virtual Characters

Mingxing Liu, Alain Micaelli, Paul Evrard, and Adrien Escande

CEA, LIST, Interactive Simulation Laboratory, France

Abstract

This paper presents a generic approach to find optimal postures, including contact positions, for manipulation tasks. It can be used in either the preparation for a task, or the evaluation of the feasibility of a task during planning stages. With such an approach, an animator can control a virtual character from a high level by just specifying a task, such as moving an object along a desired path to a desired position; the animator does not need to manually find suitable postures for the task. For each task, an optimization problem is solved, which considers not only geometric and kinematic constraints, but also force and moment constraints. The optimized postures allow the virtual character to apply manipulation forces as strongly as possible, and meanwhile to avoid foot slipping. Moreover, potential perturbation forces can be taken into account in the optimization to make postures more robust. The realism of our approach is demonstrated with different types of manipulation tasks.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

A prime functionality of a virtual character is to perform manipulation tasks. The choice of postures can impact the possibility of fulfilling a task successfully. Here we call *posture* a set of body configurations including contact positions. Optimal postures usually vary from task to task. For example, the virtual character may have to lean forward to push an object, but lean backward to pull it; the feet may have to be separated from each other to be able to generate manipulation forces that are sufficiently strong. Even for the same kind of tasks, for example pushing, the optimal posture should be adjusted to adapt to different object's physical properties. Moreover, available contact positions are sometimes restricted due to environment constraints. For example, the virtual character may need to choose foot positions that allow the end-effector to manipulate an object without moving the feet. These questions suggest that before performing a task, it is important and beneficial for the virtual character to choose postures that are optimal for the task.

In the context of computer animation where motion capture has become an essential technique, an operator's postures can be taken as references for the virtual character. Captured motions are lifelike, but they need to be adjusted to handle manipulation forces and to deal with disturbances. Moreover, the operator may not be skillful enough for providing suitable reference postures for some tasks. Consequently, his postures can be inappropriate for the virtual character to balance the interaction forces or to improve task performances. Many existing methods focus on the generation of foot contact positions for locomotion tasks. The choice of contact positions with the purpose of improving manipulation task performances by taking into account contact forces remains a challenge.

This paper introduces a generic approach that can automatically find optimal postures for a wide variety of manipulation tasks. For each manipulation task, a constrained optimization problem is solved off-line to find a sequence of optimal postures associated with a desired manipulation path, in the neighborhood of a given initial posture. The optimization problem is formulated based on a simplified model of the character. This simplified model takes into account interaction forces with the environment and the kinematic relations between control frames. Here *control frames* are some coordinate frames attached to the character's body, the positions of which are to be optimized. Once a solution is found, one can use a motion controller to make the character adjust the contact positions, then to perform object manipulation by following the desired motions of the control frames and by applying desired forces. Our approach considers quasi-static cases where dynamic effects can be ignored; therefore, we suggest applying a quasi-static task controller for object manipulation, such as the one described in [LME*11].

The main contributions of our approach are as follows: (*i*) It is a generic posture optimization that couples **geometric and kinematic constraints (G-K constraints)** with **force and moment constraints (F-M constraints)**. (*ii*) It can improve task performance by choosing suitable postures in a preparation stage before actually performing the task. Contact configurations for manipulation tasks are optimized for user-specified manipulation paths and forces. (*iii*) It deals with the redundancy of poses, and can make contact positions as robust as possible. The structure of our optimization problem allows us to take precaution against mechanical interactions and possible perturbations. By adopting the optimized postures, the risk of failures either due to poor postures or due to perturbations can be greatly decreased.

2. Related Work

Task-based constraints should take into account kinematic constraints related to the character's body structure and geometric relations between the character and the environment. For example, the character may need to find postures that allow its hand to move an object along a desired trajectory without violating the constraint of its body structure. The constraints related to a body structure can be formulated based on forward kinematics, which provides a mapping between a body frame and the joint angles. One example is the virtual kinematic loop equation [Smi10], which can be used to force the frame of a link to coincide with another frame in the environment. Researchers in robotics have adopted such a constraint in the path tracking of robot manipulators and wheeled robots [dSRA*05, DSDLR*07, Sti10]. Our approach applies this constraint on a virtual character to find kinematically feasible motions for manipulation tasks. To perform specified interactions with the environment, some kinematic-based motion editing approaches modify input human motions [PMM*09, JL09]. In [LP02], environmental restrictions are represented as positional and sliding constraints, and linear and angular momentum constraints are used to improve the realism of motions. Constrained inverse kinematics has been combined with a database of example postures to synthesize motions that satisfy a set of G-K constraints for manipulation tasks [YKH04]. However, in a physics-based simulation environment, these approaches are limited when the character needs to react to interaction forces during manipulation tasks; and moreover, approaches based on a motion database usually cannot generate certain behaviors to handle interactions if such behaviors are not included in the motion database.

To adapt output motions to interaction forces, F-M constraints should be taken into account. For example, the equilibrium of forces and moments should be considered to ensure balance; and contact forces should be handled for manipulation tasks, because the character's body is underactuated and it needs to use contact forces to perform desired motions. In some optimization based motion synthesis approaches, physical constraints based on forces and torques are included in the optimization to ensure physical realism [FP03, JYL09]. F-M constraints have been considered in many task control frameworks [KSPW04, AdSP07, CMAL07, LME*11], where the desired foot contact positions are either given a priori, or computed without considering interaction forces during object manipulation. Our posture optimization can be considered as a preparation step that can be executed before applying these task controllers. In fact, our approach provides these task controllers with a precomputed solution of suitable postures for given manipulation tasks. These postures, including contact positions such as foot positions, are optimized with respect to user-specified manipulation paths and forces. We achieve this by taking into account F-M constraints to handle physical interactions, and G-K constraints to generate kinematically feasible motions with environment awareness.

