2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

A Grasping Force Optimization Algorithm with Dynamic Torque Constraints Selection for Multi-Fingered Robotic Hands

Vincenzo Lippiello, Bruno Siciliano and Luigi Villani

Abstract— **The problem of grasping force optimization (GFO) for a multi-fingered robotic hand is considered in this paper. The GFO problem is cast in a convex optimization problem, considering also joint torque constraints. A new algorithmic solution is proposed here, which is suitable to be implemented online. The proposed formulation allows a substantial reduction of the computational load of the problem by dynamically decreasing the number of active torque constraints. Moreover, differently from other approaches, the algorithm does not require the evaluation of a new initial point at the beginning of each optimization cycle. The effectiveness of the proposed method has been tested in a simulation case study where the grasping forces of a five-fingers robotic hand are modified online to cope with time-varying external forces applied to the object.**

I. INTRODUCTION

An important issue in controlling a multi-fingered robotic hand grasping an object is the synthesis of the optimal contact points and the evaluation of the minimal contact forces able to guarantee the stability of the grasp and its feasibility. This latter problem, known as grasping force optimization (GFO) can be formulated as a constrained optimization problem.

Due to the considerable computational time requested to find the solution, the GFO is usually performed off-line. However, during the execution of a manipulation task, the position of the contact points on the object, or the wrench (force and moment) to be balanced by the contact forces, may change with time and cannot be planned in advance. The contact forces must be compatible with the friction constraints depending on the type of contact as well as with the joint torque limits of the fingers. In these cases, suitable algorithms for online computation of the solution of the constrained GFO problem must be devised.

Notice that the force closure problem [1] and, more in general, the problem of computing the optimal grasp configurations [2], are not considered here because it is assumed that the contact points, which guarantees the force closure property, are assigned by the path planner.

The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

The authors are with PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, via Claudio 21, 80125, Naples, Italy vincenzo.lippiello@unina.it, siciliano@unina.it, lvillani@unina.it

The nonlinearity of the contact friction models (point contact with friction or soft-finger contact) is one of the main reasons of the computational complexity of GFO problem. The analysis and synthesis of frictional force-closure grasps has been initially studied by linearizing the friction cone constraints and then applying linear programming techniques [3], [4], [5]. The corresponding problems, however, are ill-conditioned. Nonlinear programming techniques have been also investigated [6], but with results not suitable for real-time applications.

The use of Lagrangian neural networks is investigated in [7], [8]. These neural networks are capable to take into account the nonlinearity of the friction constraints and the joint torque limits, and asymptotically converge to a set of optimal grasping forces. In [9], [10] a method based on the minimization of a cost function, which gives an analytical solution but does not ensure by itself the satisfaction of the friction constraints is presented. An iterative correction algorithm allows to modify this function until the internal forces enter the friction cone, resulting in a fast sub-optimal solution suitable for real-time applications.

The GFO problem was formulated as a convex optimization problem on a Riemannian manifold with linear constraints in [11]. A number of gradient flow type algorithms have been proposed to provide solutions suitable for real-time applications [12]; to reduce the complexity of the matrix inversion, the computation of the solution can be split into one on-line and one off-line phase and sparse matrix techniques can be adopted [13].

In [14] the friction cone constraints have been formulated in terms of linear matrix inequalities (LMIs), and the grasping optimization problem is addressed as a convex optimization problem involving LMIs with the max–det function as objective function. This problem can be efficiently solved with the interior point algorithm for a small number of fingers. Moreover, joint torque limits can be considered and formulated in the same framework as LMIs.

A further improvement has been presented in [15], consisting in a new compact semidefinite representation of the friction cone constraints which allows a significant reduction of the dimension of the optimization problem. Moreover, an estimation technique and a recursion method for selecting the step size in the gradient algorithm are proposed, together with the proof of the quadratic convergence of the algorithm.

The method proposed in [11] has the main disadvantage that it requires the on-line pseudo-inversion of a structurally constrained matrix whose dimension linearly increases with the number of fingers, of a factor that depends on the

978-1-4577-0079-8/11/$26.00 ©2011 AACC

contact type. By adopting the frictional cone constraint matrix representation proposed in [15], the dimension of the problem decreases considerably so that the solution can be computed in real time. However, if torque limits constraints are considered, the complexity of the problem increases more than quadratically with the number of joints, making it unsuitable for real-time applications. Moreover, all the proposed solutions require, at each iteration, the evaluation of an initial point that satisfies the frictional cone constraints and the joint torque limits. The initial point can be computed with the method proposed in [16], but at the expense of a significant computational effort.

The algorithm proposed here, based on the compact formulation of [15] and on the solution of a convex optimization problem as in [12], allows to consider also torque joint constraints. These constraints are activated dynamically, with a minimum increase of computation complexity, compatible with real-time applications. As a further improvement, the proposed iterative formulation does not require the evaluation at each step of a new initial point.

The feasibility and the effectiveness of the proposed method have been tested in a simulation case study, where the contact forces of a five-fingers hand grasping an object are computed online to cope with time-varying external forces applied to the object.

