

Interactive dynamic simulator for humanoid with haptic feedback

Jean-Rémy Chardonnet^{1,2}, François Keith², Abderrahmane Kheddar^{1,2}, Kazuhito Yokoi² and François Pierrot¹

¹ CNRS Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, France

² AIST/CNRS Joint Japanese-French Robotics Laboratory, Japan

Abstract. In this paper we present an interactive dynamic simulator for humanoid. In our software, the user can interact directly with the robot through a haptic probe and meanwhile feed back with reaction forces subsequent to its actions. The simulator allows also creation and manipulation of virtual objects by the robot, by the human or in a joint collaborative way. This dynamic simulator uses fast computation dynamics and constraint-based methods with friction. It is part of a general framework that is being devised for general prototyping and collaborative scenario studies with haptic feedback. The different ingredients of the simulator are presented with an simulation scenario.

1 Introduction

Simulation is a research topic that is being more and more developed since many years as it has many applications in various fields such as robotics, computer games, virtual reality, medical and chemical science, etc. Dynamics simulation is needed to validate physical models, and adding interaction by means of haptic behaviours highly increases realism and allows performing manipulation or collaborative tasks. But interacting with the environment supposes to deals with problems of modeling properly contact, friction, impact and deformation and it still remains a challenging research.

In this paper we present an interactive dynamic simulator for humanoids with force feedback and solving contact forces with friction using constraint-based methods. Our goal is to provide high quality and fast simulations. We present a new computation method of the operationnal space inertia matrix as well as a simulation scenario. We integrated our simulator in a general framework that is being developed for general prototyping and collaborative scenario purposes.

2 Previous work

Concerning the resolution of contact forces with friction, many proposed dynamics simulators use penalty-based methods, like in Yamane and Nakamura (2006). Even if these methods are easy and fast to implement, they require parameters tuning which

can affect robustness of the simulation. An alternative to penalty-based methods is constraint-based methods, in which contact constraints are integrated to the dynamics. Constrained dynamics is generally expressed as a linear expression (Baraff (1994)). These methods are more and more used as accuracy is highly increased, but adding friction adds non linearity, requiring friction cones to be discretized (Anitescu and Potra (1996)). However this discretization can be avoided by using iterative algorithms (Liu and Wang (2005)). Renouf and Acary (2006) made a comparison between different methods for solving constraint-based contact forces. These methods are implemented in a dedicated software SICONOS¹.

Many interactive dynamic simulators have been presented. Son et al. (2000) proposed a general framework for interactive dynamic simulation. They included haptic interaction. However they do not explain how they take into account contact forces and their main application is for a fixed-base manipulator. Ruspini and Khatib (1999) presented an impressive framework for interactive dynamic simulation with contact using constraint-based methods but it is not clear how friction is integrated. Duriez et al. (2006) showed interactive simulation for deformable objects using constraint-based methods and solving contact with friction using iterative algorithms. Chardonnet et al. (2006) proposed a fast dynamic simulator for humanoids, based on Ruspini and Khatib's work and taking into account correct Coulomb's friction, meaning without discretizing the friction cones. They however did not included any haptic device. We are using their work as the basis of the proposed simulator. There are few dynamic softwares simulating complex systems such as humanoid robots, taking into account good contact forces and in which the user can interact directly with all the elements in the environment.

3 Constraint-based method architecture

3.1 Overview

The computation of the contact forces follows Chardonnet et al. (2006)'s approach: first, we compute the free acceleration of the bodies using Featherstone (1987)'s algorithm, then the acceleration due to contact forces. The total joint acceleration $\ddot{\mathbf{q}}$ is described by the following equation:

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{\text{free}} + \ddot{\mathbf{q}}_c = \mathbf{A}^{-1}(\mathbf{q})[\mathbf{\Gamma} - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})] + \mathbf{A}^{-1}(\mathbf{q})\mathbf{J}_c^T \mathbf{f}_c \quad (3.1)$$

or, in the operational space

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_{\text{free}} + \ddot{\mathbf{x}}_c = \mathbf{J}\mathbf{A}^{-1}(\mathbf{q})[\mathbf{\Gamma} - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})] + \mathbf{\Lambda}_c^{-1} \mathbf{f}_c \quad (3.2)$$

where \mathbf{A} is the inertia matrix of whole robot, $\mathbf{\Gamma}$ the vector of joint torques, \mathbf{b} the centrifugal and Coriolis effects, \mathbf{g} the gravity, \mathbf{f}_c the contact forces, \mathbf{J} is the Jacobian matrix and $\mathbf{\Lambda}_c^{-1}$ the inverse of the operational space inertia matrix introduced by Khatib (1987).

In order to compute the acceleration due to contact, we calculate the forces corresponding to each contact point using an iterative Gauss-Seidel approach that has been

¹<http://siconos.gforge.inria.fr>

recently applied to robotics by Liu and Wang (2005), combined with a Newton-Coulomb method (Jean (1993)). This requires to find Λ_c^{-1} that we will note Λ^{-1} from now on.

For this purpose, Chang and Khatib (2000) defined $\Omega_{i,j}$, the 6×6 inertia matrix that relates the spatial acceleration of link j and the force acting on link i :

$$\ddot{\mathbf{x}}_j = \begin{pmatrix} a_j \\ \alpha_j \end{pmatrix} = \Omega_{i,j} \mathbf{f}_i \quad (3.3)$$

The inverse of the operational space inertia matrix, Λ_{e_i,e_j}^{-1} can be related to $\Omega_{i,j}$ by:

$$\Lambda_{e_i,e_j}^{-1} = {}^i \mathbf{X} \Omega_{i,j} {}^j \mathbf{X}^T \quad (3.4)$$

where ${}^i \mathbf{X}$ relates the frame change from i to j and end-effector frames e_i and e_j are at the tips of links i and j .

Chang and Khatib's computation of Ω is divided into three loops: the first one computes additional data, the second one goes from the root to the contacting bodies to compute the diagonal part of Ω and the last one starts from the nearest common mother of all contacting bodies to compute the off-diagonal parts $\Omega_{i,j,i \neq j}$.

To avoid computing new data and searching the common mother, we propose a different method: we compute again the second and the third recursions of Featherstone's algorithm for each contacting body, but without gravity ($\mathbf{g} = 0$), torques ($\mathbf{\Gamma} = 0$) or joint velocity ($\dot{\mathbf{q}} = 0$); thus the free joint acceleration ($\ddot{\mathbf{q}}_{\mathbf{f}=0}$) becomes null. Besides, we use the data already computed during the calculation of the free acceleration. $\Omega_{i,j}$ is computed by applying six times Featherstone's algorithm:

$$\Omega_{i,j} = ({}_0 \ddot{\mathbf{x}}_j \dots {}_5 \ddot{\mathbf{x}}_j) \quad (3.5)$$

where ${}_u \ddot{\mathbf{x}}_j$ is the spatial acceleration of body j associated to the spatial unit force ${}_u \mathbf{f}_i$ applied on body i with ${}_u \mathbf{f}_i[y] = 1$ if $u = y$, else 0, and $y = \{0, 1, 2, 3, 4, 5\}$.

3.2 Implementation

The size of Λ^{-1} depends on the number of contact points m : if we handle friction, it becomes a $3m \times 3m$ matrix. This matrix may have some null elements if two colliding bodies do not interact (for exemple they are on the floor but they are not in contact together). In order to have a full rank matrix, we sort the colliding bodies into collision groups, each of them having its own matrix Λ^{-1} . Two bodies belong to the same collision group if there is a sequence of unfixed bodies in collision that connect them. This allows to reduce the size of the Λ^{-1} matrices and thus reduce their computational cost and so accelerate the computation of the forces using Gauss-Seidel like algorithm. The overall flow chart for one step computation is presented in Figure 1.