The configuration of contacts is a major aspect that is focused on in this paper. A support polygon reshaping approach has been proposed in [YKEL06], the idea of which is to first try to reach the target with an initial support polygon, and then reshape the support polygon according to the feedback task error. An approach to plan foot placements according to kinematic tasks has been described in [KLY11]. Compared with these approaches, ours is more general in that besides G-K constraints, we also take into account F-M constraints in the optimization of contact positions. Moreover, under many circumstances, the feet of the character are fixed when its hands are manipulating objects along a segment of manipulation path [SSF09, HNTH11]. Our approach can provide a support polygon that is suitable for a segment of end-effector motion path instead of for only one fixed end-effector task target; therefore the character will not need to adjust its foot positions frequently during task execution.

Besides, we also optimize contact positions to generate robust postures with respect to perturbation forces. An optimal control which allows the adaptation of walking motions to physical perturbations has been proposed in [YL10]. Contact forces are first generated off-line to reproduce reference motions, and then adjusted on-line to maintain contacts and balance during perturbations. But such contact forces that satisfy current contact configuration may not exist. As our approach considers manipulation tasks with the desired interaction forces known a priori, contact forces and contact configurations are optimized simultaneously before task execution. Possible perturbation forces can also be taken into account in our optimization to make the solution more robust. Moreover, kinematic relations between contacts and control frames are not considered in [YL10], which may generate kinematically unfeasible motions. Such kind of relations is taken into account in our approach.

3. Overview

The postures of the virtual character are adjusted to be adaptable to a manipulation task, which is defined by the manipulation path, the manipulation force direction, as well as some G-K and F-M constraints. According to these task requirements, posture optimization is used to find suitable postures for the task. Before the task execution, the contact configuration and the center of mass (CoM) position of the character will be adjusted according to the optimized posture. The character finally start to perform the manipulation task by using a manipulation task controller, which takes as inputs the optimized positions of control frames. The whole framework is illustrated in Figure 1.

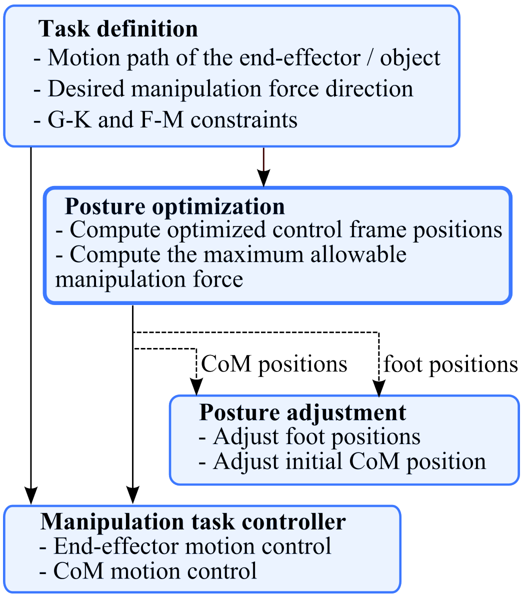


Figure 1: *Overview of the posture adaptation framework.*

4. Posture optimization

The goal of this section is to formulate an optimization problem, which will be solved to find optimal postures in the neighborhood of a given one. Such postures should allow the hand to follow a manipulation path defined by the task, and to apply sufficiently strong interaction forces on the object. We seek to improve the robustness of the posture against external perturbation forces, and in addition, to make the postures as comfortable as possible. Besides, the optimization result also provides possible contact forces on the feet and the maximum allowable interaction force between the end-effector and the object.

4.1. Simplified model

Our posture optimization problem uses a simplified model, which considers some essential elements of the character, such as contact forces, the positions of the CoM and some other control frames, as well as the kinematic relations between these control frames. This simplified model consists of a punctual mass (m) at the CoM, one massless back, one massless arm and two massless legs (see Fig.2). The model has 15 degrees of freedom (DoF), including 6 root DoF (3 translational DoF and 3 rotational DoF), 2 DoF for each hip and shoulder and 1 DoF for each knee and elbow. It has two contact points, one on each foot; and it has one end-effector, which is the hand. The contacts on each foot and hand are abstracted into one frictional point contact as in [BL08], which means only the net contact force between each body segment and the environment is considered. These simplifications help to reduce the dimension of the optimization problem while retaining the important characteristics of the interaction model.

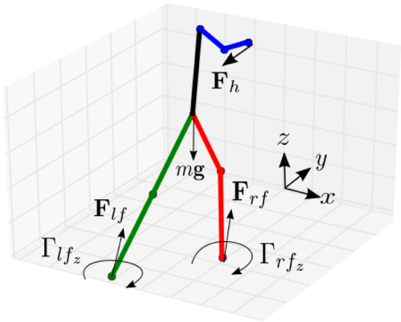


Figure 2: *Simplified model.*

The following notation is used in this paper.

- Positions are denoted as $\mathbf{P}$. All the position vectors are defined with respect to a global reference frame with axis x, y, and z. The z axis points upwards.
- Joint angles are denoted as q, and the α-th joint angle is denoted as $q\alpha$.
- Forces in Cartesian space are denoted as $\mathbf{F}$. The gravity force applied at the CoM is denoted as $\mathbf{F}_G$.
- Moments in Cartesian space are denoted as Γ.
- Each frame j on the body of the virtual character is generally denoted by subscript j. More specifically, frames are denoted by subscripts: c for the CoM, lf and rf for the left and right foot respectively, h for the hand, and g for the ground.
- The upper and lower limits of a variable v are denoted by v^U and v^L respectively.

The manipulation path is discretized into sampled points. A term associated with a discretized point i is denoted by the superscript i. The desired position of the object at each discretized point i is denoted as $\mathbf{P}^i_{\text{obj}}$. The hand force at the position $\mathbf{P}^i_{\text{obj}}$ is defined as $\mathbf{F}^i_h = k^i \hat{\mathbf{F}}^i_h$, where $\hat{\mathbf{F}}^i_h$ is a unit vector indicating the desired force direction at point i, and k^i is its magnitude.

4.2. Optimization with respect to one discretized point on the manipulation path

For clarity, we first describe a posture optimization problem with respect to one pair of object position and desired manipulation force direction ($\mathbf{P}^i_{\text{obj}}$, $\hat{\mathbf{F}}^i_h$). The optimization takes into account the objectives of increasing the maximum allowable manipulation forces, minimizing the risk of foot slipping, and reducing joint discomfort, subject to all the G-K and F-M constraints.

