Robotica (2013) volume 31, pp. 1169–1194. © Cambridge University Press 2013
doi:10.1017/S0263574713000647

Grasp planning and parallel control of a redundant dual-arm/hand manipulation system

Fabrizio Caccavale†, Vincenzo Lippiello‡, Giuseppe Muscio†, Francesco Pierri†, Fabio Ruggiero‡,* and Luigi Villani‡

†*Scuola di Ingegneria, Università della Basilicata, via dell'Ateneo Lucano 10, Potenza 85100, Italy*
‡*PRISMA Lab, Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università di Napoli Federico II, via Claudio 21, Naples 80125, Italy*

(Accepted June 3, 2013. First published online: July 19, 2013)

SUMMARY
In this paper, a kinematic model of a dual-arm/hand robotic system is derived, which allows the computation of the object position and orientation from the joint variables of each arm and each finger as well as from a suitable set of contact variables. On the basis of this model, a motion planner is designed, where the kinematic redundancy of the system is exploited to satisfy some secondary tasks aimed at ensuring grasp stability and manipulation dexterity without violating physical constraints. To this purpose, a prioritized task sequencing with smooth transitions between tasks is adopted. Afterwards, a controller is designed so as to execute the motion references provided by the planner and, at the same time, achieve a desired contact force exerted by each finger on the grasped object. To this end, a parallel position/force control is considered. A simulation case study has been developed by using the dynamic simulator GRASPIT!, which has been suitably adapted and redistributed.

KEYWORDS: Grasping; Redundant manipulators; Motion planning; Robotic hands; Force control.

List of Symbols

$\boldsymbol{C}_i$	Matrix collecting the centrifugal and Coriolis terms of the ith finger
$\mathcal{C}$	Cost function related to a constraint
$\mathcal{C}_\Sigma$	Weighted sum of the constraint cost functions
$\overline{\mathcal{C}}$ ($\underline{\mathcal{C}}$)	Threshold values for the cost function
$\nabla_{\widetilde{\boldsymbol{q}}}\mathcal{C}$	Gradient of function $\mathcal{C}$ with respect to $\widetilde{\boldsymbol{q}}$
$d_{ii'}$	Distance between the ith and the i'th finger
$\boldsymbol{e}_o$	Error between desired and actual object pose
f_{n_i}	Contact force along the normal to the object's surface at the ith contact point
$\boldsymbol{g}_i$	Vector of generalized gravity forces acting on the ith finger
$\boldsymbol{G}$	Grasp matrix
$\boldsymbol{h}_\alpha$	($\alpha \times 1$) vector of ones
$\boldsymbol{I}_\alpha$	($\alpha \times \alpha$) identity matrix
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{J}^A$	Augmented Jacobian
l_i	Rest position of the spring modelling the elastic contact of the i finger
k_∇, k_F, k_I, k k_n, k_j, ρ, γ_j	Positive definite scalar gains

* Corresponding author. E-mail: fabio.ruggiero@unina.it

$\boldsymbol{K}_o,\ \boldsymbol{K}_{t_h}$ $\boldsymbol{K}_P,\ \boldsymbol{K}_D$	Positive definite gain matrices
$\boldsymbol{M}_i$	Inertia matrix of the ith finger
$\mathcal{M}_{t_h}$	Compatibility index between the hth task and the constraints
$\hat{\boldsymbol{n}}$	Unit vector representing the outward normal to the object's surface expressed with respect to the frame Σ_b
$\hat{\boldsymbol{n}}^o$	Unit vector representing the outward normal to the object's surface expressed with respect to the frame Σ_o
$\boldsymbol{N}_o(\cdot)$	Projector onto the null space of a matrix
$\boldsymbol{o}_*$	Origin of frame Σ_* expressed with respect to frame Σ_b
$\boldsymbol{O}_\alpha$	$(\alpha \times \alpha)$ null matrix
$\boldsymbol{q}$	Joint position vector
$\widetilde{\boldsymbol{q}}$	Augmented state vector given by $[\boldsymbol{q}^{\mathrm{T}}\ \boldsymbol{\eta}^{\mathrm{T}}]^{\mathrm{T}}$
$\boldsymbol{R}_*$	Rotation matrix denoting the orientation of frame Σ_* with respect to frame Σ_b
$\boldsymbol{R}_*^\star$	Rotation matrix denoting the orientation of frame Σ_* with respect to frame $\Sigma_\star$
$\boldsymbol{S}(\cdot)$	Skew symmetric operator
$\boldsymbol{u}_i$	Vector of driving generalized forces
V	Positive definite Lyapunov function
$\boldsymbol{W}$	Matrix of positive weights
$\boldsymbol{x}_o$	Object pose vector
Δl_i	Compression of the spring modelling the elastic contact of the i finger
ϵ_f (ϵ_m)	Frictionless force (momentum) residuals
$\boldsymbol{\eta}$	Vector of contact variables
$\lambda_m(\cdot)$ $(\lambda_M(\cdot))$	Minimum (maximum) eigenvalue of a matrix
$\frac{1}{\mu}$	Time constant $[s]$
σ_i	Task function referred to the ith task
Σ_*	Coordinate frame attached at point *
τ_i	Joint torques of the ith finger
$\boldsymbol{v}_*$	Generalized velocity of frame Σ_* expressed with respect to frame Σ_b
ω_*	Angular velocity of frame Σ_* expressed with respect to frame Σ_b

Subscripts and Superscripts

b	Base of torso
c_i	ith contact point
d	Desired
o	Object
r (l)	Right (left) arm
rf (lf)	Distal phalanx of finger belonging to the right (left) hand
rh (lh)	Palm of right (left) hand
t_i	ith secondary task
∞	Equilibrium
$\dagger$	Pseudoinverse

1. Introduction

1.1. Introduction to the problem

Service robotic applications are increasingly relying on dual-arm/hand object manipulation with multi-fingered mechanical hands. This is a challenging scenario which has not been investigated as extensively as required.

In order to ensure grasp stability, the execution of grasping and manipulation tasks requires control of interaction forces and motion synchronization of arms and fingers.

An object manipulation task can be generally assigned in terms of the motion of the fingertips and/or in terms of the desired object motion. Thus, a motion planner has to map the desired task into

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

the corresponding joint trajectories of the arms and the fingers, requiring the solution of an inverse kinematic problem. Then the controller has to ensure tracking of the planned trajectory.

1.2. Proposed solution

In this paper, a kinematic model for a dual-arm/hand robotic system is derived. Such a model allows the computation of the object pose (i.e., position and orientation) from the joint variables of each arm and finger that can be actuated (active joints) as well as from a set of unactuated contact variables (passive joints). On the basis of this model, a motion planning approach is devised where the kinematic redundancy of the system is exploited to fulfill a number of secondary tasks having lower priority with respect to the primary task (i.e., the motion of the manipulated object). The lower priority tasks are aimed at ensuring grasp stability and dexterity without violating physical constraints. To this aim, a prioritized task sequencing algorithm with smooth transitions between tasks is employed.

Moreover, a controller is designed to execute the motion references provided by the planner and, at the same time, maintain a desired contact force exerted by each finger on the grasped object. To this end, a parallel position/force control law is adopted. Stability of the control law is proven for a non-planar object surface.

The work described here extends the results given in Caccavale *et al.*[5] and Lippiello *et al.*[19] by considering all the details and proofs for the presented model and control. Moreover, the framework devoted to the sub-tasks switching is formalized and a new criterion for tasks' removal is introduced. A simulation case study is developed by using the dynamic simulator GRASPIT!,[23] which has been suitably adapted and redistributed.

1.3. Related work

In the literature several works dealing with the problem of object grasping and manipulation can be found. One of the first attempts trying to formalize grasp properties and the related control laws is reported in Murray *et al.*[26] Useful surveys focused on cooperative manipulators, contact modelling, multi-fingered robotic hands, and grasp properties.[4,13,22,33]

Different from what is proposed in this paper, a few works have considered exploitation of kinematic redundancy via a task-priority approach for object grasping and manipulation.[20] Instead, the task priority approach has been successfully applied to robotic manipulators[1,16,35] and visual servoing.[21] On the other hand, in the field of object manipulation via multi-fingered hands, the focus has been put on manipulability analysis[2,32] and constrained kinematic control.[12,25]

Impedance control[38] is one of the most adopted control laws for robot manipulators in contact with the environment, and has been also employed in object manipulation with multi-fingered hands. An impedance control approach for an arm–hand system is presented in refs. [28, 44], while in ref. [17] it is adopted to control the motion of the fingertips reaching the planned grasp points on the boundary of an unknown object. The passivity property of impedance-controlled systems is used in ref. [40] to design an Intrinsically Passive Control (IPC) that can be used both in free space (i.e., when the fingers approach the object) and for grasping (i.e., the fingers apply forces to the object). In detail, a virtual object is defined, which is connected to each finger, via a variable rest length spring, and to a virtual point, via another spring; all the springs are six-dimensional spatial springs.[3,41] Further developments of IPC control for grasping can be found in refs. [42,43,45]. An impedance control scheme is adopted in ref. [36] as well, combined with an algorithm for grasp forces' optimization[18] that allows the execution of different phases of a manipulation task, including re-grasping.

However, as considered in this paper, the execution of object grasping or manipulation requires also controlling of interaction forces so as to ensure grasp stability.[29] To this end, an alternative to impedance control could be the adoption of a hybrid force/position control, especially if force and position are measured and the corresponding control actions are properly decoupled. The approach proposed in Nagai and Yoshikawa[27] starts with the consideration that the force on the fingers can be seen as the sum of two orthogonal components: the manipulation force, necessary to impose object motion, and the grasping force, necessary to fulfill friction cone constraints. An alternative approach based on feedback linearization is proposed in DasGupta and Hatwal.[9] A decentralized control law is proposed in Remond *et al.*,[34] where each finger is independently controlled via a hybrid force/position control scheme. This approach is similar to the one presented in this paper, but the redundancy of the whole system is not addressed in the planning stage and fingers' elastic pads are not included. The latter are instead considered in refs. [10,11], where the problem of stable grasping and manipulation

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

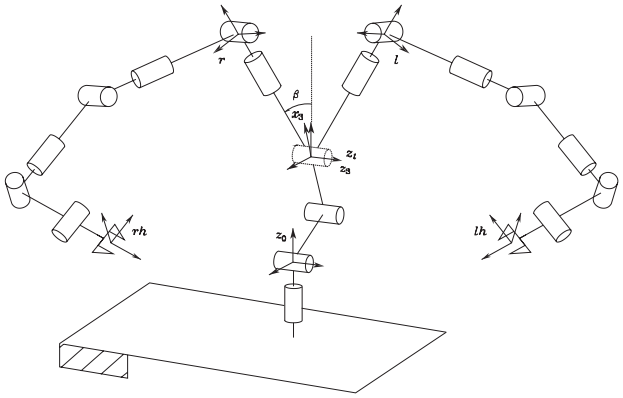


Fig. 1. Kinematic structure of a humanoid manipulator with torso and arms inspired by the DLR robot Justin.

using finger pairs covered by a soft compressible layer material is studied. Instead, the controller developed in this paper is a parallel force/position control[6,7] that can fall into the hybrid force/position category described above. The aim of such a controller is to achieve the reference position in the unconstrained directions and the reference forces in the constrained ones. This has been usually done by supposing planar surfaces: this limitation has been overcome in this paper.

2. Modelling

2.1. Kinematics of a dual-arm/hand system

Consider the dual-arm/hand manipulation system, schematically depicted in Fig. 1, composed by a torso having 3-degree of freedom (DOF) and two 7-DOF manipulators. The direct kinematics of such a system can be computed as reported in ref. [38].

Let Σ_b be the frame attached at the base of the torso and Σ_r and Σ_l be the frames attached at the base of the right and left arms respectively. Let Σ_{rh} and Σ_{lh} be the frames attached at the palms of the right and left hand respectively, and $\boldsymbol{o}_{rh}$ and $\boldsymbol{o}_{lh}$ their origins with respect to the base frame. Moreover, by assuming that each arm ends with an N-fingered robotic hand, it is useful to introduce frames Σ_{rf_i} and Σ_{lf_i} attached at the distal phalanx of finger i ($i = 1, \ldots, N$) of right and left hands respectively. The pose of Σ_{rf_i} with respect to the fixed base frame Σ_b can be thus expressed by the (4×4) homogeneous transformation matrix

$$\boldsymbol{T}_{rf_i} = \begin{bmatrix} \boldsymbol{R}_{rf_i} & \boldsymbol{o}_{rf_i} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{R}_{rf_i} \in SO(3)$ is the rotation matrix denoting the orientation of Σ_{rf_i} with respect to Σ_b, $\boldsymbol{o}_{rf_i}$ is the (3×1) position vector of the origin of Σ_{rf_i} with respect to Σ_b, and $\boldsymbol{0}_3$ denotes the (3×1) null vector.

Hence, the direct kinematics can be expressed as

$$\boldsymbol{T}_{rf_i} = \boldsymbol{T}_r(\boldsymbol{q}_t)\boldsymbol{T}_{rh}^{r}(\boldsymbol{q}_{rh})\boldsymbol{T}_{rf_i}^{rh}(\boldsymbol{q}_{rf_i}), \tag{2}$$

where $\boldsymbol{T}_r$ is the homogeneous transformation matrix expressing the pose of Σ_r with respect to Σ_b, $\boldsymbol{T}_{rh}^{r}$ is the homogeneous transformation matrix relating Σ_{rh} to Σ_r, and $\boldsymbol{T}_{rf_i}^{rh}$ is the homogeneous transformation matrix relating Σ_{rf_i} to Σ_{rh}. Note that these matrices depend on the torso joint vector, $\boldsymbol{q}_t$, the right arm joint vector, $\boldsymbol{q}_{rh}$, and the right hand fingers joint vector, $\boldsymbol{q}_{rf_i}$ respectively. The dimensions of such joint vectors depend on a particular setup. An equation similar to (2) holds for the left-hand fingers, with subscript l in place of subscript r.

Due to the branched structure of the manipulator, the kinematic equations of both the arms depend on the joint vector $\boldsymbol{q}_t$ of the torso, thus they are not independent. Hereafter it is assumed that the torso is motionless, i.e., $\boldsymbol{q}_t$ is constant; therefore, the kinematics of right and left hands can be considered

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

separately. Hence, in the following, the superscripts r and l will be dropped and used explicitly only when it is strictly required.

