Static Multi-Contact Inverse Problem for Multiple Humanoid Robots and Manipulated Objects

Karim Bouyarmane, Abderrahmane Kheddar

Static Multi-Contact Inverse Problem for Multiple Humanoid Robots and Manipulated Objects

Karim Bouyarmane and Abderrahmane Kheddar

***Abstract*— In this paper we solve the static-equilibrium constrained inverse kinematics problem for a system made of multiple humanoid robots and manipulated objects given a set of contacts between any surfaces of the robots, any surfaces of the manipulated objects, and any surfaces of the environment. In particular, inter-robots contacts are possible. The contacts considered here are neither necessarily coplanar, nor necessarily horizontal, frictional, might be unilateral (support) or bilateral (grasp). We solve both the geometric variables (configurations) and the statics variables (contact forces) simultaneously within one optimization query. In the resulting configurations all the robots and the manipulated objects are in static equilibrium under the action of gravity and actuator torques that are constrained to stay within their bounds. The main focus of the paper is on the formulation of the problem rather than the optimization algorithm, as we consider the latter as a black box that only requires a mathematical model providing algorithms to compute the values of the objective function, the constraints functions, and their derivatives. We apply this work to quasi-static multi-contact legged locomotion planning on irregular terrain, multi-fingered dexterous manipulation planning, and collaborative manipulation planning.**

I. Introduction

Solving the static multi-contact inverse problem is a core issue in acyclic multi-contact motion planning. Existing acyclic multi-contact motion planning algorithms [1][2] explore the workspace environment by growing a stances tree; a stance being a set of contacts between surfaces of the robot's cover and surfaces of the environment. To validate a stance and add it to the exploration tree, the algorithm needs to test the feasibility of the stance by finding a configuration of the robot that realizes the stance. This is what we call here the stance inverse problem. For a given stance σ, let us denote $\mathscr{Q}(\sigma)$ the solution set of this inverse problem, i.e. the set of all configurations that geometrically realize the stance. $\mathscr{Q}(\sigma)$ is a sub-manifold of the configuration space of strictly lower dimension. Let us denote $\mathscr{F}(\sigma)$ the subset of $\mathscr{Q}(\sigma)$ made of all the configurations that realize the stance while being in static equilibrium. $\mathscr{F}(\sigma)$ is a closed subset of $\mathscr{Q}(\sigma)$ provided with its subspace topology. For $q \in \mathscr{F}(\sigma)$, let us denote $\Lambda_\sigma(q)$ the set of all admissible contact forces that maintain the configuration in static equilibrium. If the stance is made of n surface contacts, each surface $i \in \{1, \ldots, n\}$ being modeled by a polygon with V_i vertices, then $\Lambda_\sigma(q)$ is a subset of $\mathbb{R}^{3\sum_i V_i}$.

K. Bouyarmane and A. Kheddar are with CNRS-AIST JRL (Joint Robotics Lab) UMI3218/CRT, AIST, Tsukuba, Japan; and with CNRS-University of Montpellier 2 LIRMM, Montpellier, France. {karim.bouyarmane,abderrahmane.kheddar}@aist.go.jp

The fundamental problem we would like to solve is to test the feasibility of a given stance σ, i.e. to test whether

$$\mathscr{F}(\sigma) \neq \varnothing \ ? \tag{1}$$

Then if (1) is true, it would also be convenient to exhibit one solution, i.e. to solve

$$\text{find } q \in \mathscr{F}(\sigma) \text{ and } \lambda \in \Lambda_\sigma(q). \tag{2}$$

Last, we would like a more refined version of (2), which is to minimize a criterion over all the feasible stances and associated forces, i.e. to solve the following non-linear constrained optimization problem

$$\min_{(q,\lambda)} \text{obj}(q, \lambda) \quad \text{subject to} \begin{cases} q \in \mathscr{F}(\sigma) \\ \lambda \in \Lambda_\sigma(q). \end{cases} \tag{3}$$

II. Related work

A lot of effort has been dedicated to solving inverse geometric queries on closed kinematic chains in the field of randomized path planning, e.g. [3][4][5]. Using the notations introduced in the previous section, these works solve the query

$$\text{find a random } q \in \mathscr{Q}(\sigma) \tag{4}$$

with no other constraints, thus without considering static equilibrium (fixed-base robots). Then, given a particular $q_0 \in \mathscr{Q}(\sigma)$, for example as returned by solving the problem (4), works like [6][7] are concerned with testing the static equilibrium of q_0, i.e. solving the problem

$$q_0 \in \mathscr{F}(\sigma) \ ? \tag{5}$$

If the answer to the problem (5) is true, other methods, e.g. [8], allow to compute optimal contact forces, and thus solve the following problem