The set of the optimization variables is defined as

$$\Theta^i = \left\{\mathbf{P}^i_c, \mathbf{q}^i, \mathbf{F}^i_{\text{lf}}, \mathbf{F}^i_{\text{rf}}, \Gamma^i_{\text{lf}z}, \Gamma^i_{\text{rf}z}, k^i\right\} \tag{1}$$

The optimization problem is written as follows:

$$\begin{aligned} &\min_{\Theta^i} . w_h G^i_h(\Theta^i) + w_f G^i_f(\Theta^i) + w_q G^i_q(\Theta^i) \\ &\text{s.t. } \Psi^i_{\text{GK}}(\mathbf{P}^i_c, \mathbf{q}^i) \\ &\quad\;\; \Psi^i_{\text{FM}}(\Theta^i) \end{aligned} \tag{2}$$

where G_h, G_f and G_q are the objectives, and Ψ_{GK} and Ψ_{FM} are G-K and F-M constraints respectively. The optimization weights w are chosen based on different task requirements. We will discuss how to make the choice of these weights in section 6.

4.2.1. Objectives

To improve the manipulation ability, the hand force magnitude along the given direction is maximized by setting the following hand force objective.

$$G^i_h(k^i) = -k^i \tag{3}$$

To avoid foot slipping, the foot contact forces should remain inside their friction cones. This non-sliding constraint will be referred to later in this paper. As this constraint is not sufficient to fully determine the tangential foot contact forces, these forces are minimized by the following objective function, which helps to reduce the risk of foot slipping.

$$G^i_f(\mathbf{F}^i_{\text{lf}}, \mathbf{F}^i_{\text{rf}}) = \frac{1}{2}\left\|\mathbf{S}_{\text{lf}}\mathbf{F}^i_{\text{lf}}\right\|^2 + \frac{1}{2}\left\|\mathbf{S}_{\text{rf}}\mathbf{F}^i_{\text{rf}}\right\|^2 \tag{4}$$

where $\mathbf{S}_{\text{lf}}$ and $\mathbf{S}_{\text{rf}}$ denote matrices to select the directions of the tangential friction forces.

A real human always intend to reduce joint discomfort during manipulation tasks. An objective function of joint discomfort G_q is used so as to imitate such human behaviors. The discomfort measure [YMK*04, MZC*09] is applied here. The objective function is defined as follows.

$$\begin{aligned} G^i_q(q^i) &= \sum_{\alpha=1}^{DoF}\left[\gamma_\alpha(\Delta q^n_\alpha)^2 + QU^i_\alpha + QL^i_\alpha\right] \\ \Delta q^n_\alpha &= \frac{q^i_\alpha - q^N_\alpha}{q^U_\alpha - q^L_\alpha} \\ QU^i_\alpha &= (0.5cos\frac{\frac{3\pi}{2}(q^U_\alpha - q^i_\alpha)}{q^U_\alpha - q^L_\alpha} + 0.5)^{100} \\ QL^i_\alpha &= (0.5cos\frac{\frac{3\pi}{2}(q^i_\alpha - q^L_\alpha)}{q^U_\alpha - q^L_\alpha} + 0.5)^{100} \end{aligned} \tag{5}$$

This objective guides the optimization to choose joint angles based on their neutral values and limits. It attempts to push joint angles q^i away from their upper limits q^U and lower limits q^L, and pull them towards a neutral value q^N, so as to increase posture comfort level. As mentioned in [YMK*04], the concept behind the discomfort measure is to enhance the preference of using certain joints to fulfill a motion task, by regulating the joint weight γ_α.

We will show in section 6 that a careful choice of the value of γ_α helps to improve the behaviors of the virtual character, making them closer to those of a real human.

4.2.2. Geometric and kinematic constraints

Our G-K constraints take into account the geometric relations between the character and the virtual environment, as well as the kinematic relations between the control frames. An example of the constraints Ψ^i_{GK} is listed below.

Joint angles should respect joint limit constraint:

$$\mathbf{q}^L \leq \mathbf{q}^i \leq \mathbf{q}^U \tag{6}$$

We search for position solutions within a constrained region of interest, which is a polygon around the object.

$$\begin{aligned} &\mathbf{A}_c\mathbf{P}^i_c + \mathbf{b}_c \leq \mathbf{d}_c \\ &\mathbf{A}_{\text{lf}}\mathbf{p}_{\text{lf}x,y}(\mathbf{P}^i_c, \mathbf{q}^i) + \mathbf{b}_{\text{lf}} \leq \mathbf{d}_{\text{lf}} \\ &\mathbf{A}_{\text{rf}}\mathbf{p}_{\text{rf}x,y}(\mathbf{P}^i_c, \mathbf{q}^i) + \mathbf{b}_{\text{rf}} \leq \mathbf{d}_{\text{rf}} \end{aligned} \tag{7}$$

The hand position is constrained to point $\mathbf{P}^i_{\text{obj}}$, which implies that the hand moves along the desired manipulation path, as the object does.

$$\mathbf{P}_h(\mathbf{P}^i_c, \mathbf{q}^i) - \mathbf{P}^i_{\text{obj}} = \mathbf{0} \tag{8}$$

The following constraint is imposed to prevent the feet from overlapping each other. The distance between the feet is kept lager than a minimum value.

$$\psi(\mathbf{p}_{\text{lf}x,y}(\mathbf{P}^i_c, \mathbf{q}^i), \mathbf{p}_{\text{rf}x,y}(\mathbf{P}^i_c, \mathbf{q}^i)) \geq d_f \tag{9}$$

where $\psi(\mathbf{P}_1, \mathbf{P}_2)$ denotes the distance between $\mathbf{P}_1$ and $\mathbf{P}_2$. The angle between the facing direction of the character and the direction of the object is constrained in (10), where ϑ denotes the angle between the two vectors.

$$\vartheta(\mathbf{P}^i_c, \mathbf{q}^i, \mathbf{P}^i_{\text{obj}}) \leq d_\vartheta \tag{10}$$

The positions of some control frames j may have to respect some additional geometric constraints (11) in constrained environments. For example, the optimal positions of the feet should not penetrate into an object.

$$\psi(\mathbf{P}_{\text{obj}}, \mathbf{P}_j(\mathbf{P}_c^i, \mathbf{q}^i)) \geq 0 \tag{11}$$

The contacts between the feet and the ground should be maintained (12):

$$p_{\text{lf,rf}_z}(\mathbf{P}_c^i, \mathbf{q}^i) - p_{g_z} = 0 \tag{12}$$

Since the position of each control frame can be obtained by forward kinematics, they are expressed as a function of the CoM position $\mathbf{P}_c^i$ and joint angles $\mathbf{q}^i$. As a result, constraints due to the skeleton structure are implicitly included in these G-K constraints.