In order to derive differential kinematics, it is useful to represent the velocity of the frame Σ_{f_i} with respect to Σ_b by the (6×1) twist vector $\boldsymbol{v}_{f_i} = [\dot{\boldsymbol{o}}_{f_i}^{\mathrm{T}} \;\; \boldsymbol{\omega}_{f_i}^{\mathrm{T}}]^{\mathrm{T}}$, where $\dot{\boldsymbol{o}}_{f_i}$ and $\boldsymbol{\omega}_{f_i}$ denote the linear and angular velocity of the finger frame with respect to the fixed base frame respectively. It is worth noting that $\dot{\boldsymbol{R}}_{f_i} = \boldsymbol{S}(\boldsymbol{\omega}_{f_i})\boldsymbol{R}_{f_i}$, where $\boldsymbol{S}(\cdot)$ is the skew-symmetric operator performing the vector product.[38]

The differential kinematic equations relating the joint velocities to the velocity of frame Σ_{f_i} can be thus written as

$$\boldsymbol{v}_{f_i} = \begin{bmatrix} \boldsymbol{J}_{P_i}(\boldsymbol{q}_i) \\ \boldsymbol{J}_{O_i}(\boldsymbol{q}_i) \end{bmatrix} \dot{\boldsymbol{q}}_i = \boldsymbol{J}_{F_i}(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i, \tag{3}$$

where $\boldsymbol{q}_i = [\boldsymbol{q}_h^{\mathrm{T}} \;\; \boldsymbol{q}_{f_i}^{\mathrm{T}}]^{\mathrm{T}}$, and $\boldsymbol{J}_{F_i}$ is the Jacobian of the arm, ending with finger i, in which $\boldsymbol{J}_{P_i}$ and $\boldsymbol{J}_{O_i}$ denote the Jacobian linear and rotational part respectively. The detailed expression of $\boldsymbol{J}_{F_i}$ in (3) is

$$\boldsymbol{J}_{F_i} = \left[\boldsymbol{G}_h^{\mathrm{T}}(\boldsymbol{o}_{f_i,h})\boldsymbol{J}_h(\boldsymbol{q}_h)\; \bar{\boldsymbol{R}}_h(\boldsymbol{q}_h)\boldsymbol{J}_{f_i}^h(\boldsymbol{q}_{f_i})\right], \tag{4}$$

where $\boldsymbol{J}_h$ is the Jacobian that maps the joint velocity of the arm, $\dot{\boldsymbol{q}}_h$, to the velocity of the frame Σ_h, $\bar{\boldsymbol{R}}_h = \mathrm{diag}\{\boldsymbol{R}_h, \boldsymbol{R}_h\}$, $\boldsymbol{R}_h \in SO(3)$ is the rotation matrix denoting the orientation of Σ_h with respect to the fixed base frame, $\boldsymbol{J}_{f_i}^h$ is the Jacobian that maps the joint velocity of the ith finger, $\dot{\boldsymbol{q}}_{f_i}$, to the velocity of the frame Σ_{f_i}, expressed with respect to Σ_h, $\boldsymbol{o}_{f_i,h} = \boldsymbol{o}_{f_i} - \boldsymbol{o}_h$, and $\boldsymbol{G}_h^{\mathrm{T}}(\boldsymbol{o}_{f_i,h})$ is given by

$$\boldsymbol{G}_h^{\mathrm{T}}(\boldsymbol{o}_{f_i,h}) = \begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{S}(\boldsymbol{o}_{f_i,h}) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix}, \tag{5}$$

where $\boldsymbol{I}_\alpha$ and $\boldsymbol{O}_\alpha$ denote the $(\alpha \times \alpha)$ identity and null matrix respectively.

Therefore, the differential kinematic equations of the whole arm–hand system can be written in the form

$$\tilde{\boldsymbol{v}}_f = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{6}$$

where $\tilde{\boldsymbol{v}}_f = [\boldsymbol{v}_{f_1}^{\mathrm{T}} \cdots \boldsymbol{v}_{f_N}^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{q} = [\boldsymbol{q}_h^{\mathrm{T}} \;\; \boldsymbol{q}_{f_1}^{\mathrm{T}} \cdots \boldsymbol{q}_{f_N}^{\mathrm{T}}]^{\mathrm{T}}$, and $\boldsymbol{J}$ is the Jacobian of the overall arm–hand system, whose detailed expression is

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{G}_h^{\mathrm{T}}(\boldsymbol{o}_{f_1,h})\boldsymbol{J}_h(\boldsymbol{q}_h) & \bar{\boldsymbol{R}}_h(\boldsymbol{q}_h)\boldsymbol{J}_{f_1}^h(\boldsymbol{q}_1) & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{G}_h^{\mathrm{T}}(\boldsymbol{o}_{f_2,h})\boldsymbol{J}_h(\boldsymbol{q}_h) & \boldsymbol{O} & \bar{\boldsymbol{R}}_h(\boldsymbol{q}_h)\boldsymbol{J}_{f_2}^h(\boldsymbol{q}_2) & \ddots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{G}_h^{\mathrm{T}}(\boldsymbol{o}_{f_N,h})\boldsymbol{J}_h(\boldsymbol{q}_h) & \boldsymbol{O} & \boldsymbol{O} & \cdots & \bar{\boldsymbol{R}}_h(\boldsymbol{q}_h)\boldsymbol{J}_{f_N}^h(\boldsymbol{q}_N) \end{bmatrix}, \tag{7}$$

where $\boldsymbol{O}$ denotes a null matrix of proper dimensions.

2.2. Contact kinematics

Both the object and the robotic fingers are often smooth surfaces and, depending on the contact type, manipulation involves rolling and/or sliding of the fingertips on the object's surface. If the fingers and object shapes are completely known, the contact kinematics can be described by introducing contact coordinates defined on the basis of a suitable parametrization of contact surfaces.[24,26]

By assuming that the hand grasps a rigid object, it is useful to introduce a frame Σ_o, attached to the object, usually chosen with the origin in the object's center of mass. Let $\boldsymbol{R}_o$ and $\boldsymbol{o}_o$ denote, respectively, the rotation matrix and the position vector of the origin of Σ_o with respect to the base frame, and let $\boldsymbol{v}_o$ denote the object velocity twist vector.

It is assumed that the fingertips are sharp (i.e., they end with a point, denoted as tip point) and covered by an elastic pad. The elastic contact is then modelled by introducing a finger contact frame Σ_{k_i}, attached to the elastic pad with the origin in the tip point $\boldsymbol{o}_{k_i}$, and a spring-damper system connecting $\boldsymbol{o}_{k_i}$ with the origin of Σ_{f_i}. This last frame is attached to the rigid part of the finger (Fig. 2)

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

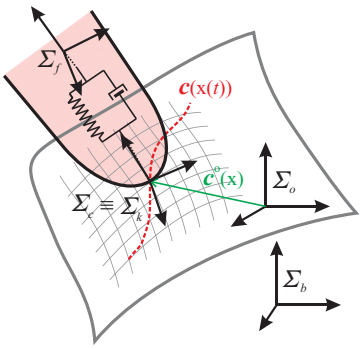


Fig. 2. (Colour online) Local parametrization of object surface with respect to Σ_o.

and has the same orientation of Σ_{k_i}. The displacement between Σ_{f_i} and Σ_{k_i} due to elastic contact force can be computed as

$$\boldsymbol{o}_{f_i} - \boldsymbol{o}_{k_i} = (l_i - \Delta l_i)\boldsymbol{R}_o \hat{\boldsymbol{n}}^o(\boldsymbol{\xi}), \tag{8}$$

where l_i and $0 \leq \Delta l_i \leq l_i$ are the rest position and the compression of the spring, respectively, and $\hat{\boldsymbol{n}}^o$ is the unit vector representing the outward normal to the object's surface at the contact point, expressed with respect to Σ_o.

Furthermore, let Σ_{c_i} be the contact frame attached to the object with the origin at the contact point, $\boldsymbol{o}_{c_i}$. Note that instantaneously the object contact point, $\boldsymbol{o}_{c_i}$, and the finger contact point, $\boldsymbol{o}_{k_i}$, coincide. One of the axes of Σ_{c_i}, e.g., the Z-axis, is assumed to be outward normal to the tangent plane to the object surface at the contact point.

The position of the contact point with respect to the object frame, $\boldsymbol{o}^o_{o,c_i} = \boldsymbol{o}^o_{c_i} - \boldsymbol{o}^o_o$, can be parametrized, at least locally, in terms of a coordinate chart, $\boldsymbol{c}^o_i : U_i \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$, which maps the chart's point $\boldsymbol{\xi}_i = [u_i \quad v_i]^{\mathrm{T}} \in U_i$ to the point $\boldsymbol{o}^o_{o,c_i}(\boldsymbol{\xi}_i)$ on the surface of the object.

By assuming that $\boldsymbol{c}^o_i$ is a diffeomorphism and the coordinate chart is orthogonal and right-handed, the contact frame Σ_{c_i} can be thus chosen as a Gauss frame,[24] where the relative orientation expressed by the rotation matrix $\boldsymbol{R}^o_{c_i}$ has the following expression:

$$\boldsymbol{R}^o_{c_i}(\boldsymbol{\xi}) = \left[\frac{\boldsymbol{c}^o_{u_i}}{\|\boldsymbol{c}^o_{u_i}\|} \quad \frac{\boldsymbol{c}^o_{v_i}}{\|\boldsymbol{c}^o_{v_i}\|} \quad \frac{\boldsymbol{c}^o_{u_i} \times \boldsymbol{c}^o_{v_i}}{\|\boldsymbol{c}^o_{u_i} \times \boldsymbol{c}^o_{v_i}\|} \right], \tag{9}$$

and hence it is computed as a function of orthogonal tangent vectors $\boldsymbol{c}^o_{u_i} = \partial \boldsymbol{c}^o_i / \partial u_i$ and $\boldsymbol{c}^o_{v_i} = \partial \boldsymbol{c}^o_i / \partial v_i$.

First, consider the contact kinematics from the object's point of view. Function $\boldsymbol{c}^o_i(\boldsymbol{\xi}_i(t))$ denotes a curve on the object's surface parametrized by the time variable t. Hence, the corresponding motion of Σ_{c_i} can be determined as a function of the object motion, the geometric parameters of the object, and the geometric features of the curve. Namely, the time derivative of equation $\boldsymbol{o}_{c_i} = \boldsymbol{o}_o + \boldsymbol{R}_o \boldsymbol{c}^o_i(\boldsymbol{\xi}_i)$, which provides the position of the object contact point in the base frame, yields

$$\dot{\boldsymbol{o}}_{c_i} = \dot{\boldsymbol{o}}_o - \boldsymbol{S}\big(\boldsymbol{R}_o \boldsymbol{c}^o_i(\boldsymbol{\xi}_i)\big)\boldsymbol{\omega}_o + \boldsymbol{R}_o \frac{\partial \boldsymbol{c}^o_i}{\partial \boldsymbol{\xi}_i} \dot{\boldsymbol{\xi}}_i, \tag{10}$$

where the first two terms on the right-hand side specify the velocity contribution due to the object motion, while the last term represents the finger velocity relative to the object surface. On the other hand, for angular velocity, the following equality holds:

$$\boldsymbol{\omega}_{c_i} = \boldsymbol{\omega}_o + \boldsymbol{R}_o \boldsymbol{\omega}^o_{o,c_i}, \tag{11}$$

where $\boldsymbol{\omega}^o_{o,c_i}$ is the angular velocity of Σ_{c_i} with respect to Σ_o and can be expressed as

$$\boldsymbol{\omega}^o_{o,c_i} = \boldsymbol{C}(\boldsymbol{\xi}_i)\dot{\boldsymbol{\xi}}_i, \tag{12}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

where $\boldsymbol{C}(\boldsymbol{\xi}_i)$ is a (3×2) matrix depending on the geometric parameters of the surface.[26] Matrix $\boldsymbol{C}$ is not necessarily full-rank (e.g., is null in the case of planar surfaces). In view of Eqs. (10)–(12), the velocity of the contact frame can be expressed as

$$\boldsymbol{v}_{c_i} = \begin{bmatrix} \dot{\boldsymbol{o}}_{c_i} \\ \boldsymbol{\omega}_{c_i} \end{bmatrix} = \boldsymbol{G}^{\mathrm{T}}_{\xi_i}(\boldsymbol{\xi}_i)\boldsymbol{v}_{o_i} + \boldsymbol{J}_{\xi_i}(\boldsymbol{\xi}_i)\dot{\boldsymbol{\xi}}_i, \tag{13}$$

where $\boldsymbol{v}_{o_i}$ is the velocity of the object computed on the basis of the kinematics of finger i, $\boldsymbol{G}_{\xi_i}(\boldsymbol{\xi}_i)$ and $\boldsymbol{J}_{\xi_i}(\boldsymbol{\xi}_i)$ are (6×6) and (6×2) full rank matrices, respectively, having the following expressions:

$$\boldsymbol{G}^{\mathrm{T}}_{\xi_i}(\boldsymbol{\xi}_i) = \begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{S}\big(\boldsymbol{R}_o \boldsymbol{c}^o_i(\boldsymbol{\xi}_i)\big) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix}, \quad \boldsymbol{J}_{\xi_i}(\boldsymbol{\xi}_i) = \begin{bmatrix} \boldsymbol{R}_o \dfrac{\partial \boldsymbol{c}^o_i}{\partial \boldsymbol{\xi}_i} \\ \boldsymbol{R}_o \boldsymbol{C}(\boldsymbol{\xi}_i) \end{bmatrix}. \tag{14}$$

Now consider the contact kinematics from the fingers' point of view. The contact can be modelled as a passive 3-DOF ball and socket kinematic pair centered at the origin $\boldsymbol{o}_{k_i}$ of Σ_{k_i}. This point is in general fixed to the elastic pad of the finger, but it may also move on the surface if sliding is allowed. Therefore, the relative orientation of Σ_{c_i} with respect to Σ_{k_i}, $\boldsymbol{R}^{k_i}_{c_i}$, can be computed in terms of a suitable parametrization of the ball and socked joint, e.g., Euler angles or angle-axis representations. If the parametrization in terms of XYZ Euler angles is adopted, a vector $\boldsymbol{\theta}_i = [\theta_{1_i}\ \theta_{2_i}\ \theta_{3_i}]^{\mathrm{T}}$ can be considered, thus $\boldsymbol{R}^{k_i}_{c_i} = \boldsymbol{R}^{k_i}_{c_i}(\boldsymbol{\theta}_i)$. In detail, θ_{1_i} and θ_{2_i} parametrize the so-called "swing" motion aligning axis Z of a moving frame to axis Z of the contact frame, while θ_{3_i} corresponds to the "twist" motion about axis Z of the contact frame. Singularities occur for $\theta_{2_i} = \pm\pi/2$, but they do not correspond to physical kinematic singularities of the kinematic pair.