$$\min_{\lambda \in \Lambda_\sigma(q_0)} \text{obj}(\lambda). \tag{6}$$

Sequentially solving problems (4) then (5) then (6) gives a rejection scheme for solving the problem (2). We propose here another scheme that does not rely on random configuration rejection sampling which might be costly especially in the case of stances made of low number of contacts where very few geometrically valid configurations are in static equilibrium. So we decide to solve problem (2) directly through the problem (3). Both [1] and [2] have chosen this approach. Our contributions with regard to these two works is in the modeling of the conditions that define $\mathscr{F}(\sigma)$ as we

try to remain as general as possible and avoid any strong hypotheses that could have allowed us to use approximations on the static equilibrium constraint, by reducing it for example to the belonging of the ground projection of the CoM to the support polygon. We also avoid hypotheses on the rigidity of the robots as we consider the specified limits on the actuators torques needed in holding the static configuration. The Iterative Constraint Enforcement method proposed in [2] considers torques limits only in a post processing rejection test once the rigid version of the problem has been solved. Once again we want to avoid this rejection scheme and input the torques limits constraint directly into the initial problem. However, both [1] and [2] consider collision avoidance constraints while, for the time being, we do not in this paper. Coming work should incorporate these collision avoidance constraints [9]. A last and original contribution of this paper is that it solves the inverse stance problem for a system made of multiple robots and objects, which is not the case in any of the previous works.

Note that our problem (2), within its static *planning* context, is different from its dynamic *control* counterpart. Precisely, we are not looking for a feasible trajectory, or a steering method, that takes us from an initial configuration and tries to reach specified contact locations. As such, optimization-based iterative inverse kinematics techniques that rely on constraints prioritization, e.g. [10][11], are not necessarily suitable for our particular purpose. Here we are not trying to satisfy constraints at best following a feasible trajectory, but rather to know whether a constrained solution exists or not.

III. Problem formulation

For the notations used in this section we refer the reader to Fig. 1.

We suppose that we have a system of N robots and objects indexed by $r \in \{1, \dots, N\}$. To this set we append an additional index 0 referring to the environment. This way we have a coherent and unified description for robot-robot contacts, robot-environment contacts, robot-object contacts, and finally object-environment contacts. For convenience we use the term *robot* when talking about either an actual robot, or a manipulated object, or the environment.

A. Optimization variables

The configuration vector for a robot $r \in \{1, \dots, N\}$ takes the form

$$q_r = (x_r, y_r, z_r, \quad \alpha_r, \beta_r, \gamma_r, \delta_r, \quad \theta_{r,1}, \theta_{r,2}, \dots, \theta_{r,j_r}),$$

which is the concatenation of the Cartesian position of the root body, the unit quaternion representing the orientation of the root body, and the vector θ_r of the j_r joint articulations. $j_r \neq 0$ for an actual robot and $j_r = 0$ for a rigid object. For a body b of the robot r we denote $O_{r,b}(q_r)$ and $R_{r,b}(q_r)$ respectively the origin's position and the orientation matrix of the frame $\mathscr{T}_{r,b}$ attached to the body b. The body $b = 0$ corresponds to the root body of r.

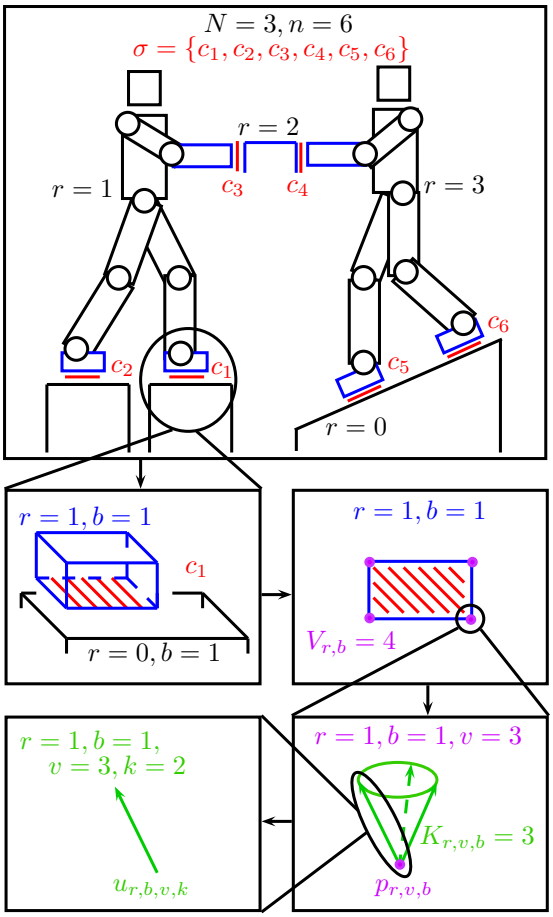


Fig. 1: Illustration of the different levels of indices used in this paper for an example made of 3 robots (4 including the environment) and a 6 contacts stance.

