Interactive Dynamics and Balance of a Virtual Character During Manipulation Tasks

Mingxing Liu, Alain Micaelli, Paul Evrard, Adrien Escande, Claude Andriot
CEA, LIST
18 route du Panorama, BP6,
FONTENAY AUX ROSES, F- 92265 France
Email: {mingxing.liu, alain.micaelli, paul.evrard, adrien.escande, claude.andriot}@cea.fr

Abstract— **This paper proposes a new framework of online hybrid control for virtual characters, which combines multi-objective control and motion capture techniques. At each time step, the motions of the operator are captured and sent to the character. These captured motions help the character understand what task the operator wants it to perform. Then, the character decides by itself how to execute proper actions while maintaining its balance, in a virtual environment which is different from the real world. Given a manipulation task, the desired virtual task wrench is computed thanks to a force control approach. The control system takes these task wrenches into consideration and solves a constrained quadratic problem for joint torques, which are then used to drive the virtual character. The whole control problem is solved online so that the virtual character can interact in real-time with the virtual environment as well as the operator. Examples using this control framework are presented as results from simulation.**

Index Terms— **Virtual character control, Optimization, Balance control, Manipulation, Force control, Interaction.**

I. NOMENCLATURE

We use the following nomenclature in this paper.

$\mathbf{M}$	Generalized inertia matrix.
$\mathbf{q}$	Vector of joint angles.
$\mathbf{T}$	Vector of velocity in generalized coordinates ($\mathbf{T} = \begin{bmatrix} \mathbf{V}_{root} \\ \dot{\mathbf{q}} \end{bmatrix}$, with $\mathbf{V}_{root}$ the twist of the root body of the virtual character).
$\dot{\mathbf{T}}$	Vector of acceleration in generalized coordinates.
τ	The set of joint torques.
$\mathbf{L}$	Matrix to select the actuated degrees of freedom ($\mathbf{L} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$).
$\mathbf{J}$	Jacobian matrix associated with different frames denoted by subscripts com (for center of mass), $waist$ (for waist body frame), t (for task frames) and c (for contact frames).
$\mathbf{W}$	Wrenches associated with different frames denoted by subscripts com, $waist$, t and c. $\mathbf{W} = \begin{bmatrix} \Gamma \\ \mathbf{F} \end{bmatrix}$, with Γ and $\mathbf{F}$ the moment and force component of $\mathbf{W}$ respectively. The superscripts d and r stand for "desired" and "real" respectively. All the "real" wrenches are defined to be applied by the environment on the character, while all the "desired" wrenches are defined to be applied by the character on environment.
γ	Gravity force in generalized coordinates.
$\mathbf{N}$	Inertial and Coriolis forces.

II. INTRODUCTION

A. Problem statement

In this paper, we are interested in the problem of online hybrid control of virtual characters, which combines multi-objective control and motion capture techniques. We focus especially on online interactive control, which means that the character is in real-time interaction with the virtual environment as well as an operator. It is important that the character should be able to understand what tasks it is asked to execute according to the intention of the operator. In our system, motion capture techniques are used for two purposes. On one hand, the captured motion of an operator can be used as a reference motion trajectory for the character. On the other hand, the character finds out the intention of the operator by observing his movements.

There are several difficulties in this research topic. Firstly, virtual characters are complex, under actuated systems, with a high number of degrees of freedom (DoF). They can have many postures which help to accomplish the same task. The controller must find the best solution among all the possible postures. Secondly, as is often the case, the character and the operator stay in two different environments. Let's consider the scenario of a character pushing a storage cabinet for example. The storage cabinet which exists in virtual environment does not exist in the real world. When the operator wants the character to push the storage cabinet, he just sends out his indication to the character by reaching out his hands. Before touching the storage cabinet, the character can simply keep tracking the operator's movements. But once the hands reach the storage cabinet, the character should be able to generate a posture, which is different from the operator's, but more suitable to handle the interaction with the storage cabinet. For example, the character will lean towards the storage cabinet to push it while the operator won't, as there is no real storage cabinet for him to push. Furthermore, it should be noticed that we need an online controller which works with the motion capture system in real-time, so as to guarantee the real-time interaction between the virtual character and the operator.

B. Related work

A lot of work has been dedicated to the control of virtual human characters in order to reproduce natural and lifelike character motions. There is a current trend in the robotics and virtual reality communities towards providing such motions by motion capture. In a motion capture system, an actor wears special markers near each joint. Each marker attached to the actor corresponds to a point on the virtual character's body. The movements of each marker are measured and collected.

