Dexterous Object Manipulation via Integrated Hand-Arm Systems

PhD Thesis
University of Genoa - Informatics Faculty
DIST - Robotics and Automation Lab

Giorgio Panin

February 2002

Contents

List of Figures

Chapter 1

Introduction

1.1 The problem of coordinated control

The problems of control and coordination of actions which may be executed by multi-robot systems (each one, in particular, equipped with anthropomorphic structured *grippers*), and then capable of executing complex cooperative operations like as, for instance, grasping, manipulation, coordinated transportation and assembly, may result of an unmanageable complexity if not being tackled by a suitably structured approach.

From this point of view, a well-structured approach to the problem should hold, in particular, an organic treatment of three of its different main aspects, strictly related with one another; that is: a) the *functional* and *algorithmic* architecture of the whole control and coordination system; b) the Hardware/Software architecture (possibly distributed), which supports the entire functional and algorithmic architecture; and c) the efficiency and quality of the design, the SW development, and the operating system devices used in order to produce and make practical the *real-time* management and the software control.

It is a matter of fact how, in the latest 10 years, the research aimed to these areas of robotics have been endured to an evolution, jointly with the more recent theoretical and technical developments in the field of robotic control; and this, in turn, jointly with the constant evolution of the technology of computing and processing systems. A complete panoramic description, together with a collection of works aimed to the three aspects before mentioned, is up to now represented by the special issue of the *International Journal of Robotics Research* magazine [54], where each one of the three aspects is presented both from the conceptual and the implementative points of view.

However, and quite obviously, even if in the subsequent years further developments have been obtained, and a greater availability of products has been registered (in particular, referring to Operating Systems and Sw development environments for Robotics), the corresponding structured approach to the mentioned problems has been almost unchanged.

Hence, in the context of the present Thesis, a deepening and an implementation of the aforesaid structured approach to the coordinated control will be proposed, by referring in particular to an integrated *hand-arm* system, which has been designed and developed at the Robotics and Automation Lab of the University of Genoa (GRAAL). In the following, an introduction and a control approach for the specific problem of manipulation via *multi-fingered* robotic devices will be given, by paying attention particularly to *tendons-actuated* systems; moreover, the overall hand-arm architecture will be presented, and the corresponding control approach will be devised, in the context of a functional and algorithmic *multi-level* (and *multi-tasking*) architecture, which will be later described.

1.2 Multi-finger manipulation

Manipulating objects by means of multi-robot systems, or articulated robotic hands, can be actually considered as one of the most representative and complete tasks, among all those generally considered in robotics.

Indeed, the employment of insudtrial robots for performing actions wich are elementary for man, like as grasping and moving small dimension objects, can be extremely difficult. Standard manipulators hold a specific *gripper* for each kind of manipulated object, and this, even if it can optimize performances, on the other hand restricts the application fields.

Instead, an anthropomorphic robotic hand has some features, like as small dimensions and reliability, which are fundamental requests for non-conventional applications: in fact, being equipped with *fingers*, it is able both to grasp and precisely position a wide class of objects. The desire that multifinger manipulators can execute tasks, both in autonomous and teleoperated modality, left a lot of development possibilities for researches in this field.

In most of the anthropomorphic manipulators, the actuators are not situated in the proximity of the joints; the reason of this design choice is due to the fact that robotic hands have a great number of degrees of freedom (*dof*) situated into a very small space, and this fact prevents the assembling of the motors (actuators) near the joints of the device. Hence, the employment of a flexible system for the transmission of the actuating forces is almost necessary.

Most of the devised systems have fluid-actuators which, like as the for the human hand muscles, are located near the joints, and they are remotely driven. Their implementation on *multilink* manipulators has been limited by their dimension; force transmission systems (like as, for instance, tapes, hollows, chains, ropes and belts) are capable of remotely actuate manipulators, but they need high-volume energy transformation systems.

In the case of multifingered robotic hands, a great number of joints is located, as said before, in a very small space; it turns out to be evident that there is not enough space for the actuators placing. So, the use of some kind of *flexible* and *small sized* transmission system is necessary, as exactly can be the tendons.

The tendons are mono-directional actuators intrinsically nonlinear; as a consequence, every analysis made on tendon-driven systems has the property of being more complex, if compared with common mechanisms using bi-directional actuators. With these data, the transmission using tendons is often difficult and requires pulleys and other routing transmission devices with low friction, in order to allow the cables passing through the joints and rotating around their axes, and moreover in order to allow possibly the insertion of other structures (for instance, position and force sensors, etc.).

The tendon system design is often an art, because they must be realized in a way such as they can suffer a high stress, have a long lifetime, and be resistant to long use and elongation. Finally, the working space of tendon-driven devices must be smaller than that commonly used for linear and rotational actuators.

Despite of those considerations, there are several advantages coming from the utilization of tendons, and in some cases there are no alternative ways. As yet said, most of actuators, like as dc motors, are relatively cumbersome; by de-locating these actuators, hence, the kinematic and structural properties of the manipulators can be improved.

It has been demonstrated that, when properly used, this kind of pre-tensioned cables have a high stiffness, and this fact, jointly with their small masses, low friction, and high geometric flexibility, makes them proper for several applications.

There exist several systems working with tendons, and most of them are similar to the human hand. Mechanical hands, like the *Utah-M.I.T. dextrous hand* [22] use *antagonist pairs* of tendons, that is, every joint is controlled by means of two opposite acting tendons, which are independent from the other joints. Later on, it has been introduced a control system such that every pair of tendons behaves like a single bi-directional actuator, then simplifying the kinematics.

By using a manipulator equipped with "fingers" it is possible both grasping (stable prehension problem) and orientating and precisely moving (manipulation problem) a wide class of objects. Just for this reason, in the last years the robotics research devoted very much attention to the study of the *grasping* problem, together with the design and development of several robot hands all over the world, making use of the most different mechanisms.

The *human grasping* study has been for a long time an interesting area for surgery, for prostheses design, and for quantifying the degree of invalidities due to defections and accidents. The results can be found in the medical literature about the prehension abilities of the human hand.

Most of the literature considers six different types of prehension, as defined by Schlesinger (1919) and later reviewed by Taylor and Schwarz (1955): cylindrical, fingertips contact, hook, palmar, spherical and lateral. This characterization induces a natural association between each grasp and each shape of the object: so, for instance, the sphere induces a spherical grasp, etc.

The experience shows that the grasps can be correctly and uniquely determined from the specification of the object features (e.g., dimension, shape) and the attributes of the grasp itself, like as dexterity, precision, sensibility, stability and ro-

bustness. But there are not fixed rules, and some factors regarding the geometric description of the object and the attributes of the grasp become more clear during the development of the system.

Manipulation is a complex task, typically comprising opening and closing of kinematic chains, nonholonomic constraints, redundancy and kinematic singularities. In the context of rigid objects manipulation by multrobot systems, or articulated robotic hands, variuos inter-related problems arise, like as, for example: the statics and kinematics of grasping, including contact problematics like as *rolling contact* [46], [24], [39]; grasp planning and allocation of the contact forces [24], [33], [3], still including rolling contact situations [34], [35]; or even the so-called *finger gaiting* (real-time reconfiguration of grasping) on the manipulated object [16], [21]; moreover, the stability of grasping and related structural problems [34], [35], [41].

The first analysts (Hanafusa and Asada (1977)) made some assumptions, and most of them are still used for multifinger manipulation study (and which will be used for our control approach) like as:

- rigid body models with point-like contact between fingertip points and the grasped object;
- no sliding or rolling of the fingertips on the object;
- assumption of the full knowledge of the object (shape, mass, friction, etc.) and the contact points, neglecting in a first time the use of force sensors during manipulation;

Obviously, the presence of tactile sensors would improve the cognitive possibilities of the system, while introducing further problematics.

More recent studies, like as those of Cutkosky and Wright (1986); Nakamura, Nagai and Yoshikawa (1987); Ji (1987); Li and Sastry (1988) have developed models which include some useful simplifications:

- idealized models of friction, which ignore the effects of the sliding velocities and of the properties of the material which constitute the robot ("skin");
- simplified dynamics of the actuators an of the tendon sheats, often by ignoring their elasticity (or by approximating it with springs of known elastic coefficient) and their friction
- simplified representation of object shape

1.3 General organization of the present work

Here it can be given a first quick description of what will be exposed in the present dissertation, neglecting in a first time the exposition details.

In Chapter 2, firstly the general theoretical approach to the control for multifinger manipulation will be described, by delineating the recent theoretical developments

here obtained, and giving the basic tools for the devisement and implementation of some simulated control schemes; the subsequent simulated results will be given next, at the end of the Chapter.

In Chapter 3, the proposed functional structure for the global hand-arm control system will be outlined, and the hardware structure of the available setup for the subsequent experimental tests will be described.

Chapter 4 contains all the algorithmic developments relevant for the control of the integrated hand-arm system, which will be devised and developed by following the functional structure previously derived; in particular, a *top-down* approach will be followed, starting from the most abstract control level, till arriving to the lowest level, "nearest" to the physical actuating and sensing system. Finally, in Chapter 5, the experimental results relevant to the DIST-Hand setup, and the simulation results relevant to the integrated hand-arm system will be given; next, some conclusive notes will follow, at the end of the Chapter.

Chapter 2

General control approach for multifingered manipulation

2.1 Introduction

In the framework of dynamics and control of multifingered manipulation, all the problematics mentioned in the previous Chapter should be taken into account, and consequently treated, to the control purposes, within a unifying approach.

To this respect, early attempts aiming to establish such a unifying approach date back to the well known work of Khatib [25] on operational space methods, then followed by the works [29], [36], [10], whose results have been successively reorganized within the book [37].

In particular, within [37], the problem of dynamically controlling a multifingered object manipulation is approached by first devising, in the operational space (i.e. in terms of a local parametrization of position and orientation of the object) a composite dynamic model strictly embedding both the object and fingers dynamics; then, the object motion is in turn controlled via computed torque techniques applied to the so devised overall composite system model. Naturally enough, a complimentary grasp force control law is also devised for guaranteeing the contact between the object and the finger tips, whithout affecting the overall system motion. Moreover, still in [37], an (off line) symbolic computation structure is also suggested for automatically devising the required composite system dynamic model.

As a matter of fact, however, since the required composite models generally result to be very complex ones, a hierarchical way of structuring the overall control systems is finally suggested for dealing with such complexities. More precisely, the underlying idea simply consists in providing each one of the robot fingers with a local (and consequently simpler) computed torque controller, compensating for the corresponding finger dynamics; thus leading to a much more simple and substantially straightforward (off-line) evaluation of the resulting composite dynamic model (as seen from a superior hierarchical level) and to a consequent very simplified structure for the upper level overall computer torque controller.

Unfortunately enough, however, if the local controllers can only partially compensate for the finger dynamics, the residual dynamic attributes must be however transferred to the superior level; thus not totally reducing the effort needed for (off line) evaluating the composite model and sucessively structuring the corresponding overall system controller.

In the present control approach, given as in [8], the problem of structuring a hierarchical controller for multifingered manipulation is instead approached in a substantially different way (and, in some sense, reverse) with respect to the one proposed in [37].

More precisely, within the here proposed approach, the motion *tracking* to be performed by the manipulated object is first considered in terms of the definition of a closed loop (kinematic only) upper level control law, accomplishing the task of generating, at each time instant, the reference velocity (linear and angular) that should be assigned to the fixed body frame, in case that the body itself could be considered as a pure *kinematic* entity (i.e. no mass, no inertia).

Then, via simple rigid body transformations, such reference velocity for the body frame is in turn real time translated into the corresponding set of reference linear velocities at the body contact points; thus coinciding with the reference linear velocities to be assigned to the corresponding *fingertips* (obviously assuming the persistency of the contact conditions). Such fingertips reference velocities are in turn real-time traslated into a related set of joint reference *velocities* for the robotic fingers, via the use of any right inverse of the *linear* part of the Jacobian matrix of the overall set of fingers.

Finally, such real time evaluated joint reference velocities are given as input to a suitable, lower level, closed loop *joint velocity* controller, accomplishing the task of guaranteeing their asymptotic tracking.

As it will be clarified in the paper, the lower level joint velocity control law will keep into account for both the object and fingers dynsmics; but in a way such that the various dynamics however remain separated, one from another; thus avoiding the need for any *off-line* symbolic manipulation devoted to the evaluation of composite models.

Naturally enough, to the aim of maintaining the contacts during motion, a complimentary force control law is also provided, not affecting the motion controlled by the above mentioned upper and lower levels control *loops*.

Hence, the present Chapter is organized as follows: in Section 2 the overall dynamic equations of the system will be derived, taking on the form of a set of DAE (*Differential-Algebraic Equations*), thus maintaining the separation between fingers and object dynamics. In Section 3, the proposed hierarchical control law for trajectories tracking will be obtained, by using Lyapunov functions, and the convergence toward zero of the errors will be demonstrated, together with the stability properties of control. Next, in Section 4 a simple complimentary force control law will be devised, in order to maintain the grasping contact conditions, while not affecting the overall system motion, based on some recent results regarding grasping forces decomposition, which are given in [3]. Finally, in Section 5, some simulative results,

preliminarly to the algorithmic developments which will be derived in the subsequent Chapters.

