



**HAL**  
open science

# Dancing Humanoid Robots: Systematic use of OSID to Compute Dynamically Consistent Movements Following a Motion Capture Pattern

Oscar Efrain Ramos Ponce, Nicolas Mansard, Olivier Stasse, Christophe Benazeth, Sovannara Hak, Layale Saab

► **To cite this version:**

Oscar Efrain Ramos Ponce, Nicolas Mansard, Olivier Stasse, Christophe Benazeth, Sovannara Hak, et al.. Dancing Humanoid Robots: Systematic use of OSID to Compute Dynamically Consistent Movements Following a Motion Capture Pattern. IEEE Robotics and Automation Magazine, 2015, 22 (4), pp.16 - 26. 10.1109/MRA.2015.2415048 . hal-01222238

**HAL Id: hal-01222238**

**<https://hal.science/hal-01222238>**

Submitted on 29 Oct 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Dancing Humanoid Robots: Systematic use of OSID to Compute Dynamically Consistent Movements Following a Motion Capture Pattern

Oscar E. Ramos, Nicolas Mansard, Olivier Stasse, Christophe Benazeth, Sovannara Hak, Layale Saab  
CNRS, LAAS, 7 av. du Colonel Roche, F-31400 Toulouse, France  
Université de Toulouse, LAAS, F-31400 Toulouse, France

**Abstract**—In October 2012, the humanoid robot HRP-2 was presented during a live demonstration performing fine-balanced dance movements with a human performer in front of more than 1000 people. This success was possible by the systematic use of operational-space inverse dynamics to compute dynamically consistent movements following a motion capture pattern demonstrated by a human choreographer. The first goal of this article is to give an overview of the efficient inverse-dynamics method used to generate the dance motion. Behind the methodological description, the second and main goal of the article is to present the robot dance as the first successful real-size implementation of inverse dynamics for humanoid-robot movement generation. This gives a proof of concept of the interest of inverse dynamics, which is more expressive than inverse kinematics and more computationally tractable than model-predictive control. It is, in our opinion, the topical method of choice for humanoid whole-body movement generation. The real-size demonstration also gave us some insight of nowadays methodological limits and the consequent future needed developments.

## I. INTRODUCTION

Motion generation for humanoid robots is a challenging task involving coordination, control and stabilization of the balance. Coordination problems are inherent to kinematically redundant robots like humanoids, which are highly redundant systems, and control difficulties arise from the complex humanoid tree-like structure as well as its unstable vertical position. **Inverse kinematics (IK)** is the most used technique to generate whole-body movements [8], but it is poorly able to deal with dynamic constraints. This lack of expressivity limits the movements that can be generated, such as multi-contact movements. However, it is well understood, very easy to implement, and computationally efficient: it can be considered as the current state of the art. **Operational-space inverse dynamics (OSID)** can be seen as an evolution of IK that can deal with most of these limitations: the extended expressivity makes the formulation of more complex constraints possible, and faster motions as well as movements with important momentum changes can be generated [13], [4], [5]. This comes with a higher computational cost. Several teams are studying the related methods but few robotic results have yet been demonstrated on complete humanoids or other complex robots. Both IK and OSID are instantaneous linearizations of the motion problem, where the non-linearity coming from the temporal evolution is neglected, leading to obvious limitations but keeping a reasonable computational cost. Numerical optimal



Fig. 1: Humanoid robot HRP-2 and the hip-hop dancer

control (also called trajectory optimization or trajectory filtering) [28], [15], is able to consider the full temporal evolution of the system inside a non linear resolution scheme, making the expressivity capabilities of the motion generator much higher. When the optimal trajectory is updated in real-time with sensor measurements, it is referred to as **model predictive control (MPC)**. Despite several recent and promising results, the underlying computational cost for whole-body motion generation currently makes this approach unfeasible for a real-time implementation and generic solvers tend to get stuck into local minima or to return trivial solutions.

While IK has poor capabilities in terms of handling the constraints due to the dynamics of the motion, MPC has yet a prohibitive cost and lacks a sufficient understanding to deploy beyond some expert-designed demonstrations. Compared to these two solutions, OSID is a very mature trade-off that provides an affordable computational cost as well as good capabilities to handle the major constraints of complex dynamic robots such as a humanoids. In this article, we first present a global overview of the OSID method that we developed during the last five years. This overview covers a sound summary of several previous technical papers [24], [14], [6] where more implementation details are given. Taken together, this methodology produces the best OSID performances of the community, providing a fast, real-time, hierarchical, inequality-prone solver. Beyond this methodological introduction, the objective of this article is to

demonstrate the maturity of the OSID approach to generate whole-body movements by reporting a complete proof of concept realized with the humanoid robot HRP-2: we have generated a 10-minute sequence of dance movements based on human motion captured trajectories for a live demonstration (see Fig. 1). Such complex movements would have been very difficult to generate with only IK, but it was fast and efficient to generate them with OSID. We believe that this proof of concept can convince that currently OSID has only advantages with respect to IK and that it is time for it to become the standard solution for complex robot motion generation.

The article first briefly describes the theoretical solver used to generate the movement. Then the technical generation of the dance sequence emphasizing the most important experimental results is described. Finally, the insights that our team gained during the realization of this proof of concept are discussed, as well as the consequences that are foreseen for future research in whole-body motion generation.

## II. DYNAMIC STACK OF TASKS

The idea of OSID is to define the motion to be executed by the robot in terms of reference movements in properly-chosen *operational* spaces [12] (sometimes called *task* spaces [26]), typically with a smaller dimension than that of the robot state space. The forward link between the state space and the operational space is given by the so-called task function, whose image maps the operational space. This function is chosen so that the movement is easier to express in the operational space than in the state space. For example, grasping a ball can be defined in the space of the distance between the hand and the ball, while looking at an object can be defined in the visual space. The stake of OSID is to back-project the reference operational motion to the state space in order to obtain a fully coordinated whole-body movement.

When several tasks have to be simultaneously performed, the motion solver should take care of the possible conflicts between them. On a humanoid robot, it is widely accepted that enforcing a hierarchy between the various objectives is the key to a safe behavior. For example, balance will always be considered a top-priority objective, while other objectives such as visibility and posture are accomplished only if possible but without disturbing the balance (or the possible manipulation or locomotion tasks). We refer to this hierarchy of motion objectives as the *stack of tasks* (SoT). In this section, we describe the key components of our current implementation that are able to handle a hierarchy of inequality objectives (Section II-B) while ensuring the dynamic consistency (Section II-C) with real-time performances (Section II-D).