A traditional way to make the virtual character follow the captured motions is to connect the markers with corresponding points on the character by virtual springs [7]. These virtual springs help to guide the character's motion. With a similar idea, our approach also uses motion capture for posture tracking (Fig.1). We consider it as one of the weighted objectives in our optimization. Furthermore, we aim at reducing the number of markers to facilitate the utilization. In our experiment of manipulation task control, the virtual character has up to 40 DoF, whereas, instead of tracking the trajectories of all the DoF, we tracked the hands motions only. The movements of the other parts of the body are generated automatically by the control framework.

Our optimization-based framework is inspired by the work of C. Collette et al. [2] and Y. Abe et al. [1], who have both proposed a framework of multi-objective control based on optimization. Their approaches use acceleration-based control, which defines the objective as:

$$\arg\min_{\dot{\mathbf{T}}} \left\| \mathbf{J}\dot{\mathbf{T}} + \dot{\mathbf{J}}\mathbf{T} - \ddot{\mathbf{x}}^d \right\| \quad (1)$$

with $\ddot{\mathbf{x}}^d$ the desired acceleration. However, such a method may lead to numerical problems or singularities for certain poses, since the process of solving for joint accelerations $\dot{\mathbf{T}}$ in (1) boils down to computing the inverse of the Jacobian. To deal with these problems, our system uses force control as an alternative approach, which renders the interaction more feasible and safe.

The force control approach suggested in this article is based on the idea of a Jacobian-transpose (JT) control method. J. Pratt et al. [9] suggested the use of this method in virtual actuator control. E. Demircan et al. [5] reconstructed human motion from motion capture data, by using a simplified version of the framework proposed in [6]. This simplified version is close to the JT control method. S. Coros et al. [4] applied the JT control method to adjust for gravity and velocity errors in a walk controller. However, neither of them used optimization to find the solution. In contrast with these approaches, our controller is formulated as a (quadratic programming) QP problem, which allows us to handle the unilateral constraints on the contact forces in a natural way.

Our work is most similar to the methods of J. Wu et al. [12] and B.J. Stephens et al. [11]. Both of these works combined the JT method and optimization to solve their control problems, as we do. J. Wu et al. [12] proposed a static resolution of forces based on the relations of some pairs of action frames (such as feet) and reaction frames (such as the root body). In the optimization problem, they used a constraint which defines the admissible domains: $\mathbf{ZW} = \mathbf{0}$, with the columns of $\mathbf{Z}$ forming a basis for the null space of $\mathbf{J}^T$. This means that they are interested only in wrenches which are achievable by control. However, this additional constraint may conflict with particular contact constraints associated with friction cones. Moreover, our method considers more generally the static equilibrium of the wrenches on all the DoF of the character instead of on the root body only. B.J. Stephens et al. [11] proposed a synthesis of control laws which involves two main steps. The first step consists of the estimation of the contact force which is compatible with the desired dynamics of the center of mass (CoM) while compensating for gravity. This step outputs the desired contact forces, which are used by the second step to compute the joint torques by solving the dynamics and constraint equations. They proposed to solve these equations in the second step by a damped pseudo-inverse, which is less time-consuming. The pseudo-inverse leads to a solution that minimizes the norm of the vector $\left[\begin{array}{cc} \dot{\mathbf{T}}^T & \tau^T \end{array} \right]^T$, which may, in most situations, risk to generate unwanted behaviors or movements. For example, if we consider a simple gravity compensation, we do not want to move the character at all, but the use of a pseudo-inverse may lead to an unwanted movement. On the other hand, our method requires solving only one QP problem to obtain a solution which satisfies all constraints.

Fig. 1. Real-time manipulation task control based on motion capture. The markers are attached on the hands of the operator.

Our contribution with regard to these works is that we propose a simple optimization-based framework for interactive control of a virtual character. All of the objectives and constraints are formulated into one weighted QP problem. Each task level controller has a force control policy. The controller solves for the optimal solution which is a compromise with regard to all the considered criteria. Our control framework has several advantages. Firstly, the whole control problem can be solved in one procedure in real-time.

Secondly, the force control policy allows the character to adapt to unknown external contact forces. Furthermore, the controller is capable of generating motions that are suitable in a virtual environment which is different from reality, rather than simply emulating a captured motion sequence. Besides, the weighted optimization framework allows to handle tasks at different weight levels.

III. OVERVIEW

Our control framework is shown in Fig. 2. The controller is applied on a virtual character with 34 joint DoF and 6 root DoF. We consider the dynamics of the virtual character (Fig. 3) as a second order system (2).

$$\mathbf{M}\dot{\mathbf{T}} + \mathbf{N} + \gamma^r = \mathbf{L}\tau + \mathbf{J}_{ext}^T \mathbf{W}_{ext}^r \tag{2}$$

Where $\mathbf{W}_{ext}$ denote all the external wrenches.