The here presented developments actually represent an extension (to the case of multirobot object manipulation) of similar results already presented in [1], [2], for the case of robot free motions. Since based on the assumption of an exact knowledge of the involved dynamic models, they represent the first necessary step to be done before passing to investigate their robustness and stability properties, in case of the presence of uncertainties (as it has been already done in [1], [2], for the free motion cases).

2.2 Overall system equations

The system considered here consists in a tree-structured robotic system composed by a number h of rigid end-effector tips, which all together handle a rigid body (see fig. 2.1).

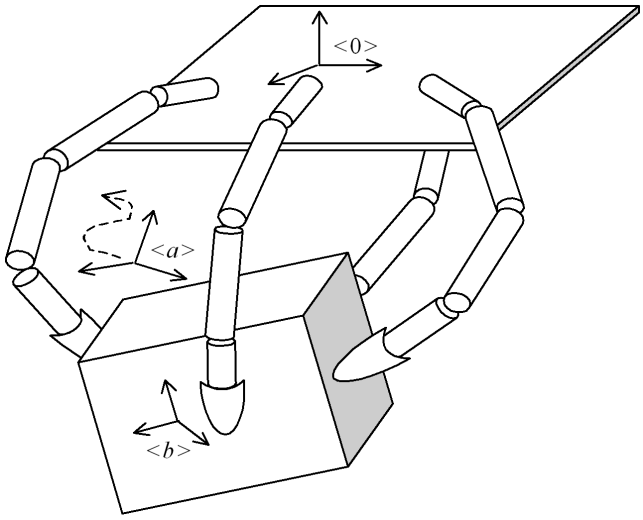

Figure 2.1: System overview (frame $< a >$ is the *goal*, that has to be followed by the object frame $< b >$).

The contact model considered is the ideal *point contact with friction* (see [45]); in other words, it is assumed that the h contact points remain fixed on the object surface (and, off course, that the h end-effector tips are point-like and they do not slip during the manipulation task).

Within this system, first consider the subsystem represented by manipulated object and its assigned fixed frame $< b >$, whose orientation and position with respect to a given (inertial) base frame $< 0 >$ are, at each time instant, respectively given by the corresponding orthogonal matrix R and distance vector $\mathbf{x}$ (the latter projected

on $< 0 >$), generally grouped within the homogenous transformation matrix

$$T \triangleq \left[\begin{array}{c|c} R & \mathbf{x} \\ \hline \mathbf{0} & 1 \end{array}\right] \tag{2.1}$$

By letting $\mathbf{v}$ and ω be the projections (on frame $< 0 >$) of the linear and angular velocities of $< b >$ with respect to $< 0 >$, respectively, we can first of all recall the *kinematic* equations of such a rigid body; i.e., the well known equations

$$\dot{\mathbf{x}} = \mathbf{v} \tag{2.2}$$

$$\dot{R} = [\omega\wedge]R \tag{2.3}$$

the second one representing the so-called "*Strapdown Differential Equation*", which relates the time derivative of R with vector ω via the skew-symmetric matrix $[\omega\wedge]$ (i.e. the cross-product operator matrix).

Note that the differential representation (2.3) has been here preferred (instead of any of the most commonly used 3-D parametrizations like, for instance, Euler angles, etc.), in order to avoid any one of the existing *representation singularities,* that could otherwise occur.

Also note that throughout the following, and with a little abuse of notation, we shall always term the vector collection of the body velocities $\mathbf{v}, \omega$ with the compact symbol

$$\dot{\mathbf{q}}_b \triangleq col(\mathbf{v}, \omega) \tag{2.4}$$

where the notational abuse stands, obviously, from the well known fact that, generally speaking, the angular velocity ω *is not* the time derivative of any pre-existing rotation vector (thus implying that the adopted notation (2.4) should be considered as an *intrinsic* one, i.e., not related to any true time derivative operation).

As a second aspect of the object kinematics, it can be also reminded that, owing to its rigidity and the assumed invariance of the h contact points on it, a direct *linear* relationship actually exists between the velocities of such contact points and the body velocities.

More precisely, by letting $\mathbf{X} \triangleq col(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_h)$ be the vector collection of the h contact points on the body, each one projected on $< 0 >$, it is actually an easy task to verify that

$$\dot{\mathbf{X}} = J_b(R)\dot{\mathbf{q}}_b \tag{2.5}$$

being $J_b(R)$ the so-called Jacobian matrix of the body at the contact points, having the form

$$J_b(R) = \left[\begin{array}{c|c} I_3 & -[\mathbf{s}_1\wedge] \\ \hline \multicolumn{2}{c}{\cdots} \\ \multicolumn{2}{c}{\cdots} \\ \hline I_3 & -[\mathbf{s}_h\wedge] \end{array}\right] \tag{2.6}$$

with submatrices $[\mathbf{s}_i\wedge]$ representing the (skew-symmetric) cross product operator for the difference vectors $\mathbf{s}_i \triangleq (\mathbf{x}_i - \mathbf{x})$, $i = 1, ..., h$, and where the dependence on

R simply stands from the fact that each vector $\mathbf{s}_i$ results projected on the inertial frame $< 0 >$.

In particular, we can make such dependence on R explicit, by simply expressing each $\mathbf{s}_i$ in terms of its *constant* projection σ_i on the object frame $< b >$ (that is, $\mathbf{s}_i = R\sigma_i$, and consequently $[\mathbf{s}_i\wedge] = R\,[\sigma_i\wedge]\,R^T$), thus obtaining

$$J_b(R) = R^{(h)} G R^{(2)T} \tag{2.7}$$

being $R^{(k)}$ the $k-th$ order block diagonal organization of matrix R, and G the now *constant* Jacobian matrix of the body, projected on the body frame $< b >$, and built on the basis of constant vectors $\sigma_i, i = 1, ..., h$, with a structure identical to that of (2.6).

Passing now to consider the *dynamic* aspects of the manipulated rigid object, let us first define $\mathbf{F} \triangleq col(\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_h)$ as the vector collection of the h contact forces (each one projected on $< 0 >$), acting on the object at the corresponding contact points. Moreover, consider its relevant wrench reduction to the origin $\mathbf{x}$ of body frame $< b >$ (i.e., its equivalent representation in terms of the resultant force $\mathbf{f}$ applied to point $\mathbf{x}$, and resulting torque τ evaluated with respect to $\mathbf{x}$, both projected on $< 0 >$), given by the dual relationship

$$\mathbf{p} \triangleq col[\mathbf{f}, \tau] = J_b^T(R)\mathbf{F} \tag{2.8}$$

(note that, by expressing the above quantities on frame $< b >$, we shall use the matrix G^T, which is the well-known *Grasp Matrix* for the grasp configuration $\sigma_i, i = 1, ..., h$ [45]).

Then, by recalling the well known general form assumed by the *body dynamic* equations, whenever written with respect to any given point $\mathbf{x}$ on it (i.e. not necessarily coinciding with its mass center), projected on the inertial frame $< 0 >$, and by keeping into account relationship (2.8), we can consequently write such dynamic equations directly in the form

$$A_b(R)\ddot{\mathbf{q}}_b + B_b(R, \dot{\mathbf{q}}_b)\dot{\mathbf{q}}_b + C_b(R) = J_b^T(R)\mathbf{F} \tag{2.9}$$

where matrix $A_b(R)$ (positive definite) represents the so called *generalized inertia* matrix of the body, evaluated at point $\mathbf{x}$ on it, and where vectors $B_b(R, \dot{\mathbf{q}}_b)\dot{\mathbf{q}}_b$ and $C_b(R)$ keep into account the centrifugal-Coriolis and gravitational effects, respectively (being such effects also evaluated at point $\mathbf{x}$ on the body.

At this point, also consider the subsystem represented by the robotic manipulating structure. Then, by denoting with $\mathbf{q}_r$ the vector of its joint coordinate, and letting $\mathbf{Y} \triangleq col(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_h)$ be the vector collection of its h end effector tip points (each one projected on $< 0 >$) we can primarily recall their general *kinematic* behaviour, expressed by the relationship

$$\dot{\mathbf{Y}} = J_r(\mathbf{q}_r)\dot{\mathbf{q}}_r \tag{2.10}$$

being $J_r(\mathbf{q}_r) \triangleq col[J_1(\mathbf{q}_1), J_2(\mathbf{q}_2), ..., J_h(\mathbf{q}_h)]$ the column collection of the h linear Jacobian matrices (one for each end-effector tip) of the robotic fingers.

Moreover, by keeping into account the assumed contact constraint conditions between the body and the fingertips, i.e. the equality constraint

$$\mathbf{X} = \mathbf{Y} \tag{2.11}$$

by differentiating it with respect to time (and keeping into account (2.5) and (2.10)) it follows that

$$\dot{\mathbf{X}} = \dot{\mathbf{Y}} \Longleftrightarrow J_b(R)\dot{\mathbf{q}}_b = J_r(\mathbf{q}_r)\dot{\mathbf{q}}_r \tag{2.12}$$

which makes explicit the *linear* constraint conditions existing between the body velocities and the joint velocities of the robotic structure.

Finally, by also recalling that the set of the h counteracting contact forces $-\mathbf{f}_1, ..., -\mathbf{f}_h$ (each one acting on the corresponding end-effector tip) globally reflect on the manipulating structure as an equivalent set of additional *joint* torques η, expressed by the dual relationship

$$\eta = -J_r^T(\mathbf{q}_r)\mathbf{F} \tag{2.13}$$

we can consequently write down the *dynamic* equations of the manipulating structure itself, directly in the form

$$A_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + B_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)\dot{\mathbf{q}}_r + C_r(\mathbf{q}_r) = \mathbf{m} - J_r^T(\mathbf{q}_r)\mathbf{F} \tag{2.14}$$

where matrix $A_r(\mathbf{q}_r)$ (positive definite) represents the inertia matrix of the whole multifinger structure, and where vectors $B_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)\dot{\mathbf{q}}_r$,$C_r(\mathbf{q}_r)$ keep into account for the corresponding centrifugal-Coriolis and gravitational effects, respectively. Moreover, vector $\mathbf{m}$ represents the set of *external input* joint torques, which can be applied to the robotic structure for control purposes.

The set of equations (2.9) (body dynamics), (2.14) (robot dynamics) and also (2.3) (for representing R, since not given by a pure integration process) completely defines the dynamics of the *entire*, contact constrained, system.

Due to the presence of the algebraic constraints (2.11), such equations take the form of a set of *DAE* (Differential-Algebraic Equations), which obviously maintain their validity within the fulfilment of the so-called "static friction conditions" for each one of the contacting forces $\mathbf{f}_i, i = 1, ...h$ (i.e., each $\mathbf{f}_i$ acting toward the body and located inside the corresponding *friction cone*).

Throughout the sequel, we shall very often refer to the above set of the DAE's (2.9), (2.14) and (2.11) by using the following, more compact representation (arguments avoided for ease of notations)

$$\begin{cases} A\ddot{\mathbf{q}} + B\dot{\mathbf{q}} + C = \mu + J^T\mathbf{F} \\ \\ \mathbf{X} = \mathbf{Y} \end{cases} \tag{2.15}$$

where we have posed

$$\begin{cases} \dot{\mathbf{q}} \triangleq col(\dot{\mathbf{q}}_b, \dot{\mathbf{q}}_r) \\ \mu \triangleq col(\mathbf{0}, \mathbf{m}) \end{cases} \tag{2.16}$$

and consequently

$$\begin{cases} A \triangleq blockdiag(A_b, A_r) \\ B \triangleq blockdiag(B_b, B_r) \\ C \triangleq col(C_b, C_r) \\ J^T = col(J_b^T, -J_r^T) \end{cases} \tag{2.17}$$

Finally observe that, as a consequence of the above definitions, also the previously noted existing velocity constraint (2.12) (second form) can be more concisely rewritten as

$$J\dot{\mathbf{q}} = \mathbf{0} \tag{2.18}$$

thus showing that the set of admissible overall velocities $\dot{\mathbf{q}}$ always belong to the null space of the above defined overall Jacobian matrix J.

2.3 Definition of the two-levels hierarchical trajectory tracking control law

Within the considered overall system, first of all suppose the existence of at least *three distinct* contact points among the h assumed ones (thus $h \geq 3$) which are distributed on the body surface as a consequence of the grasping process.

Then, consider the fixed body frame $< b >$ and consequently note how, under the above conditions, its actual position $\mathbf{x}$ and attitude R can always be algebraically reconstructed via the knowledge (from the actual posture of the robotic structure) of both contact points absolute positions $\mathbf{x}_1, ..., \mathbf{x}_h$ and constant vectors $\sigma_1, ..., \sigma_h$ on the body, with simple geometric considerations.