### A. Task Function Approach

Let  $\mathcal{C}$  be the configuration space,  $\mathcal{M}$  the task space,  $T_e\mathcal{M}$  the tangent space of  $\mathcal{M}$  at some point  $e$ , and  $\mathcal{U}$  an arbitrary control space linearly linked to the configuration tangent space  $T_q\mathcal{C}$ . The definition of a task comprises three components: the task function  $e : \mathcal{C} \rightarrow \mathcal{M}$ , the reference

behavior given by a vector field of  $\mathcal{M}$ ,  $\dot{e}^* \in T_e\mathcal{M}$ , and the differential map  $G : \mathcal{U} \rightarrow T_e\mathcal{M}$  relating the task to the control input. This map defines the direct relation between the reference vector field and the control space as  $\dot{e} + \delta = Gu$ , where  $u$  is the control signal and  $\delta$  is the task drift. The control law  $u^*$  meeting the reference behavior is obtained as  $u^* = G^\#(\dot{e}^* + \delta)$ , where  $\{\cdot\}^\#$  is any reflexive generalized inverse. Among the possible inverses, the Moore-Penrose pseudoinverse  $\{\cdot\}^+$  is generally chosen, but other inverses (such as weighted inverses) can also be used.

This generic formulation directly corresponds to IK, with  $\mathcal{U} = T_q\mathcal{C}$ ,  $u = \dot{q}$ , and  $G = \frac{\partial e}{\partial q}$  the task function tangent application. The formulation also covers inverse dynamics [12]: the reference behavior is the expected task acceleration  $\ddot{e}^* \in T_e\mathcal{M}$ , and the control input is the joint torques vector  $\tau \in \mathcal{U}$ . In this case, the approach consists in finding the desired torque control input  $\tau^*$  that generates the reference behavior  $\ddot{e}^*$ , the joint acceleration  $\ddot{q}$  being a side variable that might not need to be explicitly computed [24].

### B. Hierarchical Quadratic Program Solver

Finding the control that satisfies instantaneous constraints is a quadratic problem (QP) whose least-norm solution is given by the pseudoinverse. The redundancy of the robot with respect to the task is the linear set of controls that do not cause any change in the task spaces, and is given by the null-space of the tangent application  $G$ . A hierarchy is typically obtained by projecting the secondary objectives onto the null spaces of the higher-priority objectives. Such a solution is well-known for handling tasks written as equalities (typically, reaching or pointing tasks).

But robot control also needs to handle tasks like joint angle, speed and power limits, auto-collisions, obstacle avoidance, singularity avoidance, balance constraints, or visibility of landmarks in the field of view or behind occlusion. All of these tasks are inequalities. Thus, inequalities represent important tasks and need to be considered. Some solutions regarding inequalities are potential functions [11], damping functions [3] or clamping [23] for joint angle limits. The problem with these methods is their inability to deal with inequality constraints at arbitrary levels of the hierarchy.

Consider a generic linear system with  $n$  linear equalities or inequalities of the form  $\underline{b}_i \leq C_i x \leq \bar{b}_i$ , where  $x$  is the optimization variable,  $C_i$  is a coefficient matrix,  $\underline{b}_i$ ,  $\bar{b}_i$  are the lower and upper bounds respectively, and  $i = 1, \dots, n$  represents the priority in the hierarchy<sup>1</sup>. The lexicographic order  $(i) \prec (ii) \prec (iii) \prec \dots$ , where  $(i)$  has the highest priority, is used to represent the hierarchy. When considering only one objective, the optimum  $x$  satisfying at best the objective is simply obtained by solving the associated QP. For a hierarchy of multiple objectives, the system is written as a hierarchical quadratic problem (HQP) so that if level  $i$  has higher priority over level  $i + 1$ , then level  $i + 1$  should be fulfilled as close as possible but without interfering with level  $i$  and the other higher priority levels.

<sup>1</sup>With this representation equalities are a special case when  $\underline{b}_i = \bar{b}_i$  and single-sided inequalities are special cases when  $\underline{b}_i = -\infty$  or  $\bar{b}_i = +\infty$ .

Our team has shown that the optimum of an HQP can be defined by a cascade of QP's [10]: at each level  $i$  of the hierarchy, the QP gives the solution that satisfies at best level  $i$  under the constraint to preserve the optimum found for the  $i - 1$  first levels. The method is more a formal definition than an efficient algorithm: if the cascade can be computed using iterative calls to a QP solver, such an implementation is computationally inefficient. In [6], we have proposed an HQP solver based on a variation of one classical QP resolution algorithm, the primal active-set solver. First, a classical primal active set algorithm finds the active inequality constraints and turns them into equalities; then, a hierarchized complete orthogonal decomposition is computed to find the optimum of the partial equality problem. The two steps are iterated until convergence.

### C. Stack of Tasks using Inverse Dynamics

One of the major interests of OSID is to handle limited-actuation systems. Humanoid robots (and other floating-based systems) are a particular case of them due to their partial actuation that is completed by contacting with the environment. The configuration of a humanoid robot is represented by *generalized coordinates*  $q = (x_b, q_a)$ , where  $x_b$  represents the position and orientation of the robot free-floating body, and  $q_a$  represents the  $n$  actuated joints of the robot. When the robot is in contact with the environment, the dynamic equation of the system is

$$M\ddot{q} + b + J_c^T f_c = S^T \tau \quad (1)$$

where  $M$  is the generalized inertia matrix,  $b$  is the dynamic drift vector including Coriolis, centrifugal and gravity forces,  $f_c$  is the vector of the 3D contact forces applied at the contact points  $x_c$ ,  $J_c = \frac{\partial x_c}{\partial q}$  is the Jacobian of those contact points,  $\tau$  is the actuated torque vector, and  $S = [0 \ I]$  is a matrix selecting the actuated joints. Let  $f_c^\perp$  and  $x_c^\perp$  be the vectors containing only the components of  $f_c$  and  $x_c$  normal to the contact surface. The complementarity conditions to avoid interpenetration (e.g. between the foot and the ground) are  $\ddot{x}_c^\perp \geq 0$  and  $f_c^\perp \geq 0$ , with  $\ddot{x}_c^\perp f_c^\perp = 0$ . The case that guarantees contact is:

$$f_c^\perp \geq 0 \quad (2)$$

and  $\ddot{x}_c^\perp = 0$ , or equivalently (since  $\dot{x}_c = J_c \dot{q}$ )

$$J_c \ddot{q} + \dot{J}_c \dot{q} = 0. \quad (3)$$

In this work we assumed that large friction coefficients are present and, thus, only the normal force constraint is used instead of more general friction cones. The generalization to friction cones is straightforward [4] but computationally expensive and should be used if the sliding components are not negligible.