Note that in the presence of a contact force tip elasticity allows mutual translation of Σ_{k_i} and Σ_{f_i} according to Eq. (8), while the mutual orientation does not change. Therefore, $\boldsymbol{R}^{k_i}_{c_i} = \boldsymbol{R}^{f_i}_{c_i}$. Moreover, the angular velocity of Σ_{c_i} relative to Σ_{f_i} can be expressed as $\boldsymbol{\omega}^{f_i}_{f_i,c_i} = \boldsymbol{H}(\boldsymbol{\theta}_i)\dot{\boldsymbol{\theta}}_i$, where $\boldsymbol{H}$ is a transformation matrix depending on the chosen parametrization.[38] In view of the decomposition $\boldsymbol{\omega}_{c_i} = \boldsymbol{\omega}_{f_i} + \boldsymbol{R}_{f_i}(\boldsymbol{q}_i)\boldsymbol{\omega}^{f_i}_{f_i,c_i}$ and Eq. (3), the angular velocity of Σ_{c_i} can be computed as a function of joint and contact variables, i.e.,

$$\boldsymbol{\omega}_{c_i} = \boldsymbol{J}_{O_i}(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{R}_{f_i}(\boldsymbol{q}_i)\boldsymbol{H}(\boldsymbol{\theta}_i)\dot{\boldsymbol{\theta}}_i, \tag{15}$$

where $\boldsymbol{J}_{O_i}$ is defined in Eq. (3). Moreover, since the origins of Σ_{c_i} and Σ_{k_i} coincide, the following equality holds

$$\boldsymbol{o}_{c_i} = \boldsymbol{o}_{k_i} = \boldsymbol{o}_{f_i} - (l_i - \Delta l_i)\boldsymbol{R}_o \hat{\boldsymbol{n}}^o_i(\boldsymbol{\xi}_i), \tag{16}$$

while the time derivative of (16) yields

$$\dot{\boldsymbol{o}}_{c_i} = \boldsymbol{J}_{P_i}(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \Delta\dot{l}_i \boldsymbol{R}_o \hat{\boldsymbol{n}}^o_i(\boldsymbol{\xi}_i) + (l_i - \Delta l_i)\left[\boldsymbol{S}\big(\boldsymbol{R}_o \hat{\boldsymbol{n}}^o_i(\boldsymbol{\xi}_i)\big)\boldsymbol{\omega}_o - \boldsymbol{R}_o \frac{\partial \hat{\boldsymbol{n}}^o_i(\boldsymbol{\xi}_i)}{\partial \boldsymbol{\xi}_i}\dot{\boldsymbol{\xi}}_i\right], \tag{17}$$

where $\boldsymbol{J}_{P_i}$ is defined in (3).

By considering (15) and (17), the velocity of the contact frame can be expressed as

$$\boldsymbol{v}_{c_i} = \boldsymbol{J}_{F_i}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}_{\theta_i}(\boldsymbol{\theta}_i, \boldsymbol{q}_i)\dot{\boldsymbol{\theta}}_i + \boldsymbol{J}_{\Delta l_i}(\boldsymbol{\xi}_i)\Delta\dot{l}_i - \boldsymbol{J}'_{\xi_i}(\boldsymbol{\xi}_i, \Delta l_i)\dot{\boldsymbol{\xi}}_i - \boldsymbol{G}^{\mathrm{T}}_{\Delta l_i}(\boldsymbol{\xi}_i, \Delta l_i)\boldsymbol{v}_o, \tag{18}$$

where $\boldsymbol{J}_{F_i}$ is defined in (4), $\boldsymbol{J}_{\theta_i}$ is a (6×3) full column rank matrix

$$\boldsymbol{J}_{\theta_i} = \begin{bmatrix} \boldsymbol{O}_3 \\ \boldsymbol{R}_{f_i}(\boldsymbol{q}_i)\boldsymbol{H}(\boldsymbol{\theta}_i) \end{bmatrix}, \tag{19}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

$\boldsymbol{J}_{\Delta l_i}$ is a (6×1) vector

$$\boldsymbol{J}_{\Delta l_i} = \begin{bmatrix} \boldsymbol{R}_o \hat{\boldsymbol{n}}_i^o(\boldsymbol{\xi}_i) \\ \boldsymbol{0_3} \end{bmatrix}, \tag{20}$$

$\boldsymbol{J}'_{\xi_i}$ is a (6×2) full column rank matrix

$$\boldsymbol{J}'_{\xi_i} = \begin{bmatrix} (l - \Delta l_i)\boldsymbol{R}_o \dfrac{\partial \hat{\boldsymbol{n}}_i^o(\boldsymbol{\xi}_i)}{\partial \boldsymbol{\xi}_i} \\ \boldsymbol{O}_{3\times 2} \end{bmatrix}, \tag{21}$$

$\boldsymbol{O}_{3\times 2}$ is the (3×2) null matrix, and $\boldsymbol{G}^{\mathrm{T}}_{\Delta l_i}$ is the (6×6) matrix

$$\boldsymbol{G}^{\mathrm{T}}_{\Delta l_i} = \begin{bmatrix} \boldsymbol{O}_3 & (\Delta l_i - l_i)\boldsymbol{S}\big(\boldsymbol{R}_o \hat{\boldsymbol{n}}_i^o(\boldsymbol{\xi}_i)\big) \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 \end{bmatrix}. \tag{22}$$

Therefore, from (13) and (18), the contact kinematics of finger i has the form

$$\boldsymbol{J}_{F_i}(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{J}_{\eta_i}(\boldsymbol{\eta}_i, \boldsymbol{q}_i, \Delta l_i)\dot{\boldsymbol{\eta}}_i + \boldsymbol{J}_{\Delta l_i}(\boldsymbol{\xi})\Delta \dot{l}_i = \boldsymbol{G}_i^{\mathrm{T}}(\boldsymbol{\eta}_i, \Delta l_i)\boldsymbol{v}_o, \tag{23}$$

where $\boldsymbol{\eta}_i = [\boldsymbol{\xi}_i^{\mathrm{T}} \;\; \boldsymbol{\theta}_i^{\mathrm{T}}]^{\mathrm{T}}$ is the vector of contact variables, $\boldsymbol{J}_{\eta_i} = [-(\boldsymbol{J}_{\xi_i} + \boldsymbol{J}'_{\xi_i})\,\boldsymbol{J}_{\theta_i}]$ is a (6×5) full rank matrix, and $\boldsymbol{G}_i = \boldsymbol{G}_{\xi_i} + \boldsymbol{G}_{\Delta l_i}$ is a (6×6) full rank grasp matrix.

Hence, to summarize, in case of a motionless torso, convex object surface and sharp fingertips covered by an elastic pad, Eq. (23) can be interpreted as the differential kinematic equation of an "extended" finger corresponding to the kinematic chain, including the arm, the finger joint variables (*active joints*), and the contact variables (*passive joints*). It is worth noting that (23) involves all the six components of the velocity, while grasp constraints adopted in the literature usually consider only those transmitted by the contact.[25,26]

Depending on the considered contact type, some of the parameters $\boldsymbol{\xi}_i$ and $\boldsymbol{\theta}_i$ are constant. Hence, by assuming that such contact types remain unchanged during the task, the variable parameters at each contact point are grouped in a $(n_{c_i} \times 1)$ vector, $\boldsymbol{\eta}_i$, of contact variables, with $n_{c_i} \leq 5$.

Different form the classical grasp analysis, in this work the elasticity of the elastic pad has been explicitly modelled, although using a simplified model. This means that the force along the normal to the contact surface is always of elastic type. The quantity Δl_i, at steady state, is related to the contact normal force f_{n_i} by the equation $\Delta l_i = f_{n_i}/k_i$, where k_i is the elastic constant of the elastic pad of finger i.

2.3. Kinematic analysis of the grasp

Object manipulation is, in general, a difficult task, since the number of control variables (the active joints) is lower than the number of configuration variables (active and passive joints). However, by considering only the kinematics of the system, it is possible to simplify the analysis. As will be detailed in Section 3.2, a force control strategy is adopted to ensure the desired constant contact force f_{d_i} along the direction normal to the contact point; hence, $\Delta l_i = \Delta l_{d_i} = f_{d_i}/k_i$ can be assumed to be fixed ($\Delta \dot{l}_i = 0$) and Eq. (23) can be rewritten as

$$\boldsymbol{J}_{F_i}(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{J}_{\eta_i}(\boldsymbol{\eta}_i, \boldsymbol{q}_i, \Delta l_i)\dot{\boldsymbol{\eta}}_i = \boldsymbol{G}_i^{\mathrm{T}}(\boldsymbol{\eta}_i, \Delta l_i)\boldsymbol{v}_o, \tag{24}$$

On the basis of (24), it is possible to achieve a kinematic classification of the grasp.[33]

A grasp is said to be *redundant* if the null space of the matrix $[\boldsymbol{J}_{F_i}\; \boldsymbol{J}_{\eta_i}]$ is non-empty for at least one finger i. In this case, the mapping between the joint variables of the "extended" finger i and the object velocity is many to one: motion of active and passive joints of the extended finger is possible when the object is locked. Note that a single finger could be redundant if the null space of $\boldsymbol{J}_i$ is non-empty, i.e., in the case of a redundant arm–finger kinematic chain. In this last case, motion of the active joints is possible when both the passive joints and the object are locked. On the other hand, for the type of contacts considered here (point contact), the null space of $\boldsymbol{J}_{\eta_i}$ is always empty: this implies

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

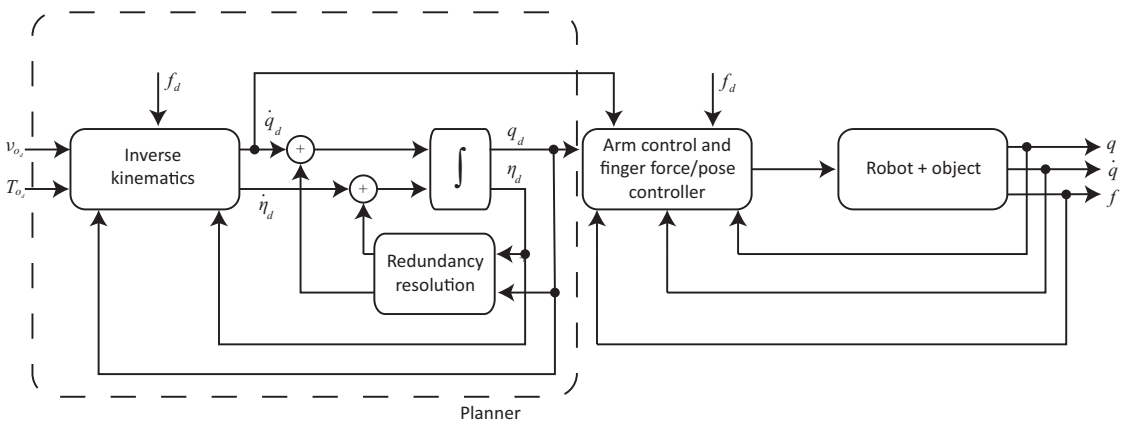


Fig. 3. Block scheme of the control architecture.

that motion of the passive joints is not possible when the active joints and the object are locked. In typical situations, the fingers of the robotic hand are not redundant, but the extended fingers (even not considering the joints of the arm) may be redundant, thanks to the presence of additional DOFs provided by the passive joints.

A grasp is *indeterminate* if the intersection of null spaces of $[-\boldsymbol{J}_{\eta_i}\ \boldsymbol{G}_i^{\mathrm{T}}]$, for all $i = 1, \ldots, N$, is non-null. In this case, motion of the object and passive joints is possible when the active joints of all the fingers are locked. The kinematic indetermination is derived from the fact that the object motion cannot be completely controlled by finger motions, but it depends on the dynamics of the whole (hands plus object) system.[26] An example of indeterminate grasp is that of a box grasped by two hard-finger opposite contacts: in this case the box may rotate about the axis connecting the two contact points while the fingers are locked.

It is worth noting that, also in the case of redundant and indeterminate grasps, the values of the contact variables are uniquely determined for a given object pose and fingers' configuration.

3. Control Scheme with Redundancy Resolution

In the case of kinematically not indeterminate and, possibly, redundant grasp, the following two-stage control architecture is proposed (Fig. 3):

- The first stage is a *motion planner*, given by a closed-loop inverse kinematic algorithm with redundancy resolution; the algorithm computes joint references for the active joints corresponding to the desired object's motion – assigned in terms of the homogeneous transformation matrix $\boldsymbol{T}_{o_d}$ and the corresponding twist velocity vector $\boldsymbol{v}_{o_d}$– and the desired contact normal force $\boldsymbol{f}_d = [f_{d_1} \cdots f_{d_N}]^{\mathrm{T}}$ for the fingers.
- The second stage is a *parallel control* scheme, composed by a proportional-derivative (PD) position controller and a proportional-integral (PI) tip force controller; the controller ensures tracking of the desired joint motion references computed in the first stage and the desired contact forces.

In ideal conditions, the joint references computed by the inverse kinematic stage ensure the tracking of the desired object motion. Tracking of the desired contact forces is guaranteed by force control, assuming that force sensors at the fingertips are available. In principle, the joint references of the overall manipulation system could be involved; however, it is reasonable to design a force controller acting only on the joints of the fingers.

3.1. Motion planner

Starting from (24), it is useful to write differential kinematic equations of the whole (right or left) arm–hand system as

$$\widetilde{\boldsymbol{J}}(\widetilde{\boldsymbol{q}}, \Delta\boldsymbol{l})\dot{\widetilde{\boldsymbol{q}}} = \boldsymbol{G}^{\mathrm{T}}(\boldsymbol{\eta}, \Delta\boldsymbol{l})\widetilde{\boldsymbol{v}}_o, \tag{25}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

where $\widetilde{\boldsymbol{q}} = [\boldsymbol{q}^{\mathrm{T}} \;\; \boldsymbol{\eta}^{\mathrm{T}}]^{\mathrm{T}}$, $\widetilde{\boldsymbol{J}} = [\boldsymbol{J} \;\; \boldsymbol{J}_{\eta}]$, $\boldsymbol{J}$ is the Jacobian of the arm–hand system defined in (6), $\boldsymbol{J}_{\eta} = \mathrm{diag}\{\boldsymbol{J}_{\eta_1}, \ldots, \boldsymbol{J}_{\eta_N}\}$ is a block-diagonal matrix corresponding to the vector of passive joint velocities, $\dot{\boldsymbol{\eta}} = [\dot{\boldsymbol{\eta}}_1^{\mathrm{T}} \cdots \dot{\boldsymbol{\eta}}_N^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{G}$ is the block-diagonal grasp matrix $\boldsymbol{G} = \mathrm{diag}\{\boldsymbol{G}_1, \ldots, \boldsymbol{G}_N\}$, $\Delta\boldsymbol{l} = [\Delta l_1 \cdots \Delta l_N]^{\mathrm{T}}$, and $\widetilde{\boldsymbol{v}}_o = [\boldsymbol{v}_o^{\mathrm{T}} \cdots \boldsymbol{v}_o^{\mathrm{T}}]^{\mathrm{T}}$.