While keeping in mind the above considerations, let us now introduce a "goal frame" $< a >$, characterized by a known absolute time evolution

$$T_a(t) \triangleq \left[\begin{array}{c|c} R_a(t) & \mathbf{x}_a(t) \\ \hline \mathbf{0} & 1 \end{array}\right] \tag{2.19}$$

and, consequently, consider the problem of ensuring that the solidal object frame $< b >$ asymptotically follows the given frame $< a >$ (Fig. 2.1).

Since $\mathbf{x}_a, R_a$ in (2.19) are, at each time instant, known quantities, as well as $\mathbf{x}$ and R in (2.1) (due to the above introduced assumption and related considerations), it then primarily follows that a couple of *directly measurable* error vectors representing the mismatch between frames $< b >$ and $< a >$ actually exist (to be possibly considered for control purposes), which are respectively defined as

$$\mathbf{d} \triangleq \mathbf{x} - \mathbf{x}_a, \text{ position error} \tag{2.20}$$

$$\rho \triangleq \mathbf{r}\theta, \text{ rotation error} \tag{2.21}$$

with the second one (projected on frame $< 0 >$) representing the so called "eigenaxis vector" of frame $< b >$ with respect to $< a >$. This is the vector whose unit versor $\mathbf{r}$ and relevant positive component $\theta \in [-\pi, \pi]$ respectively specify, at each time

instant, the axis and the corresponding angle of rotation which are ideally needed for transporting $< b >$ to the actual attitude $R(R_a)^T$, with respect to $< a >$, starting from a condition where $< b >$ itself is thought to be parallel to frame $< a >$.

As it is well known, rotation error vector ρ can always be evaluated via the use of the following formulas (*Versors Lemma*)

$$\begin{cases} [\mathbf{i}_a\wedge]\mathbf{i}_b + [\mathbf{j}_a\wedge]\mathbf{j}_b + [\mathbf{k}_a\wedge]\mathbf{k}_b = 2\mathbf{r}\sin\theta; \\ \\ (\mathbf{i}_a^T\mathbf{i}_b) + (\mathbf{j}_a^T\mathbf{j}_b) + (\mathbf{k}_a^T\mathbf{k}_b) = 1 + 2\cos\theta \end{cases} \tag{2.22}$$

being $(\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a)$ and $(\mathbf{i}_b, \mathbf{j}_b, \mathbf{k}_b)$ the columns of rotation matrices R_a and R, respectively.

Since the considered tracking problem is equivalent to that of making both error vectors $\mathbf{d}$ and ρ eventually approaching zero for increasing time, then, in order to investigate about such possibility (if any), we may start by first considering the following positive definite scalar quantity

$$V = \frac{1}{2}(\mathbf{d}^T\mathbf{d} + \rho^T\rho) \triangleq \frac{1}{2}\mathbf{e}^T\mathbf{e} = \frac{1}{2}\left\|\mathbf{e}\right\|^2 \tag{2.23}$$

as a possible candidate Lyapunov function, measuring (one half of) the squared norm of the body global error vector:

$$\mathbf{e} \triangleq col(\mathbf{d}, \rho) \tag{2.24}$$

Then, by differentiating V with respect to time, while keeping into account that

$$\dot{\rho} = \dot{\mathbf{r}}\theta + \mathbf{r}\dot{\theta} \tag{2.25}$$

with $\dot{\mathbf{r}}$ obviously orthogonal to ρ, and with $\dot{\theta}$ given (after some algebra) by the expression

$$\dot{\theta} = \mathbf{r}^T(\omega - \omega_a) \tag{2.26}$$

being ω_a the angular velocity of the target frame $< a >$, we directly get

$$\dot{V} = \mathbf{d}^T(\dot{\mathbf{x}} - \dot{\mathbf{x}}_a) + \rho^T(\omega - \omega_a) = \mathbf{e}^T(\dot{\mathbf{q}}_b - \dot{\mathbf{q}}_a) \tag{2.27}$$

where we have obviously posed

$$\dot{\mathbf{q}}_a \triangleq col(\dot{\mathbf{x}}_a, \omega_a) \tag{2.28}$$

At this point, by assuming *also* the knowledge of the velocity reference vector $\dot{\mathbf{q}}_a$, and provided that we could directly assign a body velocity vector $\dot{\mathbf{q}}_b$of the form

$$\dot{\mathbf{q}}_b = \dot{\bar{\mathbf{q}}}_b \triangleq -\Pi\mathbf{e} + \dot{\mathbf{q}}_a; \Pi > 0 \tag{2.29}$$

which would, in turn, require a joint velocity vector $\dot{\mathbf{q}}_r$ of the form (consider the constraint (2.12), for $\dot{\mathbf{q}}_b$=$\dot{\bar{\mathbf{q}}}_b$, assume J_r full row rank, and then invert with respect to $\dot{\mathbf{q}}_r$)

$$\dot{\mathbf{q}}_r = \dot{\bar{\mathbf{q}}}_r \triangleq J_r^{\#}J_b\dot{\bar{\mathbf{q}}}_b = J_r^{\#}J_b(-\Pi\mathbf{e} + \dot{\mathbf{q}}_a) \tag{2.30}$$

being $J_r^{\#}$ any right inverse of Jacobian matrix J_r (e.g., in this case, the Moore-Penrose *pseudoinverse*); then, by substituting (2.29) into (2.27), it is readily seen that we could consequently make

$$\dot{V} = -\mathbf{e}^T \Pi \mathbf{e} \leq -\underline{\pi} \left\| \mathbf{e} \right\|^2 < 0 \tag{2.31}$$

being $\underline{\pi}$ the minimum eigenvalue of the positive definite *gain* matrix Π.

From (2.23), (2.31), the asymptotic convergence of the error vector $\mathbf{e}$ toward zero would then follow directly, with a convergence rate no slower than $\exp(-\underline{\pi}t)$.

In practice, however, since we cannot generally ensure the exact fulfilment of requirement (2.30) for all the time instants (this is due, at least, to the overall system dynamics), we are naturally led to consider $\dot{\overline{\mathbf{q}}}_r$, as given by (2.30) (and consequently $\dot{\overline{\mathbf{q}}}_b$, as given by (2.29)) as nothing more than appropriate reference signals that should be tracked "at best" by the actual robot joint velocities; being however aware of the fact that, generally, we shall have

$$\dot{\mathbf{q}}_r = \dot{\overline{\mathbf{q}}}_r + \delta\dot{\overline{\mathbf{q}}}_r = J_r^{\#} J_b(-\Pi\mathbf{e} + \dot{\mathbf{q}}_a) + \delta\dot{\overline{\mathbf{q}}}_r \tag{2.32}$$

which, in turn, will imply (substitute into constraint (2.12) while keeping (2.30) into account, and then invert with respect to $\dot{\mathbf{q}}_b$)

$$\dot{\mathbf{q}}_b = (-\Pi\mathbf{e} + \dot{\mathbf{q}}_a) + J_b^{+} J_r \delta\dot{\overline{\mathbf{q}}}_r \triangleq \dot{\overline{\mathbf{q}}}_b + \delta\dot{\overline{\mathbf{q}}}_b \tag{2.33}$$

with J_b^{+} the (now unique) *left* inverse matrix of the body Jacobian J_b, and the error term $\delta\,\dot{\overline{\mathbf{q}}}_b$ obviously defined.

This will, in turn, lead to a time derivative $\dot{V}$ taking on the perturbed form (no more guaranteed to be unconditionally negative definite)

$$\dot{V} = -\mathbf{e}^T \Pi \mathbf{e} + \mathbf{e}^T J_b^{+} J_r \delta\dot{\overline{\mathbf{q}}}_r \leq -\underline{\pi} \left\| \mathbf{e} \right\|^2 + (\overline{\sigma}_r / \underline{\sigma}_b) \left\| \mathbf{e} \right\| \left\| \delta\dot{\overline{\mathbf{q}}}_r \right\| \tag{2.34}$$

being $\underline{\sigma}_b$ the (constant) minimum singular value of matrix J_b, and $\overline{\sigma}_r$ the maximum singular value of matrix J_r, this one evaluated among all possible postures.

As an additional comment to the above considerations, we can also observe that, by substituting the above representations (2.32), (2.33) (it is sufficient to consider only the first and second right hand sides, respectively), within the velocity constraints (2.12), we get te conditions

$$J_b\dot{\overline{\mathbf{q}}}_b = J_r\dot{\overline{\mathbf{q}}}_r;\;\; J_b\delta\dot{\overline{\mathbf{q}}}_b = J_r\delta\dot{\overline{\mathbf{q}}}_b \tag{2.35}$$

thus bringing into evidence that the same constraints (2.12) also holds *separately* for both reference signals $\dot{\overline{\mathbf{q}}}_b$, $\dot{\overline{\mathbf{q}}}_r$ and the corresponding tracking errors $\delta\,\dot{\overline{\mathbf{q}}}_b, \delta\,\dot{\overline{\mathbf{q}}}_r$.

As a consequence, by simply letting

$$\dot{\overline{\mathbf{q}}} \triangleq col(\dot{\overline{\mathbf{q}}}_b, \dot{\overline{\mathbf{q}}}_r);\;\; \delta\dot{\overline{\mathbf{q}}} \triangleq col(\delta\dot{\overline{\mathbf{q}}}_b, \delta\dot{\overline{\mathbf{q}}}_r) \;\Rightarrow\; \dot{\mathbf{q}} = \dot{\overline{\mathbf{q}}} + \delta\dot{\overline{\mathbf{q}}} \tag{2.36}$$

and recalling the definition (2.17), we can trivially rewrite the above additional conditions (2.35) in the same concise form as in (2.18); that is

$$J\dot{\overline{\mathbf{q}}} = \mathbf{0}; J\delta\dot{\overline{\mathbf{q}}} = \mathbf{0} \tag{2.37}$$

thus showing that also $\dot{\overline{\mathbf{q}}}$ and $\delta\,\dot{\overline{\mathbf{q}}}$ separately belong to the null space of the overall Jacobian matrix J.

At this point, in order to face with the additional problem of tracking the joint velocity reference signal $\dot{\overline{\mathbf{q}}}_r$, let us now introduce the quadratic form

$$U = \frac{1}{2}(\dot{\mathbf{q}} - \dot{\overline{\mathbf{q}}})^T A(\dot{\mathbf{q}} - \dot{\overline{\mathbf{q}}}) = \frac{1}{2}\delta\dot{\overline{\mathbf{q}}}^T A \delta\dot{\overline{\mathbf{q}}} \tag{2.38}$$

which results in a positive definite candidate Lyapunov function of the sole vector $\delta\,\dot{\overline{\mathbf{q}}}_r$, once we keep into account that, due to (2.35) and definition (2.17), we have

$$\begin{aligned} U &= \frac{1}{2}\delta\dot{\overline{\mathbf{q}}}_r^T [A_r + (J_b^+ J_r)^T A_b (J_b^+ J_r)]\delta\dot{\overline{\mathbf{q}}}_r \triangleq \\ &\triangleq \frac{1}{2}\delta\dot{\overline{\mathbf{q}}}_r^T \Sigma \delta\dot{\overline{\mathbf{q}}}_r \end{aligned} \tag{2.39}$$

where, being Σ positive definite, it is $U > 0$.

The existing analytical advantages in choosing a candidate Lyapunov function of the above form (written as in (2.38)) instead of a simpler one, will readily appear in the following.