Let us now start from a stance σ made of n contacts

$$\sigma = \{c_1, \dots, c_n\}.$$

Each contact c_i is defined between the surface $S_{r_{i1},b_{i1}}$ rigidly attached to the body b_{i1} of the robot $r_{i1} \in \{1, \dots, N\}$, and the surface $S_{r_{i2},b_{i2}}$ rigidly attached to the body b_{i2} of the robot $r_{i2} \in \{0, \dots, N\}$. A surface $S_{r,b}$ is a convex polygon[1] with $V_{r,b}$ vertices

$$S_{r,b} = \text{conv}\left(\{p_{r,b,1}, \dots, p_{r,b,V_{r,b}}\}\right).$$

For each point $p_{r,b,v}$ fixed in the local frame of the body b we denote $P_{r,b,v}(q_r)$ its position in the world frame and $P^0_{r,b,v}(\theta_r)$ its position in the root frame of the robot r. At each point $p_{r,b,v}$ we specify a polyhedral cone $\mathscr{C}_{r,b,v}$ with finite number $K_{r,b,v}$ of generators that approximate the friction cone, the axis of which is the inward normal to the surface $S_{r,b}$

$$\mathscr{C}_{r,b,v} = \text{pos}\left(\{u_{r,b,v,1}, \dots, u_{r,b,v,K_{r,b,v}}\}\right).$$

[1]This is one assumption of our work. Non-convex polygonal surfaces of the robot are decomposed into a finite set of convex polygons. Non-polygonal convex surfaces are conservatively approximated by polygons.

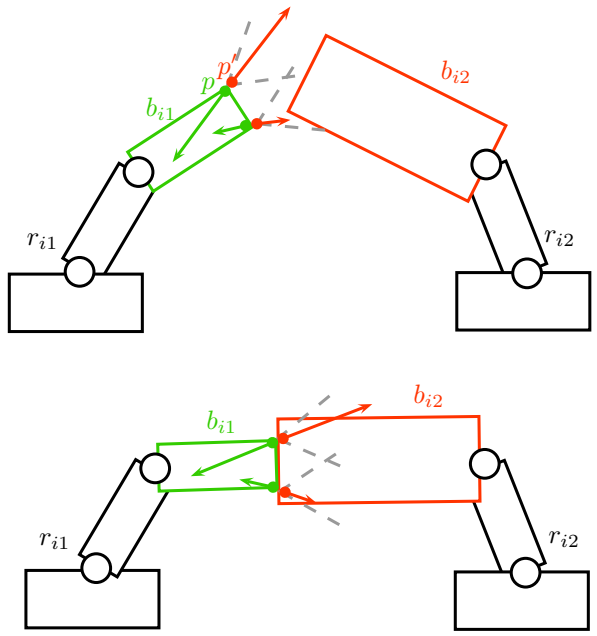


Fig. 2: In green the minimum area surface's body, in red the maximum area surface's body. The contact forces applied on a body are drawn in the same color as the body. Before the contact is established at the solution (top figure), the forces applied on the red body have their application points p originally expressed in the local frame of the green body. To compute the torques resulting from the application of these forces on the red body we have to consider the virtual point p' of the red body's local frame that instantaneously coincides with p at every configurations $q_{r_{i1}}$ and $q_{r_{i2}}$ of the robots.

The case of a bilateral contact is simply handled by setting

$$\mathscr{C}_{r,b,v} = \mathbb{R}^3,$$

and, in this case, the vectors u are simply the three basis vectors of $\mathscr{T}_{r,b}$ with no positivity constraints on their coefficients. For each unit vector $u_{r,b,v,k}$ fixed in the local frame of the body b we denote $U_{r,b,v,k}(q_r)$ its coordinates in the global frame.

We can now introduce the statics variables λ. We first suppose without loss of generality (we can permute the indexes 1 and 2) that the area of the surface $S_{r_{i1},b_{i1}}$ is less than the area of the surface $S_{r_{i2},b_{i2}}$, so that when the contact c_i occurs, at the solution, we can write

$$S_{r_{i1},b_{i1}} \subset S_{r_{i2},b_{i2}}.$$

The surface contact, at the solution, is thus $S_{r_{i1},b_{i1}}$, and the continuous surface force distribution over this surface can be reduced to a finite force distribution over its vertices. At each vertex $p_{r_{i1},b_{i1},v}$, $v \in \{1,\dots,V_{r_{i1},b_{i1}}\}$, The resulting contact force $f_{r_{i1},b_{i1},v}$ is a non-negative linear combination of the polyhedral friction cone generators

$$f_{r_{i1},b_{i1},v} = \sum_{k=1}^{K_{r_{i1},b_{i1},v}} \lambda_{r_{i1},b_{i1},v,k}\ U_{r_{i1},b_{i1},v,k}(q_{r_{i1}}).$$