II. PROBLEM FORMULATION

Consider a multi-fingered robotic hand grasping an object with n contacts between the object and the fingertips, the links of the fingers and the palm. Denote the contact wrench of the grasp with $\boldsymbol{c} = \left[\begin{array}{ccc} \boldsymbol{c}_1^{\mathrm{T}} & \dots & \boldsymbol{c}_n^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^{nm}$, where $\boldsymbol{c}_i \in \mathbb{R}^m$ is the wrench vector of the i-th contact with dimension m depending from the adopted contact model.

The grasping force optimization problem (GFO) consists in finding the set of contact wrenches balancing the generalized external force $\boldsymbol{h}_e \in \mathbb{R}^6$ acting on the object (including object inertia and weight), which are feasible with respect to the kinematic structure of the hand and to the corresponding joint torque limits, and minimize the overall stress applied the object, i.e, the internal forces. Moreover, to avoid the slippage of the fingers on the object surface, each contact wrench has to be confined within the friction cone.

The balance equation for the generalized forces applied to the object can be written in the form

$$\boldsymbol{h}_e = \boldsymbol{G}\boldsymbol{c}, \tag{1}$$

where $\boldsymbol{G} \in \mathbb{R}^{6\times nm}$ is the grasp map, which is full rank for force-closure grasps [1]. It is assumed that the contact point configurations ensuring the force-closure constraint are assigned at each time by the planning system.

Although several contact models can be used, the two usually adopted are the *point contact with friction* (PCWF) model and the *soft finger contact* (SFC) model.

In the PCWF case, the contact wrench has three degree-of-freedom ($m = 3$): the normal component $c_{i,z}$ to the object surface and the two components $c_{i,x}$, $c_{i,y}$ on the tangent plane. The friction constraint is represented by the law

$$\frac{1}{\mu_i^2}\left(c_{i,x}^2 + c_{i,y}^2\right) \leq c_{i,z}^2 \text{ and } c_{i,z} > 0, \tag{2}$$

where μ_i denotes the friction coefficient at the i-th contact point.

In the SFC case, the contact wrench has an additional degree-of-freedom $c_{i,t}$ ($m = 4$), corresponding to the torsional component of the moment about the contact normal. In this case, the friction constraint in an elliptic approximation can be expressed in the form

$$\frac{1}{\mu_i}\left(c_{i,x}^2 + c_{i,y}^2\right) + \frac{1}{\mu_{t,i}}c_{i,t}^2 \leq c_{i,z}^2 \text{ and } c_{i,z} > 0, \tag{3}$$

where μ_i and $\mu_{i,t}$ denote the tangential and the torsion friction coefficients at the i-th contact point, respectively.

The balance equation for the torques applied to fingers joints of the hand can be written in the form

$$\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{c} + \boldsymbol{\tau}_e = \boldsymbol{\tau}, \tag{4}$$

where $\boldsymbol{\tau}_e$ is the external torque, including gravity, Coriolis, centripetal and inertia effects at the fingers joints, $\boldsymbol{\tau}$ is the torque provided by the actuators, and $\boldsymbol{J}(\boldsymbol{q})$ is the $(nm \times l)$ hand Jacobian matrix, depending on the (l-dimensional) vector $\boldsymbol{q}$ of the joint variables, being l the total number of the hand joints. For simplicity, it is assumed that $\mathcal{N}(\boldsymbol{J}^{\mathrm{T}}) = \emptyset$, meaning the absence of structurally dependent forces, namely, contact forces not caused by joint torques but depending on hand mechanics (see, e.g., [1]).

To ensure that the joint actuators are able to provide the required torques, a joint torque constraint must also be considered

$$\boldsymbol{\tau}_L \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_U, \tag{5}$$

where $\boldsymbol{\tau}_L$ and $\boldsymbol{\tau}_U$ denote the lower and upper joint torque bound, respectively.

The simultaneous satisfaction of the force balance equation (1), with the friction constraints (2) and (3), and of the joint torque balance equation (4) with constraint (5), implies that the grasp is stable and feasible.

The GFO problem considered here consists in finding the optimal grasp wrench that minimize the internal forces acting on the object, under the above constraints. The internal forces are contact wrenches that satisfy the friction cone constraints and belong to the null space of the grasp matrix $\boldsymbol{G}$. These wrenches $\boldsymbol{c}_{int}$ do not contribute to the balance equation (1), being $\boldsymbol{G}\boldsymbol{c}_{int} = 0$, but are used to satisfy the friction cone constraints at the contact points.