3.3 Algorithm complexity

Computation of the operational space matrix is time-consuming, so different methods have been proposed to reduce its cost. We propose a new method that is faster. The following arrays compares the number of mathematical operations required to compute

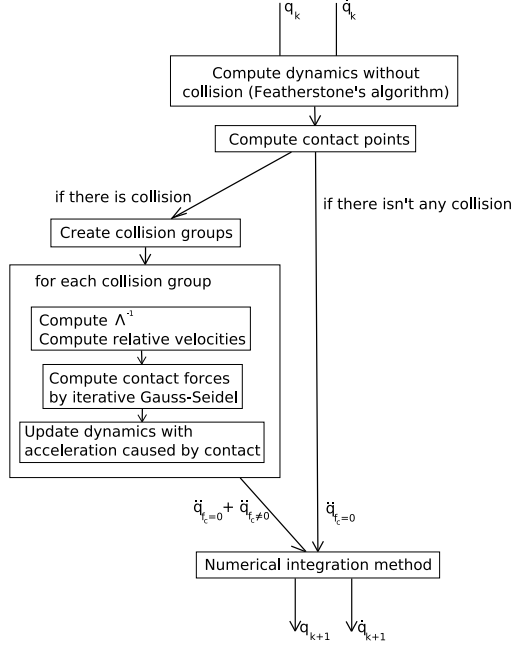


Figure 1. Algorithm of the dynamic model with collision.

Λ^{-1} for a n -dof multibody with m contact points. We note C the number of contacting bodies of the multibody, b the number of bodies that are between the base and a contacting body and $d_{(i,j)}$ the number of joints between bodies i and j ($d_{(i,j)} = d_{(j,i)}$).

Chardonnet et al. (Table 1) Chardonnet et al.'s method computes Λ^{-1} by applying the second and the third recursions of Featherstone's algorithm three times for each contact point. The complexity is thus in $\mathcal{O}(m^2 + mn)$.

Table 1. Operations required to compute Λ^{-1} using Chardonnet et al.'s method.

$\times \div$	$144m$	$+39bm$	$+36\beta_m$	$+27m \frac{m+1}{2}$
$+ -$	$99m$	$+36bm$	$+21\beta_m$	$+27m \frac{m+1}{2}$

where $\beta_m = \sum_{i \in m} d_{i,0}$, $\beta_m < nm$.

Chang and Khatib (Table 2) The algorithm proposed by Chang and Khatib computes Λ^{-1} in two steps. First it computes Ω , then projects it into the contact space. The complexity of the first step depends on the number of contacting bodies, in the worst case (all bodies contacting) it is in $\mathcal{O}(n^2)$. The second step is always in $\mathcal{O}(m^2 + mn)$.

This method has a lower complexity than Chardonnet's algorithm, but the computa-

tion of Ω is more complex to implement.

Table 2. Operations required to compute Λ^{-1} using Chang and Khatib’s method.

$\times \div$	286	+153b	+72 α_C	+54Cm + 27m $\frac{m+1}{2}$
+ -	235	+179b	+66 α_C	+54Cm + 27m $\frac{m+1}{2}$

where $\alpha_C = \sum_{i,j \in C \mid j > i} d_{i,j}$.

New method (Table 3) Our new method presents the same steps as Chang and Khatib’s one. The computation of Ω however changes. It remains in $\mathcal{O}(n^2)$ in the worst case, but is easier to implement. Besides, compared to Chardonnet et al.’s method, this method has a lower complexity in $\mathcal{O}(mn)$.

Table 3. Operations required to compute Λ^{-1} using the proposed method.

$\times \div$	252C	+78Cb	+72 β_C	+54Cm + 27m $\frac{m+1}{2}$
+ -	180C	+72Cb	+42 β_C	+54Cm + 27m $\frac{m+1}{2}$

with $\beta_C = \sum_{i \in C} d_{i,0}$, $\beta_C < n\frac{n+1}{2}$

The m^2 coefficient is the same for the three methods. Thus we will prefer methods that reduce the cost in mn .