At the acceleration level needed for the dynamic control, the operational constraints are expressed as a relation between the task acceleration and the joint acceleration:

$$\ddot{e}_i = \dot{J}_i \dot{q} + J_i \ddot{q} \quad (4)$$

where  $e_i$  is the task function, and  $J_i = \frac{\partial e_i}{\partial q}$  is the Jacobian of the  $i^{th}$  task. The OSID problem is reduced to finding the variables  $(\ddot{q}, \tau, f_c)$  that are consistent with the dynamic equations and that minimize the distance to the task reference. With the lexicographic order introduced in Sec. II-B, the dynamic SoT based on the HQP is (1)  $\prec$  (2)  $\prec$  (3)  $\prec$  (4-1)  $\prec \dots \prec$  (4- $n_t$ ), where  $n_t$  tasks of the form (4) have been considered. More details can be found in [24] and the implementation used is freely available<sup>2</sup>.

The computation of these variables in the optimization solver has the advantage that forces are obtained in a straightforward way and no consistency verification or projection is necessary to guarantee their feasibility. Explicit constraints on any of the variables can then be formulated, and either a torque-controlled or a position-controlled (by integrating the acceleration) robot can be used. However, since the computation cost of the HQP solver scales with the cube of the number of variables, the explicit formulation requires some care to preserve a good computational efficiency.

### D. Decoupling Motion and Actuation

For a faster computation, the optimization variables can be decoupled. In fact, the space given by  $(\ddot{q}, \tau, f_c)$  in the previous HQP can be divided in three subspaces. The first one is the motion space, where joint accelerations  $(\ddot{q})$  can be freely chosen and the corresponding forces  $(f_c)$  and torques  $(\tau)$  are accordingly set. The second is the actuation space, where the acceleration is fixed and only forces can be freely chosen, as forces and torques are related. The third space is useless since motion variables can be theoretically chosen, but the resulting forces are impractical.

To explicitly distinguish the actuation space from the motion space, two decoupled spaces can be introduced. To this end, an automatic formulation has been proposed in [14]. This formulation states that instead of using the original optimization variables, the bases of the two decoupled spaces are used and the dynamic model as well as the tasks are reformulated in terms of these decoupled variables. This allows for a faster computation since the decoupled spaces present a lower dimension than the original coupled variables. The reduction of the dimensionality while keeping the capabilities of the dedicated HQP solver is numerically stable making a fast resolution possible. Considering 30 actuated joints and 6 free-floating degrees of freedom, with 4 to 8 contact points (which gives a total of 36 joint accelerations, 30 joint torques and 12 to 24 forces), the decoupled control scheme for the whole robot dynamics with a complex set of equality and inequality constraints using a 2.9 GHz desktop computer (iCore 5 mono thread) takes around 4ms per control cycle, achieving real time at 200 Hz. A similar computer is available on-board HRP-2.

### E. Balance of the Robot

OSID enforces actuation constraints, in particular contacts, which can not be obtained with IK. However, this is not

<sup>2</sup><https://github.com/stack-of-tasks>

equivalent to enforcing robot balance, as it can lead the robot to fall (i.e. moving towards the ground without possibility to prevent the descent) while keeping the feet flat in contact. The balance constraint can only be comprehended through a time horizon [27] and is therefore not a direct output of the OSID. In this section, we quickly discuss the balance aspect to offer an intuition of what OSID brings in addition to IK and what it lacks of with respect to MPC.

Two aspects of the robot dynamics have to be considered to maintain balance: the linear momentum variation given by the Center of Mass (CoM) acceleration, and the angular momentum variation. Their control is achieved through the selection of proper contact forces. To keep the presentation simple, we focus on the case of flat non-sliding contacts (e.g. feet on a horizontal concrete floor). In that case, (2) projects down to the constraint that the Zero Moment Point (ZMP) should stay within the support polygon. The ZMP, neglecting the vertical CoM with respect to the acceleration of gravity  $g$ , is completely defined by [9]:

$$z^x = c^x - \frac{1}{\omega^2} \ddot{c}^x - \frac{\dot{\sigma}^y}{mg} \quad (5)$$

where  $z$  is the ZMP,  $c$  the CoM,  $\sigma$  the angular momentum,  $\omega = \sqrt{\frac{g}{h}}$  the frequency of the equivalent inverted pendulum,  $h$  the CoM altitude, and  $m$  the mass. A similar equation holds for the  $y$  coordinate. This intuitively shows that the ZMP is the actuator of two different phenomena: the CoM acceleration  $\ddot{c}$  and the variations of the angular momentum  $\dot{\sigma}$ . Since the OSID approach considers an instantaneous linearization of the system dynamics, there is very little it can do to avoid any overshoot due to a large CoM acceleration  $\ddot{c}$ . Typically, if the robot has to quickly stop because the CoM will leave the support polygon, it is in general too late for the system to react. In that case, the only solution is to perform a step in order to capture the CoM later with a different support polygon [18]. To overcome this limitation, the solution in this article is classical: to use a CoM walking pattern generator (WPG) to preview the effect of future CoM movements.

On the other hand, OSID successfully handles dynamic overshoots due to large angular momentum variations  $\dot{\sigma}$ , which arise even with limited movements of the CoM; for example when the arms move quickly or when the chest bends. In these situations, an IK solver would not be able to preserve robot balance, and the robot would tip over.

In conclusion, OSID is very suitable when a WPG is available to handle the linear momentum. It then handles the angular momentum and therefore guarantees robot balance despite any large manipulation or postural motion, which is more difficult to do with only IK. In the following experiments, a WPG is used to handle the walking phase, while the angular momentum due to large arm, flying foot and chest movements is handled by the OSID.

### III. FROM MOTION CAPTURE TO ROBOT MOTION

The dynamic SoT is a generic motion generator that can be used to execute any robot motion defined as a sequence

of tasks. We have demonstrated its use for sitting in an armchair [24], predicting human behavior [21] or walking on uneven terrain [20]. We now propose to combine it with a motion capture system to facilitate robot programming and to quickly produce some complex and dynamic movements. The objective of this application is to show the interest of OSID for quickly generating complex whole-body movements without writing dedicated methods or software. In the selected movements, we mostly focused on the capability to handle angular momentum.