At each time step of simulation, the optimization computes the joint torques from a motion capture sequence, the given manipulation tasks and the constraints to respect. All the objectives and constraints are considered as a QP problem.

IV. FORCE CONTROL

Force control is introduced in order to achieve objectives such as the tracking of captured motions, desired CoM position, waist orientation and movements of the end effectors. Furthermore, it allows us to handle contacts during the manipulation by setting desired contact forces.

First of all, let's remind the reader of the JT control method [5], [9]. Given a desired wrench $\mathbf{W}^d$ in the space of Cartesian coordinates, the equivalent joint torques τ can be obtained by $\tau = \mathbf{J}^T\mathbf{W}^d$, with $\mathbf{J}$ the jacobian matrix at the point where $\mathbf{W}^d$ is supposed to be applied. The goal of our control system is to compute joint torques based on given tasks. For each task, we imagine that a virtual wrench $\mathbf{W}^d$ is applied at a certain point on the body of the character to guide its motion (Fig. 3). We define the desired wrench using a proportional-derivative (PD) feedback control law:

$$\mathbf{W}^d = k_p D(\mathbf{X}^d, \mathbf{X}^r) + k_d D(\mathbf{V}^d, \mathbf{V}^r) \tag{3}$$

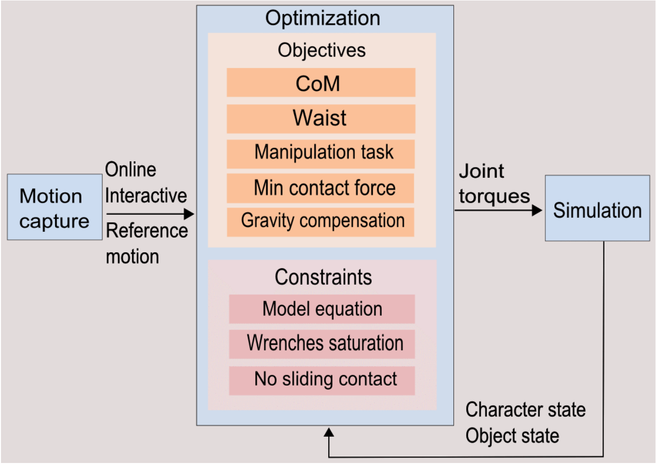


Fig. 2. Block diagram of the control framework

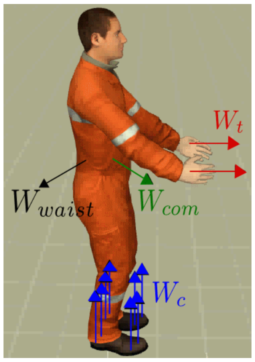


Fig. 3. Virtual humanoid model

with $\mathbf{X}^r \in \mathbf{SE(3)}$, $\mathbf{X}^d \in \mathbf{SE(3)}$, $\mathbf{V}^r \in \mathbf{se(3)}$ and $\mathbf{V}^d \in \mathbf{se(3)}$, where $\mathbf{SE(3)}$ is the special Euclidean group and $\mathbf{se(3)}$ is the Lie algebra of $\mathbf{SE(3)}$. $D(\mathbf{X}^d, \mathbf{X}^r)$ stands for the error between desired and current position and orientation, while $D(\mathbf{V}^d, \mathbf{V}^r)$ stands for the error between desired and current linear and angular velocity.

If a contact is established between the end effector and an object during the manipulation, an additional offset value can be added to the result of (3).

$$\mathbf{W}^d = k_p D(\mathbf{X}^d, \mathbf{X}^r) + k_d D(\mathbf{V}^d, \mathbf{V}^r) + \mathbf{W}^{offset} \tag{4}$$

This offset $\mathbf{W}^{offset}$ helps to enhance the contact between the end effector and the object. It can be defined according to the physical characteristic of the object, such as the weight and the friction coefficient.

The object of our force control is to obtain the joint torques τ which are as close as possible to $\mathbf{J}^T\mathbf{W}^d$.

V. OPTIMIZATION

In our multi-objective control, the optimization variables are joint torques τ, all the wrenches $\mathbf{W}$ and gravity force γ.