III. GRASPING CONSTRAINTS AND SEMIDEFINITE PROGRAMMING

As shown in [15], the frictional inequalities (2) and (3) are equivalent to the the positive definiteness of the block-diagonal matrix

$$\boldsymbol{F}(\boldsymbol{c}) = \mathrm{diag}\left(\boldsymbol{F}_1(\boldsymbol{c}_1), \dots, \boldsymbol{F}_n(\boldsymbol{c}_n)\right) > 0, \tag{6}$$

where $\boldsymbol{F}_i(\boldsymbol{c}_i)$ is the symmetric (2×2) matrix

$$\boldsymbol{F}_i(\boldsymbol{c}_i) = \begin{bmatrix} c_{i,z} + \frac{c_{i,x}}{\mu_i} & \frac{c_{i,y}}{\mu_i} \\ \frac{c_{i,y}}{\mu_i} & c_{i,z} - \frac{c_{i,x}}{\mu_i} \end{bmatrix} \tag{7}$$

in the PCWF case, while it is the Hermitian (2×2) matrix

$$\boldsymbol{F}_i(\boldsymbol{c}_i) = \begin{bmatrix} c_{i,z} + \frac{c_{i,x}}{\sqrt{\mu_i}} & \frac{c_{i,y}}{\sqrt{\mu_i}} - j\frac{c_{i,t}}{\sqrt{\mu_{i,t}}} \\ \frac{c_{i,y}}{\sqrt{\mu_i}} + j\frac{c_{i,t}}{\sqrt{\mu_{i,t}}} & c_{i,z} - \frac{c_{i,x}}{\sqrt{\mu_i}} \end{bmatrix}, \tag{8}$$

in the SFC case.

Similarly, the torque limit constraint (5), in view of the torque balance equation (4), is equivalent to the positive definiteness of the diagonal matrix

$$\boldsymbol{T}(\boldsymbol{c}, \boldsymbol{q}, \boldsymbol{\tau}_e) = \mathrm{diag}\,(\boldsymbol{\tau}_B) > 0, \tag{9}$$

where

$$\boldsymbol{\tau}_B = \begin{bmatrix} \boldsymbol{\tau}_{B,L} \\ \boldsymbol{\tau}_{B,H} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{c} - \boldsymbol{\tau}_L + \boldsymbol{\tau}_e \\ -\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{c} + \boldsymbol{\tau}_H - \boldsymbol{\tau}_e \end{bmatrix} \tag{10}$$

contains the gaps of actuator torques from the lower ($\boldsymbol{\tau}_{B,L}$) and upper ($\boldsymbol{\tau}_{B,H}$) limit, respectively.

Hence, the simultaneous satisfaction of both frictional and joint torque constraints is equivalent to the positive definiteness of the linearly constrained block-diagonal matrix

$$\boldsymbol{P} = \mathrm{diag}\,(\boldsymbol{F}, \boldsymbol{T}) > 0. \tag{11}$$

Notice that the elements of the matrices $\boldsymbol{F}$ and $\boldsymbol{T}$ are linearly dependent, because both depend on $\boldsymbol{c}$. Moreover, the force balance equation (1) and the torque balance equation (4) corresponds to linear constraints imposed on matrix $\boldsymbol{P}$.

By denoting with $\boldsymbol{c}(\boldsymbol{F})$ the contact wrench vector extracted from the frictional constraint matrix, with $\boldsymbol{\tau}_B(\boldsymbol{T})$ the vector composed by the diagonal elements of $\boldsymbol{T}$, and defining vector $\boldsymbol{\xi}(\boldsymbol{P}) = \left[\boldsymbol{c}(\boldsymbol{F})^{\mathrm{T}}, \boldsymbol{\tau}_B(\boldsymbol{T})^{\mathrm{T}}\right]^{\mathrm{T}}$, the linear constraints on matrix $\boldsymbol{P}$ imposed by (1) and (4) can be represented in the following affine general form

$$\boldsymbol{A}\boldsymbol{\xi}(\boldsymbol{P}) = \boldsymbol{b} \tag{12}$$

with

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{0}_{6\times 2l} \\ \boldsymbol{A}_\tau & \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{h}_e \\ \boldsymbol{\tau}_L - \boldsymbol{\tau}_e \\ \boldsymbol{\tau}_H - \boldsymbol{\tau}_e \end{bmatrix}, \tag{13}$$

where $\boldsymbol{A}_\tau$ is a $(2l \times nm + 2l)$ matrix defined as follows

$$\boldsymbol{A}_\tau = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{q})^{\mathrm{T}} & -\boldsymbol{I}_l & \boldsymbol{0}_l \\ \boldsymbol{J}(\boldsymbol{q})^{\mathrm{T}} & \boldsymbol{0}_l & \boldsymbol{I}_l \end{bmatrix}, \tag{14}$$

being $\boldsymbol{0}_\times$ the null matrix and $\boldsymbol{I}_\times$ the identity matrix of the indicated dimensions.

The optimization procedure is based on the minimization of the cost function $\Phi(\boldsymbol{P}) : \mathcal{P}(r) \to \mathbb{R}$, being $\mathcal{P}(r)$ the set of positive definite symmetric $r \times r$ matrices $\boldsymbol{P} = \boldsymbol{P}^{\mathrm{T}} > 0$, defined as

$$\Phi(\boldsymbol{P}) = \mathrm{tr}\left(\boldsymbol{W}_p\boldsymbol{P} + \boldsymbol{W}_b\boldsymbol{P}^{-1}\right), \tag{15}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator, $\boldsymbol{W}_p$ and $\boldsymbol{W}_b$ are symmetric positive definite matrices. Notice that Φ is a strictly convex twice continuously differentiable function on $\mathcal{P}(r)$ and $\Phi(\boldsymbol{P}) \to +\infty$ for $\boldsymbol{P} \to \partial\mathcal{P}(r)$, being $\partial\mathcal{P}(r)$ the boundary of $\mathcal{P}(r)$.