#### A. Related work

Motion imitation is a special case of motion generation where the specified goal configuration is obtained from human motion. However, the kinematic and dynamic differences between humans and robots (structure, power, shape, weight distribution) make the *direct* mapping from human joint angles to robot joint angles impossible. In computer animation, retargeting motion to characters is a well-established application, and methods like dynamic filters [28] have been implemented. In robotics, several methods such as optimization before imitation [25] and the Learning from Observation (LFO) paradigm [17] have been proposed. In particular, dance motions have been applied to small humanoid robots like NAO and SONY's SDR, but motion for light-weight robots is easier than for human-sized robots due to the design, light material, small size, low weight, and larger feet sole to leg length relation of the former ones (see App. I for more details).

For human-sized robots, a pioneering work, and the current state of the art, has been done in [17] for a traditional Japanese dance based on IK and the LFO. It allows the interaction with a motion designer who can set up some key-frame postures that stabilize and guide the numerical optimization [16]. The method leads to impressive dynamic movements, coming from its non-causal methodology. A consequence is the difficulty to bring it to real time, for example, when dynamic walking is needed. This non-causality is needed to capture a prediction horizon, which makes the achievement of particular dynamic movements possible, as emphasized by the study of walking [1]: a new step must be anticipated, implying an action on the past trajectory when looking at the motion as a trajectory optimization problem.

Being OSID an instantaneous-linearization method, it presents similar limitations, but for most movements it can be applied in real-time without interactions with an expert during the movement adaptation. Moreover, different constraints such as joint limits and collision avoidance can be taken into account, which is a significant advantage over existing methods. As explained in the introduction, this application shows that OSID is a very appealing trade-off between IK and trajectory optimization, providing both the control of a large range of movements and fast computations. However, as described in section II-E, the method is not able to independently generate movements that require a prediction [27], such as a walking step, due to the instantaneous linearization. Like for IK, this type of effects should be

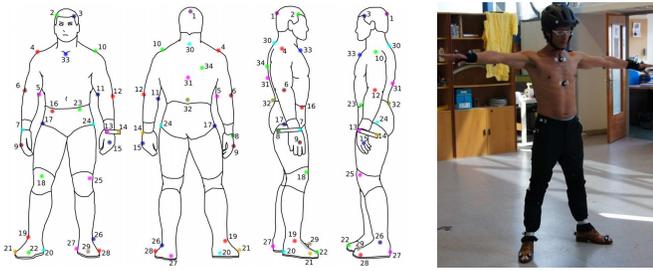


Fig. 2: Position of the markers in the human body and calibration position

specifically integrated [18], which implies that the trigger of a new step must be anticipated by an according prediction filter [1] if OSID-based imitation should be implemented in real time.

### B. Data acquisition

The motion capture system provides the spatial trajectory for each of the markers attached to the human body as Fig. 2 shows. The markers are distributed on the human body in a proper way to minimize the motion of the skin with respect to the bones and to facilitate their temporal tracking. These markers are manually associated to form a skeleton in which every link (or bone) is characterized by its position and orientation in a fixed frame. The robot configuration is then adjusted to fit at best the measured body positions and orientations. For each time frame, the geometry is retargeted by solving a non-linear least-squares problem minimizing the distance between the observations and the model [22].

The human motion was acquired using a *Motion Analysis*<sup>3</sup> motion capture system. More than thirty minutes of movements were captured for the preparation of the show. The captured data for the most important motions in this work is freely available<sup>4</sup> along with the geometrical retargeting and the results of the dynamic SoT, explained in the following section.

Joint trajectories obtained by geometrical retargeting are not dynamically consistent with the robot model: nothing guarantees that the robot is stably balanced, or that auto-collision or joint limits are avoided. Moreover, some important aspects of the original motion can be damaged since the retargeting is obtained by a trade-off among the positions of all the bodies. If a given body is more relevant than the others, this importance is not reflected in the obtained motion due to the differences between the two kinematic chains. For example, if both hands are clamping in the demonstration, their resulting positions are not likely to satisfy the clamp.

### C. Dynamical retargeting

The SoT is a means to enforce dynamic consistency while simultaneously editing specific aspects of the resulting motion. It is possible to make the robot more precisely track

<sup>3</sup>Ten infrared cameras distributed around the experimentation zone and calibrated by the *Motion Analysis* software with acquisition frequency of 200Hz and precision of 2 mm of nearly pure white noise and negligible bias (<http://www.motionanalysis.com/>).

<sup>4</sup><http://projects.laas.fr/gepetto/novela/noveladb>

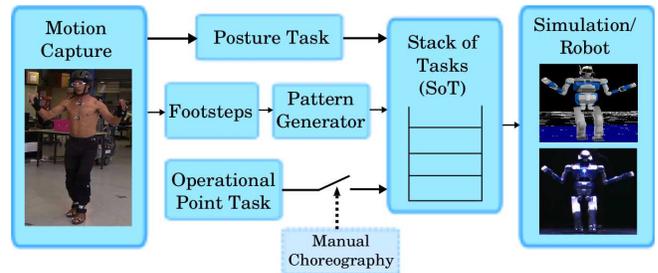
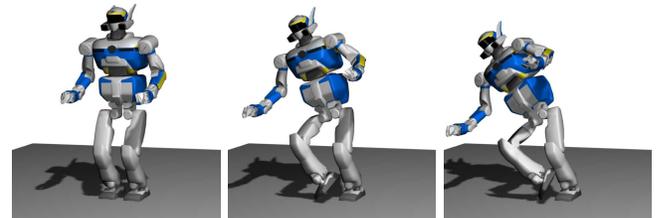


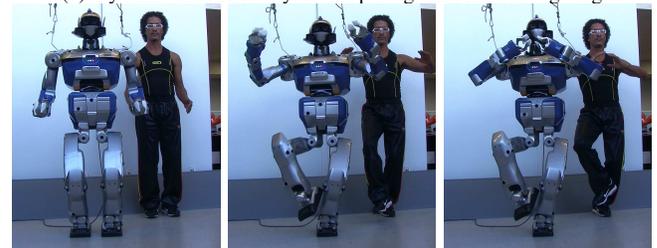
Fig. 3: Dynamical retargeting scheme. Motion capture information passes to the SoT through some tasks. Tasks for operational points can be manually activated according to the choreography. The motion is finally executed in a dynamic simulator or the real robot.