1) CoM objective: Our control system takes the CoM as the stability criteria and maintains balance by controlling its position. For CoM tracking objective, we consider the force component only. $\mathbf{F}_{com}^d$ is obtained by employing PD control in $\mathbb{R}^3$ to measure the error between the actual and desired CoM positions.

$$\mathbf{F}_{com}^d = k_p(x_{com}^d - x_{com}^r) + k_d(v_{com}^d - v_{com}^r) \tag{5}$$

It should be noticed that, if the character is simply standing on the horizontal ground, then the vertical projection of CoM should lie inside the supporting polygon for static balance. However, if multiple non-coplanar contacts exist, for example the character pushes an object with his hands, then the contact force due to this new interaction must be considered in the computation of the desired CoM configuration.

2) Waist objective: For some motions, the waist configuration should be considered. Here we take into account the orientation error of the waist, so Γ_{waist}^d is obtained by PD control in $\mathbf{SO(3)}$ and $\mathbf{so(3)}$.

$$\Gamma_{waist}^d = k_p D(R_{waist}^d, R_{waist}^r) + k_d D(\omega_{waist}^d, \omega_{waist}^r) \tag{6}$$

with $R_{waist}^r \in \mathbf{SO(3)}$, $R_{waist}^d \in \mathbf{SO(3)}$, $\omega_{waist}^r \in \mathbf{so(3)}$ and $\omega_{waist}^d \in \mathbf{so(3)}$, where $\mathbf{SO(3)}$ is the rotation group on $\mathbb{R}^3$ and $\mathbf{so(3)}$ is the Lie algebra of $\mathbf{SO(3)}$. $D(R_{waist}^d, R_{waist}^r)$ stands for the error between desired and current waist orientation, while $D(\omega_{waist}^d, \omega_{waist}^r)$ stands for the error between desired and current waist angular velocity.

3) Task objectives: The task objectives can be either tracking the captured motion trajectory of a frame $\mathbf{p}_i$ attached on an operator, or performing some specific manipulation tasks. The desired task wrench $\mathbf{W}_{t_i}^d$ is computed by PD control in $\mathbf{SE(3)}$ and $\mathbf{se(3)}$.

$$\mathbf{W}_{t_i}^d = k_p D(\mathbf{X}_{t_i}^d, \mathbf{X}_{t_i}^r) + k_d D(\mathbf{V}_{t_i}^d, \mathbf{V}_{t_i}^r) \tag{7}$$

with $\mathbf{X}_{t_i}^r \in \mathbf{SE(3)}$, $\mathbf{X}_{t_i}^d \in \mathbf{SE(3)}$, $\mathbf{V}_{t_i}^r \in \mathbf{se(3)}$ and $\mathbf{V}_{t_i}^d \in \mathbf{se(3)}$. $D(\mathbf{X}_{t_i}^d, \mathbf{X}_{t_i}^r)$ stands for the error between desired and current position and orientation of $\mathbf{p}_i$, while $D(\mathbf{V}_{t_i}^d, \mathbf{V}_{t_i}^r)$ stands for the error between desired and current linear and angular velocity of $\mathbf{p}_i$.

Additionally, an offset wrench can be added to adjust the desired wrench as in (4), so as to enhance the contact during the manipulation.

4) Contact force objective: As we want to produce stable motion with smaller contact forces, we use an optimization objective to minimize the contact forces between the feet and the ground. To realize this, the desired foot contact force $\mathbf{F}_{c_j}^d$ is set to zero.

5) Gravity compensation: In a virtual environment with existence of gravity, if no joint actuation forces are applied on the character, it will certainly fall onto the ground. One method to keep the character upright is to compensate the gravity force. The gravity compensation objective decouples open-loop gravity compensation on one hand and closed-loop correction of task errors on the other hand. Thus, the PD gains for the tasks can be set to lower values when the virtual character moves without disturbance. In our method, we achieve gravity compensation by adding a gravity variable γ in our optimization problem. The corresponding objective is to minimize the difference between the value of γ and the real gravity force γ^r.

A. Constraints

The control system optimizes the above objectives subject to the following constraints.

First of all, the following model equation (8) should be respected, which describes the static equilibrium of the system under $\mathbf{F}_{com}$, Γ_{waist}, $\mathbf{W}_{t_i}$, $\mathbf{F}_{c_j}$, γ and τ.

$$\begin{aligned}\mathbf{L}\tau = &\mathbf{J}_{com}^T \mathbf{F}_{com} + J_{waist}^T \Gamma_{waist} \\ &+ \sum_i \mathbf{J}_{t_i}^T \mathbf{W}_{t_i} + \sum_j \mathbf{J}_{c_j}^T \mathbf{F}_{c_j} + \gamma\end{aligned} \tag{8}$$

Secondly, the task wrenches are bounded.