By noting that the sum of the elements of $\boldsymbol{T}$ (i.e. of $\boldsymbol{\tau}_B$) is constant for each $\boldsymbol{c}$, because the sum of the two joint torque constraints for the i-th joint is constant and equal to $\tau_{H,i} - \tau_{L,i}$, the diagonal weighting matrix $\boldsymbol{W}_p = \mathrm{diag}(w_p\boldsymbol{I}_6, \boldsymbol{0}_{2l})$, with $w_p > 0$, is considered. In this way, the term $\boldsymbol{W}_p\boldsymbol{P}$ weights only the normal forces $\boldsymbol{c}_{i,z}$ at each contact point, i.e. the pressure forces on the object. If required, different weights can be used allowing higher contact forces for strongest fingers.

The second term $\boldsymbol{W}_b\boldsymbol{P}^{-1}$ represents a barrier function, which goes to infinity when $\boldsymbol{P}$ tends to a singularity, i.e. when friction or torque limits are approached. The barrier weight matrix is also chosen diagonal $\boldsymbol{W}_b = \mathrm{diag}(\boldsymbol{W}_{b,F}, \boldsymbol{W}_{b,T})$, with

$$\begin{aligned} \boldsymbol{W}_{b,F} &= w_{b,F}\mathrm{diag}\,(\mu_1, \ldots, \mu_n) \\ \boldsymbol{W}_{b,T} &= w_{b,T}\mathrm{diag}\,(\tau_{H,1} - \tau_{L,1}, \ldots, \tau_{H,l} - \tau_{L,l}, \\ &\qquad \tau_{H,1} - \tau_{L,1}, \ldots, \tau_{H,l} - \tau_{L,l}), \end{aligned} \tag{16}$$

being $w_{b,F} > 0$ and $w_{b,T} > 0$.

Hence, the minimization of the cost function (15) with the linear constraint (12), corresponds to the minimization of the normal contact wrench components applied to the object while satisfying the friction and torque constraints. This problem can be solved using the linearly constrained gradient flow approach on the smooth manifold of positive definite matrices presented in [17], and applied in [11] to a GFO problem. In particular, it is possible to prove that $\Phi(\boldsymbol{P})$ presents a unique minimum that can be reached through the linear constrained exponentially convergent gradient flow

$$\dot{\boldsymbol{\xi}}(\dot{\boldsymbol{P}}) = \boldsymbol{Q}\boldsymbol{\xi}(\boldsymbol{P}^{-1}\boldsymbol{W}_b\boldsymbol{P}^{-1} - \boldsymbol{W}_p), \tag{17}$$

where $\boldsymbol{Q} = (\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})$ is the linear projection operator onto the tangent space of $\boldsymbol{A}$, and $\boldsymbol{A}^\dagger = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$ is the pseudo-inverse of $\boldsymbol{A}$. Consequently, $\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{0}$ and $\boldsymbol{A}\dot{\boldsymbol{\xi}}(\dot{\boldsymbol{P}}) = \boldsymbol{0}$; hence, if the solution satisfies the constraint (12) at $t = 0$, it will satisfy the constraint for all $t > 0$.

A discrete-time version of (17) based on the Euler numerical integration algorithm is

$$\boldsymbol{\xi}(\boldsymbol{P}_{k+1}) = \boldsymbol{\xi}(\boldsymbol{P}_k) + \alpha_k\boldsymbol{Q}\boldsymbol{\xi}(\boldsymbol{P}_k^{-1}\boldsymbol{W}_b\boldsymbol{P}_k^{-1} - \boldsymbol{W}_p), \tag{18}$$

where the step-size α_k is chosen to ensure down hill steps. Notice that the choice of α_k strongly affects the performance of the optimization algorithm. A wrong choice could determine a very slow convergence or the break of the barrier. Several strategies have been proposed for the self-tuning of α_k at each iteration (see [16] for details). The sensitivity to the step-size choice can be reduced by adopting a Dikin-type recursive algorithm [12], [18], that leads to the discrete flow

$$\boldsymbol{\xi}(\boldsymbol{P}_{k+1}) = \boldsymbol{\xi}(\boldsymbol{P}_k) - \alpha_k\boldsymbol{Q}\frac{\boldsymbol{\xi}(\boldsymbol{P}_k^{-1}\boldsymbol{W}_b\boldsymbol{P}_k^{-1} - \boldsymbol{W}_p)}{\|\boldsymbol{P}_k^{-1}\boldsymbol{W}_b\boldsymbol{P}_k^{-1} - \boldsymbol{W}_p\|_{\boldsymbol{P}_k}}, \tag{19}$$

where $\|\boldsymbol{X}\|_{\boldsymbol{Y}} = \mathrm{tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}\boldsymbol{Y}^{-1}\boldsymbol{X})$, and $0 \leq \alpha_k \leq 1$ can be evaluated with a bounded line search minimizing $\Phi(\boldsymbol{P}_{k+1})$.