4.2.3. Force and moment constraints

This paper is interested in manipulation tasks where the major perturbation comes from mechanical interactions. The character should be able to keep its balance under external contact forces. To achieve this goal, the following F-M constraints Ψ_{FM}^i are imposed. Only the quasi-static cases are considered here, and the dynamic effects such as acceleration are neglected.

To maintain the static equilibrium, the constraint of force and moment balance (13) is imposed.

$$\begin{aligned} &\mathbf{F}_{\text{lf}}^i + \mathbf{F}_{\text{rf}}^i + \mathbf{F}_h^i(k^i) + \mathbf{F}_G = \mathbf{0} \\ &\mathbf{P}_{\text{lf}}(\mathbf{P}_c^i, \mathbf{q}^i) \times \mathbf{F}_{\text{lf}}^i + \mathbf{P}_{\text{rf}}(\mathbf{P}_c^i, \mathbf{q}^i) \times \mathbf{F}_{\text{rf}}^i \\ &\quad + \mathbf{P}_h(\mathbf{P}_c^i, \mathbf{q}^i) \times \mathbf{F}_h^i(k^i) + \mathbf{P}_c^i \times \mathbf{F}_G + \Gamma_{\text{lf}z}^i + \Gamma_{\text{rf}z}^i = \mathbf{0} \end{aligned} \tag{13}$$

In order to avoid foot slipping, each foot contact force is constrained to remain inside a friction cone in (14).

$$\left\| \left[\mathbf{F}_{\text{lf,rf}_x}{}^i, \mathbf{F}_{\text{lf,rf}_y}{}^i \right]^T \right\| \leq \mu \left\| \mathbf{F}_{\text{lf,rf}_z}{}^i \right\| \tag{14}$$

with μ denoting the friction coefficient between the feet and the ground. The hand force magnitude is constrained as follows.

$$k^L \leq k^i \leq k^U \tag{15}$$

The lower bound k^L is defined by the task. It stands for the minimum magnitude of the interaction force that is necessary for performing the object manipulation.

4.3. Optimization with respect to a manipulation path

Given a manipulation path and the desired force directions along the path, the whole posture optimization problem is solved with respect to each discretized point i. We want the foot positions to be fixed during the manipulation task (Figure 3), so the following constraints are used between each discretized step.

$$\mathbf{P}_{\text{lf,rf}}(\mathbf{P}_c^i, \mathbf{q}^i) - \mathbf{P}_{\text{lf,rf}}(\mathbf{P}_c^{i-1}, \mathbf{q}^{i-1}) = \mathbf{0}, i > 2. \tag{16}$$

In this way, the final solution of foot positions satisfy not only the constraints associated with a local point i, but also those for the whole motion path. It should be noticed that we did not impose similar constraints on the CoM position and joint angles. Therefore the virtual character is allowed to move its body and change its posture during manipulation, even though its feet are fixed.

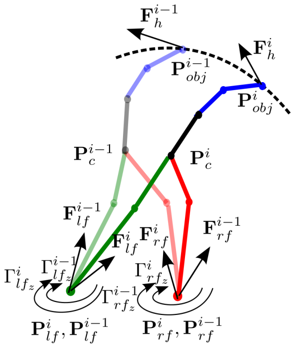


Figure 3: *Postures along a motion path with fixed foot positions.*

Moreover, the minimum value of k^i is maximized to maximize the hand force along the whole path. Hence, instead of using (3) as the hand force objective function, the following one is used for the whole path:

$$G_h(\left\{k^i\right\}) = -\min_i \left\{k^i\right\} \tag{17}$$

An advantage of our posture optimization is that it can be used to improve the robustness by taking account of perturbations in the optimization problem. This can be done by adding more pairs of hand positions and hand force directions. For example, at point i of the original motion path, some perturbations ($\delta\mathbf{P}_{\text{obj}}$, $\delta\mathbf{F}_h$) of different magnitudes and directions can be added to $\mathbf{P}_{\text{obj}}^i$ and $\mathbf{F}_h^i$:

$$\begin{aligned} \tilde{\mathbf{P}}_{\text{obj}}^i &= \mathbf{P}_{\text{obj}}^i + \delta\mathbf{P}_{\text{obj}} \\ \tilde{\mathbf{F}}_h^i &= \mathbf{F}_h^i + \delta\mathbf{F}_h, \end{aligned} \tag{18}$$

The optimization problem is solved with respect to a set of possible hand positions $\tilde{\mathbf{P}}_{\text{obj}}^i$ and force directions $\tilde{\mathbf{F}}_h^i$, so that the posture solution can better cope with the perturbations.

The optimization problem is summarized as follows:

$$\begin{aligned} \min_{\Theta = \cup_i \Theta^i} . \; & w_h G_h + \sum_i (w_f G_f^i + w_q G_q^i) \\ \text{s.t.} \; & \left\{ \Psi_{\text{GK}}^i(\mathbf{P}_c^i, \mathbf{q}^i) \right\} \\ & \left\{ \Psi_{\text{FM}}^i(\Theta^i) \right\} \end{aligned} \tag{19}$$

The above optimization problem contains several equality and inequality constraints, most of which are nonlinear. We solve it by using CFSQP algorithm [CLT97].

Note that the optimization problem presented here is just an example to explain the idea. It can be generalized to handle other problems with different contact configurations, such as a character moving one foot using both hands and the other foot as fixed contacts.