In addition, contact constraints should be imposed. Here we apply a linearized Coulomb friction model as was used in [3], [1]. Each foot has four contact points. The friction cone of each contact is approximated by a four faced polyhedral convex cone. The contact force $\mathbf{F}_{c_j}$ should remain inside the friction cone. We formulate this constraint as

$$\mathbf{E}_{c_j} \mathbf{F}_{c_j} - \mathbf{d}_{c_j} > 0 \tag{9}$$

where

$$\mathbf{E}_{c_j} = \begin{bmatrix} \lambda_2 \times \lambda_1 & \lambda_3 \times \lambda_2 & \lambda_4 \times \lambda_3 & \lambda_1 \times \lambda_4 \end{bmatrix}^T, \tag{10}$$

with λ_1,λ_2,λ_3 and λ_4, the unit edge vectors of the approximated friction cone. $\mathbf{d}_{c_j}$ is a customer defined margin vector, so that the projection of $\mathbf{F}_{c_j}$ on the normal vector of each facet of the friction cone should be kept larger than $\mathbf{d}_{c_j}$.

B. Optimization summary

The optimization problem introduced in this section is summarized in (11). We combine all the optimization objectives by the weight matrix $\mathbf{Q}$.

$$\hat{\mathbf{S}} = \underset{\substack{\mathbf{F}_{com}, \Gamma_{waist}, \\ \mathbf{W}_{t_i}, \mathbf{F}_{c_j}, \tau, \gamma}}{\arg\min} \left\| \begin{bmatrix} \mathbf{F}_{com}^d \\ \Gamma_{waist}^d \\ \mathbf{W}_{t_i}^d \\ \mathbf{F}_{c_j}^d \\ \tau^d \\ \gamma^r \end{bmatrix} - \begin{bmatrix} \mathbf{F}_{com} \\ \Gamma_{waist} \\ \mathbf{W}_{t_i} \\ \mathbf{F}_{c_j} \\ \tau \\ \gamma \end{bmatrix} \right\|_{\mathbf{Q}}^2$$

$$\text{s.t.} \begin{cases} \mathbf{L}\tau = \mathbf{J}_{com}^T \mathbf{F}_{com} + \mathbf{J}_{waist}^T \Gamma_{waist} \\ \quad + \sum_i \mathbf{J}_{t_i}^T \mathbf{W}_{t_i} + \sum_j \mathbf{J}_{c_j}^T \mathbf{F}_{c_j} + \gamma \\ \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \mathbf{W} \leq \begin{bmatrix} \mathbf{W}_{max} \\ -\mathbf{W}_{min} \end{bmatrix} \\ \mathbf{E}_{c_j} \mathbf{F}_{c_j} - \mathbf{d}_{c_j} > 0, \forall j \end{cases} \tag{11}$$

This QP problem is solved at each time step during simulation. The optimized joint torques are obtained from the solution $\hat{\mathbf{S}}$.

VI. RESULTS

The proposed framework has been implemented on a virtual humanoid character executing manipulation tasks with the hands.

The experiment examples are intended to demonstrate that the proposed control system is capable of driving the virtual character to perform a desired manipulation task in an interactive manner, and to change its posture automatically so as to handle unknown external contact forces.

The virtual character's actions are determined based on events from the operator and the simulation environment. The interaction between the character and the operator was realized by observing the captured positions of the operator's hands with regard to the position of the object to manipulate. Currently, the actions of the character are fixed for a given task. We divided the whole manipulation task into five states: *approaching, tightening, manipulating, releasing* and *idle state*, so as to explain the control strategy in detail. We tell the character how to react according to **Algorithm 1**. In the future, we will develop a more general method for intention detection. Hereby, the goal of the experiments is to assess the validity of our control framework, as well as to illustrate the problematics of our future works on the topic.

Algorithm 1 Action decision

```
Input: the distance (in virtual environment) between the operator's hands and the object ($d^{op}_{hand}$) and its threshold value ($\eta$), the operator's hand velocity towards the object to manipulate ($v^{op}_{hand}$), the contact between the character's hand and the object ($c_{hand}$), the desired hand task force ($F^{d}_{hand}$) and its threshold value ($\Psi$).
Output: the actions that the virtual character execute.
  if $d^{op}_{hand} > \eta$ and $v^{op}_{hand} > 0$ and $c_{hand} ==$ false then
    return approaching
  else if $d^{op}_{hand} \leq \eta$ and $c_{hand} ==$ true and $F^{d}_{hand} < \Psi$ then
    return tightening
  else if $d^{op}_{hand} \leq \eta$ and $c_{hand} ==$ true and $F^{d}_{hand} \geq \Psi$ then
    return manipulating
  else if $d^{op}_{hand} > \eta$ and $c_{hand} ==$ true then
    return releasing
  else
    return idle state
  end if
```