IV. ALGORITMIC IMPROVEMENTS

The online implementation of the proposed algorithm presents the disadvantage of requiring the inversion of a $(6+2l)$ square matrix $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ needed for the evaluation of $\boldsymbol{A}^{\dagger}$ at each iteration, also when the grasping configuration is unchanged, i.e. when $\boldsymbol{G}$ is constant, due to the variation of $\boldsymbol{J}(\boldsymbol{q})$. Moreover, at each iteration, the evaluation of a valid initial point is required for the correct initialization of the gradient flow algorithm.

In the following, some improvements are proposed to cope with the previous issues for an efficient recursive implementation of the proposed algorithm.

A. Affine Constraint Decomposition

Starting from the discrete version of the gradient flow (18), the following new formulation can be derived

$$\boldsymbol{c}_{k+1} = \boldsymbol{c}_k + \alpha_k \bar{\boldsymbol{Q}} \boldsymbol{\xi} (\boldsymbol{P}^{-1}(\boldsymbol{c}_k) \boldsymbol{W}_b \boldsymbol{P}^{-1}(\boldsymbol{c}_k) - \boldsymbol{W}_p), \quad (20)$$

where $\bar{\boldsymbol{Q}} = (\boldsymbol{I} - \boldsymbol{G}^{\dagger}\boldsymbol{G})[\boldsymbol{I}_{nm} \boldsymbol{0}_{2l}](\boldsymbol{I} - \boldsymbol{A}_{\tau}^{\dagger}\boldsymbol{A}_{\tau})$ is the result of the projection onto the tangent space of matrix $\boldsymbol{A}_{\tau}$ in (14), which guarantees the coherence of the elements of matrix $\boldsymbol{P}$, and of the subsequent projection onto the tangent space of the grasp matrix, ensuring the force balance constraint (1). Therefore, the evaluation of the inverse of a $6+2l$ matrix is decomposed into the evaluation of the inverse of two matrices of lower dimensions (6 and $2l$, respectively). Moreover, if the grasp configuration remains unchanged, the projector depending on $\boldsymbol{G}$ can be evaluated off-line.

A similar decomposition can be easily achieved for the gradient flow (19).

B. Dynamic Joint-Torque Constraints

In the reasonable hypothesis that the solutions of the recursive algorithm evaluated at successive sampling times are quite close, the above problem can be further simplified observing that not all the joint torque constraints can be effective simultaneously. For example, if, for the current solution, the actuator of joint i provides a torque close to the upper bound $\tau_{H,i}$, the constraint on the lower bound $\tau_{L,i}$ can be deactivated at the next sampling time, the corresponding barrier term in the cost function being negligible. Similarly, if, for a grasp configuration, it is required a given contact force along a certain direction, it is reasonable to assume that the corresponding joint torques will not change sign at the next sampling time, and thus the constraints on the bounds of opposite sign can be deactivated.

Starting from this observation, the number of the joint torque constraints can be dynamically reduced from $2l$ to l at each sampling time, by using the distance of the torque evaluated at the previous sampling time from the lower and upper bounds as the criterion for selecting the constraint (the lower or the upper one) that needs to be activated. A further simplification can be achieved by activating a constraint only if the corresponding distance is higher than a torque threshold, that can be choose as a fraction $\sigma_{\tau} > 0$ of the corresponding torque limit.

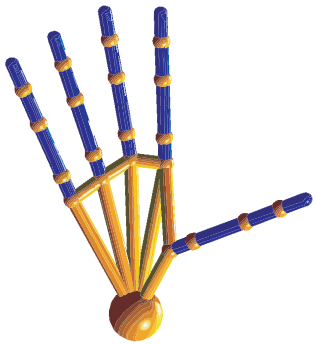

Fig. 1. DEXMART hand skeleton.

C. Initial Point Self-Evaluation

A new technique for the on-line evaluation, at each sampling time, of the initial point —the initial solution $\boldsymbol{P}_0$ for the optimization gradient flow algorithm— is proposed here, based on the optimal solution at the previous sampling time. The quantities that can vary between successive sampling times are the hand configuration $\boldsymbol{q}$, the external torque $\boldsymbol{\tau}_e$, and the grasp map $\boldsymbol{G}$, while they are taken constant during the iterations of the optimization algorithm between two consecutive sampling times (optimization cycle).