5. Implementation for manipulation task control

The implementation of the aforementioned posture optimization in the control of a character for a manipulation task is divided into three steps:

Step I: Off-line posture optimization. Before task execution, the posture optimization problem is solved to find suitable postures for the manipulation task. The optimization result provides us with the optimal solution Θ^*, from which the joint angles (q^*) and the positions of control frames, especially the CoM positions ($\mathbf{P}_c{}^*$) and the foot positions ($\mathbf{P}_{\text{lf}}{}^*$, $\mathbf{P}_{\text{rf}}{}^*$), as well as the maximum allowable value of the manipulation force magnitude (k^*), can be obtained.

Step II: On-line posture adjustment. The postures, especially contact configurations, are adjusted before task execution. In our implementation, a walking controller is applied to make the character walk to the optimized position ($\mathbf{P}_{\text{lf}}{}^*$, $\mathbf{P}_{\text{rf}}{}^*$). The walking motion generator presented in [HDW*10] is applied here, which generates automatically the reference trajectories of the CoM and the feet according to their initial states and their desired states ($\mathbf{P}_c{}^*$, $\mathbf{P}_{\text{lf}}{}^*$, and $\mathbf{P}_{\text{rf}}{}^*$).

Step III: On-line manipulation task control. The character starts to perform the manipulation task by using a manipulation task controller, which takes as inputs the optimized positions of control frames and manipulation task requirements. The position of each control frame $\mathbf{P}_j(\mathbf{P}_c^*, q^*)$ can be used as a reference position in the controller. In the next section, we will show that the manipulation task performance can be improved using the optimized postures.

To perform manipulation tasks, the force-based motion control proposed in [LME*11] is applied. The motion of each control frame is associated with a motion task force. The controller takes as inputs the desired values of these motion task forces $\mathbf{F}^d$ and outputs joint torques. The joint torques are computed based on the comprehensive consideration of the desired control frame positions, joint angles, foot contact forces and the gravity force. The solution of joint torques τ is used to drive the virtual character.

The manipulation task controller solves the following constrained Quadratic Programming problem.

$$\begin{aligned} \min_{\mathbf{F}_j,\tau} . \sum_j \|\mathbf{F}_j^d - \mathbf{F}_j\|^2_{\mathbf{W}_1} + \|\tau^d - \tau\|^2_{\mathbf{W}_2} & \\ \text{s.t. } \Phi(\tau, \mathbf{F}_j) = \mathbf{0} & \qquad (20) \\ \left\| \left[\mathbf{F}_{\text{lf,rf}_x}, \mathbf{F}_{\text{lf,rf}_y} \right]^T \right\| \leq \mu \left\| \mathbf{F}_{\text{lf,rf}_z} \right\|, & \end{aligned}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices whose values are chosen according to the importance levels or the priorities of different objectives. The static equilibrium at each time step is ensured by

$$\Phi(\tau, \mathbf{F}_j) = \mathbf{L}\tau - \sum_j \mathbf{J}_j^T \mathbf{F}_j - \mathbf{f}_G, \qquad (21)$$

where $\mathbf{L} = \left[\begin{array}{c} \mathbf{0}\,\mathbf{I} \end{array} \right]^T$ is a matrix to select the actuated DoF, $\mathbf{J}_j$ is the Jacobian matrix evaluated at frame j, and $\mathbf{f}_G$ denotes the gravity force in generalized coordinates. Each foot contact force is constrained inside a friction cone to ensure a non-sliding contact.

A proportional-derivative (PD) control law is applied to compute the desired motion task force based on the state error (position error δ_P and velocity error δ_v) of each frame j.

$$\mathbf{F}_j^d = \mathbf{K}_{p\,j} \delta_{P\,j} (\mathbf{P}_j{}^*, \mathbf{P}_j^r) + \mathbf{K}_{d\,j} \delta_{v\,j} \qquad (22)$$

where $\mathbf{P}_j^r$ denotes the actual position of frame j. The optimized joint angles may also be taken into account by the computation of desired joint torques τ_q^d.

$$\tau_q^d = \mathbf{K}_{p_q} \delta_{P\,q} (q^*, q^r) + \mathbf{K}_{d_q} \delta_{v\,q} \qquad (23)$$

Moreover, additional interaction force $\bar{k}\hat{\mathbf{F}}_h$ for manipulating the object is added to the desired motion task forces of the hand.

$$\mathbf{F}_h^d = \mathbf{K}_{p_h} \delta_{P_h} (\mathbf{P}_h, \mathbf{P}_h{}^r) + \mathbf{K}_{d_h} \delta_{v_h} + \bar{k}\hat{\mathbf{F}}_h \qquad (24)$$

with $k^L \leq \bar{k} \leq k^*$. Since our posture optimization problem is solved for each discretized point i, the smoothness of the overall motion is achieved by using a smooth interpolation trajectory that connecting the discretized desired positions ($\mathbf{P}_j^i{}^*$) in the controller.

6. Results

The proposed method has been tested on a virtual character performing different manipulation tasks in simulation. The character consists of 45 DoFs, including 6 DoFs for the root position and orientation, 8 DoFs for each leg, 7 DoFs for each arm, 3 DoFs for the thorax, 3 DoFs for the chest, and 3 DoFs for the head.

Each of the following tests are divided into two parts. In the first part, the virtual character tries to perform the task without using posture optimization results, with its feet remaining at their initial positions. In the second part, the character performs the task using posture optimization results. A walking controller is used to adjust the foot positions according to their optimal values before performing the task. The optimal trajectory of each control frame, especially the CoM trajectory, is used as the reference trajectory during manipulation. The experiment setup for each manipulation task is in Table 1. Real-time animations can be seen in the accompanying video.