(a) Original demonstrated motion



(b) Dynamic inconsistency of the pure geometrical retargeting



(c) Executed motion

Fig. 4: Yoga motion: (a) original demonstration, (b) dynamic replay of the kinematic retargeting: the motion is not stable, (c) robot execution after adding the CoM and hands tasks.

some part of the demonstration by adding a task on specific operational points to follow exactly the corresponding trajectory of the human performer. This section describes the generic retargeting and edition process, illustrated in Fig. 3.

From the demonstration, the contact points are extracted by detecting clusters of static points in the feet trajectories. The first constraint of the SoT is to enforce dynamic consistency of the corresponding contact model. This also guarantees local balance: all the contact points remain stable. Other robot constraints are added to enforce joint limits (or collision avoidance). The least-priority task, often named the posture task, tracks at best the reference configuration

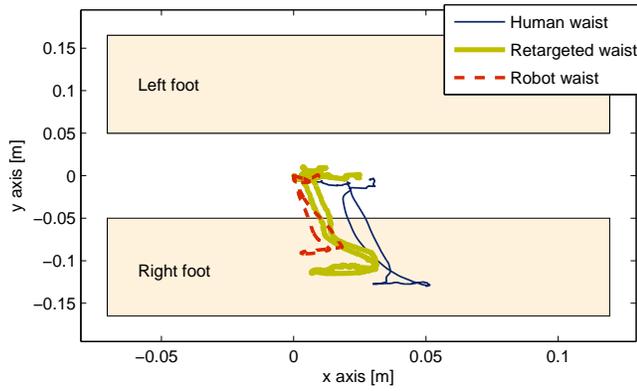


Fig. 5: Comparison of the human and robot waist trajectories. The robot waist is not directly constrained, but moves to meet CoM and posture objectives. However, the obtained trajectory is similar to that of the human. The artificial CoM pattern imposed to the robot is thus not an obstacle to the proper imitation.

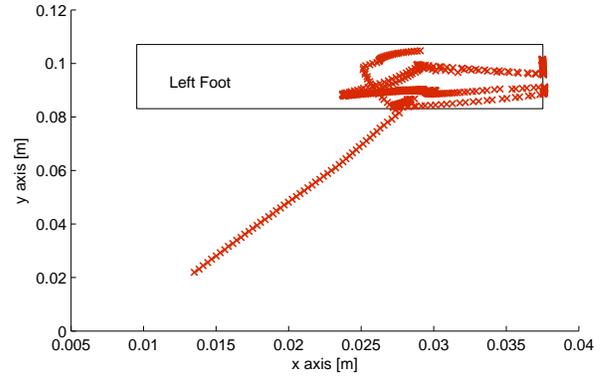
coming from the geometrical retargeting. It ensures that the global structure of the movement looks similar to the one that has been demonstrated. However, since the kinematic structures of the demonstrator and the robot are different, the posture similarity (corresponding to a “least-norm” trade-off) is sometimes not satisfactory. Typically, some important aspects of each dance figure are given by the choreography and should be accurately tracked. They can also be automatically extracted by a frequency analysis [2], by studying the motion model of the human [7] or, like in the following, manually determined from the choreography specifications. A set of tasks is added to track some particular aspects of the demonstration. The tasks used in this work are technically described in App. II. Similarly, new motion features that are not in the original demonstration can be added with an appropriate task.

#### D. A first example: the yoga figure

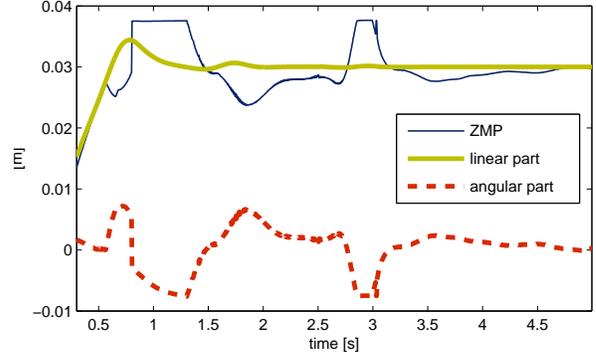
The *standing lotus* yoga motion is a typical example of motion easily achieved with the proposed method. The motion starts in a double-support position. Then, a sweeping motion of both arms is executed while reaching a stable single-support posture with the free foot close to the support knee and both hands joined in front of the chest. The movement is summarized in Fig. 4-(a).

The important features of the demonstrated motion are: balance *i.e.* the position of the CoM at the balanced position; the relative position of the hands and the chest; and the position of the free foot. When directly executing the geometrically-retargeted motion with a realistic (dynamic) simulator, the robot does not reach a stable single-support posture, as shown in Fig. 4-(b). Indeed, the mass distribution of the human and the humanoid are very different: the legs of the robot are much heavier. Thus, posture alone does not ensure robot balance. Also, the hands are colliding with each other and with the chest, and the flying foot is badly positioned.

The SoT is used to enforce balance and the proper placements for the hands and chest. Four tasks are added to



(a) Spatial evolution of the ZMP



(b) Temporal evolution of the components

Fig. 6: ZMP evolution during the yoga motion. (a) Motion of the ZMP from double to single (left foot) support. The ZMP is restricted to a reduced part of the foot, the resulting safety margin being used to improve the movement robustness. (b) The ZMP saturates when there is a greatest acceleration of the arms and during the transition from double to single support in the yoga motion. The ZMP overshoot is mostly caused by angular momentum variations.

respectively control the CoM, each hand and the free foot. The three latter follow the demonstrated trajectories of the corresponding human-body points. The CoM is more difficult to observe on the human: it can be estimated by the waist position or using an inertial model with the correct mass distribution, but even if recovered, it cannot be guaranteed that the human CoM generates stable motions in the robot due to mass differences. Thus, an artificial pattern for the CoM was imposed to enforce balance since CoM accelerations might destabilize the robot and the approach cannot predict those states. This pattern constrains the CoM to lie inside the support polygon and is obtained by experimentally determining the timings for the change in the supports of the human performer. The SoT is finally composed of the dynamics and contact constraints, a task tracking the CoM pattern, tasks tracking the demonstrated right hand, left hand and right foot, and a task tracking the geometrically-retargeted configuration. The task that tracks the free foot is only added when the foot leaves the ground. The robot motion is shown in Fig. 4-(c).