The forces applied on the body b_{i2} robot r_{i2} will be, at the solution, equal to $-f_{r_{i1},b_{i1},v}$, applied at the same application points. See Fig. 2. For each contact c_i we denote λ_i the vector of $(\mathbb{R}^+)^{K_{r_{i1},b_{i1},v}}$ made of all the $\lambda_{r_{i1},b_{i1},v,k}$

$$\lambda_i = (\lambda_{r_{i1},b_{i1},v,1},\dots,\lambda_{r_{i1},b_{i1},v,K_{r_{i1},b_{i1},v}}).$$

Finally, the variables of our optimization problem (3) can be split into:

- geometric variables $q = (q_r)_{r\in\{1,\dots,N\}}$.
- statics variables $\lambda = (\lambda_i)_{i\in\{1,\dots,n\}}$.

B. Geometric constraints

For each contact c_i of the stance σ, a geometric constraint sets the relative position of the frame $\mathscr{T}_{r_{i1},b_{i1}}$ in the frame $\mathscr{T}_{r_{i2},b_{i2}}$. For each couple (r,b) we choose the frame $\mathscr{T}_{r,b}$ so that its origin is inside the surface $S_{r,b}$ and its third basis vector coincides with the inward normal to the surface $S_{r,b}$. Let us denote $(\vec{x}_{r,b}(q_r), \vec{y}_{r,b}(q_r), \vec{z}_{r,b}(q_r))$ the coordinates of the basis vectors of $\mathscr{T}_{r,b}$ in the global frame. A surface contact c_i needs the realization of at least the two following constraints

$$\vec{z}_{r_{i1},b_{i1}}(q_{r_{i1}}) + \vec{z}_{r_{i2},b_{i2}}(q_{r_{i2}}) = 0 \tag{7}$$
$$O_{r_{i1},b_{i1}}(q_{r_{i1}})^T \vec{z}_{r_{i2},b_{i2}}(q_{r_{i2}}) = 0. \tag{8}$$

This leaves us with three degrees of freedom that we denote $(x_{c_i}, y_{c_i}, \theta_{c_i})$, corresponding to the three following constraints

$$O_{r_{i1},b_{i1}}(q_{r_{i1}})^T \vec{x}_{r_{i2},b_{i2}}(q_{r_{i2}}) = x_{c_i} \tag{9}$$
$$O_{r_{i1},b_{i1}}(q_{r_{i1}})^T \vec{y}_{r_{i2},b_{i2}}(q_{r_{i2}}) = y_{c_i} \tag{10}$$
$$\vec{x}_{r_{i1},b_{i1}}(q_{r_{i1}})^T \vec{x}_{r_{i2},b_{i2}}(q_{r_{i2}}) = \cos(\theta_{c_i}), \tag{11}$$

which can be fixed as equality constraints if we specify a fixed contact location or left as inequality constraints if we wish to realize the contact and leave its location to be decided by the optimization process as a component of the objective cost function.

C. Static equilibrium constraints

We will write N static equilibrium constraints, one for each robot $r \in \{1,\dots,N\}$. Let us denote g the gravity field vector, m_r the total mass of the robot, $C_r(q_r)$ the coordinates of the CoM of the robot in the global frame. We partition the index set $I = \{1,\dots,n\}$ of the stance contacts $\sigma = \{c_1,\dots,c_n\}$ into three different subsets. $I_1(r)$ is the subset of I made of the contacts in which a surface from r is involved as the minimum area surface. $I_2(r)$ is the subset of I made of the contacts in which a surface from r is involved as the maximum area surface. $I_3(r)$ is the subset of I made of the contacts in which no surface from the robot r is involved.

$$I_1(r) = \{i \mid r_{i1} = r\}$$
$$I_2(r) = \{i \mid r_{i2} = r\}$$
$$I_3(r) = \{i \mid r \notin \{r_{i1}, r_{i2}\}\}.$$

A fundamental remark in our approach is that the forces acting on $r = r_{i2}$ resulting from the contacts i indexed

$$\sum_{i\in I_1(r)}\sum_{v=1}^{V_{r_{i1},b_{i1}}} f_{r_{i1},b_{i1},v} - \sum_{i\in I_2(r)}\sum_{v=1}^{V_{r_{i1},b_{i1}}} f_{r_{i1},b_{i1},v} + m_r g = 0 \tag{13}$$

$$\sum_{i\in I_1(r)}\sum_{v=1}^{V_{r_{i1},b_{i1}}} P_{r_{i1},b_{i1},v} \times f_{r_{i1},b_{i1},v} - \sum_{i\in I_2(r)}\sum_{v=1}^{V_{r_{i1},b_{i1}}} P_{r_{i1},b_{i1},v} \times f_{r_{i1},b_{i1},v} + C_r \times m_r g = 0 \tag{14}$$