Fig. 1 shows the experimental setup. The operator wears motion tracking markers on the hands. Thus, there are only two task objectives, one for each hand. The time step used for simulation is 0.005s. The optimization weights for the different objectives are: 10^4 for CoM, 1000 for waist, 2000 for hand task, 1 for contact force and 10^5 for gravity compensation. These weights values are chosen according to the importance and priorities of different objectives. Additionally, there is no dynamic friction in our simulation environment, and we simulate the Coulomb friction only.

A. Push an object

We made our virtual character move a storage cabinet by pushing it with the hands (Fig. 4).

1) Approaching: At the beginning of the scene, the character's body is upright and its hands are not in contact with the storage cabinet. During the simulation, the character keeps observing the movements of the operator's hands. Once the operator reaches out the hands, this movement is detected by the virtual character which considers it as an order for pushing the storage cabinet. So the character follows the movements of the operator's hands and reaches them out towards the storage cabinet. During this state, we ask the character to move the right foot one step towards the storage cabinet in order to prepare a more robust posture for the pushing task.

2) Tightening: At the moment when the character's hands touch the storage cabinet, new contacts are established between the hands and the surface of the storage cabinet. From then on, the operator's hands may continue their motion as there is no storage cabinet in the real world, while the character has to deal with the interaction between itself and the storage cabinet in virtual environment. During tightening state, as can be seen in Fig. 5a, the contact force between the character's hand and the storage cabinet increases gradually up to a threshold value. After that, the storage cabinet will start to move.

3) Manipulating: As the operator's hands continue reaching out, the character pushes more strongly so that the storage cabinet starts to slip on the ground. During this state, the character's hands keep in contact with the storage cabinet.

4) Releasing: The operator sends out an interruption order for the pushing task by moving back his hands. This movement is detected by the character, which then pushes the storage cabinet less and less strongly until the contacts are invalidated.

5) Idle state: The character's hands follow the movements of the operator's hands.

Fig. 5a illustrates the optimization solution of the right hand task force. Left hand task force takes similar values. During the simulation, the desired task wrench is computed by PD control with $k_p = 400N/m$, $k_d = 40N \cdot s/m$ for position error, and $k_p = 2N \cdot m/rad$, $k_d = 0.1N \cdot m \cdot s/rad$ for orientation error. The values of the parameters of the PD controllers are firstly computed based on the closed loop bandwidth and the sampling frequency, and then they are adjusted through experimentation. The result in Fig. 5a also shows the storage cabinet motion during simulation. The character is able to push storage cabinets which are of different weights and influenced by different friction coefficients. The weight of storage cabinet in Fig. 5a is 20kg and 40kg, and the friction coefficient between the storage cabinet and the ground is 0.1, while the friction coefficient between the foot and the ground is 1.0. The accompanying video demonstrates that the character can also handle a storage cabinet which is heavier and influenced by a larger friction coefficient.

Balance control is achieved by regulating the position of CoM. Fig. 5b depicts the trajectory of CoM and storage cabinet in the sagittal plan during simulation. The actions of the character can be recognized according to the movement of the storage cabinet. We also illustrate the position of each foot to provide a better idea of CoM position with respect to the support polygon. In Fig. 5b, the weight of the storage