Although an artificial CoM pattern is imposed, the resulting motion is still globally similar to the demonstration. Fig. 5 shows the trajectory of the waist for three cases: the

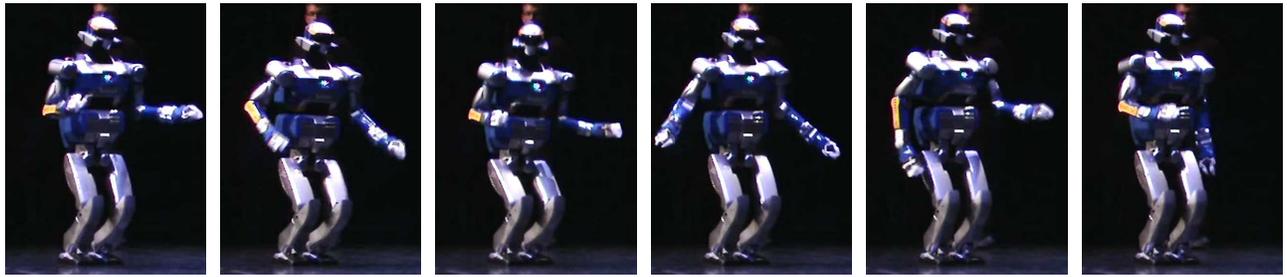


Fig. 7: Example of a motion in double support: fast arms motion

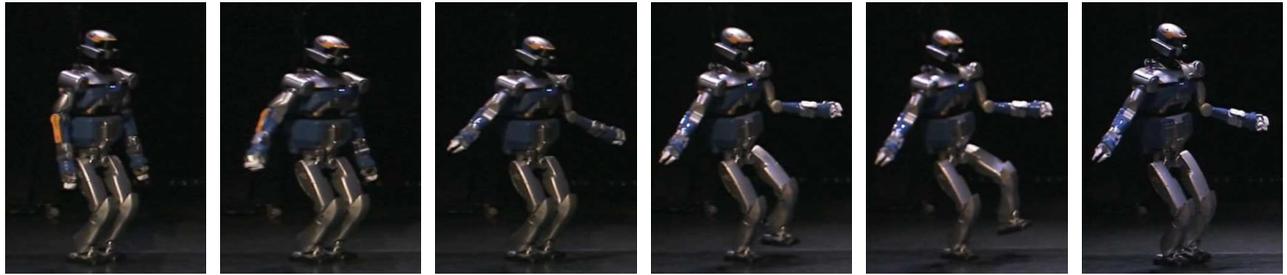


Fig. 8: Example of motion in single support: raising the left leg upwards

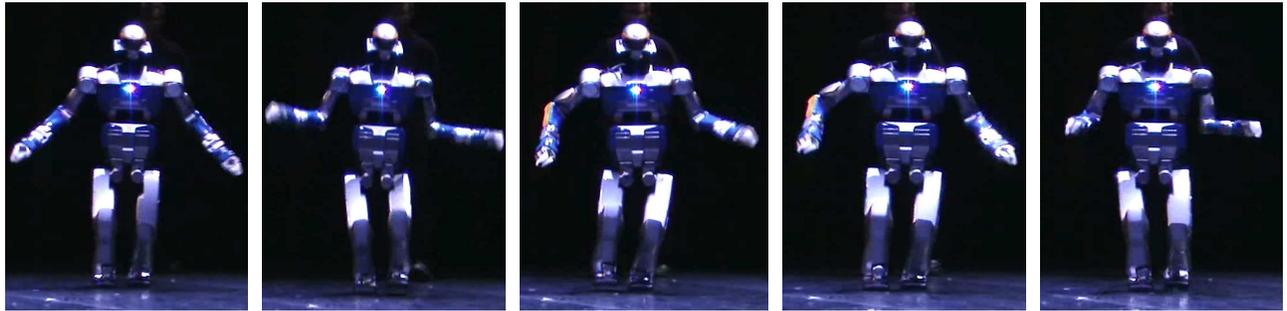


Fig. 9: Walking while arbitrarily moving the upper part of the body

human waist acquired by the motion capture system, the waist retargeted with geometrical methods as described in Section III-B and the robot waist obtained as output of the SoT. Even if the waist is not explicitly controlled, the original waist pattern is kept during the robot replication.

This movement perfectly illustrates the discussion of Sec. II-E: the linear momentum of the robot is controlled in a preview horizon using a WPG, whereas the angular momentum is locally regulated by the OSID. Fig. 6 shows the trajectory of the  $x$  component of the ZMP caused by the linear ( $c - \frac{1}{\omega^2} \ddot{c}$ ) and angular ( $\frac{\sigma}{mg}$ ) momentum variations. The ZMP saturates to enforce the contact constraint, which mostly corresponds to a bounding of the angular components. The WPG is not sufficient to enforce the balance due to large arm movements. Using IK instead of OSID, the robot would have tipped over and this motion would not have been achieved.

#### E. Generation of the 10-minute dance movement

The generic task sequence can be applied to automatically treat several sequences. From the choreography designed for the robot, three typical execution patterns are detailed below.

1) *Double-Support Motion*: The simplest case consists in moving the whole body fast while keeping both feet in double support. Even though some parts of the body move fast and might generate undesired angular moments, the proposed method keeps the robot balance by automatically compensating with the appropriate control. A task can be added only when specifically requested by the choreographer, but there is no task for the CoM. Fig. 7 shows an example of this motion.

2) *Single-Support Motion*: A second type of motion consists in moving the body while keeping a single leg in contact with the ground. Like for the yoga, the CoM is driven by a task following an artificial pattern for both reaching a stable single-support position and going back to a double-support stance. An example is shown in Fig. 8 where the robot first moves its arms and then raises the left leg.

3) *Dynamic Walk*: Finally, we consider the case where the movement implies to walk while moving the upper body at the same time. As previously, the footprints (positions and timings) are extracted from the demonstration by grouping clusters of points on the foot trajectories. The footprints are used to compute a walking trajectory for both the feet and the

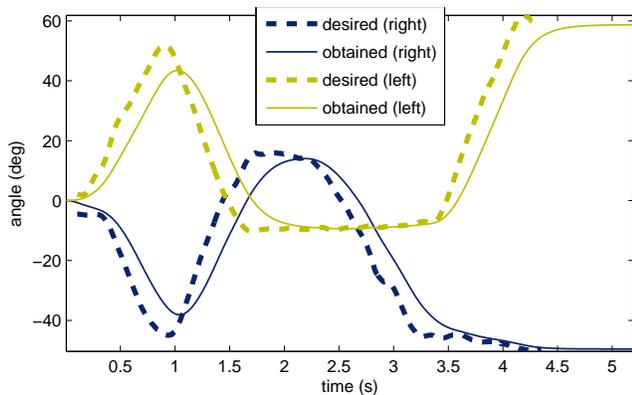


Fig. 10: Desired and obtained joint trajectories for the right and left shoulders

CoM using a walking pattern generator [8]. The CoM task is added as the top-priority task, while a task is added to drive the position of the flying foot during each step. Eventually, additional tasks are added to improve the hand or the head placements depending on the choreography.