$$\tau_r + \sum_{i\in I_1(r)}\sum_{v=1}^{V_{r_{i1},b_{i1}}} J_{r_{i1},b_{i1}}(q_{r_{i1}}, p_{r_{i1},b_{i1},v})^T f_{r_{i1},b_{i1},v} - \sum_{i\in I_2(r)}\sum_{v=1}^{V_{r_{i1},b_{i1}}} J_{r_{i2},b_{i2}}(q_{r_{i2}}, p'_{r_{i1},b_{i1},v})^T f_{r_{i1},b_{i1},v} + \left(\frac{\partial C_r}{\partial \theta_r}\right)^T m_r g = 0 \tag{15}$$

in I_2 have their application points $(p_{r_{i1},b_{i1},v})_v$ fixed in the frame $\mathscr{T}_{r_{i1},b_{i1}}$ of the other robot r_{i1}. To calculate the torques resulting on the joints of $r = r_{i2}$ we thus need to transform the points p in the frame $\mathscr{T}_{r_{i2},b_{i2}}$. Let us denote the transformed points p' such that, for each v,

$$p'_{r_{i1},b_{i1},v}(q_{r_{i1}}, q_{r_{i2}}) = R_{r_{i2},b_{i2}}(q_{r_{i2}})^T \left(P_{r_{i1},b_{i1},v}(q_{r_{i1}}) - O_{r_{i2},b_{i2}}(q_{r_{i2}})\right).$$

For $p \in \mathbb{R}^3$ let us denote $J_{r,b}(q_r, p)$ the following Jacobian matrix

$$J_{r,b}(q_r, p) = \frac{\partial\Big[R_{r,0}(q_r)^T\Big(\big(O_{r,b}(q_r) + R_{r,b}(q_r)p\big) - O_{r,0}(q_r)\Big)\Big]}{\partial\theta_r}. \tag{12}$$

We can finally write the static stability constraint for r, which are the constraints (13)-(14)-(15) appearing at the top of this page, where $\tau_r \in \mathbb{R}^{j_r}$ denotes the actuators torques vector. Equation (15) gives us the expression of τ_r as a function of the optimization variables q and λ, $\tau_r(q,\lambda)$, and allows us to write the inequality constraint on the maximum torques, denoting $\tau_{r,\mu}$ the μ-th component of τ,

$$\forall \mu \in \{1,\dots,j_r\} \quad |\tau_{r,\mu}(q,\lambda)| \leq \tau_{r,\mu,\max}. \tag{16}$$

D. Objective function

The objective function to minimize in problem (3) $\mathrm{obj}(q,\lambda)$ can be chosen in different ways depending on the application we are targeting. One typical choice is a quadratic form

$$\mathrm{obj}(q,\lambda) = (q - q_{\mathrm{ref}})^T A (q - q_{\mathrm{ref}}) + \lambda^T B \lambda,$$

if we want to minimize contact forces, or,

$$\mathrm{obj}(q,\lambda) = (q - q_{\mathrm{ref}})^T A (q - q_{\mathrm{ref}}) + \sum_r \tau_r(q,\lambda)^T C\, \tau_r(q,\lambda),$$

if we want to minimize actuators torques, q_{ref} being a reference configuration given manually as an input and used to drive the solution towards a goal as well as to produce natural-looking solutions, which is a fundamental concern for a humanoid robot. Within the planning context this reference configuration is taken from a guide path as computed in [12]. A, B, C are positive semi-definite matrices. Practically we choose diagonal matrices, the coefficients of which are tuned to weight the different objectives.

IV. Gradients derivations

Both state-of-the-art non-linear constrained optimization algorithms we have used, feasible sequential quadratic programming [13] and interior-point filter line-search [14], require that we provide them with the gradients of the objective and constraints functions. In this section we give details on these non-trivial gradient derivations. The gradients of all the functions with respect to λ are straightforward to derive, let us focus on the gradients with respect to q.