For the sake of clarity, a minimal representation has been adopted for the parametrization of both object and finger orientation. Hence, by considering an Euler angles' representation, from (25) the following closed-loop inverse kinematic algorithm can be derived:

$$\dot{\widetilde{\boldsymbol{q}}}_d = \widetilde{\boldsymbol{J}}^{\dagger}(\widetilde{\boldsymbol{q}}_d, \Delta\boldsymbol{l}_d)\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{\eta}_d, \Delta\boldsymbol{l}_d)\widetilde{\boldsymbol{\Gamma}}(\widetilde{\boldsymbol{x}}_o)(\dot{\widetilde{\boldsymbol{x}}}_{o_d} + \boldsymbol{K}_o\widetilde{\boldsymbol{e}}_o) + \boldsymbol{N}_o\boldsymbol{\sigma}, \tag{26}$$

where the symbol † denotes a weighted right pseudo-inverse, $\boldsymbol{K}_o$ is a diagonal and positive definite matrix gain, $\boldsymbol{N}_o = \boldsymbol{I} - \widetilde{\boldsymbol{J}}^{\dagger}\widetilde{\boldsymbol{J}}$ is a projector onto the null space of the Jacobian matrix $\widetilde{\boldsymbol{J}}$, and

$$\widetilde{\boldsymbol{x}}_{o_d} = \begin{bmatrix} \boldsymbol{x}_{o_d} \\ \vdots \\ \boldsymbol{x}_{o_d} \end{bmatrix}, \;\; \widetilde{\boldsymbol{x}}_o = \begin{bmatrix} \boldsymbol{x}_{o_1} \\ \vdots \\ \boldsymbol{x}_{o_N} \end{bmatrix}, \;\; \widetilde{\boldsymbol{e}}_o = \begin{bmatrix} \boldsymbol{e}_{o_1} \\ \vdots \\ \boldsymbol{e}_{o_N} \end{bmatrix}, \;\; \widetilde{\boldsymbol{\Gamma}}(\widetilde{\boldsymbol{x}}_o) = \mathrm{diag}\{\boldsymbol{\Gamma}(\boldsymbol{x}_{o_1}), \ldots, \boldsymbol{\Gamma}(\boldsymbol{x}_{o_N})\}, \tag{27}$$

where $\boldsymbol{x}_{o_d}$ and $\boldsymbol{x}_{o_i}$ are the planned and the actual object poses, respectively, $\boldsymbol{e}_{o_i} = \boldsymbol{x}_{o_d} - \boldsymbol{x}_{o_i}$, and $\boldsymbol{\Gamma}(\boldsymbol{x}_{o_i})$ is the transformation between $\dot{\boldsymbol{x}}_{o_i}$ and the object velocity $\boldsymbol{v}_{o_i}$, computed on the basis of the kinematics of finger i. The quantity $\Delta\boldsymbol{l}_d$ in (26) is the vector collecting the finger elastic pad deformations $\Delta l_{d_i} = f_{d_i}/k_i$ corresponding to the desired contact force f_{d_i}.

Since the system may be highly redundant, multiple tasks could be fulfilled, provided that they are suitably arranged in a priority order. Consider m secondary tasks, each expressed by a task function $\boldsymbol{\sigma}_{t_h}(\widetilde{\boldsymbol{q}})$ $(h = 1, \ldots, m)$. According to the *augmented projection method*,[1] the null projection can be better detailed as

$$\dot{\widetilde{\boldsymbol{q}}}_d = \widetilde{\boldsymbol{J}}^{\dagger}(\widetilde{\boldsymbol{q}}_d, \Delta\boldsymbol{l}_d)\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{\eta}_d, \Delta\boldsymbol{l}_d)\widetilde{\boldsymbol{\Gamma}}(\widetilde{\boldsymbol{x}}_o)(\dot{\widetilde{\boldsymbol{x}}}_{o_d} + \boldsymbol{K}_o\widetilde{\boldsymbol{e}}_o) + \sum_{h=1}^{m} \boldsymbol{N}\big(\boldsymbol{J}_{t_h}^A\big)\boldsymbol{J}_{t_h}^{\dagger}\boldsymbol{K}_{t_h}\boldsymbol{e}_{t_h}, \tag{28}$$

where $\boldsymbol{J}_{t_h}$ is the hth task Jacobian, and $\boldsymbol{J}_{t_h}^A$ is the augmented Jacobian given by

$$\boldsymbol{J}_{t_h}^A = \Big[\widetilde{\boldsymbol{J}}^{\mathrm{T}} \; \boldsymbol{J}_{t_1}^{\mathrm{T}} \ldots \boldsymbol{J}_{t_{h-1}}^{\mathrm{T}}\Big]^{\mathrm{T}}. \tag{29}$$

$\boldsymbol{N}(\boldsymbol{J}_{t_h}^A)$ is a null projector of the matrix $\boldsymbol{J}_{t_h}^A$, $\boldsymbol{K}_{t_h}$ is a positive definite gain matrix, and $\boldsymbol{e}_{t_h} = \boldsymbol{\sigma}_{t_{hd}} - \boldsymbol{\sigma}_{t_h}$ is the task error, being $\boldsymbol{\sigma}_{t_{hd}}$ the desired value of the hth task variable.

The augmented projection method can be also adopted to fulfill mechanical or environmental constraints, such as joint limits and obstacle (i.e., other fingers or the grasped object) avoidance. To this aim, each constraint can be described by means of a cost function, $\mathcal{C}(\widetilde{\boldsymbol{q}})$, which increases when the manipulator is close to violate the constraint. In order to minimize the cost function, the manipulator could be moved according to $-\nabla_{\widetilde{\boldsymbol{q}}}^{\mathrm{T}}\mathcal{C}(\widetilde{\boldsymbol{q}})$ that could be considered as a fictitious force moving the manipulator away from configurations violating the constraints. In order to include the constraints in (28), an overall cost function $\mathcal{C}_{\Sigma}$, given by

$$\mathcal{C}_{\Sigma}(\widetilde{\boldsymbol{q}}) = \sum_{s} \gamma_s\mathcal{C}_s(\widetilde{\boldsymbol{q}}), \tag{30}$$

is introduced, where γ_s and $\mathcal{C}_s$ are positive weight and cost function, respectively, referred to the sth constraint. Therefore, the following term can be added to (28),

$$\dot{\widetilde{\boldsymbol{q}}}_c = -k_{\nabla}\boldsymbol{N}\big(\boldsymbol{J}_{t_{m+1}}^A\big)\nabla_{\widetilde{\boldsymbol{q}}_d}^{\mathrm{T}}\mathcal{C}_{\Sigma}, \tag{31}$$

where k_{∇} is a positive gain.

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

If the system is close to violate a constraint, a high-level supervisor has to remove some secondary tasks and relax enough DOFs to fulfill the constraints.[21] To manage in a correct way removal/insertion of tasks from/into the stack (task sequencing), a task supervisor, based on a two-layer architecture, can be designed: the lower layer determines which and when some tasks must be removed from the stack; then the upper layer verifies if the previously removed tasks can be pushed back into the stack.

3.1.1. Removal and insertion of the tasks. The first layer verifies if the planned trajectory will cause a constraint violation at the next time step. Hence, a task must be removed from the stack when the predicted value of the overall cost function at the next time step is above a suitably defined threshold, $\overline{\mathcal{C}}$. Let T be the sampling time and κT be the actual time (where κ is an integer), the configuration at the time instant $(\kappa + 1)T$ can be estimated as follows:

$$\widehat{\widetilde{\boldsymbol{q}}}_d(\kappa + 1) = \widetilde{\boldsymbol{q}}_d(\kappa) + T\dot{\widetilde{\boldsymbol{q}}}_d(\kappa). \tag{32}$$

Hence, a task must be removed from the stack if

$$\mathcal{C}_\Sigma\left(\widehat{\widetilde{\boldsymbol{q}}}_d(\kappa + 1)\right) \geq \overline{\mathcal{C}}. \tag{33}$$

Once it has been ascertained that a task must be removed from the stack, the problem is to detect which task has to be removed. To this purpose, different criteria have been proposed in ref. [21], with the aim of verifying the conflict between the constraints and each task. In detail, in ref. [21] two criteria are presented: The first one compares the velocities induced by a subtask and by the gradient of $\mathcal{C}_\Sigma$; the second criterion considers the projection of the gradient onto the null space of Jacobian task. In this paper, a new criterion is presented. Given two generic tasks, whose Jacobians are $\boldsymbol{J}_{t_x}$ and $\boldsymbol{J}_{t_y}$, respectively, they are defined as *annihilating*[1] if

$$\boldsymbol{J}_{t_x}\boldsymbol{J}_{t_y}^{\dagger} = \boldsymbol{O}, \tag{34}$$

where $\boldsymbol{O}$ is the null matrix of suitable dimensions. The annihilation condition can be considered as a compatibility condition between the tasks, since it is equivalent to the orthogonality condition between the subspaces spanned by $\boldsymbol{J}_{t_x}^{\mathrm{T}}$ and $\boldsymbol{J}_{t_y}^{\mathrm{T}}$. Therefore, in order to select the secondary task less compatible with the constraints, the following compatibility metric can be introduced:

$$\mathcal{M}_{t_h} = \left\| \nabla_{\widetilde{\boldsymbol{q}}_d}^{\mathrm{T}} \mathcal{C}_\Sigma \; \boldsymbol{J}_{t_h}^{\dagger} \right\|, \quad h = 1, \ldots, m. \tag{35}$$

The more $\mathcal{M}_{t_h}$ is close to zero, the more the hth task is compatible with the constraints; hence, the task having the maximum value of $\mathcal{M}_{t_h}$ is removed.

The tasks removed by the first layer must be reinserted into the stack as soon as possible, provided that the reinsertion does not cause constraint violation. To this aim, a prediction of the evolution of $\mathcal{C}_\Sigma$ at the next time step is evaluated by considering the effect of each task currently out of the stack, i.e.,

$$\widehat{\widetilde{\boldsymbol{q}}}_{t_h}(\kappa + 1) = \widetilde{\boldsymbol{q}}_d(\kappa) + \boldsymbol{J}_{t_h}^{\dagger}\boldsymbol{e}_{t_h}(\kappa). \tag{36}$$

Therefore, let $\underline{\mathcal{C}} < \overline{\mathcal{C}}$ be a suitably chosen threshold, a task is pushed back into the stack if

$$\mathcal{C}_\Sigma\left(\widehat{\widetilde{\boldsymbol{q}}}_{t_h}(\kappa + 1)\right) \leq \underline{\mathcal{C}}. \tag{37}$$

3.1.2. Smooth transition. Task sequencing might cause discontinuities in the planned joint velocities due to the change of active tasks in the stack.[21,39] In order to achieve a smooth behavior of the motion planner output, for each task a variable gain, ρ_{t_h}, is defined as

$$\rho_{t_h}(t) = \begin{cases} 1 - \mathrm{e}^{-\mu(t-\tau)} & \text{if the } h\text{th task is in the stack,} \\ \mathrm{e}^{-\mu(t-\tau')} & \text{if the } h\text{th task is out of the stack,} \end{cases} \tag{38}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

where τ and τ' are the time instant in which the task is inserted into the stack and the time instant in which it is removed, respectively, and $1/\mu$ is a time constant. These gains guarantee the continuity of the planned joint velocity, $\tilde{\dot{\boldsymbol{q}}}_d$, during the insertion and removal of the tasks.

To sum, the planned joint reference vector for the controller is computed via

$$\tilde{\dot{\boldsymbol{q}}}_d = \widetilde{\boldsymbol{J}}^{\dagger}(\tilde{\boldsymbol{q}}_d, \Delta \boldsymbol{l}_d)\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{\eta}_d, \Delta \boldsymbol{l}_d)\widetilde{\boldsymbol{\Gamma}}(\tilde{\boldsymbol{x}}_o)(\tilde{\dot{\boldsymbol{x}}}_{o_d} + \boldsymbol{K}_o\tilde{\boldsymbol{e}}_o) + \sum_{h=1}^{m} \rho_{t_h} \boldsymbol{N}\left(\boldsymbol{J}^A_{t_h}\right)\boldsymbol{J}^{\dagger}_{t_h}\boldsymbol{K}_{t_h}\boldsymbol{e}_{t_h} - k_{\nabla}\boldsymbol{N}\left(\boldsymbol{J}^A_{t_{m+1}}\right)\nabla^{\mathrm{T}}_{\tilde{\boldsymbol{q}}_d}\mathcal{C}_{\Sigma}. \tag{39}$$

3.2. *Parallel force/pose control*

Since the motion planner provides joint references (i.e., $\boldsymbol{q}_d$ and $\dot{\boldsymbol{q}}_d$) of the overall dual-arm/hand system, any kind of joint motion control can be adopted for the arms, while joint torques for the ith finger are computed according to the following parallel force/pose control law in the operational space,

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^{\mathrm{T}}(\boldsymbol{q}_i)\left(\boldsymbol{K}_P\Delta\boldsymbol{x}_i - \boldsymbol{K}_D\dot{\boldsymbol{x}}_i + \boldsymbol{f}_{di} + k_F\Delta\boldsymbol{f}_{ni} + k_I\int_0^t \Delta\boldsymbol{f}_{ni}\mathrm{d}\zeta + \boldsymbol{g}_i(\boldsymbol{q}_i)\right), \tag{40}$$

where $\boldsymbol{g}_i(\boldsymbol{q}_i)$ is the vector of the generalized gravity force acting on finger i, $\Delta\boldsymbol{x}_i$ denotes the pose error of finger i between the desired value $\boldsymbol{x}_{i_d}$, corresponding to $\boldsymbol{q}_{d_i}$, and the current one, $\boldsymbol{x}_i$, with respect to the palm frame Σ_{rh} (or Σ_{lh}), $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are gain matrices, k_F and k_I are positive scalar gains, and $\Delta\boldsymbol{f}_{n_i} = [\Delta f_{n_i}\hat{\boldsymbol{n}}_i^{\mathrm{T}} \quad \boldsymbol{0}^{\mathrm{T}}]^{\mathrm{T}}$, being Δf_{n_i} the projection of the force error along the normal to the object surface, $\hat{\boldsymbol{n}}_i$, at the contact point i. The control law (40) allows to track the assigned contact forces which are, in turn, imposed to avoid contact breaks or excessive stresses on the manipulated object, even in the presence of uncertainties.