To avoid the evaluation of an initial point at each sampling time, the following approach is proposed. Initially, at time t_0, the method proposed in [16] (or an equivalent one) is used to evaluate off-line a first valid initial solution, which is employed for the first optimization cycle. For the next sampling times t_k, the initial point is computed from the optimal solution $\boldsymbol{c}_{k-1}$ computed at the end of the previous optimization cycle, through the iterative algorithm

$$\begin{aligned} \bar{\boldsymbol{c}}_j &= (\boldsymbol{I} - \boldsymbol{G}_k^{\dagger}\boldsymbol{G}_k)\bar{\boldsymbol{c}}_{j-1} + \gamma_j \boldsymbol{G}_k^{\dagger}\boldsymbol{h}_{e,k} + (1-\gamma_j)\boldsymbol{G}_{k-1}^{\dagger}\boldsymbol{h}_{e,k-1} \\ \bar{\boldsymbol{\tau}}_j &= \boldsymbol{J}^{\mathrm{T}}(\gamma_j \boldsymbol{q}_k + (1-\gamma_j)\boldsymbol{q}_{k-1})\bar{\boldsymbol{c}}_j + \gamma_j \boldsymbol{\tau}_{e,k} + (1-\gamma_j)\boldsymbol{\tau}_{e,k-1}, \end{aligned} \quad (21)$$

with initial condition $\bar{\boldsymbol{c}}_0 = \boldsymbol{c}_{k-1}$, where the subscript k is referred to the sampling time (i.e., to the current optimization cycle) while the subscript j and the variables with the bar are referred to the iterations within the cycle. The coefficient $\gamma_j \in (0,1]$ is chosen at each iteration according to a monotone sequence, using a simple linear search algorithm, as the maximum value that does not produce invalid solutions ($\boldsymbol{P}_0 \leq 0$). In the worst case, γ_0 must be set to a value close to zero.

In detail, at each step of the optimization cycle, the first equation of (21) gradually modifies the external wrench component of the current solution until the full external wrench $\boldsymbol{h}_{e,k}$ is balanced (i.e., $\gamma_j = 1$). Obviously, the optimization cycle cannot be terminated until γ_j does not reach 1. If the solution evaluated at the previous sampling time ($\boldsymbol{c}_{k-1}$) is sufficiently far from the boundaries (the distance depends also from the weights assigned to $\boldsymbol{W}_b$), γ_0 can be set to 1 at the first iteration, and thus the initial point has the same internal wrench component of the previous optimal solution. On the other hand, when $\gamma_0 < 1$, the effect of the barrier

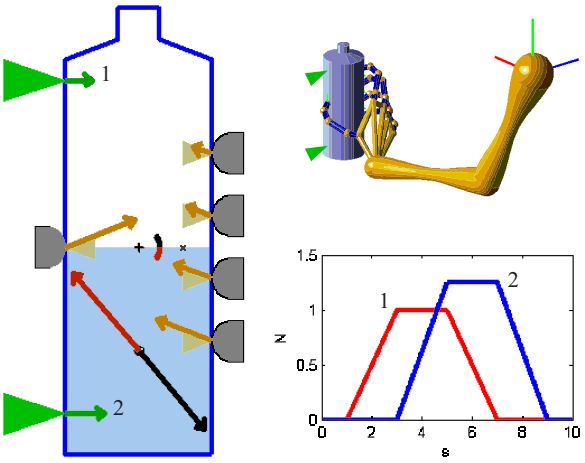

Fig. 2. *Left*: section of the bottle with graphical representation of the gravity force and torque (black arrows), of the external forces (green arrows), of the resultant force and torque applied by the fingers (red arrows), of the optimal contact forces (orange arrows), and of the friction cones (yellow triangles); the intensity of the forces and torques are proportional to the arrow lengths. *Top right*: Arm and hand grasping the bottle. *Bottom right*: Time history of the external forces acting on the bottle.

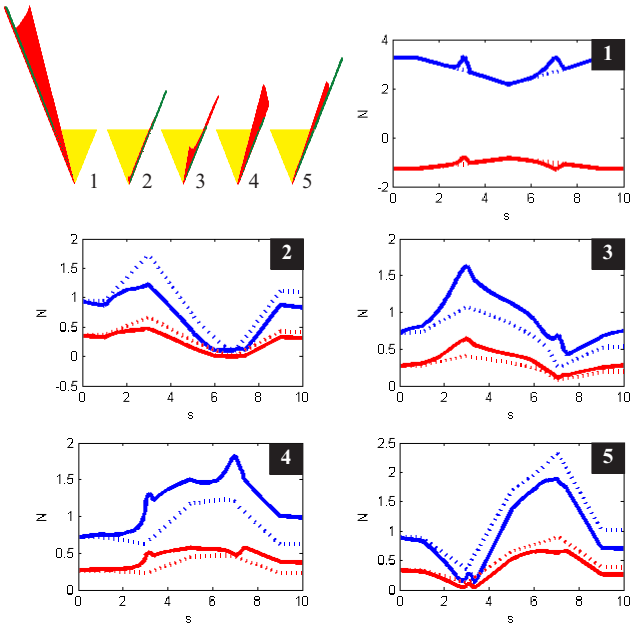

Fig. 3. Time histories of the contact forces ([N]) for the 5 fingers of the hand without (dotted lines) and with (continuous lines) torque constraints. Top left: Areas covered by the contact forces within the friction cones without (green color) and with (red color) torque constraints; the friction cones are represented in yellow.

function produces a new solution that, at each iteration of the optimization cycle, goes away from the boundaries; this guarantees that γ_j increases at each step, until $\gamma_j = 1$. The second equation is required to modify the joint torque with the same rationale of the first equation.