Table 1: *Experiment setup: desired motion path* $\mathbf{P}_h$ *and manipulation force direction* $\hat{\mathbf{F}}_h$ *(applied by the objects on the character) associated with each discretized point, where* $\mathbf{O}_{obj}$ *denotes the origin of the valve, and* $\mathbf{P}^0_{obj}$ *denotes the initial contact position between the hand and the object.*

Task	$\mathbf{P}^i_h$, $\hat{\mathbf{F}}^i_h$
open a valve	$\mathbf{O}_{\mathrm{obj}}+[0,0,0.25]^T$, $[-1,0,0]^T$
	$\mathbf{O}_{\mathrm{obj}}+[0.177,0,0.177]^T$, $[-0.707,0,0.707]^T$
	$\mathbf{O}_{\mathrm{obj}}+[0.25,0,0]^T$, $[0,0,1]^T$
move box	$\mathbf{P}^0_{\mathrm{obj}}$, $[0,0,-1]^T$
	$\mathbf{P}^0_{\mathrm{obj}}+[0,-0.2,0.6]^T$, $[0,0,-1]^T$
	$\mathbf{P}^0_{\mathrm{obj}}+[0,0.2,0.6]^T$, $[0,0,-1]^T$
push storage cabinets	$\mathbf{P}^0_{\mathrm{obj}}$, $[0,-1,0]^T$
	$\mathbf{P}^0_{\mathrm{obj}}+[0,0.4,0]^T$, $[0,-1,0]^T$

6.1. Follow a desired motion path

Our approach can choose suitable postures that allow the character's hand to follow a curved motion path. To demonstrate this, the character is required to open a valve with a radius of $0.25m$ to 90 degrees with his right hand (Figure 4). The hand should follow exactly the given motion path as quarter of a circle, because the valve can only rotate around its rotation axle which is fixed. The desired hand force is tangential to the motion path. The initial foot positions are not optimized for the task, as they can cause a break of foot contacts during manipulation.

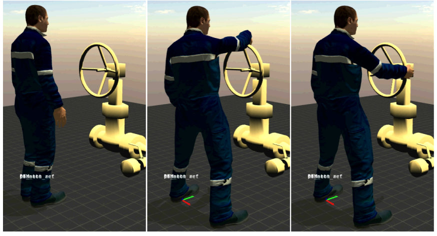


Figure 4: *Character opening a valve.*

The optimized foot positions make the character walk rightward before starting manipulation. This foot contact configuration enables the hand to open the valve along the given motion path without breaking foot contacts. It is observed that less upper body movement is generated by using the optimal postures than not using them, which makes the whole body motion more natural.

6.2. Obstacle avoidance

An experiment of moving an object while avoiding obstacles has been conducted. The character should change postures so as to allow the hands to easily approach a box located between a table and a shelf above the table, then move it around the shelf, and finally put it on the shelf (Figure 5). The desired manipulation path goes around the shelf with a safety margin. The postures optimized according to this manipulation path can allow the character to successfully fulfill the task without causing collisions with the shelf.

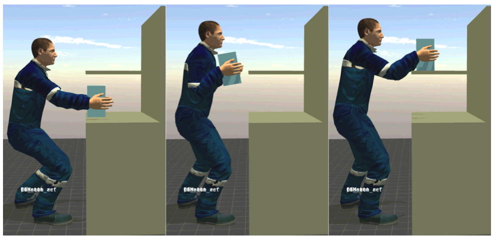


Figure 5: *Character moving a box while avoiding obstacles.*

6.3. Joint comfort

Our posture optimization provides posture solutions that can take into account the joint discomfort measure. The function of the joint discomfort objective is to enhance the preference of using certain joints to fulfill a motion task. This function can be observed by comparing the behaviors of the character taking up light and heavy box. In our experiments, the neutral values q^N are set to joint angles of an erect standing posture. A higher value of a joint weight γ_α reinforces the value of q_α to be closer to q^N_α, which means we prefer to use joints with weights lower than γ_α to make the end-effector attain the desired position. Similar to [YMK*04], we set higher weights for joints on the back of the character, and lower weights for joints on the arms and the legs. In addition, we adjust certain joint weights to adapt to different task requirements. For a task of lifting a box, the character can either lean over with the back then lift it up using the back, or crouch down while keeping the back straight then lift it by standing up. People tend to choose the latter one to take up a heavy object. This is what physical therapists usually suggest people to do in order to protect their backs. Such behaviors can be achieved by tuning the joint weights. For heavy box, high values are assigned to γ_{back} associated

(a) $m = 5kg, \gamma_{back} = 0.4$ (b) $m = 15kg, \gamma_{back} = 10$

Figure 6: *Character lifting box of different mass (m).*

with joints on the back of the body. As a result, the character just slightly crouches down to pick up a light box; whereas it crouches down more and carefully keep the back straight to take up a heavy box (Figure 6).

6.4. Handle interaction forces

When searching for suitable postures for the task of lifting a box, the weight of the box can be taken into account by k^L in (15), which indicates the minimum force that is necessary to lift the box. It is observed that when the box is heavy, the optimized CoM position is obviously behind its initial position. This result is consistent with the needs of the character to lean backward so as to balance the interaction force due to the weight of the box.

Similar results have been observed during pushing tasks. The character is required to push forward a storage cabinet (Figure 7) a distance of up to $0.4m$. Storage cabinets of different mass m (from $30kg$ to $50kg$) and different friction coefficients with the ground μ (from 0.1 to 0.4) are used.

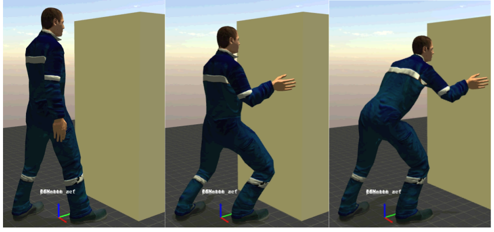


Figure 7: *Character pushing a storage cabinet.*

The results of optimal foot positions tell the character to separate the feet along the pushing direction, so as to generate a robust posture against the pushing force. Moreover, the optimal CoM and shoulder positions tell the character to lean forward. Similar behaviors can be observed when a real human attempts to push strongly.

More fluctuation of the interaction force has been observed without the use of optimal postures, which suggests that optimal postures help to generate more coherent motions during manipulation. The magnitudes of the forces applied by the hand pushing a storage cabinet are shown in Figure 8. It can be seen that the interaction forces resulting from optimal postures is more stable than those resulting from non-optimal postures. Without an optimization before task execution, the character may find his posture not quite adaptable for the task from time to time. If continuing pushing forward as strongly as before will result in the loss of balance, then the character will sacrifice the hand task performance to ensure its balance; because the balance task is of higher priority than all the other motion tasks. Consequently, the pushing force will be reduced at this moment. However, task performance can be improved by using the optimized CoM positions as the reference positions in the manipulation task controller, because the optimized CoM positions provided by our posture optimization are suitable for the contact configurations and the interaction forces.