3.2.1. Stability analysis. In order to prove stability of the system under the control law (40) the dynamic model in the operational space[38] of the ith finger should be considered

$$\boldsymbol{M}_i(\boldsymbol{x}_i)\ddot{\boldsymbol{x}}_i + \boldsymbol{C}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i)\dot{\boldsymbol{x}}_i + \boldsymbol{g}_i(\boldsymbol{x}_i) = \boldsymbol{u}_i - \boldsymbol{f}_i, \tag{41}$$

where $\boldsymbol{M}_i$ is the (6×6) inertia matrix of the ith finger, $\boldsymbol{C}_i$ is the (6×6) matrix collecting the centrifugal and Coriolis terms, $\boldsymbol{f}_i$ is the (6×1) vector of generalized contact forces (acting at the fingertip), $\boldsymbol{u}_i$ is the (6×1) vector of driving generalized forces, through which the control torques can be obtained via

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^{\mathrm{T}}(\boldsymbol{q}_i)\boldsymbol{u}_i. \tag{42}$$

Hereafter the subscript i will be dropped for notation compactness. The following properties hold:[15,37,38]

1. $\boldsymbol{M}$ is symmetric and positive definite; therefore, if $\lambda_m(\cdot)$ ($\lambda_M(\cdot)$) denotes the minimum (maximum) eigenvalue, it is

$$0 < \lambda_m(\boldsymbol{M})\boldsymbol{I}_6 \leq \boldsymbol{M}(\boldsymbol{x}) \leq \lambda_M(\boldsymbol{M})\boldsymbol{I}_6, \tag{43}$$

   where $\lambda_M(\boldsymbol{M}) < \infty$ if all joints are revolute.
2. There always exists a choice of $\boldsymbol{C}$ such that

$$\dot{\boldsymbol{M}}(\boldsymbol{x}) = \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{x}, \dot{\boldsymbol{x}}), \tag{44}$$

   moreover, $\boldsymbol{C}$ can be upper-bounded as follows $\dot{\boldsymbol{x}}$

$$\|\boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\| \leq k_c\|\dot{\boldsymbol{x}}\|, \tag{45}$$

   with $k_c > 0$.

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

The following assumptions have been considered:

Assumption 1. Pose and force references are constant, i.e., $\dot{\boldsymbol{x}}_d = \dot{\boldsymbol{f}}_d = \boldsymbol{0}$.

Assumption 2. Quasi-static object manipulation, i.e., $\dot{\boldsymbol{v}}_o = \boldsymbol{v}_o = \boldsymbol{0}$.

Assumption 3. The force along the normal to the contact surface is assumed of elastic type, i.e., $\boldsymbol{f}_n = k\Delta l\hat{\boldsymbol{n}} = f_n\hat{\boldsymbol{n}}$.

Assumption 4. The object has a convex surface. For this kind of object and for quasi-static manipulation the time derivative of the unit vector normal to object surface at contact point (see Appendix A for further details) can be norm-bounded as follows:

$$\|\dot{\hat{\boldsymbol{n}}}\| \leq k_n \|\dot{\boldsymbol{p}}_f\|, \tag{46}$$

where $\boldsymbol{p}_f = \boldsymbol{o}_{f_i} - \boldsymbol{o}_{jh}$ is the position of Σ_{f_i} with respect to the palm frame Σ_{jh} ($j = \{l, r\}$ for left and right respectively) expressed in base frame coordinates.

By taking into account the elasticity of the normal force (Assumption 3) and by considering the object quasi-static (Assumption 2), the following relationship between the force and position errors can be derived,

$$\Delta f_n = k\,(\Delta l_d - \Delta l) = k\boldsymbol{n}^T\Delta\boldsymbol{x}, \tag{47}$$

where $\boldsymbol{n} = [\hat{\boldsymbol{n}}^{\mathrm{T}} \quad \boldsymbol{0}^{\mathrm{T}}]^{\mathrm{T}}$ is a (6×1) unit vector. By virtue of integral action in Eqs. (40) and (47), system (41) under the control law (40) has a unique equilibrium at $\boldsymbol{x}_\infty = \boldsymbol{x}_d$ and $\boldsymbol{f}_{n_\infty} = \boldsymbol{f}_d$ (see Appendix B).

In order to study the stability of the equilibrium, it is convenient to consider a (13×1) state vector,[37]

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \Delta\dot{\boldsymbol{x}} \\ \Delta\boldsymbol{x} \\ \Delta s \end{bmatrix}, \tag{48}$$

where

$$\Delta s = s_\infty - s = s_\infty - \int_0^t \left(\Delta f_n - \frac{k}{\rho}\dot{\boldsymbol{n}}^{\mathrm{T}}\Delta\boldsymbol{x}\right) d\zeta, \tag{49}$$

ρ is a positive constant, k is the stiffness of the elastic pad, and s_∞ is the value of s at the equilibrium (the explicit expression of s_∞ is given in Appendix C). The augmented state dynamics is thus given by

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}, \tag{50}$$

with

$$\boldsymbol{A} = \begin{bmatrix} -\boldsymbol{M}^{-1}(\boldsymbol{C} + \boldsymbol{K}_D) & -\boldsymbol{M}^{-1}(\boldsymbol{K}_P + \boldsymbol{F}) & k_I\boldsymbol{M}^{-1}\boldsymbol{n} \\ \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{0} \\ 0 & -k\left(\boldsymbol{n} - \dfrac{\dot{\boldsymbol{n}}}{\rho}\right)^{\mathrm{T}} & 0 \end{bmatrix}, \tag{51}$$

$$\boldsymbol{b} = [k_I I_n(\boldsymbol{M}^{-1}\boldsymbol{n})^{\mathrm{T}} \quad \boldsymbol{0}^{\mathrm{T}} \quad 0]^{\mathrm{T}}, \tag{52}$$

where the dependencies of $\boldsymbol{M}$ and $\boldsymbol{C}$ upon $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ have been dropped, $\boldsymbol{F} = (1 + k_f)k\boldsymbol{n}\boldsymbol{n}^T$ and

$$I_n = -s_\infty - \int_0^t \frac{k}{\rho}\dot{\boldsymbol{n}}^{\mathrm{T}}\Delta\boldsymbol{x}\, d\zeta. \tag{53}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

Theorem 1. *There exists a set of parameters $\boldsymbol{K}_P$, $\boldsymbol{K}_D$, k_f and k_I such that $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are locally asymptotically convergent to* $\mathbf{0}$.

Proof of Theorem 1. Consider the candidate Lyapunov function

$$V = \frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{z}, \tag{54}$$

where $\boldsymbol{P}$ is a symmetric matrix,

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{M} & \rho\boldsymbol{M} & \mathbf{0} \\ \rho\boldsymbol{M} & \rho\boldsymbol{K}_D + \boldsymbol{K}_P & -k_I\boldsymbol{n} \\ \mathbf{0} & -k_I\boldsymbol{n}^T & \rho\dfrac{k_I}{k} \end{bmatrix}, \tag{55}$$

positive definite under the following condition

$$\rho\lambda_m(\boldsymbol{K}_D) + \lambda_m(\boldsymbol{K}_P) > \max\left\{\frac{2\rho^2\lambda_M(\boldsymbol{M})^2}{\lambda_m(\boldsymbol{M})}, \frac{k_I k}{\rho}\right\}. \tag{56}$$

Under condition (56), the function V can be bounded as

$$\frac{1}{2}\underline{\lambda}_m(\boldsymbol{P})\|z\|^2 \leq V \leq \frac{1}{2}\overline{\lambda}_M(\boldsymbol{P})\|z\|^2, \tag{57}$$

where, since $\boldsymbol{P}$ is time varying, $\underline{\lambda}_m = \min_{t\geq 0}\{\lambda_m(\boldsymbol{P}(t))\}$ and $\overline{\lambda}_M = \max_{t\geq 0}\{\lambda_M(\boldsymbol{P}(t))\}$.

Consider the state–space domain defined as $\mathcal{D} = \{z : \|z\| < \Phi\}$. It can be recognized that the following inequality holds in the domain $\mathcal{D}$:

$$I_n \leq \frac{k}{\rho}\Phi k_n\|z_2\|. \tag{58}$$

Some details about inequality (58) are given in Appendix C.

The time derivative $\dot{V}$ is given by

$$\dot{V} = \boldsymbol{z}^{\mathrm{T}}\left(\boldsymbol{P}\boldsymbol{A} + \frac{1}{2}\dot{\boldsymbol{P}}\right)\boldsymbol{z} + \boldsymbol{z}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{b}, \tag{59}$$

where $\dot{\boldsymbol{P}}$ can be computed, by exploiting Property 2, as

$$\dot{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{C} + \boldsymbol{C}^{\mathrm{T}} & \rho(\boldsymbol{C} + \boldsymbol{C}^{\mathrm{T}}) & \mathbf{0} \\ \rho(\boldsymbol{C} + \boldsymbol{C}^{\mathrm{T}}) & \boldsymbol{O} & -k_I\dot{\boldsymbol{n}} \\ \mathbf{0} & -k_I\dot{\boldsymbol{n}}^{\mathrm{T}} & 0 \end{bmatrix}. \tag{60}$$

After some algebraic steps, Eq. (59) becomes

$$\begin{aligned}\dot{V} = &-\boldsymbol{z}_1^{\mathrm{T}}(\boldsymbol{K}_D - \rho\boldsymbol{M})\boldsymbol{z}_1 - \boldsymbol{z}_2^{\mathrm{T}}\left(\rho(\boldsymbol{K}_P + \boldsymbol{F}) - k_I k\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}} + \frac{k_I k}{\rho}\boldsymbol{n}\dot{\boldsymbol{n}}^{\mathrm{T}}\right)\boldsymbol{z}_2 \\ &-\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{F}\boldsymbol{z}_2 + \rho\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{z}_1 + k_I I_n(\boldsymbol{z}_1 + \rho\boldsymbol{z}_2)^{\mathrm{T}}\boldsymbol{n}.\end{aligned} \tag{61}$$

By exploiting Assumption 4 and Property 2, the time derivative $\dot{V}$ can be upper-bounded as follows,

$$\begin{aligned}\dot{V} \leq &-(\lambda_m(\boldsymbol{K}_D) - \rho\lambda_M(\boldsymbol{M}) - \rho k_c\Phi)\|\boldsymbol{z}_1\|^2 - (\rho\lambda_m(\boldsymbol{K}_P) - k_I k - k_I k\Phi k_n)\|\boldsymbol{z}_2\|^2 \\ &+ \left(k(1 + k_f) + 2\frac{k_I k}{\rho}k_n\Phi\right)\|\boldsymbol{z}_1\|\|\boldsymbol{z}_2\|,\end{aligned} \tag{62}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

and rearranged in a suitable quadratic form as

$$\dot{V} \leq -[\,\|z_1\|\ \|z_2\|\,]\,\boldsymbol{Q}\begin{bmatrix}\|z_1\|\\ \|z_2\|\end{bmatrix}, \tag{63}$$

where $\boldsymbol{Q}$ is the (2×2) matrix,

$$\mathbf{Q} = \begin{bmatrix} \lambda_m(\boldsymbol{K}_d) - \rho\lambda_M(\boldsymbol{M}) - \rho k_c\Phi\,, & -\frac{1}{2}\left(k(1+k_f) + 2\frac{k_I k}{\rho}k_n\Phi\right) \\ -\frac{1}{2}\left(k(1+k_f) + 2\frac{k_I k}{\rho}k_n\Phi\right), & \rho\lambda_m(\boldsymbol{K}_p) - k_I k(1+\Phi k_n) \end{bmatrix}. \tag{64}$$

On the basis of (63) and (64), $\dot{V}$ is negative semi-definite in the domain $\mathcal{D}$ provided that $\boldsymbol{Q}$ is positive definite, i.e., if the following inequality holds,

$$\lambda_m(\boldsymbol{K}_d) \geq \rho\lambda_M(\boldsymbol{M}) + \rho k_c\Phi + \max\left\{0, \frac{\varphi}{4}\right\}, \tag{65}$$

where

$$\varphi = \left(k(1+k_f) + 2\frac{k_I k}{\rho}k_n\Phi\right)^2 (\rho\lambda_m(\boldsymbol{K}_P) - k_I k\,(1+\Phi k_n))^{-1}. \tag{66}$$

Moreover, since V is a non-increasing function along the system trajectories, inequality (57) guarantees that all $z(t)$ trajectories starting in the domain

$$\mathcal{D}_0 = \left\{ z : \|z(0)\| < \Phi\sqrt{\frac{\underline{\lambda}_m(\boldsymbol{P})}{\overline{\lambda}_m(\boldsymbol{P})}} \right\} \tag{67}$$

remain in the domain $\mathcal{D}$, $\forall t > 0$.

Finally, since $\dot{V} = 0$ only if $z_1 = \mathbf{0}$ and $z_2 = \mathbf{0}$, by invoking the La Salle's theorem,[14] it can be recognized that if $z(0) \in \mathcal{D}_0$, z_1 and z_2 asymptotically converge to $\mathbf{0}$ while z_3 is only bounded. ■

Since $\Delta\dot{\boldsymbol{x}}$ and $\Delta\boldsymbol{x}$ are asymptotically convergent to $\mathbf{0}$, by recalling (47) it can be seen that Δf_n asymptotically converges to 0 as well.

It is worth noting that, different from Siciliano and Villani,[37] it has been proven that system (41), under the control law (40), is locally stable even when a non-planar convex surface is considered.

4. Case Study

4.1. Set-up configuration

The proposed scheme has been tested in simulation on the dual-arm/hand manipulation system (Fig. 4) grasping a cardboard box and composed by two identical planar grippers, each composed by two branches and 7-DOF, resulting in a total of $N = 4$ fingers and 14 active joints. It is assumed that in its initial configuration the system grasps the object with tips 1 and 2 ensuring force closure since the contact normal forces are acting on the same straight line,[26] while tips 3 and 4 are also in contact but in arbitrary way. The main task consists in keeping the object still, thanks to fingers 1 and 2, while tips 3 and 4 move in order to achieve a force closure condition upon the object in a dexterous configuration without violating a certain number of limits and constraints. Then fingers 1 and 2 can leave the object, simulating in this way an hand-to-hand object passing. The force control loop ensures that the planned forces are applied on the object. In this case study, the desired forces for tips 3 and 4 are set close to zero, since they have to slide, but not exactly zero because contact continuity should be ensured during the whole motion. Concerning fingers 1 and 2, higher values have been considered in such a way to hold the object without excessive stresses.

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

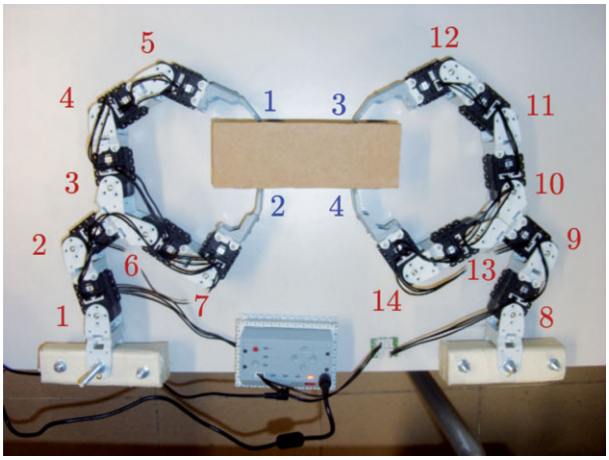


Fig. 4. (Colour online) Dual-arm/hand experimental setup built by using the Bioloid© Expert Kit. The red numbers label the joints. The blue numbers indicate the tips of the fingers.