In fact, by differentiating (2.38) with respect to time along (2.15), after some simple algebra we primarily get

$$\begin{aligned} &\dot{U} = \delta\dot{\overline{\mathbf{q}}}^T A(\ddot{\mathbf{q}} - \ddot{\overline{\mathbf{q}}}) + \tfrac{1}{2}\delta\dot{\overline{\mathbf{q}}}^T \dot{A}\delta\dot{\overline{\mathbf{q}}} = \\ &= \delta\dot{\overline{\mathbf{q}}}^T(\mu - A\ddot{\overline{\mathbf{q}}} - B\dot{\mathbf{q}} - C + J^T\mathbf{F}) + \tfrac{1}{2}\delta\dot{\overline{\mathbf{q}}}^T \dot{A}\delta\dot{\overline{\mathbf{q}}} = \\ &= \delta\dot{\overline{\mathbf{q}}}^T(\mu - A\ddot{\overline{\mathbf{q}}} - B\dot{\overline{\mathbf{q}}} - C) + \delta\dot{\overline{\mathbf{q}}}^T J^T\mathbf{F} - \delta\dot{\overline{\mathbf{q}}}^T B\delta\dot{\overline{\mathbf{q}}} + \tfrac{1}{2}\delta\dot{\overline{\mathbf{q}}}^T \dot{A}\delta\dot{\overline{\mathbf{q}}} \end{aligned} \tag{2.40}$$

Then, by exploiting the well known property of matrices $B, \dot{A}$ (i.e. $\mathbf{z}^T B\mathbf{z} = 1/2(\mathbf{z}^T\ \dot{A}\ \mathbf{z})$) and also noting that the second term in the right hand side of (2.40) is actually zero (due to the second of (2.37)) we can consequently simplify (2.40) as follows

$$\dot{U} = \delta\dot{\overline{\mathbf{q}}}^T(\mu - A\ddot{\overline{\mathbf{q}}} - B\dot{\overline{\mathbf{q}}} - C) \tag{2.41}$$

thus yelding (by making explicit the dependence on $\delta\,\dot{\overline{\mathbf{q}}}_r$ while recalling definitions (2.17).

$$\dot{U} = \delta\dot{\overline{\mathbf{q}}}_r^T[\mathbf{m} - (A_r\ddot{\overline{\mathbf{q}}}_r + B_r\dot{\overline{\mathbf{q}}}_r + C_r) - (J_b^+ J_r)^T(A_b\ddot{\overline{\mathbf{q}}}_b + B_b\dot{\overline{\mathbf{q}}}_b + C_b)] \tag{2.42}$$

At this point, by assuming also the knowledge of the body acceleration reference vector $\ddot{\overline{\mathbf{q}}}_a$(which, as it can be shown, allows the on-line evaluation of vector signals $\ddot{\overline{\mathbf{q}}}_b, \ddot{\overline{\mathbf{q}}}_r$) from (2.42) we can readily see that, with the adoption of a joint torque control signal having the form

$$\mathbf{m} = -\Lambda\delta\dot{\overline{\mathbf{q}}}_r + (A_r\ddot{\overline{\mathbf{q}}}_r + B_r\dot{\overline{\mathbf{q}}}_r + C_r) + (J_b^+ J_r)^T(A_b\ddot{\overline{\mathbf{q}}}_b + B_b\dot{\overline{\mathbf{q}}}_b + C_b) \tag{2.43}$$

with $\Lambda > 0$, and where all the appearing quantities are actually measurable ones (in particular refer to (2.33) for $\dot{\mathbf{q}}_b$ appearing inside B_b); so, we can actually make

$$\dot{U} = -\delta\dot{\overline{\mathbf{q}}}_r^T \Lambda \delta\dot{\overline{\mathbf{q}}}_r < 0 \tag{2.44}$$

which guarantees the asymptotic convergence of $\delta\,\dot{\overline{\mathbf{q}}}_r$ (and, then, of $\delta\,\dot{\overline{\mathbf{q}}}_b$ due to (2.33) or, equivalently, to the second of (2.35) solved for $\delta\,\dot{\overline{\mathbf{q}}}_b$) toward zero.

As it concerns the convergence rate, it is instead sufficient to observe that, by factorizing the positive definite matrix Σ in terms of its square root matrix H; that is

$$\Sigma = H^T H \tag{2.45}$$

and introducing the auxiliary vector

$$\mathbf{z} \triangleq H\delta\dot{\overline{\mathbf{q}}}_r \Leftrightarrow \delta\dot{\overline{\mathbf{q}}}_r = H^{-1}\mathbf{z} \tag{2.46}$$

we can actually rewrite (2.39) as

$$U = \frac{1}{2}\left\|\mathbf{z}\right\|^2 \tag{2.47}$$

and consequently (2.44) as

$$\dot{U} = -\mathbf{z}^T(H^{-T}\Lambda H^{-1})\mathbf{z} \leq -(\underline{\lambda}/\overline{\sigma})\left\|\mathbf{z}\right\|^2 \tag{2.48}$$

being $\underline{\lambda}, \overline{\sigma}$ the minimum and maximum eigenvalue (the latter evaluated among all possible postures) of positive definite matrices Λ and Σ, respectively. Then, since (2.47), (2.48) together imply that auxiliary vector $\mathbf{z}$ converges to zero with a rate no slower than $\exp[-(\underline{\lambda}/\overline{\sigma})t]$, we can immediately conclude that the same also holds for $\delta\,\dot{\overline{\mathbf{q}}}_r$, due to the fact that (from the second of (2.46))

$$\left\|\delta\dot{\overline{\mathbf{q}}}_r\right\| \leq (1/\underline{\sigma})^{1/2}\left\|\mathbf{z}\right\| \tag{2.49}$$

being $\underline{\sigma}$ the minimum eigenvalue of the matrix Σ among all possible postures. Naturally enough, the same behaviour also extends to $\delta\,\dot{\overline{\mathbf{q}}}_b$, as a consequence of the existing condition (see (2.33), or equivalently the second of (2.35) solved for $\delta\,\dot{\overline{\mathbf{q}}}_b$, and keep into account the upper bound adopted in (2.34))

$$\left\|\delta\dot{\overline{\mathbf{q}}}_b\right\| \leq (\overline{\sigma}_r/\underline{\sigma}_b)\left\|\delta\dot{\overline{\mathbf{q}}}_r\right\| \tag{2.50}$$

Finally, the ensured exponential convergence of $\delta\,\dot{\overline{\mathbf{q}}}_r$ toward zero, also guarantees the asymptotic convergence to zero of the tracking error $\mathbf{e}$ (as just established by (2.34), whenever considered with an exponentially zeroing term $\delta\,\dot{\overline{\mathbf{q}}}_r$).

A first comment which can be made to the developed position control law is that we deal with a 2-level control scheme: in fact, we have a first "Medium Level" loop (MLC), which calculates the joint velocity reference $\dot{\overline{\mathbf{q}}}_r$, to be imposed in order to follow the reference position and orientation of the object (law (2.30)); and, in turn, by law (2.43) we see that we have a "Low Level" control loop (LLC), closed on the joint velocity error $\delta\,\dot{\overline{\mathbf{q}}}_r$, in order to ensure that the joint velocity goes asymptotically to follow the reference $\dot{\overline{\mathbf{q}}}_r$.

As an additional comment to the previous developments, it is last worth noting how the proposed control law (2.43) can actually be considered as a special case of the following more general form

$$\begin{cases} \mathbf{m} = -\Lambda\delta\dot{\bar{\mathbf{q}}}_r + (A_r\ddot{\bar{\mathbf{q}}}_r + B_r\dot{\bar{\mathbf{q}}}_r + C_r) + (J_b^+ J_r)^T(A_b\ddot{\bar{\mathbf{q}}}_b + B_b\dot{\bar{\mathbf{q}}}_b + C_b) + J_r^T\mathbf{\Phi} \\ \\ \mathbf{\Phi} \in Ker(J_b^T) \end{cases} \tag{2.51}$$

where the presence of the additional term $J_r^T\mathbf{\Phi}$ (with $\mathbf{\Phi}$ above specified) cannot in any case influence the evolution of $\delta\dot{\bar{\mathbf{q}}}_r$ (and, consequently, of $\delta\dot{\bar{\mathbf{q}}}_b$), as imposed by the control law already mentioned.

In fact, by simply substituting (2.51) into (2.42), instead of (2.43), it is straightforward verifying that, with respect to (2.44), the corresponding expression for $\dot{U}$ simply modifies itself with the addition of the, however *null*, term

$$\delta\dot{\bar{\mathbf{q}}}_r^T J_r^T\mathbf{\Phi} = \delta\dot{\bar{\mathbf{q}}}_b^T J_b^T\mathbf{\Phi} = \mathbf{0} \tag{2.52}$$

where the equalities directly follow from kinematic constraints (2.35), and the assumed choice for $\mathbf{\Phi}$ in (2.51).

As it will be better clarified in the next section, the presence of the additional term $J_r^T\mathbf{\Phi}$, with $\mathbf{\Phi} \in Ker(J_b^T)$, will actually play an important role within the definition of an "ideal" complimentary force control law for contact constraints maintenance, in presence of dry friction.

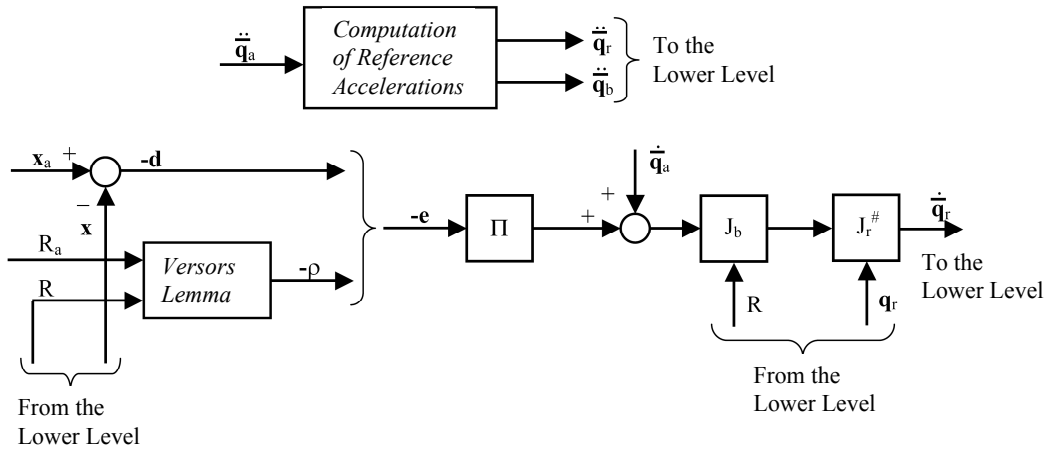

Figure 2.2: Upper level of the control scheme.

In fig. (2.2) and (2.3), for better clarity for the reader, the upper and lower level control schemes developed in this Section are shown.

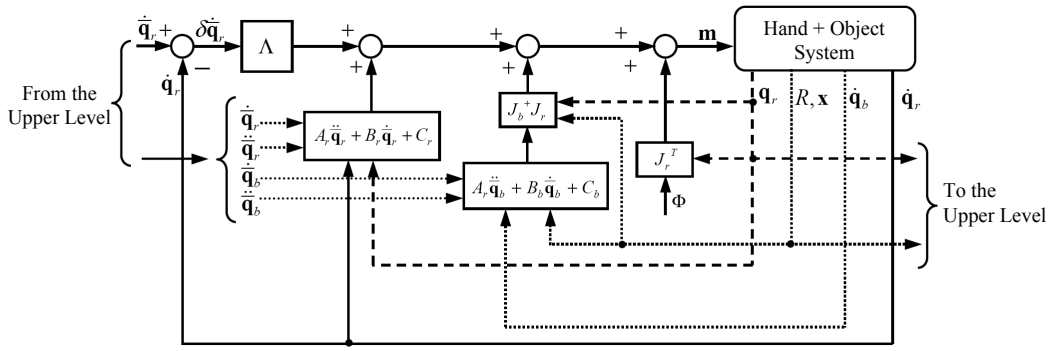

Figure 2.3: Lower level controller (*Computed-Torque*).

2.3.1 Acceleration reference vectors computation

In this paragraph the calculation of the acceleration reference vectors, given by the quantities $\ddot{\bar{\mathbf{q}}}_b, \ddot{\bar{\mathbf{q}}}_r$ which appear in the control law (2.51) (and depicted in fig. 2.2) will be developed. As it will be clarified by the following developments, their computation requires a little effort, and will be here described only for completeness of the theory so far devised; neverhteless, for simulation purposes it will be useful having these quantities, in order to completely show the effectiveness of the here proposed two-levels control.

Evaluation of $\ddot{\bar{\mathbf{q}}}_b$

First of all, consider the definition (2.29), here re-written for clarity of explaining:

$$\dot{\bar{\mathbf{q}}}_b \triangleq -\Pi\mathbf{e} + \dot{\mathbf{q}}_a; \Pi > 0 \tag{2.53}$$

By deriving this relationship with respect to time, we have immediately

$$\ddot{\bar{\mathbf{q}}}_b = -\Pi\dot{\mathbf{e}} + \ddot{\mathbf{q}}_a \tag{2.54}$$

hence, it becomes evident the presence of 2 terms, given respectively by:

- A term containing the time derivative of the global position/orientation error $\mathbf{e} \triangleq col(\mathbf{d}, \rho)$, obviously evaluated with respect to the base frame $< 0 >$, whose analytic calculation will require some attention, and will be hereinafter developed;
- An acceleration term, relative to the reference trajectory $\ddot{\mathbf{q}}_a \triangleq col(\dot{\mathbf{v}}_a, \dot{\omega}_a)$; this term is supposed to be known a priori, at every time instant, and given together with the position and velocity terms of the goal frame $< a >$;

At this point, in order to compute the vector $\dot{\mathbf{e}}$, we have

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{\mathbf{d}} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}} - \dot{\mathbf{x}}_a \\ \dot{\rho} \end{bmatrix} \tag{2.55}$$

being $\mathbf{d}$ and ρ as defined at the beginning of the Chapter; now, while the computation of $\dot{\mathbf{x}}$ (linear velocity of the body $< b >$) and $\dot{\mathbf{x}}_a$ (linear velocity of the reference $< a >$, known a priori) does not present difficulties, the calculation of the vector $\dot{\rho}$, i.e. the time derivative of the rotation error between the two frames $< a >$ and $< b >$, requires some computational effort, as it will become readily clear.