V. CASE STUDY

The proposed GFO algorithm has been tested in simulation using Matlab/Simulink and employing the DEXMART [19] anthropomorphic hand mounted on a 7-dof anthropomorphic arm as shown in Fig. 1 and Fig. 2.

It is assumed that the hand grasps a cylinder representing a bottle half filled with water with a total weight of 0.25 N with the main axis aligned to the vertical (gravitational) direction. The task consists in holding the bottle with an human-like grasp also in the presence of two external forces applied at the top-left and bottom-left side of the bottle with a time varying magnitude (see Fig. 2). Notice that a PCWF model is considered with a friction coefficient $\mu = 0.4$ equal for each contact point.

A section of the bottle half filled with water is shown in Fig. 2 where the intensity of the resultant force, sum of gravitational and external forces, is proportional to the black vertical arrow applied to the instantaneous center of mass, while the intensity of the resultant torque with respect to the center of the bottle is proportional to the black circular arrow. The external forces and the corresponding contact points are represented with green arrows and triangles, respectively. The red arrows are the force and torque resulting from the contact forces applied by the fingers, which are the orange arrows. The sections of the friction cones in the contact points are colored in yellow.

Two different simulations have been done. In the first simulation only the friction constraint is considered, without any constraint on the joint torque limits. In the second simulation, instead, different torque limits are set for the fingers, considering the thumb actuators 5 times stronger than the actuators of the other fingers (± 0.5 vs ± 0.1 Nm).

In Fig. 3 the time histories of the two significant contact force components for each finger are shown, represented in a frame fixed to the bottle and labeled from 1 to 5 from the thumb to the little finger. The dotted lines correspond to the first simulation, while the solid lines correspond to the second simulation. At the top left of the figure, the areas covered by the contact force vector of each contact point during the bottle motion is shown, in green color for the first simulation and in red color for the second simulation.

The time histories of the corresponding joint torques are shown in Fig. 4, where the dotted lines correspond to the first simulation, the solid lines correspond to the second simulation, and the gray horizontal lines represent the joint limits. The torques of the same finger are grouped together. Notice that there are 3 actuators for each fingers, being the last two joints of each finger coupled.

When only the friction constraints are imposed, fingers 2 (index) and 5 (little finger) are loaded more than the others by the GFO algorithm during the action of the external forces, while fingers 3 and 4 remain almost inactive. This can be easily explained because fingers 2 and 5 have moment arms with respect to the center of mass larger than the two central fingers, and are able to produce a large moment with a reduced contact force to balance the moment produced by the external forces. However, the joint torque limits are exceeded for both the fingers 2 and 5, resulting in an unfeasible grasp. When torque constraints are considered, fingers 3 and 4 are used more actively to sustain part of the load in charge to

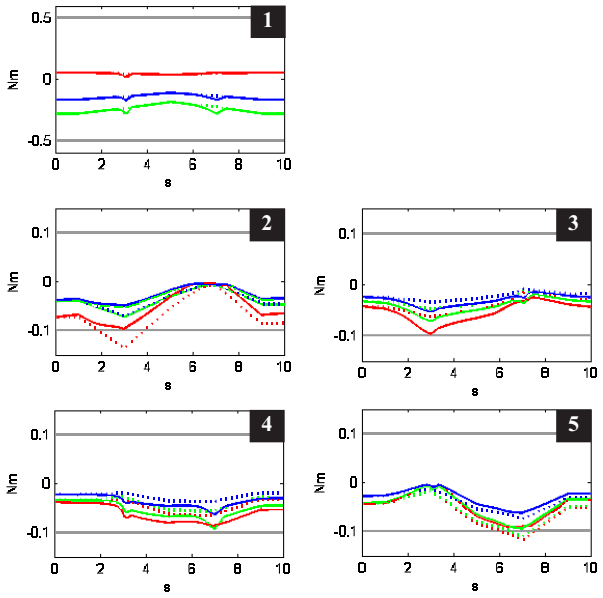

Fig. 4. Time history of the joint torques ([Nm]), for the 5 fingers of the hand without (dotted lines) and with (continuous lines) torque constraints. The gray horizontal lines represent the joint limits.

the adjacent fingers when the torque limits are approached, and thus all the torque constraints can be respected.

The benefits resulting from the adoption of the dynamic joint-torque constraints selection are showed in Fig. 5, where the time history of the computational time effort are represented on the left and the number of employed constraints are represented on the right. To remove the dependence from the employed hardware, all the considered cases are normalized with respect to the maximum value of the fully constrained case (black line) to the value 100. In particular, four different cases are compared: all constraints (black lines), $\sigma_\tau = 0$ (red lines), $\sigma_\tau = 0.5$ (green lines), $\sigma_\tau = 0.8$ (blue lines), and unconstrained (gray lines), where σ_τ is the threshold for the activation of the joint torque constraints. The achieved reduction of the mean of the computational time varies between a minimum of about 10% for $\sigma_\tau = 0$ to a maximum of about 80% for $\sigma_\tau = 0.8$.