A. Geometric Jacobians

All the geometric gradients that we need to compute are down to the expressions of the $\mathbb{R}^{3\times(7+j_r)}$ matrices

$$\frac{\partial O_{r,b}(q_r)}{\partial q_r}, \frac{\partial [R_{r,b}(q_r)\, u]}{\partial q_r},$$

where u is any fixed vector of $\mathbb{R}^3$. The objective here is to derive these expressions relying only on the kinematic Jacobian of the body b with respect to the root body 0 of the robot r, for which algorithms can be found in standard textbooks such as [15]. Let us denote this kinematic Jacobian $J^k_{r,b} \in \mathbb{R}^{3\times j_r}$, its μ-th column

$$J^{k,\mu}_{r,b}(q_r) = \begin{bmatrix} \xi^\mu_{r,b}(q_r) \\ \omega^\mu_{r,b}(q_r) \end{bmatrix}$$

is the concatenation of the linear and angular velocities of the frame $\mathscr{T}_{r,b}$ with respect the the frame $\mathscr{T}_{r,0}$ expressed in this latter frame, corresponding to a unit velocity of the joint μ, $\dot\theta_{r,\mu} = 1$. If ρ denotes the mapping from unit quaternions to rotation matrices, i.e.

$$\rho(\alpha,\beta,\gamma,\delta) = \begin{pmatrix} 2(\alpha^2+\beta^2)-1 & 2(\beta\gamma-\alpha\delta) & 2(\beta\delta+\alpha\gamma) \\ 2(\beta\gamma+\alpha\delta) & 2(\alpha^2+\gamma^2)-1 & 2(\gamma\delta-\alpha\beta) \\ 2(\beta\delta-\alpha\gamma) & 2(\gamma\delta+\alpha\beta) & 2(\alpha^2+\delta^2)-1 \end{pmatrix},$$

then we can write

$$\frac{\partial O_{r,b}(q_r)}{\partial [x_r, y_r, z_r]} = \mathbb{1}_{3\times 3}$$
$$\frac{\partial O_{r,b}(q_r)}{\partial [\alpha_r, \beta_r, \gamma_r, \delta_r]} = \left(\frac{\partial \rho}{\partial \alpha} O^0_{r,b}, \frac{\partial \rho}{\partial \beta} O^0_{r,b}, \frac{\partial \rho}{\partial \gamma} O^0_{r,b}, \frac{\partial \rho}{\partial \delta} O^0_{r,b} \right)$$
$$\frac{\partial O_{r,b}(q_r)}{\partial \theta_r} = R_{r,0}\, \xi_{r,b}$$
$$\frac{\partial [R_{r,b}(q_r)\, u]}{\partial [x_r, y_r, z_r]} = \mathbb{O}_{3\times 3}$$
$$\frac{\partial [R_{r,b}(q_r)\, u]}{\partial [\alpha_r, \beta_r, \gamma_r, \delta_r]} = \left(\frac{\partial \rho}{\partial \alpha} R^0_{r,b} u, \frac{\partial \rho}{\partial \beta} R^0_{r,b} u, \ldots, \frac{\partial \rho}{\partial \delta} R^0_{r,b} u \right)$$
$$\frac{\partial [R_{r,b}(q_r)\, u]}{\partial \theta_r} = \left(R_{r,0} \left[\omega^\mu_{r,b} \times (R^0_{r,b}\, u) \right] \right)_{\mu \in \{1,\ldots,j_r\}},$$

where we have used the following notation

$$R^0_{r,b} = R^T_{r,0}\, R_{r,b}.$$

B. Torques gradients

Let us now derive the gradient of the constraint (16) for which the main difficulty resides in the derivation of

$$J_1 = \frac{\partial J^\mu_{r_{i2},b_{i2}} \left(q_{r_{i2}}, p'_{r_{i1},b_{i1},v}(q_{r_{i1}}, q_{r_{i2}}) \right)}{\partial q_{r_{i1}}},$$

$$J_2 = \frac{\partial J^\mu_{r_{i2},b_{i2}} \left(q_{r_{i2}}, p'_{r_{i1},b_{i1},v}(q_{r_{i1}}, q_{r_{i2}}) \right)}{\partial q_{r_{i2}}}.$$

where $J^\mu_{r,b}(q_r, p)$ is the μ-th column of the matrix defined in (12). Let us denote $D_{q_r} J^\mu_{r,b}$ and $D_p J^\mu_{r,b}$ respectively the partial derivatives of $J^\mu_{r,b}(q_r, p)$ with respect to q_r and to p. We can write (we temporarily drop the subscripts of p')

$$J_1 = D_p J^\mu_{r_{i2},b_{i2}}(q_{r_{i2}}, p') \frac{\partial p'(q_{r_{i1}}, q_{r_{i2}})}{\partial q_{r_{i1}}},$$
$$J_2 = D_{q_r} J^\mu_{r_{i2},b_{i2}}(q_{r_{i2}}, p') + D_p J^\mu_{r_{i2},b_{i2}}(q_{r_{i2}}, p') \frac{\partial p'(q_{r_{i1}}, q_{r_{i2}})}{\partial q_{r_{i2}}}.$$