It is firstly necessary to specify that the computation algorithm for the rotation vector ρ, based upon the Versors Lemma (2.22), gives as output the 2 quantities

$$\mathbf{u} \equiv \frac{\rho}{\|\rho\|}; \theta \equiv \|\rho\| \tag{2.56}$$

which are the versor of the rotation axis and the (positive) angle, only in case that the angle itself be a significant quantity, that is, when $\|\rho\| \geq \epsilon$, being ϵ a (sufficiently small) positive threshold. If this is not true, being impossible to calculate accurately the versor $\mathbf{u}$, the approximation

$$\rho \simeq \tilde{\rho} \triangleq \mathbf{u} \sin\theta \tag{2.57}$$

can be adopted, where the last quantity comes directly form the first of the Versors Lemma (see again (2.22)), and it is however well-defined (like the vector ρ) for every value of θ.

In order to distinguish between the two cases, the condition to be verified before any subsequent calculation (both for ρ and its derivative $\dot{\rho}$) is then the following: the vector $\mathbf{u} \sin\theta$ is evaluated via the first Versors Lemma, whose norm is $|\sin\theta|$, and then the comparison

$$|\sin\theta| \geq \epsilon \tag{2.58}$$

is done, as a starting point for distinguish between the 2 ways (being the former exact, and the latter approximated but valid for $\theta \to 0$) for the calculation of ρ and its time derivative $\dot{\rho}$. In what follows, there will not be given the details for the yet mentioned algorithm for the computation of ρ, using directly its results, while the computation of the derivative $\dot{\rho}$ will be explained.

Consider now the deivative computation in the first case, which is generally valid starting from the knowledge of the axis-angle pair $\mathbf{u}, \theta$, and of the angular velocities $\omega_{b/0}, \omega_{a/0}$, indicated in the following by ω and ω^*, respectively. Moreover, when not explicitely said, all the considered vectors and the time derivative operations will be referred to the base frame $< 0 >$.

Hence, given the vector $\rho \equiv \mathbf{u}\theta$, we have

$$\dot{\rho} = \dot{\mathbf{u}}\theta + \mathbf{u}\dot{\theta} \tag{2.59}$$

and, then, we can separately consider the computation of the two terms in (2.59).

Starting with the vector $\dot{\mathbf{u}}$, some preliminary observations can be made; in particular, because $\mathbf{u}$ is the versor of the rotation axis between $< b >$ and $< a >$, obviously results

$$^{b}\mathbf{u} \equiv \ ^{a}\mathbf{u} \triangleq \ ^{*}\mathbf{u} \tag{2.60}$$

where, from now on, $^{*}\mathbf{u}$ is the projection of $\mathbf{u}$ on anyone of the two frames. But, as output of the Versors Lemma algorithm we have $^{0}\mathbf{u}$, and then it must be calculated, for instance,

$$^{*}\mathbf{u} = {}^{0}_{b}R^{T} \ ^{0}\mathbf{u} \tag{2.61}$$

Being now available the vector $^{*}\mathbf{u}$, it can be noted the fact that, as a consequence of (2.60), the relationship

$$^{b}_{a}R \ ^{*}\mathbf{u} = \ ^{*}\mathbf{u} \tag{2.62}$$

holds; then, by deriving with respect to time

$$\dot{R} \ ^{*}\mathbf{u} + R \ ^{*}\dot{\mathbf{u}} = \ ^{*}\dot{\mathbf{u}} \tag{2.63}$$

where R denotes the matrix $^{b}_{a}R = (^{0}_{b}R^{T})^{0}_{a}R$, which then can be evaluated starting from the known quantities $^{0}_{b}R, ^{0}_{a}R$. Now, it is important to note that the time derivative vector $^{*}\dot{\mathbf{u}}$ is the derivative of $\mathbf{u}$ with respect to $< b >$ (or $< a >$) frame, that is $^{*}\dot{\mathbf{u}} \equiv \frac{d_{*}}{dt}\mathbf{u}$, being, as before, $< * > \equiv < b >$ or $< a >$, and then

$$\dot{\mathbf{u}} \triangleq \frac{d_0}{dt}\mathbf{u} = \ ^{*}\dot{\mathbf{u}} + [\omega\wedge]\mathbf{u} \tag{2.64}$$

From the *Strapdown Equation* we have

$$\dot{R} = [^{b}\omega_{a/b}\wedge]R \tag{2.65}$$

being $\omega_{a/b}$ the angular velocity between the two frames, obtainable by $\omega_{a/b} = \omega^{*} - \omega$, being also known the 2 angular velocities as defined before; because it is necessary to have the expression of this vector on the $< b >$ frame, it will be $^{b}\omega_{a/b} = {}^{0}_{b}R^{T}(\omega^{*} - \omega)$; moreover, by considering the (2.62) we become to

$$[{}^{0}_{b}R^{T}(\omega^{*} - \omega)\wedge] \ ^{*}\mathbf{u} = (I_3 - R) \ ^{*}\dot{\mathbf{u}} \tag{2.66}$$

which turns out to be a linear algebraic system with $^{*}\dot{\mathbf{u}}$ as the unknown variable; nevertheless, the coefficients matrix $(I_3 - R)$ cannot be inverted; indeed, more precisely, it is

$$rank(I_3 - R) = 2, \forall R \neq I_3 \tag{2.67}$$

In order to justify this assertion, it can just be observed the well-known fact that the orthogonal matrices have always one unitary eigenvalue (except for the trivial

case of $R = I_3$), and then the null space of the matrix $(I_3 - R)$ have dimension given by

$$\dim \ker(I_3 - R) = 1 \tag{2.68}$$

at this point, it can be observed that also the following condition holds, for every versor (or every constant norm vector):

$${}^*\dot{\mathbf{u}} \perp {}^*\mathbf{u} \Leftrightarrow {}^*\mathbf{u}^T {}^*\dot{\mathbf{u}} = 0 \tag{2.69}$$

and this, together with the (2.66), gives the following algebraic system

$$A\, {}^*\dot{\mathbf{u}} = b \tag{2.70}$$

being:

$$A \triangleq \left[\frac{I_3 - R}{{}^*\mathbf{u}^T}\right];\ b \triangleq \left[\frac{[{}^0_b R^T(\omega^* - \omega)\wedge]\ {}^*\mathbf{u}}{0}\right] \tag{2.71}$$

where $A \in \Re^{4\times 3}, b \in \Re^{4\times 1}$; but the matrix A is full-column rank ($rank(A) = 3$) because, being ${}^*\mathbf{u} \in \ker(I_3 - R)$, it will be ${}^*\mathbf{u}^T \perp span((I_3 - R)^T)$, and then $rank(A^T) = 2 + 1 = 3 = rank(A)$.

This fact allows us to compute the solution of eq. (2.70), certainly uniquely existing, by using the pseudoinverse of matrix A:

$${}^*\dot{\mathbf{u}} = A^+ b \tag{2.72}$$

being $A^+ \triangleq (A^T A)^{-1} A^T$.

Regarding the second term in (2.59), that is, $\mathbf{u}\dot{\theta}$, here it can be noted the result (yet seen before) that

$$\dot{\theta} = \mathbf{u}^T(\omega - \omega^*) \tag{2.73}$$

and, finally, we get the following formula for the computation of the vector $\dot{\rho}$:

$$\dot{\rho} = ({}^*\dot{\mathbf{u}} + [\omega\wedge]\mathbf{u})\theta + \mathbf{u}\mathbf{u}^T(\omega - \omega^*) \tag{2.74}$$

where the vector ${}^*\dot{\mathbf{u}}$ is calculated by using (2.72).

The expression (2.74) is valid, as yet observed, under the assumption of being the angle-axis pair ($\mathbf{u}$,θ) available, and this is equivalent to the fulfilment of the condition (2.58); if, instead, that condition is not satisfied, we must proceed to approximate the rotation vector ρ with the expression (2.57), and then the following derivative will be calculated

$$\dot{\tilde{\rho}} = \frac{d}{dt}(\mathbf{u}\sin\theta) \tag{2.75}$$

The resultant expression, which here is given without proof (the details can be found in the literature), is the following:

$$\frac{d}{dt}\tilde{\rho} = \frac{1}{2}(\alpha I_3 - RR^{*T})(\omega - \omega^*) + \frac{1}{2}(R^* R^T - RR^{*T})\omega^* \tag{2.76}$$

being here $R \triangleq {}^0_bR$; $R^* \triangleq {}^0_aR$, and having defined $\alpha \triangleq (1 + 2\cos\theta)$; the latter scalar quantity can be obtained from the second Versors Lemma equation (2.22)

$$\alpha = \sum_{i=1}^{3} \mathbf{v}_i^T \mathbf{v}_i^* = \mathbf{i}^T\mathbf{i}^* + \mathbf{j}^T\mathbf{j}^* + \mathbf{k}^T\mathbf{k}^* \tag{2.77}$$

Summarizing what has up to now explained, starting from the condition (2.58), the computation of the vector $\dot{\rho}$ can be done by using one of the 2 formulas (2.74), (2.76) and, as a consequence, the desired acceleration reference vector $\ddot{\bar{\mathbf{q}}}_b$ can be obtained as

$$\ddot{\bar{\mathbf{q}}}_b = -\Pi \left[\frac{\dot{\mathbf{x}} - \dot{\mathbf{x}}_a}{\dot{\rho}} \right] + \ddot{\mathbf{q}}_a \tag{2.78}$$

Evaluation of $\ddot{\bar{\mathbf{q}}}_r$

Now the computation of the reference acceleration vector $\ddot{\bar{\mathbf{q}}}_r$, for which the results obtained before for $\ddot{\bar{\mathbf{q}}}_b$, will be performed. From (2.30) we have

$$\dot{\bar{\mathbf{q}}}_r \triangleq J_r^{\#} J_b \dot{\bar{\mathbf{q}}}_b \tag{2.79}$$

and, as a consequence,

$$\ddot{\bar{\mathbf{q}}}_r = (\frac{d}{dt} J_r^{\#}) J_b \dot{\bar{\mathbf{q}}}_b + J_r^{\#} \dot{J}_b \dot{\bar{\mathbf{q}}}_b + J_r^{\#} J_b \ddot{\bar{\mathbf{q}}}_b \tag{2.80}$$

Hence, it will be necessary to consider separately the calculation of the 3 distinct terms in (2.80), and the first of them, in particular, needs some computational effort, but it can be obtained with some straightforward considerations.

Starting from the Jacobian matrix expression, given in (2.10), it can be obviously noted that

$$J_r^{\#} = blockdiag[J_1^{\#}(\mathbf{q}_1), J_2^{\#}(\mathbf{q}_2), ..., J_h^{\#}(\mathbf{q}_h)] \tag{2.81}$$

and, then, we must deal with single separated blocks, anyone of them of dimension $(n_i \times 3)$ (being n_i the dof of the respective $i-th$ finger, $n_i \geq 3, i = 1, 2, ..., h$). In what follows, the assumption that the matrices $J_i(\mathbf{q}_i)$ are always *full-rank* will be done, and then their pseudoinverses can be calculated. Hence, by indicating with J one of these matrices, it will be

$$\dot{J}^{\#} = \frac{d}{dt}[J^T(JJ^T)^{-1}] = \dot{J}^T(JJ^T)^{-1} + J^T \frac{d}{dt}[(JJ^T)^{-1}] \tag{2.82}$$

Since for a generic, square and invertible matrix A the following relationship holds

$$\frac{d}{dt}(A^{-1}) = -A^{-1}\dot{A}A^{-1} \tag{2.83}$$

we can apply equation (2.83) to (2.82), thus obtaining

$$\dot{J}^{\#} = \dot{J}^T(JJ^T)^{-1} - J^T(JJ^T)^{-1}(\dot{J}J^T + J\dot{J}^T)(JJ^T)^{-1} = \tag{2.84}$$

$$= [\dot{J}^T - J^T(JJ^T)^{-1}(\dot{J}J^T + J\dot{J}^T)](JJ^T)^{-1}$$

which, in turns, requires the possibility of evaluating the temporal derivative $\dot{J}$ of the Jacobian matrix.