VI. CONCLUSION

A new algorithm for dextrous-hand grasping force optimization has been presented in this paper, considering also joint torque limits. The computational charge of the algorithm has been considerably reduced by adopting an iterative formulation based on a dynamic set of active constraints and by avoiding the evaluation of the initial point at the beginning of each iteration, as required by existing approaches. A simulation case study has been proposed to show the feasibility and the effectiveness of the proposed technique in a case study, assuming that time-varying and unknown forces and moments are applied to the grasped object. Future work will be devoted to test the proposed technique in combination with a compliant control strategy for the arm-hand system in the presence of interaction with the environment or a human being.

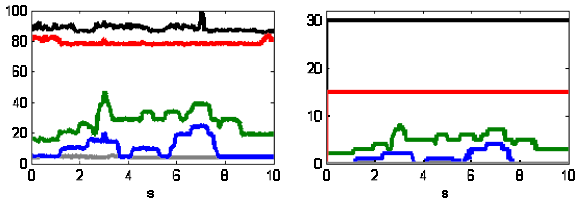

Fig. 5. Time history of the normalized computational-time effort (left) and of the number of employed joint torque constraints (right) for the cases of all constraints (black), $\sigma_\tau = 0$ (red), $\sigma_\tau = 0.5$ (green), $\sigma_\tau = 0.8$ (blue), and unconstrained (gray).

REFERENCES

[1] R. Murray, Z. X. Li and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, Boca Raton, FL: CRC, 1994.

[2] C. Xiong, H. Ding and Y. Xiong, *Fundamentals of Robotic Grasping and Fixturing*, World Scientific Publishing Company 2007.

[3] J. Kerr and B. Roth, "Analisys of Multifingered Hands", *International Journal of Robotics Research*, vol. 4, no. 4, pp. 3-17, 1986.

[4] F. T. Cheng and D.E. Orin, "Efficient Algorithm for Optimal Force Distribution–The Compact-dual LP Method", *IEEE Transaction on Robotics and Automation*, vol. 6, no. 2, pp. 178-187, 1990.

[5] Y. H. Liu, "Qualitative Test and Force Optimization of 3D Frictional Form-Closure Grasps Using Linear Programming", *IEEE Transaction on Robotics and Automation*, vol. 15, no. 1, pp. 163-173, 1999.

[6] Y. Nakamura, k. Nagai and T. Yoshikawa, "Dynamics and Stability in Coordination of Multiple Robotic Mechanisms", *International Journal of Robotics Research*, vol. 8, no. 2, pp. 45-61, 1989.

[7] W. S. Tang and J. Wang, "A Lagrangian Network for Multifingered Hand Grasp Force Optimization", *IEEE International Joint Conference on Neural Networks*, Hawaii, USA, 2002.

[8] L. M. Fok and J. Wang, "Two Recurrent Neural Networks for Grasping Force Optimization of Muulti-fingered Robotic Hands", *IEEE International Joint Conference on Neural Networks*, Hawaii, USA, 2002.

[9] C. Remond, V. Perdereau and M. Drouin, "A Multi-fingered Hand Control Structure with On-line Grasping Force Optimization", *IEEE/ASME International Confonference on Advanced Intelligent Mechatronics*, Como, Italy, 2001.

[10] J. P. Saut, C. Remond, V. Perdereau and M. Drouin, "Online computation of Grasping Force in Multi-fingered Hands", *IEEE/RSJ International Confonference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, 2005.

[11] M. Buss, H. Hashimoto and J. B. Moore, "Dextrous Hand Grasping Force Optimization", *IEEE Transaction on Robotics and Automation*, vol. 12, no. 3, pp. 406-418, 1996.

[12] M. Buss, L. Faybusovich and J. B. Moore, "Dikin-type Algorithms for Dextrous Grasping Force Optimization", *International Journal of Robotics Research*, vol. 17, no. 8, pp. 831-839, 1998.

[13] Z. X. Li, Z. Quin, S. Jiang and L. Han, "Coordinated Motion Generation and Real-Time Grasping Force Control for Multifingered Manipulation", *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[14] L. Han, J. C. Trinkle and Z. X. Li, "Grasp Analysis as Linear Matrix Inequality Problems", *IEEE Transaction on Robotics and Automation*, vol. 16, no. 6, pp. 663-674, 2000.

[15] U. Helmke, K. Hper and J. B. Moore, "Quadratically Convergent algorithms for Optimal Dexterous Hand Grasping", *IEEE Transaction on Robotics and Automation*, vol. 168 no. 2, pp. 138-146, 2002.

[16] G. Liu, J. Xu and Z. Li, "On Geometric Algorithms for Real-Time Grasping Force Optimization", *IEEE Transaction on Control System Technology*, vol. 12, no. 6, pp. 843-859, 2004.

[17] U. Helmke and J. B. Moore, *Optimization and Dynamic Systems*, Springer-Verlag, New York, 1993.

[18] L. Faybusovich, *Dikin's algorithm for matrix linear programming problems*, in LNCIS 197: System Modelling and Optimization (J. Henry and J.-P. Yvon, eds.), pp. 237-247, Springer, 1994.

[19] http://www.dexmart.eu