The planner (Eq. (39)) and the controller (Eq. (40)) have been developed in the Matlab© environment, while GRASPIT! has been used as a dynamic simulator.

4.2. Dynamic simulation environment

GRASPIT! offers a dynamic engine which allows to deal with contact mechanics in a realistic way, since it is possible to simulate hard finger contacts (as well as point contacts without friction) respecting non-penetration constraint. Frictional forces and non-penetration constraints are expressed via inequalities; thus, a Linear Complementary Problem (LCP) is solved by GRASPIT! at each time step by using Lemke's algorithm.[23] Moreover, a collision detection system acts in such a way to prevent collisions within bodies as well as to identify and mark contact regions.

GRASPIT! also provides C-MEX functions which allow communication with Matlab©: It is possible to assign joint torques (only when the dynamic mode is enabled) to the manipulation system as well as read joint positions and contact forces. Some modifications to the source code have been done to retrieve end-effector pose, choose the reference frame in which contact forces are provided to Matlab©, and include prismatic dynamic joint class.[1]

The dual-arm/hand system model has been added to the GRASPIT! robot library; accurate values of mass and geometric parameters have been set on the basis of available datasheets.

The elastic contact, described in Section 2.2, has been modelled by using a rotational joint and a prismatic one, acting like a spring-damper systems, in such a way to ensure the elasticity in the direction of the object surface normal at each contact point.

4.3. Secondary tasks and constraints

Different secondary tasks have been considered: The first two, aimed at choosing the optimal contact points, are related to the grasp force-closure condition, the other one is related to a measure of the grasp quality, while the last one is regarding the manipulability of the dual-arm/hand system. On the other hand, two physical constraints have been considered: joint limits and collision avoidance.

4.3.1. Unit frictionless equilibrium. By moving the contact points on the object surface until the unit frictionless equilibrium is reached, it is possible to guarantee the grasp force-closure condition.[26] Such equilibrium is satisfied when two positive indices, called frictionless force (ε_f) and moment

[1] The modified source code of GRASPIT! is available for download for Linux platform at `http://www.unibas.it/automatica/laboratory.html`

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

(ε_m) residuals, are zero,[8,30]

$$\begin{aligned} \varepsilon_f &= \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{f} \qquad \boldsymbol{f} = \sum_{i=1}^{N} \hat{\boldsymbol{n}}_i^o, \\ \varepsilon_m &= \frac{1}{2}\boldsymbol{m}^{\mathrm{T}}\boldsymbol{m} \qquad \boldsymbol{m} = \sum_{i=1}^{N} \boldsymbol{c}_i^o \times \hat{\boldsymbol{n}}_i^o, \end{aligned} \tag{68}$$

where $N = 4$ is the number of fingers, and $\hat{\boldsymbol{n}}_i^o(\boldsymbol{\xi}_i)$ is the surface normal to the ith contact point, referred to the object frame. It has been shown that, for two or more contact points, unit frictionless equilibrium is a force closure condition for any nonzero friction coefficient.[30,31]

The Jacobian matrix of the unit frictionless force residual is given by

$$\boldsymbol{J}_{\varepsilon_f} = \frac{\partial \varepsilon_f}{\partial \widetilde{\boldsymbol{q}}} = \frac{\partial \varepsilon_f}{\partial \boldsymbol{f}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{q}}} = \boldsymbol{f}^{\mathrm{T}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{q}}}, \tag{69}$$

where $\boldsymbol{\xi} = [\boldsymbol{\xi}_1^{\mathrm{T}} \cdots \boldsymbol{\xi}_N^{\mathrm{T}}]^{\mathrm{T}}$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}} = [\frac{\partial \hat{\boldsymbol{n}}_1^o}{\partial \boldsymbol{\xi}_1} \cdots \frac{\partial \hat{\boldsymbol{n}}_N^o}{\partial \boldsymbol{\xi}_N}]$. As for the unit frictionless momentum residual the Jacobian can be computed as

$$\boldsymbol{J}_{\varepsilon_m} = \frac{\partial \varepsilon_m}{\partial \widetilde{\boldsymbol{q}}} = \frac{\partial \varepsilon_m}{\partial \boldsymbol{m}} \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{q}}} = \boldsymbol{m}^{\mathrm{T}} \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{q}}}, \tag{70}$$

with $\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{\xi}} = [\frac{\partial (\boldsymbol{c}_1^o \times \hat{\boldsymbol{n}}_1^o)}{\partial \boldsymbol{\xi}_1} \cdots \frac{\partial (\boldsymbol{c}_N^o \times \hat{\boldsymbol{n}}_N^o)}{\partial \boldsymbol{\xi}_N}]$.

It is worth noting that, since the considered object is rectangular and the opposite fingers of each hand are on the opposite sides of the rectangle, the force residual index is always zero during the whole case study, therefore it is not considered in the following.

4.3.2. Grasp quality. The unit frictionless equilibrium is necessary to achieve the positions of the fingertips on the object surface ensuring that the external wrenches acting on the object can be balanced by the fingers. A subset of these positions might be selected according to a grasp quality index. In general, several indices can be considered: In this case study, the fingers are commanded to reach a symmetric position with respect to the object's center. In detail, the following task function is considered:

$$\sigma_{s_i} = \begin{cases} |\xi_{d_i} - \xi_i| & \text{if } |\xi_{d_i} - \xi_i| > \overline{\xi}_i \\ 0 & \text{otherwise} \end{cases}, \tag{71}$$

where $\overline{\xi}_i$ is a threshold for the task activation and ξ_{d_i} is the desired value for the ith finger contact variable, with $i = 3, 4$. The desired value, $\sigma_{d_{s_i}}$, is zero. The meaning of (71) is that the contact variables for fingers 3 and 4, the only fingers that can slide, should reach the desired position on the object, represented by the values ξ_{d_3} and ξ_{d_4}, on the basis of the positions of fingers 1 and 2 on the object, denoted by the constant values ξ_1 and ξ_2 respectively.

Let $\boldsymbol{\sigma}_s = [\sigma_{s_3}\ \sigma_{s_4}]^{\mathrm{T}}$, the Jacobian $\boldsymbol{J}_{\sigma_s}(\boldsymbol{\xi})$ for the symmetric grasp subtask can be computed as $\partial \boldsymbol{\sigma}_s / \partial \widetilde{\boldsymbol{q}}$.

4.3.3. Manipulability. In order to keep the manipulation system far from singularities, the manipulability index presented in ref. [38] can be considered for the ith finger,

$$w_i(\boldsymbol{q}_i) = \sqrt{det\left(\boldsymbol{J}_{F_i}(\boldsymbol{q}_i)\boldsymbol{J}_{F_i}^{\mathrm{T}}(\boldsymbol{q}_i)\right)}, \quad i = 1, \ldots, 4. \tag{72}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

However, a simplified manipulability index, computationally simpler than (72) but still describing in an effective way the distance from kinematic singularities, is adopted for the considered setup, i.e.,

$$\begin{aligned} w_1 &= 0.5\left(s_2^2 + s_3^2 + s_4^2 + s_5^2\right), \\ w_2 &= 0.5\left(s_2^2 + s_6^2 + s_7^2\right), \\ w_3 &= 0.5\left(s_9^2 + s_{10}^2 + s_{11}^2 + s_{12}^2\right), \\ w_4 &= 0.5\left(s_9^2 + s_{13}^2 + s_{14}^2\right), \end{aligned} \tag{73}$$

where $s_\alpha = \sin(q_\alpha)$.

Hence, the following task function is considered

$$\sigma_{w_i} = \begin{cases} |w_{d_i} - w_i| & \text{if } |w_{d_i} - w_i| > \overline{w}_i \\ 0 & \text{otherwise} \end{cases}, \tag{74}$$

where $\overline{w}_i$ is a threshold for the task activation and w_{d_i} is the desired value for the ith finger manipulability, with $i = 1, \ldots, 4$. The desired value, $\sigma_{d_{w_i}}$, is zero and a vectorial task function $\boldsymbol{\sigma}_w = [\sigma_{w_1} \quad \ldots \quad \sigma_{w_4}]^{\mathrm{T}}$ is considered.

The Jacobian $\boldsymbol{J}_{\sigma_w}(\boldsymbol{q})$ for the manipulability subtask can be computed as $\partial \boldsymbol{\sigma}_w / \partial \widetilde{\boldsymbol{q}}$.

4.3.4. Joint-limit avoidance. A physical constraint to the motion of the system is imposed by the mechanical joint limits. The system configuration is considered safe if $q_j \in [\underline{q}_j, \overline{q}_j]$, for $j = 1, \ldots, 14$, with $\underline{q}_j$ and $\overline{q}_j$ suitable chosen values far enough from the mechanical limits. The related cost function is chosen as follows

$$\begin{aligned} \mathcal{C}_{JL}(\boldsymbol{q}) &= \sum_{j=1}^{14} c_j(q_j), \\ c_j(q_j) &= \begin{cases} k_j e^{\delta(q_j - \underline{q}_j)^2} - 1, & \text{if } q_j \le \underline{q}_j, \\ 0, & \text{if } \underline{q}_j < q_j \le \overline{q}_j, \\ k_j e^{\delta(q_j - \overline{q}_j)^2} - 1, & \text{if } q_j > \overline{q}_j, \end{cases} \end{aligned} \tag{75}$$

where k_j and δ are positive constants.

4.3.5. Collision avoidance. In order to avoid collisions between the fingers, it is imposed that the distance between the fingers be larger than a safety value, d_s; hence, if $d_{ii'}$ denotes the distance between the ith and the i'th finger, the following cost function can be considered,

$$\mathcal{C}_{CA}(\widetilde{\boldsymbol{q}}) = \sum_{i,i'} c_{ii'}(\widetilde{\boldsymbol{q}}), \tag{76}$$

where the sum is extended to all the couples of fingers,

$$c_{ii'}(d_{ii'}) = \begin{cases} k_{ii'} \dfrac{d_s - d_{ii'}}{d_{ii'}^2} & \text{if } d_{ii'} \le d_s \\ 0 & \text{if } d_{ii'} > d_s \end{cases}, \tag{77}$$

and $k_{ii'}$ is a positive gain.

4.4. Simulation results

4.4.1. Parameters. The elastic contact parameters are: 1000 N/m for the spring elastic coefficients, 20 Ns/m for the spring damper coefficients of all fingers, while $l_i = 24.5 \times 10^{-3}$ m is the spring rest position, with $i = 1, \ldots, 4$. Concerning the planner (39), the gain for the object pose error has been tuned to $\boldsymbol{K}_o = 450\, \boldsymbol{I}_{12}$, while the pseudo-inverse of $\widetilde{\boldsymbol{J}}$, $\widetilde{\boldsymbol{J}}^{\dagger} = \boldsymbol{W}^{-1}\widetilde{\boldsymbol{J}}^{\mathrm{T}}(\widetilde{\boldsymbol{J}}\boldsymbol{W}^{-1}\widetilde{\boldsymbol{J}}^{\mathrm{T}})^{-1}$, has been

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

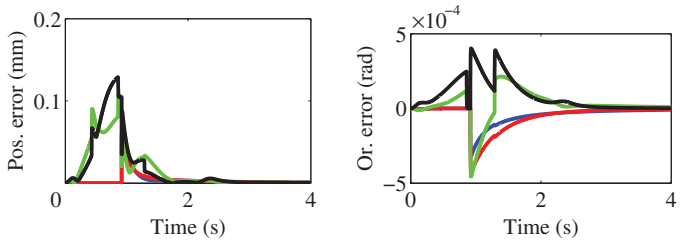


Fig. 5. (Colour online) Object's pose error computed on the basis of direct kinematics of each extended finger. Left: norm of object's position error; right: object's orientation error. Finger 1 is represented in blue, finger 2 in red, finger 3 in green, and finger 4 in black.

weighted by the matrix $\boldsymbol{W} = \text{diag}([4\ \ 4\ \ \boldsymbol{e}_{11}\ \ 4\ \ 4\ \ \boldsymbol{e}_{11}])$, where $\boldsymbol{e}_\alpha$ is a $(1 \times \alpha)$ vector of ones, in order to limit the motion of the arms with respect to that of fingers, assuming that fingers' motion is less demanding in terms of power consumption. The object is required to keep its initial position of $[0\ \ 0.1]^{\mathrm{T}}$ m and orientation of 0 rad during the whole task.

The parameters used to define the secondary tasks are chosen as follows: $\overline{\xi}_i = 0$, with $i = 3, 4$, $\xi_{d_3} = -30 \times 10^{-3}$ m, $\xi_{d_4} = 84.5 \times 10^{-3}$ m, for the quality index subtask, $\overline{w}_i = 0$, with $i = 1, \ldots, 4$, $w_{d_1} = w_{d_3} = 1.80$, $w_{d_2} = w_{d_4} = 1.30$, for the manipulability subtask. Note that both activation thresholds have been put to zero in order to precisely reach their null error conditions. Subtask gains are set as follows: $k_{t_1} = 30$, $\boldsymbol{K}_{t_2} = 73.5\, \boldsymbol{I}_2$, and $\boldsymbol{K}_{t_3} = 180\, \boldsymbol{I}_4$.

Since the mechanical limit of the joints is about ± 1.74 rad, the following safety thresholds for joint limits avoidance have been set: $\overline{q}_j = 1.6$ rad, $\underline{q}_j = -1.6$ rad; moreover, the other parameters in (75) are $\delta = 2.2$ and $k_j = 2$ for $j = 1, \ldots, 14$. As for the collision avoidance, the safety distance d_s has been set to 50×10^{-3} m and the gain $k_{ii'}$ is equal to 1 for all couples of fingers.

The task has a duration of 4 s. A Runge–Kutta integration method, with time-step of 0.2 ms, has been used to simulate the system.

The trajectories of the active joints computed by the motion planner are the references for the control law (Eq. (40)). The parameters in such equation are chosen as follows: $\boldsymbol{K}_P = 2 \times 10^5 \boldsymbol{I}_3$, $\boldsymbol{K}_D = 150\ \boldsymbol{I}_3$, $k_{F_1} = k_{F_2} = 12.5$, $k_{F_3} = k_{F_4} = 1.25$, $k_I = 10$. The desired values for the contact normal forces are 2 N, -2 N, 0.2 N, and -0.2 N for fingers 1, 2, 3, and 4 respectively. The first two contact normal forces are bigger, since the corresponding fingers have to keep the object still while the other two slide along the surface (i.e., small contact normal force values are required) in order to reach a force closure condition.

4.4.2. Motion planner. The planner performance are summarized in Figs. 5 and 6. In detail, Fig. 5 shows the time history of the norm of the object's pose error computed on the basis of direct kinematics of each finger. It can be noted that the error asymptotically goes to zero for each extended finger. This proves the performance of the planner (Eq. (39)).