Let now $J^{(k)}(\mathbf{q}_k)$ be the linear Jacobian matrix of finger $k; (k = 1, .., h)$

$$\dot{J}^{(k)}(\mathbf{q}_k) = \left[\begin{array}{c|c|c|c} \dot{J}^{(k)}_{1,1} & \dot{J}^{(k)}_{1,2} & ... & \dot{J}^{(k)}_{1,n_k} \\ \hline \dot{J}^{(k)}_{2,1} & \dot{J}^{(k)}_{2,2} & ... & ... \\ \hline \dot{J}^{(k)}_{3,1} & ... & ... & \dot{J}^{(k)}_{3,n_k} \end{array}\right] \tag{2.85}$$

where $n_k =$ is the number of *dof* of finger k; then, we have

$$\dot{J}^{(k)}_{i,j} = \left[\frac{\partial}{\partial q_{k,1}} J^{(k)}_{i,j}, ..., \frac{\partial}{\partial q_{k,n_k}} J^{(k)}_{i,j}\right] \left[\begin{array}{c} \dot{q}_{k,1} \\ ... \\ \dot{q}_{k,n_k} \end{array}\right] = \left(\frac{\partial}{\partial \mathbf{q}_k} J^{(k)}_{i,j}\right) \dot{\mathbf{q}}_k \tag{2.86}$$

$i = 1, .., 3; j = 1, .., n_k$. That means, the temporal derivative of the matrix $J^{(k)}$ is, in general, a matrix of the same dimension $(3 \times n_k)$, which is a function of both the joint position vector $\mathbf{q}_k$ and its derivative $\dot{\mathbf{q}}_k$; therefore

$$\dot{J}^{(k)} = \dot{J}^{(k)}(\mathbf{q}_k, \dot{\mathbf{q}}_k) \tag{2.87}$$

The availability of an analytic representation for the matrix $J^{(k)}(\mathbf{q}_k)$ may allow a symbolic computation of the many terms inside (2.86), which contribute to make up the expression (2.85) of $\dot{J}^{(k)}$; currently, there exist symbolic computing packages which allow to obtain this expression in a simple way, with optimized computational times.

Concerning the second term in (2.80), i.e. $J_r^{\#}\dot{J}_b\dot{\overline{\mathbf{q}}}_b$, we must compute the expression of $\dot{J}_b$, which is the temporal derivative of the rigid body Jacobian matrix.

From the expression of J_b, given by (2.6), we immediately have

$$\dot{J}_b = \frac{d}{dt}\left[\begin{array}{c|c} I_3 & -[R\sigma_1\wedge] \\ \hline \multicolumn{2}{c}{\cdots} \\ \hline \multicolumn{2}{c}{\cdots} \\ \hline I_3 & -[R\sigma_h\wedge] \end{array}\right] = \left[\begin{array}{c|c} 0_3 & -[([\omega\wedge]R)\sigma_1\wedge] \\ \hline \multicolumn{2}{c}{\cdots} \\ \hline \multicolumn{2}{c}{\cdots} \\ \hline 0_3 & -[([\omega\wedge]R)\sigma_h\wedge] \end{array}\right] \tag{2.88}$$

being, as already mentioned in the previous paragraph, $R \triangleq {}^0_bR$, $\omega \triangleq {}^0\omega_{b/0}$; in order to obtain (2.88) we made use of the Strapdown Equation, as already done several times; we notice that the structure of $\dot{J}_b$ allows obvious simplifications while computing the term $J_r^{\#}\dot{J}_b\dot{\overline{\mathbf{q}}}_b$.

Finally, the third term in (2.80), that is $J_r^{\#}J_b\ddot{\overline{\mathbf{q}}}_b$, does not require further comments, since the expression of $\ddot{\overline{\mathbf{q}}}_b$ has already been obtained in the previous Section.

2.4 Definition of the control law for contact forces

In the previous Section we developed an "ideal" control law in order to follow trajectories, under the assumption of keeping the contact constraints satisfied during the whole manipulation task.

It is quite obvious that, since this requires to keep the set of contact forces $\mathbf{F} \triangleq col(\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_h)$ all directed towards the body, and at the same time internal to the respective *static friction cone*, it immediately arises the need of having a complimentary force control law, such as to maintain the above mentioned conditioned during the system motion.

In order to derive this additional control law, a preliminary discussion is needed, concerning the structure that the contact forces $\mathbf{F}$ will assume, under the assumption of validity of the constraints, and under the action of joint torques $\mathbf{m}$ of the form (2.51).

To this aim, we begin first of all by considering the general expression characterizing $\mathbf{F}$, under non-structured input actions on the joints $\mathbf{m}$. Such expression can be easily obtained from the general dynamic model (DAE) given by (2.15), by taking the second time derivative of the algebraic contact conditions (in order to do this, one can simply take the first derivative of the compact form (2.18)), thus obtaining the condition

$$J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} = \mathbf{0} \tag{2.89}$$

Afterwards, by substituting the expression for $\ddot{\mathbf{q}}$ from the first equation in (2.15), and finally solving with respect to $\mathbf{F}$, we obtain

$$\begin{aligned}\mathbf{F} = -(JA^{-1}J^T)^{-1}[JA^{-1}(\mu - B\dot{\mathbf{q}} - C) + \dot{J}\dot{\mathbf{q}}] = -(JA^{-1}J^T)^{-1}[-J_rA_r^{-1}\mathbf{m}+ \\ +J_rA_r^{-1}(B_r\dot{\mathbf{q}}_r + C_r) - J_bA_b^{-1}(B_b\dot{\mathbf{q}}_b + C_b) + \dot{J}\dot{\mathbf{q}}]\end{aligned} \tag{2.90}$$

Then, by structuring $\mathbf{m}$ as in (2.51), we get to the following expression

$$\begin{aligned}\mathbf{F} = -(JA^{-1}J^T)^{-1}[-J_r\ddot{\bar{\mathbf{q}}}_r - J_rA_r^{-1}(B_r\dot{\bar{\mathbf{q}}}_r + C_r)+ \\ -(J_rA_r^{-1}J_r^T)(J_b^+)^T(A_b\ddot{\bar{\mathbf{q}}}_b + B_b\dot{\bar{\mathbf{q}}}_b + C_b) + J_rA_r^{-1}(B_r\dot{\mathbf{q}}_r + C_r)+ \\ -J_bA_b^{-1}(B_b\dot{\mathbf{q}}_b + C_b) + \dot{J}\dot{\mathbf{q}} + J_rA_r^{-1}\Lambda\delta\dot{\bar{\mathbf{q}}}_r - (J_rA_r^{-1}J_r^T)\mathbf{\Phi}]\end{aligned} \tag{2.91}$$

that can be rewritten as

$$\begin{aligned}\mathbf{F} = -(JA^{-1}J^T)^{-1}[(-J_r\ddot{\bar{\mathbf{q}}}_r + \dot{J}\dot{\mathbf{q}}) + J_rA_r^{-1}(\Lambda + B_r)\delta\dot{\bar{\mathbf{q}}}_r - (J_rA_r^{-1}J_r^T)\mathbf{\Phi}+ \\ -(J_rA_r^{-1}J_r^T)(J_b^+)^T(A_b\ddot{\bar{\mathbf{q}}}_b + B_b\dot{\bar{\mathbf{q}}}_b + C_b) - (J_bA_b^{-1}J_b^T)(J_b^+)^T(B_b\dot{\mathbf{q}}_b + C_b)]\end{aligned} \tag{2.92}$$

where we simply summed the second and fourth term in (2.91), and the last term in (2.92) has been obtained from the fifth term in (2.91), by using the well-known identity $J_b^+J_b = I$.

Actually, we can further develop the first term in (2.92), by rewriting it as

$$\begin{aligned}-J_r\ddot{\bar{\mathbf{q}}}_r + \dot{J}\dot{\mathbf{q}} = (-J_r\ddot{\bar{\mathbf{q}}}_r + \dot{J}\dot{\bar{\mathbf{q}}}) + \dot{J}\delta\dot{\bar{\mathbf{q}}} = \\ = -J_b\ddot{\bar{\mathbf{q}}}_b + \dot{J}\delta\dot{\bar{\mathbf{q}}} = -(J_bA_b^{-1}J_b^T)(J_b^+)^TA_b\ddot{\bar{\mathbf{q}}}_b + \dot{J}\delta\dot{\bar{\mathbf{q}}}\end{aligned} \tag{2.93}$$

where the second right-hand side of the equation directly follows from the temporal derivative of the first condition of (2.37) (or, equivalently, the first of (2.35)), while the last is again a trivial consequence of the identity $J_b^+ J_b = I$.

Therefore, by substituting (2.93) in (2.92), keeping into account the definition (2.33), and moreover considering the following equalities

$$\begin{array}{l}(JA^{-1}J^T) = (J_b A_b^{-1} J_b^T) + (J_r A_r^{-1} J_r^T); \\ (J_r A_r^{-1} J_r^T)\mathbf{\Phi} = [(J_b A_b^{-1} J_b^T) + (J_r A_r^{-1} J_r^T)]\mathbf{\Phi} = (JA^{-1}J^T)\mathbf{\Phi}\end{array} \tag{2.94}$$

where the second directly follows from $\mathbf{\Phi} \in Ker(J_b^T)$ in (2.51), we finally get to the expression

$$\begin{cases} \mathbf{F} = (J_b^+)^T(A_b\ddot{\bar{\mathbf{q}}}_b + B_b\dot{\bar{\mathbf{q}}}_b + C_b) - (JA^{-1}J^T)[J_r A_r^{-1}(\Lambda + B_r)\delta\dot{\bar{\mathbf{q}}}_r + \\ \quad + J_b A_b^{-1}(\Lambda + B_b)\delta\dot{\bar{\mathbf{q}}}_b + \dot{J}\delta\dot{\bar{\mathbf{q}}}] + \mathbf{\Phi}; \\ \\ \mathbf{F} \triangleq \overline{\mathbf{F}} + \delta\overline{\mathbf{F}} + \mathbf{\Phi} \end{cases} \tag{2.95}$$

From the last expression, we can note how the set of contact forces $\mathbf{F}$ can be interpreted as composed of 3 main terms:

- A first term $\overline{\mathbf{F}}$, clearly coincident with the minimum-norm forces required in order to give to the body the reference velocity $\dot{\bar{\mathbf{q}}}_b$ (consider the body dynamics (2.9) for $\dot{\mathbf{q}}_b = \dot{\bar{\mathbf{q}}}_b$ an solve with respect to $\mathbf{F}$ with minimum norm), in turn converging towards the desired velocity $\dot{\mathbf{q}}_a$ (consider (2.33) and recall that both errors, of position and velocity, converge to zero);
- A second term $\delta\overline{\mathbf{F}}$, converging to zero with the velocity errors, arising from the initial (transient) difference between $\dot{\mathbf{q}}_b$ ande $\dot{\bar{\mathbf{q}}}_b$;
- Finally, a third term $\mathbf{\Phi}$, consisting only of *internal*, arbitrary forces (i.e. belonging to the space $Ker(J_b^T)$), which are applied to the body because of the above mentioned presence of the arbitrary term $J_r^T\mathbf{\Phi}$ in the control law (2.51).

Now we notice how any set of contact forces $\mathbf{F}$ applied to the body, can also be expressed (in a unique way) as the sum of two orthogonal terms, one related to the pure motion forces $\mathbf{F}_m \in Span(J_b)$, and the other to the internal forces $\mathbf{F}_o \in Ker(J_b^T)$, respectively; therefore, of the form

$$\mathbf{F} = \mathbf{F}_m + \mathbf{F}_o = P_b\mathbf{F} + (I - P_b)\mathbf{F} \tag{2.96}$$

where P_b is the orthogonal *projection* matrix onto the set $Span(J_b)$, given by

$$P_b \triangleq J_b(J_b^T J_b)^{-1} J_b^T \tag{2.97}$$

It immediately follows that, by applying the above decomposition to our case (2.95), we obtain the specification

$$\mathbf{F} = \mathbf{F}_m + \mathbf{F}_o = (\overline{\mathbf{F}} + P_b\delta\overline{\mathbf{F}}) + [(I - P_b)\delta\overline{\mathbf{F}} + \mathbf{\Phi}] \tag{2.98}$$

which clearly shows how, in general, only the term $\delta\overline{\mathbf{F}}$ splits into the two above mentioned components, while the terms $\overline{\mathbf{F}}, \mathbf{\Phi}$ already belong to the respective subspaces, of pure motion and internal forces.

In (2.98), we can further note how the firs term also represents the minimum-norm forces required in order to assign the current velocity $\dot{\mathbf{q}}_b$ to the body (consider the body dynamics (2.9), and solve it w.r.t. $\mathbf{F}$ with minimum norm, and recall that such solution necessarily belongs to $Span(J_b)$, and therefore coincides with the first term in (2.98)).

Still from (2.98), we finally note that the requirement of having each vector component $\mathbf{f}_i; i = 1, ..., h$ of $\mathbf{F}$ confined within the respective friction cone, should be therefore solved by only acting on the component $\mathbf{F}_o$, and not on $\mathbf{F}_m$, by suitably modulating the arbitrary vector $\mathbf{\Phi}$ (which influences motion), since all of the remaining terms have already been imposed by the system motion, which is independently forced by the application of the structured control law (2.51).

In order to be able to define a complimentary control law for satisfying the contact constraints, in the following we will make large use of a suitable parametrization for the internal forces set $\mathbf{F}_o$ exerted on the body, that has been recently introduced in [3]. Such parametrization in fact corresponds to what hereafter explained.