3.4 Simulation results

We made some performance tests to compare each method. We took a 30 degree-of-freedom humanoid robot (HRP-2), standing on its two feet, each of them having four contact points with the ground. The processor used is a bi-AMDTM64 2.5Ghz, and the time step is 1ms. We can see that our new method is faster than the other two ones.

Table 4. Comparison of the computation time of each method.

	Chardonnet et al.	Chang	New method
Computation of Ω (μs)	\emptyset	11.171	8.219
Total computation of Λ^{-1} (μs)	85.688	15.8570	12.875

4 Haptic interaction

Interaction with the virtual environment allows for example the realization of collaborative manipulation tasks with force feedback or testing external force perturbation on a walking pattern or on a given implementation of a task controller. Since our simulator is centered on haptic interaction, we need to integrate a haptic device. For our purpose we

interfaced the PHANTOM©OmniTM device commercialized by SensAble Technologies². This device has six degrees of freedom with a three degree-of-freedom force feedback.

We need to consider two different ways of interacting: simply touching and dragging. In both cases, interaction will add an external force \mathbf{f}_e to the dynamics of the objects:

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}(\mathbf{q})[\mathbf{\Gamma} - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})] + \mathbf{A}^{-1}(\mathbf{q})\mathbf{J}_c^T \mathbf{f}_c + \mathbf{A}^{-1}(\mathbf{q})\mathbf{J}_e^T \mathbf{f}_e \quad (4.1)$$

When touching an object, an external force will be applied at the contact point between the object and the virtual tip of the haptic device. This force is given directly by the device via the Sensable OpenHapticsTM Toolkit, and weighted with an arbitrary chosen coefficient. The contact point is obtained via the collision detection algorithm provided by this library.

Dragging objects is useful for tasks like pick-and-place or collaborative tasks. As the user moves the device, the object must behave accordingly while the user feels its weight. To achieve this, the most common way used in interactive simulations is to model a virtual spring-damper system between the device and the object, that is:

$$\mathbf{f}_e = k_p(\mathbf{x}_{\text{device}} - \mathbf{x}_{\text{contact}}) + k_v(\dot{\mathbf{x}}_{\text{device}} - \dot{\mathbf{x}}_{\text{contact}}) \quad (4.2)$$

where $\mathbf{x}_{\text{device}}$ and $\mathbf{x}_{\text{contact}}$ are the position/orientation of the haptic device and the one of the end-effector respectively. Although this method is very fast and easy to implement, the most difficult part is the choice of k_p and k_v . k_v is chosen so that $k_v = \sqrt{2mk_p}$, with m the mass of the object.

5 Simulation scenarios and results

5.1 Computation time of $\mathbf{\Lambda}^{-1}$

In order to highlight the advantage of sorting the colliding bodies into multiple collision groups, we realized the following simulations. We took $k = [1, 30]$ cubes contacting only the floor, each cube having $m = 4$ contact points. For a fixed k , we compared the following cases: 1) all cubes belong to the same collision group; 2) each cube is a collision group. In the first case, the size of $\mathbf{\Lambda}^{-1}$ is $mk \times mk$, meaning computation time will be in $(mk)^2$, whereas in the second case, the size of each $\mathbf{\Lambda}^{-1}$ is $m \times m$, meaning computation time will be in km^2 . The results are depicted in Figure 2. This preliminary sorting allows a more efficient calculation of contact forces.

5.2 Simulation scenario

For the following scenario, we use the 30-dof HRP-2 robot built by Kawada Industries. We placed a table on which there is an object to be handled by the robot. The HRP-2 robot grasps it and handles it. Once it has taken the object, we pick it using the haptic device and play with the robot by pulling/pushing in all directions. This is somehow a very simple collaborative task. The screenshots are shown in Figure 3. For clarity, we did not display contact forces. The thin orange bar in Figure 3 represents the virtual tip of the haptic device.