Figure 6(a) depicts the time history of the stack status. The main task, with priority 1, is never removed from the stack, while the other tasks, numbered from 2 to 4 in the same order as they have been described above, are removed when some constraints are near to be violated. Note that task 3 is never removed from the stack since, in this case, it never affects the constraints. When the system is in a safe condition with respect to the constraints, the tasks are reinserted in the stack, maintaining their previous priorities. Moreover, it can be noted that the peaks in the time histories of the object's pose error correspond to task insertion and/or removal.

Figures 6(b) and 6(c) show the cost functions related to the joint limits and collision-avoidance constraints respectively. In the first phase, their values increase and for this reason the tasks farthest from the annihilating condition are removed from the stack. When their values become almost zero, the removed tasks are reinserted into the stack.

Figure 6(d) shows the moment residual ϵ_m. This asymptotically converges to zero, i.e., fingers 3 and 4 reach a force closure condition. Figure 6(e) depicts the time histories of the grasp quality indices σ_{s_3} and σ_{s_4}. Such values converge to zero since both fingers 3 and 4 reach a symmetric position with respect to the object's center, from fingers 1 and 2, respectively. Finally, Fig. 6(f) shows the time

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

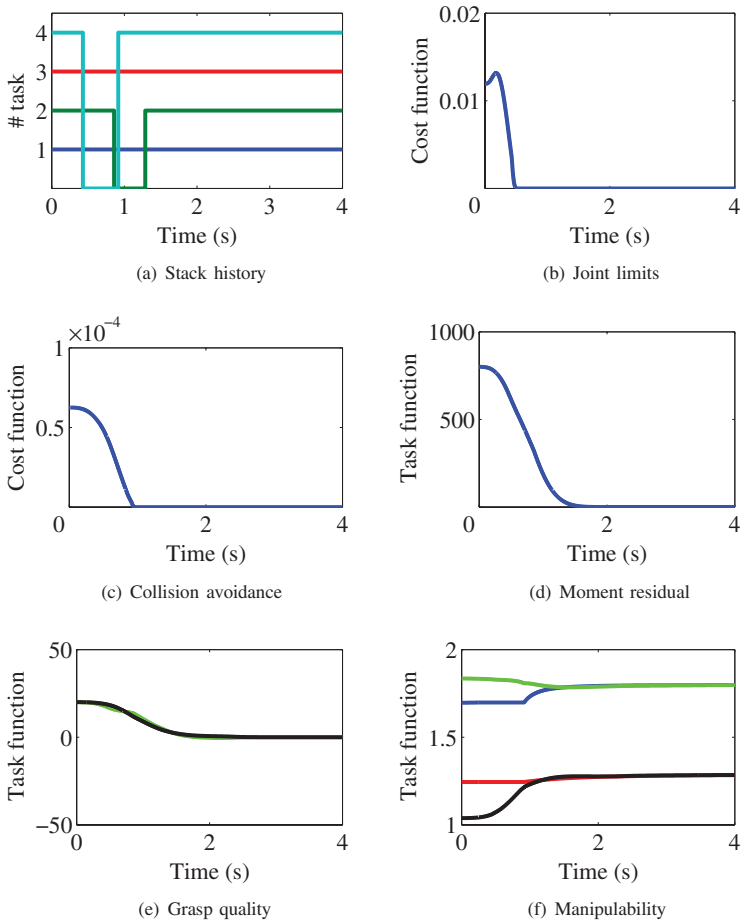


Fig. 6. (Colour online) Time histories of the constraints and secondary tasks. Sub-figures (e) and (f) use the same color legend as Fig. 5.

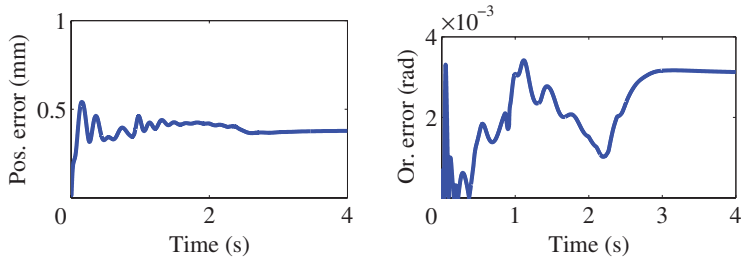


Fig. 7. (Colour online) Object's pose error. Left: norm of object's position error; right: object's orientation error.

history of the manipulability measures w_i, with $i = 1, \ldots, 4$, for each finger. The depicted values are equal to or above the desired ones w_{d_i}.

4.4.3. Controller. The controller performances are summarized in Figs. 7–9. In detail, Fig. 7 shows the time history of the norm of the object's pose error. It can be noted that the errors do not converge to zero, but they present a constant offset. This is due to the absence, in the control scheme, of the feedback of the object pose. In fact, sliding of fingers 3 and 4 affects the object's motion, while the off-line planner cannot take into account these disturbances.

Figure 8 depicts the time histories of the errors of the normal contact forces with respect to the desired ones. It could be noted that all the errors converge asymptotically to zero. Some peaks occur in correspondence to task removal/insertion from/in the stack.

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

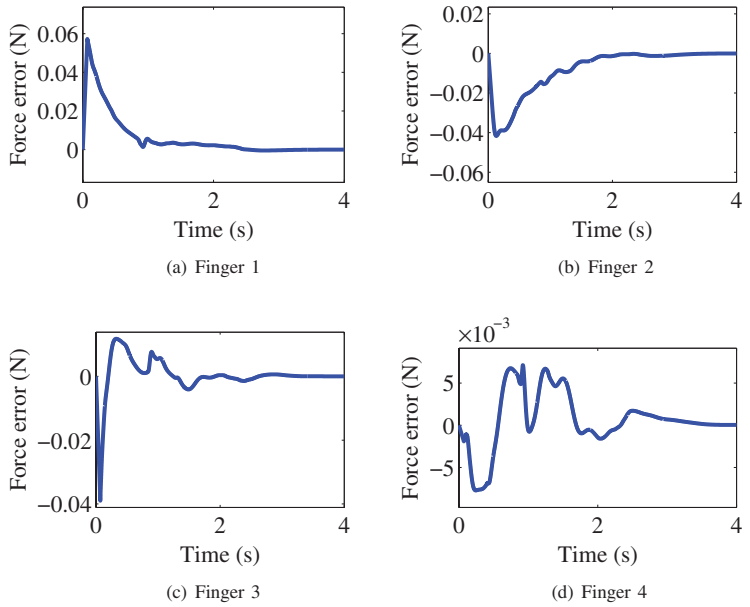


Fig. 8. (Colour online) Time histories of the contact normal forces errors.

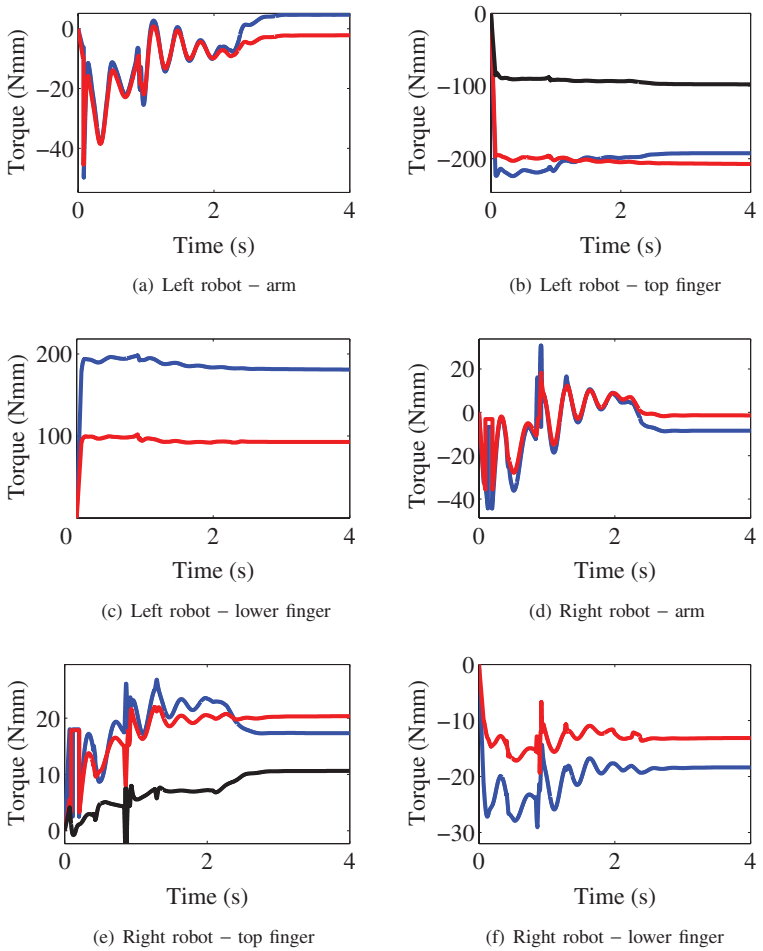
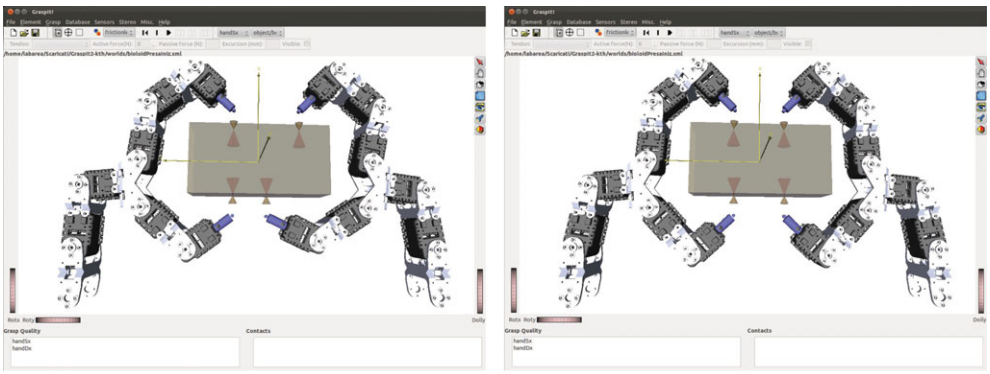


Fig. 9. (Colour online) Time histories of joint torques. Color legend for sub-figures (a) and (d): blue are joint torques 1 and 8, and red are joint torques 2 and 9. Color legend for sub-figures (b) and (e): blue are the joint torques 3 and 10, and red are joint torques 4 and 11, black are joint torques 5 and 12. Color legend for sub-figures (c) and (f): blue are joint torques 6 and 13, and red are joint torques 7 and 14.

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press



(a) Initial configuration (b) Final configuration

Fig. 10. (Colour online) GRASPIT! screenshots depicting the system in its initial and final configurations. Prismatic joints that model the fingers' elastic pads have not been drawn.

Figure 9 shows the time histories of the joint actuation torques. Their values are smooth and suitable with respect to common available motors in the market.

Finally, Fig. 10 shows the initial and final configurations of the system. It can be noted that fingers 3 and 4 move along the object surface until their tips are on the same straight line on the opposite sides of the object so as to ensure both force closure and symmetric position with respect to the object's center.

5. Conclusion

In this paper the kinematic model of a redundant dual-arm/hand robotic manipulation system has been derived. This model allows to compute the position and orientation of a grasped object from the joint variables of each arm and finger that can be actuated (active joints) as well as from a set of contact variables. A kinematic planner and a parallel position/force controller have been designed to achieve the desired object motion and the desired contact normal forces. The redundancy of the whole system has been managed at the kinematic level in order to fulfill a set of prioritized constraints and secondary tasks. The latter are aimed at ensuring grasp stability and dexterity, without violating physical constraints. To this aim, a prioritized task sequencing algorithm with smooth transitions between tasks has been employed. The controller has been designed to execute motion references provided by the planner and, at the same time, maintain a desired contact force exerted by each finger on the grasped object. Simulation results show that the adopted control scheme ensures successful achievement of the main task without violating any imposed constraint.

The contribution of the paper can be summarized as follows: The work presented in ref. [5] has been extended with all the details and the proofs; the parallel force/position control has been proven to converge even with non-planar surfaces; the framework devoted to the sub-tasks switching has been formalized and a new criterion for tasks' removal has been introduced; and the simulator GraspIt! has been suitably adapted and redistributed.

Acknowledgments

The research leading to these results has received funding from the Italian Government under grant PRIN 2009 no. 20094WTJ29 (project RoCoCo), and from the European Community's Seventh Framework Programme under grant agreement no. 287617 (collaborative project ARCAS).

The research activity of Giuseppe Muscio is supported by Agenzia Spaziale Italiana (ASI).

References

1. G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.* **25**, 985–994 (2009).
2. A. Bicchi and D. Prattichizzo, "Manipulability of cooperative robots with unactuated joints and closed-chain mechanisms," *IEEE Trans. Robot. Autom.* **16**, 336–345 (2000).