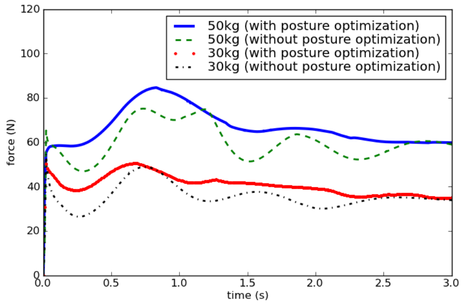


Figure 8: *Forces applied by the hand on the storage cabinet.*

7. Discussion

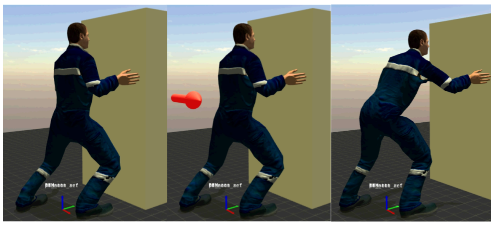
7.1. Compromise between objectives

The objective weights are chosen according to task requirements. If the character has to apply a strong manipulation force, for example, taking up a heavy box or pushing against a heavy obstacle, the weight w_h should be set to a high value to enhance the hand force objective, and to ensure that the maximum allowable manipulation forces are sufficient for the task. If we want to reinforce non-sliding contacts on the feet, then w_f should be assigned with a high value to reduce tangential contact forces on the feet; however, the maximum allowable value of k might be limited as a compromise. The objective weights used in our experiments are: $w_h = 1000$, $w_f = 1$, and $w_q = 500$.

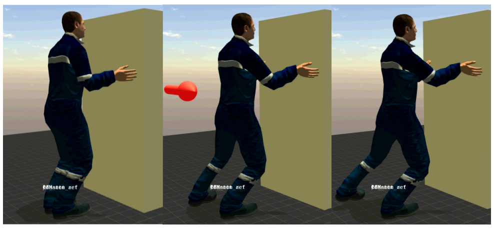
7.2. Robustness to mechanical interactions

Generally, the optimal postures tell the character to increase the distance between its feet, along the direction of the manipulation force, so as to generate a robust posture against this force. For example, when opening a valve, its feet are separated from each other mainly in the lateral direction (x axis). However, when moving an object forward or backward, the feet are much more separated in the sagittal direction (y axis).

As mentioned before, perturbations can be taken into account in posture optimization. In our experiment, some perturbation forces, including those which are perpendicular to the manipulation force, are considered. During task execution, external perturbation forces (up to $120N$ during $0.3s$ or up to $90N$ during $1s$) have been applied on the character (Figure 9). When pushed by strong perturbation forces, the character using non-optimal postures abandons the task and sometimes loses its balance. It has to move its foot to



(a) Using optimal postures



(b) Using non-optimal postures

Figure 9: *Behaviors of the character before, during, and after perturbations from an external pushing force (shown by the red arrow) during task execution. The optimal postures can better resist the push.*

try to keep balance. However, by adopting optimal postures, the character's ability to continue task execution under some perturbation forces is enhanced. Less body movements are generated to resist strong pushes and to recover from them, and neither foot slipping nor a break of foot contacts are observed.

7.3. Physical consistency

Our approach represents the G-K and F-M constraints as hard constraints. It is possible to represent some of the constraints as soft constraints to simplify the optimization problem. However, as our goal is to produce motions that can really work in a physics-based environment, the crucial constraints for physical consistency, such as the F-M constraints, should still be strictly respected. The motions provided by our posture optimization are not the final output motions; they are used as reference motions to improve the performance of the task controller. The final motions are refined by the controller, which do not just look realistic, but are really verified in the physics-based simulation.

7.4. Limitations

The current approach has a few limitations. First, since the posture optimization problem that we are dealing with is not convex, several local minima may exist. The solution of our posture optimization is a local optimum in the neighborhood of an initial value; and the global optimum might be drastically different. This is because the CFSQP algorithm that we use to solve our problem is based on derivatives, which leads to a local minimum. However, we choose to use a derivative-based optimization algorithm because it converges the fastest. One possible solution to improve the posture solution is to build a database of captured motion for different kinds of manipulation tasks, so as to provide natural and lifelike initial postures.

Second, the computation time for solving the optimization problem is sensible to the given task, especially the complexity of the motion path. For all the experiments mentioned above, it took from $0.06s$ to $2min$ to solve the problems. Currently we first apply posture optimization off-line. Then use the optimization results in the on-line task controller.

Moreover, the current posture optimization might not always be able to find an optimal solution, especially when the manipulation path is too long for the optimization to find a suitable contact configuration that supports the whole path. We plan to handle this problem by developing some automatic segmentation techniques; so that the path can be divided into several segments automatically, and posture optimization will be executed segment by segment.

Besides, our approach considers quasi-static cases. It can be successfully applied to generate motions for a variety of manipulation tasks. However, it is still limited for synthesizing highly dynamic motions.

8. Conclusion

We have introduced a generic approach to find optimal postures for object manipulation tasks. The optimized posture can enable the end-effector to follow given manipulation path while applying the maximum manipulation forces without causing foot slipping and balance problems. Besides, constraints such as joint limits, non-sliding contacts, and geometrical relations with the environment can be satisfied.

The results of our experiments suggest that the proposed posture optimization problem based on both G-K constraints and F-M constraints can be numerically solved for a wide variety of tasks. The obtained postures are different from task to task, changing not only in favor of different motion paths, but also for different interaction forces. Task performance can be improved by choosing suitable postures before actually performing the task. The robustness of postures can be improved, so that character can cope with perturbations due to mechanical interactions.

This approach can be applied to a virtual character manipulating objects while trying to follow an operator's motions. In the implementation presented in [NWB*10], the operator needs to adjust his postures according to the character's balance features, such as the support polygon and the CoM. Our approach automatically computes adaptable postures of the character in advance for a task, and then further adjusts its postures during task execution, so that the operator does not need to compensate for the character's balance.