We skip the details of the derivations of

$$\frac{\partial p'(q_{r_{i1}}, q_{r_{i2}})}{\partial q_{r_{i1}}}, \frac{\partial p'(q_{r_{i1}}, q_{r_{i2}})}{\partial q_{r_{i2}}},$$

that can be shown to have similar structures as the geometric gradients exposed in the previous section IV-A, and we concentrate on the derivations of $D_{q_r} J^\mu_{r,b}$ and $D_p J^\mu_{r,b}$. First, for $D_{q_r} J^\mu_{r,b}$, we can write

$$\frac{\partial J^\mu_{r,b}}{\partial [x_r, y_r, z_r]} = \mathbb{O}_{3\times 3}$$
$$\frac{\partial J^\mu_{r,b}}{\partial [\alpha_r, \beta_r, \gamma_r, \delta_r]} = \left(\frac{\partial \rho}{\partial \alpha} \xi^\mu_{r,b}(p), \frac{\partial \rho}{\partial \beta} \xi^\mu_{r,b}(p), \ldots, \frac{\partial \rho}{\partial \delta} \xi^\mu_{r,b}(p) \right)$$
$$\frac{\partial J^\mu_{r,b}}{\partial \theta_r} = R_{r,0} \left(\frac{\partial \xi^\mu_{r,b}(p)}{\partial \theta_{r,\nu}} \right)_{\nu \in \{1,\ldots,j_r\}},$$

where

$$\xi^\mu_{r,b}(p) = \xi^\mu_{r,b} + \omega^\mu_{r,b} \times \left[R^0_{r,b}\, p \right]$$

TABLE I: Some figures

	Circus	Coll.	Ladder
$\dim(q)$	94	101	47
$\dim(\lambda)$	48	96	48
total dimension	142	197	95
num. of eq. constr.	34	61	27
num. of ineq. const.	80	80	40
num. of iterations	30	42	19
optim. algo. time	0.732s	1.423s	0.280s
func. & grad. eval. time	7.190s	9.515s	1.454s

is the velocity transported from the origin of the frame $\mathscr{T}_{r,b}$ to the point p, and

$$\frac{\partial \xi^\mu_{r,b}(p)}{\partial \theta_{r,\nu}} = \begin{cases} \omega^\nu_{r,b} \times \xi^\mu_{r,b}(p) & \text{if } \nu < \mu, \\ \omega^\mu_{r,b} \times \xi^\mu_{r,b}(p) & \text{if } \nu = \mu, \\ \omega^\mu_{r,b} \times \xi^\nu_{r,b}(p) & \text{if } \nu > \mu. \end{cases}$$

This latter result is a straightforward generalization of the result published in [16] from serial kinematic chains to kinematic trees such as a humanoid robot. Now that we have derived $D_{q_r} J^\mu_{r,b}$ let us derive $D_p J^\mu_{r,b}$. We can simply write

$$D_p J^\mu_{r,b} = R_{r,0}\, \tilde{\omega}^\mu_{r,b}\, R^0_{r,b}$$

where $\tilde{\omega}^\mu_{r,b}$ is the skew-symmetric matrix corresponding to the vector product by $\omega^\mu_{r,b}$. This brings our derivations to an end.

V. Results

We have tested our static stance inverse solver on different theoretic scenarios in virtual environments involving one or two humanoid robots (for the robot we used a model of HRP-2 [17]) conjointly manipulating objects and taking unilateral or bilateral contacts, see Fig. 3. Our implementation being generic and totally transparent to the robot model, any other robot could have been used with no additional model-specific implementation effort. Of course some of these scenarios are not meant to be simulated or executed on real-life robots but we choose them to illustrate the generality of our approach from the conceptual point-of-view.

Within multi-contact planning queries made with a planner similar to [1], no local minima problems were encountered. This is mainly due to the fact that during the stances exploration phase, i.e. when growing the search tree, we use the resulting configuration from the father stance node as an initial guess for testing a new stance with our solver and add it to the tree in case of success. Care should thus be taken only when choosing the very first configuration initializing the search tree.

Table I gives some figures[2] concerning queries on these scenarios made with the solver [14] on a standard 3.06 GHz computer. As we can see most of the computation time

[2]The number of inequality contraints do not include the bounds on joints articulations nor the positivity conditions on λ for unilateral contacts as these bounds are handled directly as limits on the optimization variables by the solver.

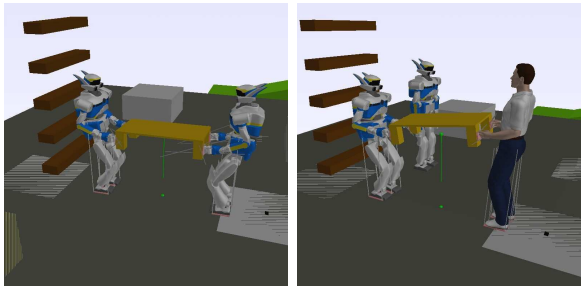
is spent on functions and gradients evaluations and can be greatly reduced, given that our current implementation splits vector constraints into individual scalar constraints and thus wastes a lot of time in redundant computations that can be factorized when using vector constraints. However, although computational time appears to be quite heavy (still being of same order of magnitude as the times reported in [1][2] for more complex problems in our case), it allows for solving multi-contact planning queries in times comparable to those of the aforementioned state-of-the-art planners, i.e. tens of minutes in average, while being more generic and handling a broader range of contact situations.