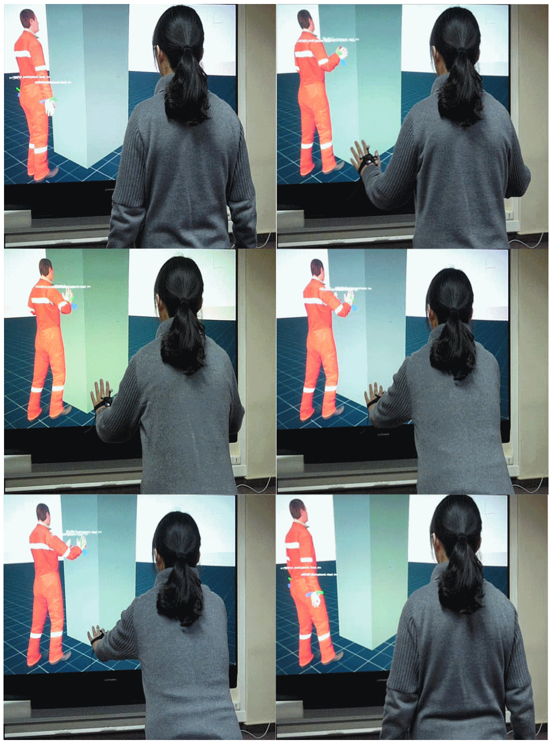
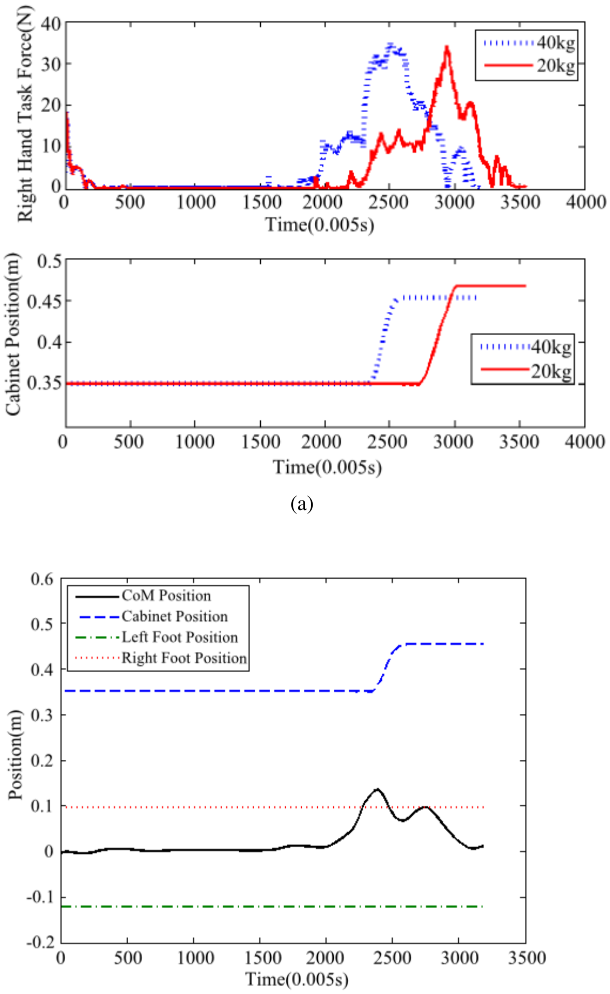


Fig. 4. Snapshots of the virtual character pushing a storage cabinet according to the interaction with the operator.

cabinet is 40kg, and the friction coefficient between storage cabinet and ground is 0.1.

Before and after pushing the storage cabinet, the only contacts between the character and the virtual environment are the contacts with the ground, which is flat. Since such contacts are coplanar, we just set the projection of the desired CoM position at the center of the support polygon.

Once the contacts between the hands and the storage cabinet have been established, the desired CoM position should be adjusted to adapt to the multiple non-coplanar contacts. This goal could be achieved by using the method proposed by C. Collette et al. [3]. Thanks to the knowledge of some physical parameters, such as the weight of the storage cabinet and the friction coefficient, the character will move its CoM to a more suitable position. In our example, the character deliberately moves its CoM towards the storage cabinet in order to generate a robust posture with regard to the new established contacts between hands and storage cabinet. It is visible on Fig. 5b that, during the simulation, the projection of CoM on the ground keeps moving towards the storage cabinet just before the storage cabinet slips, and even went out of the support polygon momentarily.



(a)

(b)

Fig. 5. The results of (a) right hand task force and storage cabinet motion with respect to different storage cabinet weights during simulation and (b) CoM and storage cabinet motion in the sagittal plan.

B. Take and carry an object

In this experiment, the character is asked to take a box up from the table, and manipulate it according to the reference hand motion of the operator (Fig. 6). The box weights 10kg. The friction coefficient between the box and the hand is 1.0.

Thanks to the force control approach, the character is able to maintain and strengthen the contacts between the hands and the box, so as to avoid from dropping the box during the manipulation. This is achieved by setting the offset values of the desired hand task wrenches as in (4), which provides strong contact forces between the hands and the box. The hand contact force offset is always set perpendicular to the