The hands produce an important momentum, especially when the choreography imposes some movements that are contrary to the natural walking motion. The momentum is however corrected by the whole-body inverse-dynamics scheme and does not disturb the balance. Fig. 9 shows an example of this type of motion. Fig. 10 shows the captured and experimental joint trajectories of two typical joints (at the shoulders). It can be observed that the trajectories obtained with the dynamic control follow closely the desired ones, but the control also acts as a ‘low pass filter’. This attenuation can be controlled by changing the gain of the task. If the gain is larger, the joints move faster and the trajectory is closer to the desired one.

#### F. Public Performance

The *Festival of Shared Knowledge (Novela Festival)* is held in Toulouse every year since 2009 and is organized by the city council. It constitutes one of the main cultural events of the city and aims at presenting the research of the several universities and laboratories of the city to the public. In this framework, the ‘robotic dance’ was presented. Tests of the whole dance sequence were initially made in the robotics room at LAAS-CNRS assuming a completely flat ground. We needed three men-month for 10 minutes of performance while simultaneously producing some of the fundamental developments. The final presentation of the robotic dance took place at a concert hall in the city center of Toulouse: the dancers were HRP-2 together with a hip-hop dancer. The night of the main presentation, the motion was performed without the security lifter and only using the power cable in front of more than a thousand people who came to see the show. Videos of the performance (summary and complete) are available<sup>5</sup>.

<sup>5</sup><http://projects.laas.fr/gepetto/novela/videos>

## IV. DISCUSSION

The objective of this article is to show the maturity of OSID for generating complex whole-body movements on real robots by the first real-size proof of concept on a real humanoid. Our specific OSID method is based on hierarchical quadratic programming with the best performances of the state of the art, providing real-time resolution, inequality enforcement and hierarchical resolution. The demonstrated motion is based on motion capture, which is a very efficient way to program the robot. But OSID can work in other general ways, for example by specifying a task sequence or by adjusting a movement coming from a motion planner (some in-lab movements that show other advantages of the proposed method such as multiple-contact movements, over existing methods can be found in [24] and [6]). The context of the dance was also a good opportunity to show the capabilities of OSID to handle large angular momentum variations and small linear momentum variations (the large CoM movements being handled by a WPG).

The realization of a long demonstration is always the opportunity to gain some insights from the integration process. The first key point is the interest of OSID with respect to the usual IK that we generally used for generating humanoid movements. OSID straightforwardly handled the dynamic variations coming from the artificial unbalanced dance movements while IK typically involves many trial-and-error iterations to choose the variation control parameters (gains and others). We also found that motion capture is a very efficient tool, already widely used in computer animation, but with less use in robotics. Motion capture provides a very intuitive solution to shape a reference movement that is particularly suitable for anthropomorphic structures. Finally, from an applicative point of view, we noticed a very high and not-so-expected enthusiasm of the public in the robot spectacle, showing the big application of robotics in the leisure industry, which is less documented than the potentiality of robotics in manufacturing or services.

The pair WPG (to handle the horizon of linear momentum) and OSID (to handle local variations of the dynamics) is very efficient. The inability of OSID to handle alone the balance of the robot due to its intrinsic instantaneous nature is sometimes not clear for the users, but the use of motion capture is a good opportunity to emphasize it. This aspect would be handled by a whole-body MPC, which is not yet available due to many scientific and technical aspects. For the deployment of well identified movements, the proposed method is satisfactory; but in order to reach total autonomy of humanoid robots, coupling the WPG with the OSID is problematic and MPC remains a desirable goal.

Finally, the HRP-2 platform has also shown some limitations during the dance execution by its lack of sensor feedback related to dynamic effects perception. For instance, there is no skin to perceive the deformations of the mechanical structure or feel the transition from friction to sliding at the contact point, no accelerometers to perceive vibrations through the mechanical structure (assumed rigid,

but oscillating in practice), no joint torque sensors, and no temperature probes to perceive if the motors are overheating. The OSID control handled the effects of the dynamics in feed-forward, while feedback dynamic control still remains to be established. The only feedback loop was came from the commercial “stabilizer” of HRP-2, which controls the movements of the flexible parts of the feet by feedback of the ankle force-torque sensors but in exchange imposes some strong limitations. For instance, the bended knees observed throughout the demonstration are a limitation of the stabilizer and not of the OSID approach. The feedback control of the the dynamic effects is, before going to MPC, the next challenge of humanoid movements.

#### ACKNOWLEDGMENTS

This work was granted by the City Council of Toulouse in the framework of the Novela science festival. The third author is funded by the european grant FP7-ICT-2013-10/611909 KOROIBOT. The authors would like to thank M. Morisawa (AIST, Tsukuba, Japan) for his contribution to the generation of some dancing steps, P. Souères and A. Mauffret for the coordination of the event, the hip hop dancer T. Benamara for the performance of the human dance, and A. El Khoury for his help during the presentation.

#### APPENDIX I BALANCE SIZE RATIO

On flat ground, it is possible to compare the intrinsic balance of two humanoid robots of different size (*e.g.* Nao and HRP-2) in single support by comparing the ratio between their soil size and their legs length. The ZMP can be computed as in (5) where the variation of the angular momentum is usually omitted. Assuming a rectangular feet of size  $(S_w, S_l)$ , the robot is dynamically unbalanced if  $(z^x, z^y)$  goes over the limits imposed by the dimensions of the feet. If we assume a rigid transformation between the waist and the CoM as in [8], and as (5) is characterized by the height of the CoM, ignoring  $\dot{\sigma}$ , then it is possible to compare two humanoid robots by comparing their following balance ratios:

$$b_w = \frac{S_w}{l}, \quad b_l = \frac{S_l}{l} \quad (6)$$

where  $l$  is the length of the robot legs at a straight position. For example,  $b_l = 0.53$  for a small robot like NAO ( $S_l \approx 0.16m$ ,  $l \approx 0.3m$ ), and  $b_l = 0.34$  for a human-sized robot like H7 ( $S_l \approx 0.23m$ ,  $l \approx 0.68m$ ) which is similar in size to HRP-2.