One future direction is to reduce the computation time when there are a large number of discretized points along a complex motion path. We can also improve our approach by taking into account objectives concerning joint torques as in [BMT96, HHBL06]. Moreover, we can make our approach more generic by optimizing the trajectory of the center of pressure (CoP) as well. So for each contact, the CoP will be optimized and will be allowed to move inside the support polygon instead of being fixed. To realize this, the foot size and the admissible shape of the support polygon should also be taken into account in the optimization.

References

[AdSP07] ABE Y., DA SILVA M., POPOVIĆ J.: Multiobjective control with frictional contacts. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 249–258. 2

[BL08] BRETL T., LALL S.: Testing static equilibrium for legged robots. *IEEE Transactions on Robotics 24*, 4 (aug. 2008), 794–807. 3

[BMT96] BOULIC R., MAS R., THALMANN D.: A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics 20*, 5 (1996), 693–701. 10

[CLT97] C. LAWRENCE J. L. Z., TITS A.: User's guide for cfsqp: A c code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints, version 2.5. *Institute for Systems Research, University of Maryland, College Park, MD* (1997). 5

[CMAL07] COLLETTE C., MICAELLI A., ANDRIOT C., LEMERLE P.: Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts. In *7th IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 81–88. 2

[DSDLR*07] DE SCHUTTER J., DE LAET T., RUTGEERTS J., DECRÉ W., SMITS R., AERTBELIËN E., CLAES K., BRUYNINCKX H.: Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research 26*, 5 (2007), 433–455. 2

[dSRA*05] DE SCHUTTER J., RUTGEERTS J., AERTBELIEN E., DE GROOTE F., DE LAET T., LEFEBVRE T., VERDONCK W., BRUYNINCKX H.: Unified constraint-based task specification for complex sensor-based robot systems. In *IEEE International Conference on Robotics and Automation (ICRA)* (april 2005), pp. 3607–3612. 2

[FP03] FANG A. C., POLLARD N. S.: Efficient synthesis of physically valid human motion. *ACM Trans. Graph. 22*, 3 (July 2003), 417–426. 2

[HDW*10] HERDT A., DIEDAM H., WIEBER P.-B., DIMITROV D., MOMBAUR K., DIEHL M.: Online Walking Motion Generation with Automatic Foot Step Placement. *Advanced Robotics -Utrecht- 24*, 5-6 (2010), 719–737. 6

[HHBL06] HARADA K., HAUSER K., BRETL T., LATOMBE J.-C.: Natural motion generation for humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (oct. 2006), pp. 833–839. 10

[HNTH11] HAUSER K., NG-THOW-HING V.: Randomized multi-modal motion planning for a humanoid robot manipulation task. *The International Journal of Robotics Research 30*, 6 (2011), 678–698. 2

[JL09] JAIN S., LIU C. K.: Interactive synthesis of human-object interaction. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), SCA '09, pp. 47–53.

[JYL09] JAIN S., YE Y., LIU C. K.: Optimization-based interactive motion synthesis. *ACM Transaction on Graphics 28*, 1 (2009), 1–10. 2

[KLY11] KANOUN O., LAUMOND J.-P., YOSHIDA E.: Planning foot placements for a humanoid robot: A problem of inverse kinematics. *The International Journal of Robotics Research 30*, 4 (2011), 476–485. 2

[KSPW04] KHATIB O., SENTIS L., PARK J., WARREN J.: Whole-body dynamic behavior and control of human-like robots. *I. J. Humanoid Robotics 1*, 1 (2004), 29–43. 2

[LME*11] LIU M., MICAELLI A., EVRARD P., ESCANDE A., ANDRIOT C.: Interactive dynamics and balance of a virtual character during manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)* (may 2011), pp. 1676–1682. 2, 6

[LP02] LIU C. K., POPOVIĆ Z.: Synthesis of complex dynamic character motion from simple animations. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), SIGGRAPH '02, ACM, pp. 408–416. 2

[MZC*09] MA L., ZHANG W., CHABLAT D., BENNIS F., GUILLAUME F.: Multi-objective optimisation method for posture prediction and analysis with consideration of fatigue effect and its application case. *Computers & Industrial Engineering 57*, 4 (Nov. 2009), 1235–1246. 4

[NWB*10] NGUYEN N., WHEATLAND N., BROWN D., PARISE B., LIU C. K., ZORDAN V.: Performance capture with physical interaction. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2010), SCA '10, Eurographics Association, pp. 189–195. 9

[PMM*09] PEINADO M., MEZIAT D., MAUPU D., RAUNHARDT D., THALMANN D., BOULIC R.: Full-body avatar control with environment awareness. *IEEE Computer Graphics and Applications 29*, 3 (may-june 2009), 62–75. 2

[Smi10] SMITS R.: Design of a constraint-based methodology and software support. *PhD thesis, Katholieke Universiteit Leuven* (2010). 2

[SSF09] SAAB L., SOUERES P., FOURQUET J.: Coupling manipulation and locomotion tasks for a humanoid robot. In *International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)* (july 2009), pp. 84–89. 2

[Sti10] STILMAN M.: Global manipulation planning in robot joint space with task constraints. *IEEE Transactions on Robotics 26*, 3 (june 2010), 576–584. 2

[YKEL06] YOSHIDA E., KANOUN O., ESTEVES C., LAUMOND J.-P.: Task-driven support polygon reshaping for humanoids. In *6th IEEE-RAS International Conference on Humanoid Robots* (dec. 2006), pp. 208–213. 2

[YKH04] YAMANE K., KUFFNER J. J., HODGINS J. K.: Synthesizing animations of human manipulation tasks. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 532–539. 2

[YL10] YE Y., LIU C. K.: Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph. 29* (July 2010), 74:1–74:9. 2

[YMK*04] YANG J., MARLER R. T., KIM H., ARORA J. S., ABDEL-MALEK K.: Multi-objective optimization for upper body posture prediction. In *10th AIAA/ISSMO multidisciplinary analysis and optimization conference* (2004). 4, 7