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

3. F. Caccavale, C. Natale, B. Siciliano and L. Villani, "Six-DOF impedance control based on angle/axis representations," *IEEE Trans. Robot. Autom.* **15**, 289–300 (1999).
4. F. Caccavale and M. Uchiyama, "Cooperative Manipulators," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer, Berlin, 2008) pp. 701–718.
5. F. Caccavale, V. Lippiello, G. Muscio, F. Pierri, F. Ruggiero and L. Villani, "Kinematic Control with Force Feedback for a Redundant Bimanual Manipulation System," **In**: *Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA, 2011) pp. 4194–4200.
6. S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Trans. Robot. Autom.* **4**, 361–373 (1993).
7. S. Chiaverini, B. Siciliano and L. Villani, "Force/position regulation of compliant robot manipulators," *IEEE Trans. Autom. Control* **39**, 647–652 (1994).
8. J. Coelho and R. Grupen, "A control basis for learning multifingered grasps," *J. Robot. Syst.* **14**, 545–557 (1997).
9. A. DasGupta and H. Hatwal, "Dynamics and nonlinear coordination control of multi-fingered mechanical hands," *ASME, J. Dyn. Syst. Meas. Control* **120**, 275–281 (1998).
10. Z. Doulgeri, J. Fasoulas and S. Arimoto, "Feedback control for object manipulation by a pair of soft tip fingers," *Robotica* **20**, 1–11 (2002).
11. J. Fasoulas and M. Sfakiotakis, "Modeling and control for object manipulation by a two DOF robotic hand with soft fingertips," *Robot Control* **10**(1), 259–264 (2012).
12. L. Han and J. C. Trinkle, "The Instantaneous Kinematics of Manipulation," **In**: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* (Lueven, Belgium, 1998) pp. 1944–1949.
13. I. Kao, K. Lynch and J. W. Burdick "Contact Modeling and Manipulation," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer, Berlin, Germany, 2008) pp. 647–670.
14. H. K. Khalil, *Nonlinear Systems* (2nd ed.) (Prentice Hall, Upper Saddle River, NJ, 1996).
15. O. Khatib, "A unified approach for motion and force control of robot manipulators," *IEEE J. Robot. Autom.* **3**, 43–53 (1987).
16. J. Lee, N. Mansard and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Trans. Robot.* **28**(6), 1260–1277 (2012).
17. V. Lippiello, F. Ruggiero, B. Siciliano and L. Villani, "Visual grasp planning for unknown objects using a multi-fingered robotic hand," *IEEE/ASME Trans. Mechatronics* **18**(3), 1050–1059 (2013).
18. V. Lippiello, B. Siciliano and L. Villani, "A grasping force optimization algorithm for multi-arm robots multi-fingered hands," *IEEE Trans. Robot.* **29**(1), 55–67 (2013).
19. V. Lippiello, F. Ruggiero and L. Villani, "Inverse Kinematics for Object Manipulation with Redundant Multi-Fingered Robotic Hands," **In**: *Lecture Notes in Control and Information Sciences*, Vol. 396 (Springer, Heidelberg, Germany, 2009) pp. 255–264.
20. V. Lippiello, F. Ruggiero and L. Villani, "Exploiting Redundancy in Closed-Loop Inverse Kinematics for Dexterous Object Manipulation," **In:** *Proceedings of the International Conference on Advanced Robotics* (Munich, Germany, 2009) pp. 1–6.
21. N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Trans. Robot. Autom.* **23**, 60–72 (2007).
22. C. Melchiorri and M. Kaneco, "Robot Hands," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer, Berlin, Germay, 2008) pp. 345–360.
23. A. T. Miller and P. K. Allen, "GraspIt! – A versatile simulator for robotic grasping," *IEEE Robot. Autom. Mag.* **11**, 110–122 (2004).
24. D. Montana, "The kinematics of contact and grasp," *Int. J. Robot. Res.* **7**(3), 17–32 (1988).
25. D. Montana, "The kinematics of multi-fingered manipulation," *IEEE Trans. Robot. Autom.* **11**, 491–503 (1995).
26. R. M. Murray, Z. X. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, Boca Raton, FL, 1993).
27. K. Nagai and T. Yoshikawa, "Dynamic Manipulation/Grasping Control of Multi-Fingered Robot Hands," **In**: *Proceedings of 1993 IEEE International Conference on Robotics and Automation* (Atlanta, GA, 1993) pp. 1027–1033.
28. K. Nagai and T. Yoshikawa, "Grasping and Manipulation by Arm/Multifingered-Hand Mechanism," **In**: *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (Nagoya, Japan, 1995) pp. 1040–1047.
29. A. M. Okamura, N. Smaby and M. R. Cutkosky, "An Overview of Dexterous Manipulation," **In**: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* (San Francisco, CA, 2000) pp. 255–262.
30. R. Platt, A. H. Fagg and R. Grupen, "Null-space grasp control: Theory and experiments," *IEEE Trans. Robot.* **26**, 282–295 (2010).
31. J. Ponce, S. Sullivan, A. Sudsang, J. Boissonnat and J. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *Int. J. Robot. Res.* **16**, 11–35 (1996).
32. D. Prattichizzo, M. Malvezzi, M. Gabiccini and A. Bicchi, "On the manipulability ellipsoids of underactuated robotic hands with compliance," *Robot. Auton. Syst.* **60**(3), 337–346 (2012).
33. D. Prattichizzo and J. C. Trinkle, "Grasping," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer, Berlin, Germany, 2008) pp. 671–700.

34. C. Remond, V. Perderau and M. Drouin, "A Hierarchical Multi-Fingered Hand Control Structure with Rolling Contact Compensation," **In**: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation* (Washington, DC, 2002) pp. 3731–3736.
35. H. Sadeghian, L. Villani, M. Keshmiri and B. Siciliano, "Dynamic multi-priority control in redundant robotic systems," *Robotica*, doi:10.1017/S0263574713000416 (2013).
36. T. Schlegl, M. Buss, T. Omata and G. Schmidt, "Fast Dexterous Regrasping with Optimal Contract Forces and Contact Sensor-Based Impedance Control," **In**: *Proceedings of 2001 IEEE International Conference on Robotics and Automation* (Seoul, Korea, 2001) pp. 103–108.
37. B. Siciliano and L. Villani, "An adaptive force/position regulator for robot manipulators," *Int. J. Adapt. Control Signal Process.* **7**, 389–403 (1993).
38. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics. Modelling, Planning and Control* (Springer, Berlin, Germany 2009).
39. P. Soueres, S. Tarbouriech and B. Gao, "A Robust Vision-Based Controller for Mobile Robots Navigation: Application to the Task Sequencing Problem," **In**: *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Edmonton, Alberta, Canada, 2005) pp. 2191–2196.
40. S. Stramigioli, C. Melchiorri and S. Andreotti, "A Passivity Based Control Scheme for Robotic Grasping and Manipulation," **In**: *Proceedings of the 38th Conference on Decision and Control* (Phoenix, AZ, 1999) pp. 2951–2956.
41. S. Stramigioli, *Modeling and IPC Control of Interactive Mechanical Systems* (Springer-Verlag, London, 2001).
42. T. Wimboeck, C. Ott and G. Hirzinger, "Passivity-Based Object-Level Impedance Control for a Multifingered Hand," **In**: *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing, China, 2006) pp. 4621–4627.
43. T. Wimboeck, C. Ott and G. Hirzinger, "Analysis and Experimental Evaluation of the Intrinsically Passive Controller (IPC) for Multi-Fingered Hands," **In**: *Proceedings of 2008 IEEE International Conference on Robotics and Automation* (Pasadena, CA, 2008) pp. 278–284.
44. T. Wimboeck, J. Reinecke and M. Chalon, "Derivation and Verification of Sinergy Coordinates for the DLR Hand Arm System," **In**: *Proceedings of 2012 IEEE International Conference on Automation Science and Engineering* (Seoul, Korea, 2012) pp. 454–460.
45. T. Wimboeck, C. Ott, A. Albu-Schaffer and G. Hirzinger, "Comparison of object-level grasp controllers for dynamic dexterous manipulation," *Int. J. Robot. Res.* **31**(1), 3–23 (2012).

Appendix A: Time Derivative of n

The linear velocity of the ith fingertip with respect to the palm frame can be expressed as follows (subscript i will be dropped for simplicity):

$$\dot{\boldsymbol{p}}_f = \boldsymbol{G}_l^{\mathrm{T}} \boldsymbol{v}_{o,p} + \boldsymbol{R}_o \frac{\partial \boldsymbol{c}^o}{\partial \boldsymbol{\xi}} \dot{\boldsymbol{\xi}} + (l - \Delta l) \boldsymbol{R}_o \dot{\hat{\boldsymbol{n}}}^o - \boldsymbol{R}_o \hat{\boldsymbol{n}}^o \Delta \dot{l}, \tag{A1}$$

where $\boldsymbol{G}_l^{\mathrm{T}}$ is the matrix composed of the first three rows of $\boldsymbol{G}^{\mathrm{T}}$, $\boldsymbol{v}_{o,p}$ is the object relative velocity with respect to the palm, and $\dot{\hat{\boldsymbol{n}}}^o(\boldsymbol{c}^o(\boldsymbol{\xi}))$ is given by

$$\dot{\hat{\boldsymbol{n}}}^o = \frac{\partial \hat{\boldsymbol{n}}^o}{\partial \boldsymbol{c}^o} \frac{\partial \boldsymbol{c}^o}{\partial \boldsymbol{\xi}} \dot{\boldsymbol{\xi}}. \tag{A2}$$

Since the time derivative of the unit normal vector belongs to the tangent plane of the object at contact point, by projecting Eq. (A2) in such a plane, a suitable expression for $\dot{\hat{\boldsymbol{n}}}$ can be derived,

$$\dot{\hat{\boldsymbol{n}}} = \boldsymbol{L}_o \boldsymbol{v}_o + \boldsymbol{L}_f \dot{\boldsymbol{p}}_f, \tag{A3}$$

with

$$\boldsymbol{A}_n = \boldsymbol{I} + (l - \Delta l) \frac{\partial \hat{\boldsymbol{n}}}{\partial \boldsymbol{c}}, \tag{A4}$$

$$\boldsymbol{P}_t = \boldsymbol{I} - \hat{\boldsymbol{n}} \hat{\boldsymbol{n}}^T,$$

$$\boldsymbol{L}_f = \frac{\partial \hat{\boldsymbol{n}}}{\partial \boldsymbol{c}} \boldsymbol{A}_n^{-1} \boldsymbol{P}_t, \tag{A5}$$

$$\boldsymbol{L}_o = [\boldsymbol{L}_f, \boldsymbol{S}(\hat{\boldsymbol{n}}) + \boldsymbol{L}_f ((l - \Delta l) \boldsymbol{S}(\hat{\boldsymbol{n}}) - \boldsymbol{S}(\boldsymbol{c}))]. \tag{A6}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

It can be noted that matrix $\boldsymbol{A}_n$ is always a full rank matrix for convex objects bounded by a smooth surface. Moreover, it could be recognized that both $\boldsymbol{L}_o$ and $\boldsymbol{L}_f$ are norm-bounded, i.e.,

$$\|\boldsymbol{L}_o\| \leq k_o, \quad k_o > 0, \tag{A7}$$

$$\|\boldsymbol{L}_f\| \leq k_n, \quad k_n > 0. \tag{A8}$$

Appendix B: System Equilibrium

System (41) under the control law (Eq. (40)) can be described by the closed-loop dynamics,

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} = \boldsymbol{K}_P\,\Delta\boldsymbol{x} - \boldsymbol{K}_D\,\dot{\boldsymbol{x}} + \Delta\boldsymbol{f}_n + k_F\,\Delta\boldsymbol{f}_n + k_I\int_0^t \Delta\boldsymbol{f}_n d\zeta. \tag{B1}$$

At the equilibrium, i.e., $\ddot{\boldsymbol{x}} = \dot{\boldsymbol{x}} = \boldsymbol{0}$, $\boldsymbol{x} = \boldsymbol{x}_\infty$, the following equality holds:

$$\boldsymbol{K}_P(\boldsymbol{x}_d - \boldsymbol{x}_\infty) + (1 + k_F)(\boldsymbol{f}_d - \boldsymbol{f}_{n_\infty}) + k_I\int_0^{+\infty} \Delta\boldsymbol{f}_n d\zeta = \boldsymbol{0}. \tag{B2}$$

Projection of (B2) onto the tangent plane and along the normal unit vector leads to

$$\left(\boldsymbol{I}_6 - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\right)\boldsymbol{K}_P\,(\boldsymbol{x}_d - \boldsymbol{x}_\infty) = \boldsymbol{0}, \tag{B3}$$

$$\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\left(\boldsymbol{K}_P\,(\boldsymbol{x}_d - \boldsymbol{x}_\infty) + (1 + k_F)(\boldsymbol{f}_d - \boldsymbol{f}_{n_\infty}) + k_I\int_0^{+\infty} \Delta\boldsymbol{f}_n d\zeta\right) = \boldsymbol{0}. \tag{B4}$$

By virtue of t*a*he integral action, which gives $\Delta\boldsymbol{f}_n = \boldsymbol{f}_d - \boldsymbol{f}_{n_\infty} = 0$, and (47), it can be seen that $\boldsymbol{n}^{\mathrm{T}}(\boldsymbol{x}_d - \boldsymbol{x}_\infty) = 0$; Eq. (B3) ensures that the tangential part of $\boldsymbol{x}_d - \boldsymbol{x}_\infty$ is null and thus $\boldsymbol{x}_\infty = \boldsymbol{x}_d$. Moreover, from (B4) it can be noted that $\int_0^\infty \Delta\boldsymbol{f}_n d\zeta = \boldsymbol{0}$ as well.

Appendix C: Proof of Inequality (58)

By considering the expression of s in Eq. (49), term s_∞ is given by

$$s_\infty = \int_0^{+\infty}\left(\Delta f_n - \frac{k}{\rho}\dot{\boldsymbol{n}}^{\mathrm{T}}\Delta\boldsymbol{x}\right)d\zeta = -\int_0^{+\infty}\frac{k}{\rho}\dot{\boldsymbol{n}}^{\mathrm{T}}\Delta\boldsymbol{x}d\zeta. \tag{C1}$$

Therefore, I_n becomes

$$I_n = \frac{k}{\rho}\int_t^{+\infty}\Delta\boldsymbol{x}^{\mathrm{T}}\dot{\boldsymbol{n}}d\zeta, \tag{C2}$$

where $\dot{\boldsymbol{n}} = [\dot{\hat{\boldsymbol{n}}}^{\mathrm{T}}\ \ \boldsymbol{0}^{\mathrm{T}}]^{\mathrm{T}}$. From the assumption of quasi-static manipulation, i.e., $\boldsymbol{v}_o = \boldsymbol{0}$, the following equality follows,

$$\dot{\boldsymbol{n}} = \begin{bmatrix}\boldsymbol{L}_f\dot{\boldsymbol{p}}_f \\ \boldsymbol{0}\end{bmatrix}, \tag{C3}$$

through which it can be recognized that

$$\|\dot{\boldsymbol{n}}\| = \|\dot{\hat{\boldsymbol{n}}}\| \leq k_n\|\dot{\boldsymbol{p}}_f\|. \tag{C4}$$

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press

Equation (C4) allows to upper-bound I_n in the domain $\mathcal{D}$ as follows:

$$I_n = \frac{k}{\rho}\int_t^{+\infty} \Delta \boldsymbol{x}^{\mathrm{T}} \dot{\boldsymbol{n}} d\zeta = \frac{k}{\rho}\int_t^{+\infty} \boldsymbol{z}_2^{\mathrm{T}} \dot{\boldsymbol{n}} d\zeta \leq \frac{k}{\rho}\int_t^{+\infty} \|\boldsymbol{z}_2\| \, \boldsymbol{h}_6^{\mathrm{T}} \dot{\boldsymbol{n}} d\zeta$$
$$\leq \frac{k}{\rho}\Phi \int_t^{+\infty} \boldsymbol{h}_3^{\mathrm{T}} \boldsymbol{L}_f \dot{\boldsymbol{p}}_f d\zeta \leq \frac{k}{\rho}\Phi \left\| \boldsymbol{L}_f \right\| \int_t^{+\infty} \boldsymbol{h}_3^{\mathrm{T}} \dot{\boldsymbol{p}}_f d\zeta \leq \frac{k}{\rho} k_n \Phi \|\boldsymbol{z}_2\|, \tag{C5}$$

where $\boldsymbol{h}_\alpha$ is a $(\alpha \times 1)$ vector of ones.

https://doi.org/10.1017/S0263574713000647 Published online by Cambridge University Press