VI. Conclusion and future work

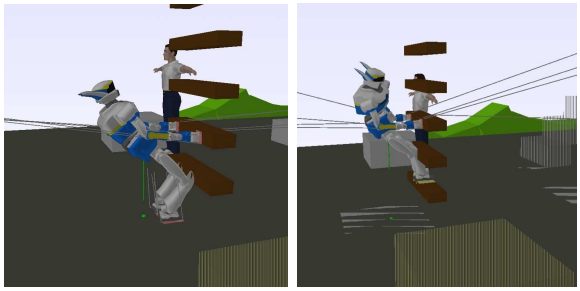
We provided a formulation for the multiple robots, multiple objects, multiple contacts, static stance inverse problem. The problem has been written as an optimization problem in the geometric and statics variables conjointly. Analytical gradients based on the kinematic Jacobian and its derivatives have been derived. We have tested our approach on very high dimensional challenging scenarios for which solutions were found in a relatively small number of iterations. These results are currently used in acyclic multi-contact motion planning for multiple agents. One missing piece of this work is the collision avoidance constraints, on the implementation of which we are currently working. A possible extension of this work is considering deformable bodies of the robots or the environment. In the longer term, non-static (kinetic) friction model can also be considered allowing displacement of the environment objects under the action of contact forces. We are currently investigating these topics.

Acknowledgement

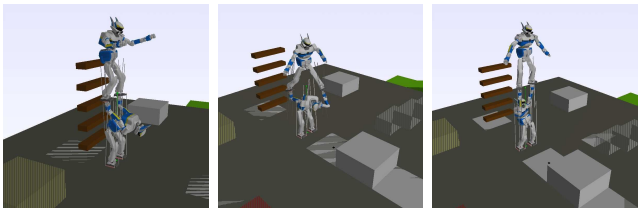
This work is partially supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (B), 22300071, 2010.



(a) Collaborative object manipulation scenarios with bilateral contacts between the hands and the table.



(b) Ladder climbing scenarios with unilateral contacts at the feet and bilateral contacts at the hands.



(c) Circus scenario involving only unilateral contacts, for different initial guesses.

Fig. 3: Example scenarios

References

[1] A. Escande, A. Kheddar, and S. Miossec, "Planning support contact-points for humanoid robots and experiments on hrp-2," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2974–2979.

[2] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, November-December 2008.

[3] S. M. LaValle, J. Yakey, and L. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.

[4] L. Han and N. M. Amato, "A kinematics-based probabilistic roadmap method for closed chain systems," in *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, 2000.

[5] J. Cortes, T. Siméon, and J.-P. Laumond, "A random loop generator for planning the motions of closed kinematic chains using prm methods," in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, 2002.

[6] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 794–807, August 2008.

[7] E. Rimon, R. Mason, J. W. Burdick, and Y. Or, "A general stance stability test based on stratified morse theory with applicaiton to quasi-static locomotion planning," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 626–641, June 2008.

[8] S. P. Boyd and B. Wegbreit, "Fast computation of optimal contact forces," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1117–1132, December 2007.

[9] M. Benallegue, A. Escande, S. Miossec, and A. Kheddar, "Fast c1 proximity queries using support mapping of sphere-torus-patches bounding volumes," in *IEEE Int. Conf. on Robotics and Automation*, 2009.

[10] O. Khatib, L. Sentis, and J. Park, "A unified framework for whole-body humanoid robot control with multiple constraints and contacts," in *European Robotics Symposium*, 2008, pp. 303–312.

[11] D. Raunhardt and R. Boulic, "Motion constraint," *The Visual Computer*, vol. 25, no. 5-7, pp. 509–518, April 2009.

[12] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar, "Potential field guide for multicontact humanoid motion planning," in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, 2009.

[13] C. T. Lawrence and A. L. Tits, "Nonlinear equality constraints in feasible sequential quadratic programming," *Optimization Methods and Software*, vol. 6, pp. 265–282, 1996.

[14] A. Wachter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[15] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*. Kogan Page Limited, 2005.

[16] H. Bruyninckx and J. D. Schutter, "Symbolic differentiation of the velocity mapping for a serial kinematic chain," *Mechanism And Machine Theory*, vol. 31, no. 2, pp. 135–148, 1996.

[17] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid robot hrp-2," in *IEEE International Conference on Robotics and Automation*, 2004.