²<http://www.sensable.com>

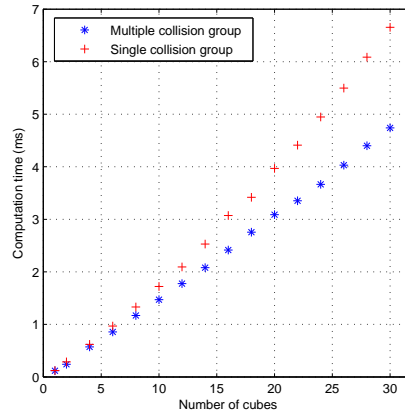


Figure 2. Computation time regarding the number of cubes.



Figure 3. Screenshots of HRP-2 handling an object and being manipulated by a user.

In this simulation once the robot has taken the object there are 40 contact points. As simulations slow down when the number of contact points increases, one step takes around 5.2ms, thus, not allowing real-time interaction. However, we are working on improving this aspect by optimizing the number of contact points and computing the resulting forces per body rather than forces per contact point.

6 Conclusion and future work

We presented a new interactive dynamics simulator for humanoids with haptic feedback. This simulator handles fast computation of contact forces with friction using constraint-based method and keeping true friction cones that guarantee accurate results. Adding a haptic module allows the user to interact with objects in the environment with force feedback and perform simple tasks, such as collaborative tasks. We showed a new method to compute the operational space inertia matrix that reduces computation time of contact forces. In future works, we plan to extend the simulation and the computation of contact forces to deformable skin by adding flexibility to the joints and to the bodies, so that

we will be able to simulate realistically human-shaped avatars such as androids. We also need to make some experimental validation of our simulator by comparing the data measured by forces sensors and the data computed in simulation.

Bibliography

- M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Reports on computational mathematics*, 03, October 1996.
- David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH*, pages 23–34, Orlando, July 24–29 1994.
- Kyong-Sok Chang and Oussama Khatib. Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms. In *IEEE International Conference on Robotics and Automation*, pages 850–856, San Francisco, CA, April 2000.
- J-R. Chardonnet, S. Miossec, A. Kheddar, H. Arisumi, H. Hirukawa, F. Pierrot, and K. Yokoi. Dynamic simulator for humanoids using constraint-based method with static friction. In *IEEE International Conference on Robotics and Biomimetics*, pages 1366–1371, Kunming, China, December 17–20 2006.
- Christian Duriez, Frédéric Dubois, Abderrahmane Kheddar, and Claude Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):36–47, January–February 2006.
- Roy Featherstone. *Robot dynamics algorithms*. Kluwer Academic Publishers, 1987.
- Michel Jean. Numerical methods for three dimensional dynamical problems. In M.H. Aliabadi and C.A. Brebbia, editors, *Conference Contact Mechanics, Southampton*, page 71, July 1993.
- Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational sapce formulation. *IEEE Journal of Robotics and Automation*, 3(1): 43–53, February 1987.
- Tong Liu and Michael Yu Wang. Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping and gauss-seidel methods. *IEEE Transactions on Automation Science and Engineering*, 2(2):19–31, January 2005.
- M. Renouf and V. Acary. Comparison and coupling of algorithms for collisions, contact and friction in rigid multi-body simulations. In *European Conference on Computational Mechanics*, Lisbon, Portugal, June 2006.
- Diego C. Ruspini and Oussama Khatib. Collision/contact models for dynamics simulation and haptic interaction. In *International Symposium of Robotics Research*, pages 185–195, Snowbird, Utah, October 9–12 1999.
- Wookho Son, Kyunghwan Kim, Nancy M. Amato, and Jeffrey C. Trinkle. Interactive dynamic simulation using haptic interaction. In *IEEE International Conference on Intelligent Robots and Systems*, pages 145–150, November 2000.
- Katsu Yamane and Yoshihiko Nakamura. Stable penalty-based model of frictional contacts. In *IEEE International Conference on Robotics and Automation*, pages 1904–1909, Orlando, Florida, May 2006.