Let $\mathbf{n}_i; i = 1, ..., h$ be the unitary normals to the body, at the contact points, directed inwards, each one projected on the body frame $< b >$ (therefore constant in time). Moreover, we define the so-called *normal matrix* $N \triangleq blockdiag(\mathbf{n}_1, ..., \mathbf{n}_h) \in \Re^{3h \times h}$, also constant in time, and denote with $\alpha \triangleq [\alpha_1, ..., \alpha_h]^T$ a vector of dimension h of suitable scalar parameters. Afterwards, we denote with $\mathbf{\Psi}_o$ the projection on the body of the set $\mathbf{F}_o$ of internal forces applied to the body (projected onto the base frame $< 0 >$).

Therefore we can show that, under mild assumptions (that is, $h \geq 3$), the set $\mathbf{F}_o$ always has a parametrization of the form

$$\mathbf{F}_o = R^{(h)}\mathbf{\Psi}_o = R^{(h)}[(N + D)\alpha + S\lambda] \tag{2.99}$$

where $R^{(h)}$ is the $h - th$ block-diagonal organization using the rotation matrix R (here used only for transferring the projection from the body frame $< b >$ to the base frame $< 0 >$), while $(N + D) \in \Re^{3h \times h}$, $S \in \Re^{3h \times (2h-6)}$ are constant matrices with full rank, only dependent on the normal matrix N, and the object Jacobian matrix G projected on the $< b >$ frame, both constant. In particular, for a given configuration of body contact points, the matrices D and, obviously, N, are unique, while S may be modified only as a consequence of a base change in the vector space of parameters λ, with dimension $(2h - 6)$.

Concerning the constant structural matrices $(N + D)$ and S, it can also be shown that, apart from "covering" the full space of internal body forces, they also decompose such space into two orthogonal subspaces. Moreover, both components of the force vector separately due to $D\alpha$ and $S\lambda$ are *tangent* to the body surface, at the respective contact points. Therefore, only the force vector due to $N\alpha$ has a non-zero component along the normals $\mathbf{n}_i$ to the contact points (such components are obviously represented, globally, by the vector α itself).

By considering the parametrization (2.99), we must also assume to have preliminarly (i.e. by means of off-line computations) determined at least one pair of finite-norm vectors $[\underline{\alpha}^T,\underline{\lambda}^T]^T$ (supposed to exist), such as to satisfy the following conditions:

$$\mu\underline{\alpha}_i \geq |D_i\underline{\alpha} + S_i\underline{\lambda}|\,, i = 1, 2, ..., h \tag{2.100}$$

where $0 < \mu < 1$ is the static friction coefficient, related to the contact (assumed to be the same for all contacts), while D_i, S_i are matrices obtained from D and S respectively, by slicing them into blocks, each one consisting of 3 consecutive lines. As it can be easily understood, satisfying (2.100) simply corresponds to having off-line established the structure of a particular set of internal forces (body projected), which takes the form

$$\underline{\mathbf{\Psi}}_o = [(N + D)\underline{\alpha} + S\underline{\lambda}] \tag{2.101}$$

and it is such that the corresponding non-zero force components $\underline{\mathbf{\Psi}}_i; i = 1, 2, ..., h$ are all contained within the respective friction cone.

It is a matter of fact how a finite-norm parameter vector $[\underline{\alpha}^T,\underline{\lambda}^T]^T$ satisfying (2.100) can always be computed by means of simple linear programming techniques; moreover, in general there exist solutions characterized by $\underline{\lambda}= \mathbf{0}$, which may be preferred because of their computational simplicity.

Apart from the particular set of internal forces above defined $\underline{\mathbf{\Psi}}_o$, projected onto the body frame, consider now the following related parameter, strictly positive

$$\underline{\beta} \triangleq \min_i\{\underline{\beta}_i; i = 1, 2, ..., h\} > 0 \tag{2.102}$$

where $\underline{\beta}_i$ represents the *distance* of each force component $\underline{\mathbf{\Psi}}_i; i = 1, 2, ..., h$ of $\underline{\mathbf{\Psi}}_o$, from the *surface* of the respective friction cone.

Finally, based upon (2.101), (2.102), we define the following set of reference vectors for the internal forces, projected on the body:

$$\overline{\mathbf{\Psi}}_o \triangleq \frac{1}{\underline{\beta}}\underline{\mathbf{\Psi}}_o \tag{2.103}$$

which results to be a set of forces characterized by a *unitary* minimum distance from the surfaces of the friction cones.

As a consequence of all of the above given considerations, we are now able to formulate a very simple control law, and structurally invariant, although slightly conservative, in order to maintain the contact conditions during the system motion. As a matter of fact, due to its intrinsic simplicity, such a control law could be preferred for real-time applications, if compared with many others of more sophisticated nature, that can be developed for the same task.

In order to obtain such control law, first of all we restrict our attention to the sets of internal forces $\mathbf{F}_o$, projected on the base $< 0 >$, which are strictly aligned with the reference ones above defined $R^{(h)}\overline{\mathbf{\Psi}}_o$ (also projected on the base frame).

Moreover, we consider the decomposition (2.98) (first right-hand side) and, after defining

$$g(\mathbf{F}_m) \triangleq \max_i\{|\mathbf{f}_{m,i}|\,; i = 1, 2, ..., h\} \tag{2.104}$$

we simply note that a sufficient condition, in order to keep each vector component $\mathbf{f}_i; i = 1, ..., h$ of $\mathbf{F}$ confined in the respective friction cone, is to have the internal part $\mathbf{F}_o$ in the worst case of the form

$$\mathbf{F}_o = \gamma g(\mathbf{F}_m) R^{(h)} \overline{\mathbf{\Psi}}_o; \gamma > 1 \tag{2.105}$$

thus getting to the following expression for the additional term of internal force $\mathbf{\Phi}$ in (2.51) (see also the expressions of $\mathbf{F}_m$ e $\mathbf{F}_o$ in (2.98), second r.h.s.)

$$\mathbf{\Phi} = \gamma g(\overline{\mathbf{F}} + P_b \delta \overline{\mathbf{F}}) R^{(h)} \overline{\mathbf{\Psi}}_o - (I - P_b) \delta \overline{\mathbf{F}}; \gamma > 1 \tag{2.106}$$

which therefore represents the desired, complimentary control law for maintaining the contact constraints during motion.

As it can be observed, such a control low operates in *real-time*, first of all by compensating for the presence of the (transient) internal forces term imposed by $(I - P_b)\delta\overline{\mathbf{F}}$, while *simultaneously*, and suitably, varying the resulting internal forces set $\mathbf{F}_o$ along the projection of $\overline{\mathbf{\Psi}}_o$ on the base frame, in order to keep the validity of the sufficient condition (2.105) during time. This is obtained on the basis of our knowledge of the terms $\overline{\mathbf{F}}, \delta\overline{\mathbf{F}}$, which in turn are functions of known quantities, measurable from the posture and motion (as established by their expression given by the first two terms in (2.95)).

By concluding, we also note how the complimentary control law (2.106) is in fact of *open-loop* type, with respect to its controlled output $\mathbf{F}_o$.

This fact is actually coherent with our assumption, of the existence of pure *rigid* contact between the body and the fingertips of the manipulating structure (that is, an infinitely fast and stable dynamics of reaction forces), which induced a DAE modeling for the whole structure, and consequently established an algebraic relationship between output contact forces $\mathbf{F}$ and input torques $\mathbf{m}$, plus the motion signals (see again the general relationship (2.90)). And this fact, together with the assumed exact knowledge of the system model, necessarily leads to purely *algebraic* control laws of the contact forces, i.e. open-loop.

We note, however, that in case of available tactile sensors on the fingertips, able to measure $\mathbf{F}$, we can use a *closed-loop* control law, of the form

$$\dot{\mathbf{\Phi}} = K[\gamma g(\mathbf{F}_m) R^{(h)} \overline{\mathbf{\Psi}}_o - \mathbf{F}_o]; K > 0 \tag{2.107}$$

where $\mathbf{F}_m$ and $\mathbf{F}_o$ are evaluated in real time, from *measurements* of $\mathbf{F}$, by applying (2.96) and (2.97). As it can be easily understood, applying (2.107) simply corresponds to control the output $\mathbf{F}_o$ (as given by the second term of (2.98)), by means of an *integral* control action, dealing with the term $(I - P_b)\delta\overline{\mathbf{F}}$ as a transient disturbance.

It is clear, however, that further investigations are required, in order to be able to assess the efficiency of the control law (2.107).

2.5 Simulation results of the manipulation system

In order to show the efficiency and feasibility of the multi-finger manipulation system, depicted in Fig. 2.2, 2.3, we will provide now some results obtained through simulations of the complete dynamical model of the hand-object system above mentioned. The fundamental points upon which we based such tests, are the following ones:

- The possibiliy of working in a numerical computing and dynamical simulation environment given by the well-known MATLAB-Simulink system, used in order to build all of our control schemes and, as we will show in the following, implement on the available experimental setup all of the previously tested control algorithms;
- The availability of a modeling and development system for robotic structures consisting of linear or multi-branch kinematic chains, such as the RDS software (*Robotics Developer Studio*), developed and implemented at the DIST-Robotics Lab of the University of Genoa, in order to allow a fast and efficient implementation of MATLAB-Simulink functions; through the RDS software it is possible to obtain, starting from a complete description of the structure given through a graphical interface, all of the main algebraic expressions of an arbitrary robotic structure, such as transformation matrices, Jacobian matrices and direct dynamics expressions; such functions are finally converted in executable files under Simulink, i.e. *S-Functions*, and then directly inserted in the control scheme;
- A suitable modeling of kinematic constraints of *contact* between fingers and object, such as the one hereafter described;
- A suitable graphical interface, allowing the direct visualization of the system state, with a 3D rendering of the virtual scene.

Having all of the required tools needed for creating a simulation of the system dynamics at disposal, we need now to clear the main issue of the contact constraint modeling fingertips-object, which has been put into evidence at the beginning of the Chapter, and given by eqs. (2.11) and (2.12), and finally associated to the global system equations (2.15).

2.5.1 Modeling the contact forces

In the subsequent developments, it will be clear how the simulation scheme implements the contact constraint, in principle purely *algebraic*, not by solving a DAE equation system, but by modeling it in a *differential* way, based on a point-like contact *almost* rigid, obtained by introducing a *spring-damper* system in 3D, *virtually* inserted at the respective contact points of each finger to the object.

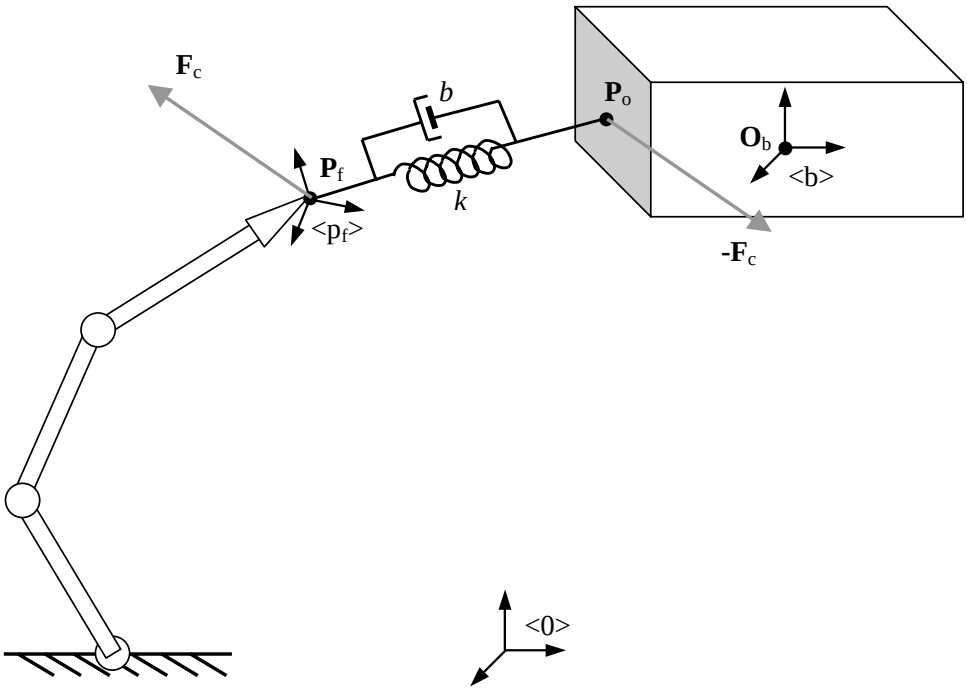

Figure 2.4: Schematics of the contact model between fingertip and object.

In order to simulate the point-like contact constraint, therefore, refer to Fig. 2.4, that shows a generic contact between the end-effector of a robot (in our case, a fingertip) and the object.