#### APPENDIX II SOME CLASSICAL TASKS

We quickly describe the tasks used to drive the motion generation. For each task, the reference behavior  $\ddot{e}^*$  can be based on a proportional derivative (PD) control law as:

$$\ddot{e}^* = -K_p e - K_v \dot{e} \quad (7)$$

with  $K_p > 0$  and  $K_v = 2\sqrt{K_p}$ . This control law imposes an exponential decay of the task tending to zero. Some examples of tasks using this principle are:

1) *Placement Task (6D)*: It is used for the position and orientation of one body of the robot whose attached frame is usually named an *operational point*. The first part of the task defines the position error as the difference between the current and the desired position:  $e_p = x - x^*$ . The second part defines the attitude error as  $e_o = r\theta \ominus r\theta^*$ , where  $r\theta$  is the angle-axis representation of the orientation and  $\ominus$  is a suitable difference group operator of  $SO(3)$ . For instance, if  $R$  and  $R^*$  are the current and desired rotation matrices, the error matrix can be defined as  $R_e = R^{-1}R^*$  or  $R_e = (R^*)^{-1}R$  and the attitude error is identified with the axis-angle representation of  $R_e$ .

2) *CoM Task (3D)*: It comprises the error between the current position of the CoM and the desired one, so that  $e_{com} = c - c^*$ . The control might involve the three position components or only some subset of them.

3) *Posture Task*: It follows the joints evolution and is expressed as  $e = q_k - q_k^*$ , where  $q_k$  is the vector containing the angular values for the joints to be controlled, and  $q_k^*$  is the desired configuration for those joints [13]. If  $K_p$  in (7) is increased, PD tasks allow to achieve the target faster but without time guarantees. To explicitly specify time constraints, an interpolation task updated in every iteration can be used [5]. For joint angle limits, constraints are imposed by a set of linear inequalities on the joint angular positions [24]. Using the Taylor expansion of the joint position  $q$ , the task for joint limits is defined as [19]:

$$\underline{q} \leq q + \Delta T \dot{q} + \frac{1}{2} \Delta T^2 \ddot{q} \leq \bar{q} \quad (8)$$

where  $\underline{q}$  and  $\bar{q}$  represent the lower and upper angular joint limit values, respectively, and  $\Delta T$  is the control loop period. From (8), the inequality constraint for  $\ddot{q}$  is straightforward. A similar constraint can be used to enforce collision avoidance or to keep an object within the field of view.

#### REFERENCES

- [1] Mehdi Benallègue, P-B Wieber, Abderrahmane Kheddar, and Bernard Espiau. A computational model for synchronous motion imitation by humans: The mirror controller applied on stepping motions. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, Nashville, USA, 2010.
- [2] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics*, 37(2):286–298, 2007.
- [3] T. Chang and R. Dubey. A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292, 1995.
- [4] 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 *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008.
- [5] M. de Lasa, I. Mordatch, and A. Hertzmann. Feature-based locomotion controllers. *ACM Transactions on Graphics (SIGGRAPH)*, 29(4):131:1–131:10, 2010.
- [6] A. Escande, N. Mansard, and P-B. Wieber. Hierarchical quadratic programming. *The Int. Journal of Robotics Research*, 33(7), 2014.
- [7] S. Hak, N. Mansard, O. Stasse, and J-P. Laumond. Reverse control for humanoid robot task recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 42(6):1524–1537, 2012.
- [8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003.

- [9] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa. Zmp-based biped running control. *IEEE Robotics & Automation Magazine*, 14(2):63–72, 2007.
- [10] O. Kanoun, F. Lamiroux, and P-B. Wieber. Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.
- [11] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The Int. Journal of Robotics Research*, 5(1):90–98, 1986.
- [12] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987.
- [13] O. Khatib, L. Sentis, J. Park, and J. Warren. Whole body dynamic behavior and control of human-like robots. *The Int. Journal of Robotics Research*, 1(1):29–44, 2004.
- [14] N. Mansard. A dedicated solver for fast operational-space inverse dynamics. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, St Paul, MN, USA, 2012.
- [15] I. Mordatch, E. Todorov, and Z. Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM SIGGRAPH'12*, 31(4):43, 2012.
- [16] S. Nakaoka, K. Miura, M. Morisawa, F. Kanehiro, K. Kaneko, S. Kajita, and K. Yokoi. Toward the use of humanoid robots as assemblies of content technologies - realization of a biped humanoid robot allowing content creators to produce various expressions. *Synthesiology*, 4(2):80–91, 2011.
- [17] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi. Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *The Int. Journal of Robotics Research*, 26(8):829–844, 2007.
- [18] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Captuability-based analysis and control of legged locomotion. *The Int. Journal of Robotics Research*, 31(10):1117–1133, 2012.
- [19] O. Ramos. *Generation of whole-body motion for humanoid robots with the complete dynamics*. PhD thesis, Université de Toulouse, LAAS-CNRS, 2014.
- [20] O. Ramos, N. Mansard, O. Stasse, and P. Souères. Walking on non-planar surfaces using an inverse dynamic stack of tasks. Osaka, Japan, November 2012.
- [21] O. Ramos, N. Mansard, O. Stasse, and P. Souères. An advanced robotics motion generation framework for inferring the organisation of human movements. *Computer methods in biomechanics and biomedical engineering*, 16(sup1):177–178, 2013.
- [22] O. Ramos, L. Saab, S. Hak, and N. Mansard. Dynamic motion capture and edition using a stack of tasks. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, Bled, Slovenia, 2011.
- [23] D. Raunhardt and R. Boulic. Progressive clamping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Roma, Italy, 2007.
- [24] L. Saab, O. Ramos, N. Mansard, P. Souères, and J-Y. Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2):346–362, 2013.
- [25] A. Safonova, N.S. Pollard, and J.K. Hodgins. Optimizing human motion for the control of a humanoid robot. In *Int. Symposium on Adaptive Motion of Animals and Machines*, Kyoto, Japan, 2003.
- [26] C. Samson, B. Espiau, and M. Le Borgne. *Robot control: the task function approach*. Oxford University Press, 1991.
- [27] P-B Wieber. Viability and predictive control for safe locomotion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, 2008.
- [28] K. Yamane and Y. Nakamura. Dynamics filter-concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, 2003.