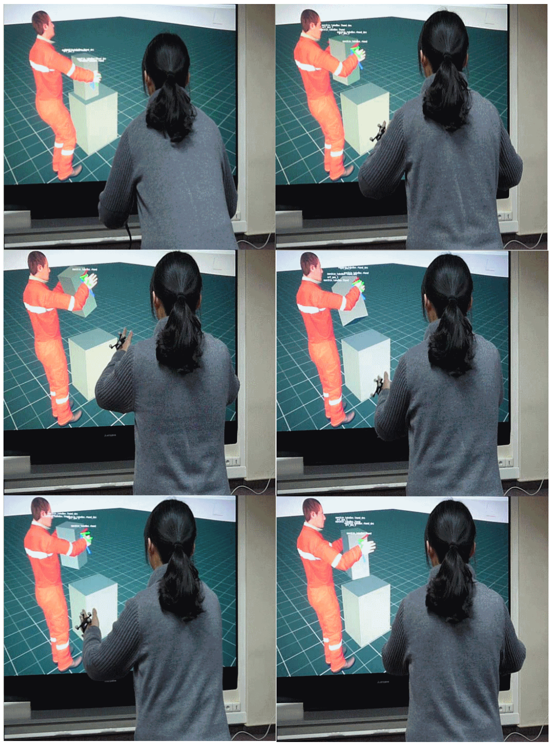


Fig. 6. Snapshots of the character taking a box up, moving it, and putting it down.

contact surface, so as to deal with different box rotation states. The associated video demonstrates the functionality of the force control approach which allows us to adjust the hand contact force during manipulation.

An advantage of the force control based method over the direct marker control method, in which the character is virtually connected to the motion tracking markers via virtual springs, is that we can enhance the ability to control the contact forces during manipulation. By using the direct marker control, it is possible to manipulate an object which is not too heavy. However, to be able to take up heavier objects, the operator has to move his hands much more closer to each other, so as to generate stronger contact forces with the object. As there is a limitation of how close the operator's hands can be to each other, this results in a limit of the weight of the object which the character is able to carry.

VII. CONCLUSIONS AND FUTURE WORKS

This paper has introduced a virtual character control framework based on optimization and force control. The character can perform manipulation tasks in the presence of unknown external wrenches. The control architecture has been used in an interactive manner, which is fast enough to allow the operator to control the character's behavior in real-time. All the tasks, posture and CoM motion objectives are realized thanks to the force control approach. Furthermore, the character is able to adjust its CoM position to generate a robust posture in the presence of non coplanar contacts. The effectiveness of the proposed control framework was demonstrated by the experiments of object manipulation control on a virtual character in simulation. Full animations can be seen in the supplemental video.

In the future, we will work on more complex behaviors such as manipulation during walking. These complex behaviors need motion planning for which we plan to adopt model predictive control [8], [10] in our framework. Moreover, we also plan to implement reinforcement learning techniques [7] for the virtual character, so that it can better understand the intention of the operator during their interaction.

REFERENCES

[1] Y. Abe, M. da Silva, and J. Popović. Multiobjective control with frictional contacts. *In Proc. ACM SIGGRAPH/EG Symposium on Computer Animation*, pp. 249-258, 2007.

[2] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle. Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts. *In Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, USA, May 2008.

[3] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle. Dynamic Balance Control of Humanoids for Multiple Grasps and non Coplanar Frictional Contacts. *Int. Symp. on Visual Computing*, pp. 734-744, 2007

[4] S. Coros, P. Beaudoin and M. van de Panne. Generalized Biped Walking Control. *ACM Transactions on Graphics*, vol. 29, no.3, 2010.

[5] E. Demircan, L. Sentis, V. DeSapio, and O. Khatib. Human motion reconstruction by direct control of marker trajectories. *In Advances in Robot Kinematics (ARK), 11th International Symposium*, Batz-sur-Mer, France, June 2008. Springer.

[6] O. Khatib, L. Sentis, and J. Park. A unified framework for wholebody humanoid robot control with multiple constraints and contacts. *In Springer Tracts in Advanced Robotics - STAR Series*, Prague, Czech Republic, March 2008.

[7] D. Lee, C. Ott, and Y. Nakamura. Mimetic Communication Model with Compliant Physical Contact in Human-Humanoid Interaction. *The International Journal of Robotics Research*, May 2010.

[8] U. Muico, Y. Lee, J. Popović, and Z. Popović. Contact-aware nonlinear control of dynamic characters. ACM Transactions on Graphics (Proc. SIGGRAPH) 28, 3, Article 81, 2009.

[9] J. Pratt, A. Torres, P. Dilworth and G. Pratt. Virtual Actuator Control. *IEEE International Conference on Intelligent Robots and Systems*, 1996.

[10] M. da Silva, Y. Abe, and J. Popović. Simulation of Human Motion Data using Short-Horizon Model-Predictive Control. Computer Graphics Forum, Volume 27 Issue 2, Pages 371-380. 2008.

[11] B.J. Stephens, C.G. Atkeson. Dynamic balance force control for compliant humanoid robots. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[12] J. Wu, Z. Popović. Terrain-adaptive bipedal locomotion control. *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 72:1-72:10, July 2010.