We the introduce of a set of cartesian frames, given by

- The inertial frame $< 0 >$, at the base of the system;
- The rigid body frame $< b >$, with origin located (for example) in the mass center, denoted by $\mathbf{O}_b$;
- A frame solidal with the fingertip, $< p_f >$, with origin located on the contact point, $\mathbf{P}_f$;
- The respective contact point on the body, $\mathbf{P}_o$;

We note how the above definitions require, obviously, a preliminary choice of the two contact points (on the fingertip and on the object), a choice that must be kept during the whole system evolution; this is a consequence coherent with the decision of assuming a point-like contact in presence of static friction as mentioned several

times, therefore without any sliding and/or rolling of the fingertip onto the object surface.

We can now proceed to formalize the differential, quasi-rigid constraint, by starting from the following equation, whose meaning will be clear in the following

$$\mathbf{F}_c^{(f)} = k(\mathbf{P}_o - \mathbf{P}_f) + b \cdot \mathbf{v}_{\mathbf{P}_o/p_f} \tag{2.108}$$

being $\mathbf{F}_c^{(f)}$ the contact force exerted *on the finger* as a consequence of two actions:

- An *elastic* term $k(\mathbf{P}_o - \mathbf{P}_f)$, proportional to the detachment between contact points, according to a proper constant k, and directed along the line joining the two points;
- A *viscous* term $b \cdot \mathbf{v}_{\mathbf{P}_o/p_f}$ proportional, according to the b constant, to the *relative* velocity of point $\mathbf{P}_o$ with respect to the frame solidal to the fingertip; this term, which exerts a "braking" action on the relative motion between the two points, is required in order to damp the *oscillations* otherwise induced by the presence of the "spring" given by the first term; we also note that its direction, obviously, is not necessarily coincident with the line joining the two points;

Assuming now to knoe the system *state*, i.e. the joint position and velocity of the finger $\mathbf{q}, \dot{\mathbf{q}}$, as well as the position and velocity of the object with respect to the base frame $< 0 >$, it will be possible first of all to compute the following quantities, all referred to the base frame

$$\overline{\mathbf{P}}_o = {}^0_bT \begin{bmatrix} \sigma \\ 1 \end{bmatrix}; \ \overline{\mathbf{P}}_f = {}^0_{p_f}T(\mathbf{q}) \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{2.109}$$

being ${}^0_bT, {}^0_{p_f}T(\mathbf{q})$ the transformation matrices of object and finger, and σ the constant position of the object point, projected on the base frame $< b >$ (note how we defined the positions of points in *homogeneous* coordinates, through the matrices T). Through the equations (2.109) we are now able to compute the elastic term in (2.108).

Concerning the viscous term, some more algebra is necessary; by recalling the rule of composition for linear velocities, we get

$$\mathbf{v}_{\mathbf{P}_o/0} = \mathbf{v}_{\mathbf{P}_f/0} + \omega_{p_f/0} \wedge r_{\mathbf{P}_o,\mathbf{P}_f} + \mathbf{v}_{\mathbf{P}_o/p_f} \tag{2.110}$$

where it appears the term $\mathbf{v}_{\mathbf{P}_o/p_f}$ of relative velocity that we desire to compute; the other terms are: the absolute velocities $\mathbf{v}_{\mathbf{P}_o/0}, \mathbf{v}_{\mathbf{P}_f/0}$ of the two contact points, the angular velocity $\omega_{p_f/0}$ between fingertip and base frame, and the vector $r_{\mathbf{P}_o,\mathbf{P}_f} \triangleq (\mathbf{P}_o - \mathbf{P}_f)$ joining the two points, already computed in the elastic term.

Therefore, by solving for the desired quantity, we have

$$\mathbf{v}_{\mathbf{P}_o/p_f} = \mathbf{v}_{\mathbf{P}_o/0} - \mathbf{v}_{\mathbf{P}_f/0} - \omega_{p_f/0} \wedge r_{\mathbf{P}_o,\mathbf{P}_f} \tag{2.111}$$

by developing the term $\mathbf{v}_{\mathbf{P}_o/0}$ in (2.111), again from the composition rule for linear velocities, we have

$$\mathbf{v}_{\mathbf{P}_o/0} = \mathbf{v}_{\mathbf{O}_b/0} + \omega_{b/0} \wedge r_{\mathbf{P}_o,\mathbf{O}_b} \tag{2.112}$$

where we take into account the fact that the relative velocity $\mathbf{v}_{\mathbf{P}_o/b}$ is of course zero, since the point $\mathbf{P}_o$ is solidal with the rigid body; finally, we have

$$\mathbf{v}_{\mathbf{P}_o/p_f} = \mathbf{v}_{\mathbf{O}_b/0} - \mathbf{v}_{\mathbf{P}_f/0} + \omega_{b/0} \wedge r_{\mathbf{P}_o,\mathbf{O}_b} - \omega_{p_f/0} \wedge r_{\mathbf{P}_o,\mathbf{P}_f} \tag{2.113}$$

At this point, the 4 term appearing on the right hand side of (2.113) can be computed in the following way:

- The term $\mathbf{v}_{\mathbf{O}_b/0}$ is given as state variable of the rigid body, in the simulated model;
- The term $\mathbf{v}_{\mathbf{P}_f/0}$ is given by

  $$\mathbf{v}_{\mathbf{P}_f/0} = {}^0J_{L,p_f}(\mathbf{q})\dot{\mathbf{q}} \tag{2.114}$$

  being ${}^0J_{L,p_f}$ the linear part of the Jacobian matrix between frame $< 0 >$ and the end-effector frame $< p_f >$;
- In the third term, the angular velocity $\omega_{b/0}$ is given as state variable of the rigid body, while the vector $r_{\mathbf{P}_o,\mathbf{O}_b}$, which we desire to express with respect to $< 0 >$, can be computed as

  $$r_{\mathbf{P}_o,\mathbf{O}_b} = {}^0_bR\sigma \tag{2.115}$$

  where the transformation matrix ${}^0_bT$ is known as state variable, as well as the constant vector σ previously defined;
- Concerning the last term, the angular velocity $\omega_{p_f/0}$ of the fingertip frame is given, analogously to the second term, by

  $$\omega_{p_f/0} = {}^0J_{A,p_f}(\mathbf{q})\dot{\mathbf{q}} \tag{2.116}$$

  where ${}^0J_{A,p_f}$ is the angular part of the Jacobian matrix of the end-effector; finally, the vector $r_{\mathbf{P}_o,\mathbf{P}_f}$ has already been computed in the elastic term through (2.109);

Given the term $\mathbf{v}_{\mathbf{P}_o/p_f}$ from (2.113), we have therefore also the viscous term in (2.108), and finally also the instantaneous contact force, exerted on the fingertip; by translating the force in terms of an equivalent *joint* torque system, we have

$$\mathbf{m}_c = J^T_{L,p_f}(\mathbf{q}) \cdot \mathbf{F}^{(f)}_c \tag{2.117}$$

being $\mathbf{m}_c$ the joint torque, used as input to the finger dynamics simulator; this torque term must obviously be *added* to the other terms of joint torque, given by the position and force controller of the whole system.

From the object point of view, by proceeding in an analogous way for all fingertips, we get a set of contact forces on each point, each one trivially given by

$$\mathbf{F}_{c_i}^{(o)} = -\mathbf{F}_{c_i}^{(f)};\ i = 1, 2, ..., h \tag{2.118}$$

By collecting all force vectors on the object in a single column vector $\mathbf{F}_c^{(o)} = col(\mathbf{F}_{c_1}^{(o)}, \mathbf{F}_{c_2}^{(o)}, ..., \mathbf{F}_{c_h}^{(o)})$, we arrive to the following relationship

$$\begin{bmatrix} \mathbf{f} \\ \tau \end{bmatrix} = J_b^T(\mathbf{q}_b)\mathbf{F}_c^{(o)} \tag{2.119}$$

which allows to compute the force/torque on the object through the Jacobian matrix of the rigid body J_b, defined by (2.6). This will be the generalized force term that, once inserted in the rigid body dynamics simulator, allows to obtain the temporal evolution of its state variables, given by ${}_b^0T$ and $\omega_{b/0}, \mathbf{v}_{\mathbf{O}_b/0}$.

2.5.2 Simulation results

Hereafter we will describe some simulations related to the hand-object system, obtained through the above mentioned software tools, and using the described dynamical contact model.

The simulated system consists of a 4-fingered hand, with overall 16 *dof*, kinematically correspondent to the experimental setup of which we will talk in the last Chapter. In the present simulations we used the first 3 fingers, that is the thumb and the second and third finger, only for simplicity of computation, and therefore without loss of generality of the control methodology, that in fact can be applied to an arbitrary numbr h of fingers, without substantial modifications.

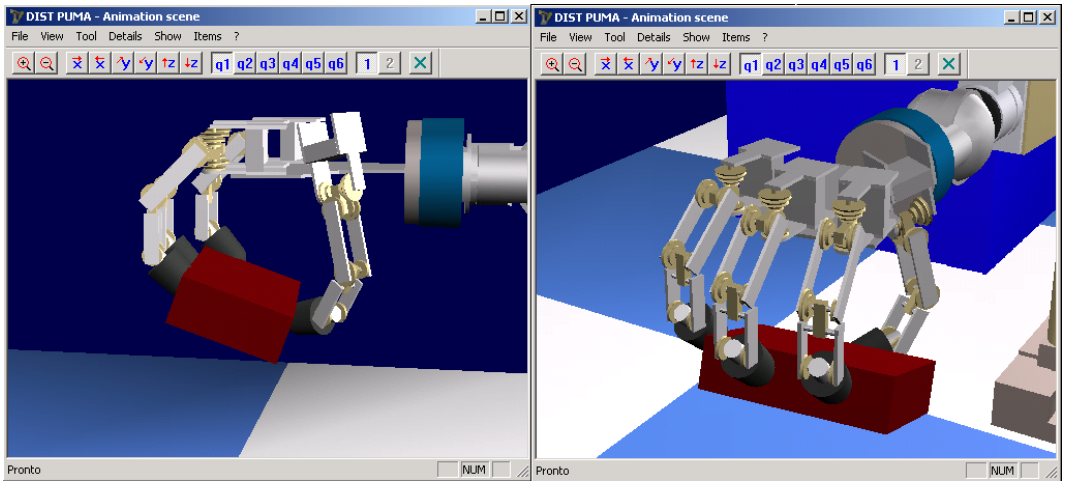

Figure 2.5: Initial grasp of an object.

Starting from suitable initial conditions, the task consists of positioning and orienting a given object, of non-negligible mass, with the shape of a box, and grasped by the hand on 3 given points, as it can be seen in Fig. 2.5.

Moreover, together with the position control, a complimentary *force* controller is present, that uses the closed-loop formula derived in (2.107). As reference vector for the internal forces, we decided to use, in an empirical way, the vector obtained by *projecting* the 3 normal versors to each contact n_i (see also the Section about the internal *grasping* forces) onto the space of internal forces, by means of the orthogonal projector onto the null-space of J_b^T, given by the term $(I - P_b)$ inside (2.96); in this way, starting from vectors which are surely contained within the static friction cones (because lying along their axes), these vectors are being projected in a such a way that they keep being within the internal forces space (i.e. such that they do not generate any motion).

We notice how this choice does not guarantee always to have, as reference vectors, forces contained within the friction cones; but this can be verified in advance, since these vectors are computed *off-line*, in order to satisfy the two above mentioned requirements (i.e. to be internal forces *and* to lie within the friction cones), and therefore these vectors can be multiplied by a unique *scalar* factor, theoretically of an arbitrary magnitude, in order to ensure the static friction conditions, during all of the manipulation task.

We also notice how, as already mentioned, the closed-loop control law (2.107), being based on *feedback* and not pre-computed like as the (2.106), does not guarantee *a priori* to keep the overall contact forces within their friction cones during all of the manipulation task; therefore, it is also necessary to monitor the behaviour of the forces during manipulation, by checking the *angle* with respect to the contact normal against the static friction coefficient threshold μ.

In Fig. 2.6 we can see the reference position signals, along the three Cartesian axes (left side) and the three orientation references (in terms of the axis-angle vector); for sake of simplicity, only the term containing the acceleration reference $\ddot{\bar{\mathbf{q}}}_b$ of the two *feed-forward* terms within (2.51) has been computed, since in this example, the mass of a single finger is assumed to be negligible, with respect to the mass of the manipulated object. Orientation references are given in terms of *Roll-Pitch-Yaw* angles with a smooth behaviour, which are internally converted into a reference rotation matrix $R^*(t)$.

As we can notice, in order to show the validity of the control law (2.51), in presence of a *feed-forward* acceleration signal, after a short transient the position and orientation of the object closely follows the reference signals, apart from small errors due, as already mentioned, to the lack of the first feed-forward term in (2.51).

Concerning manipulation forces during the task, the results are shown in Fig. 2.7. Force references are therefore given by a set of contact forces obtained, as above mentioned, by projecting the contact normals onto the internal forces space through the projector $(I - P_b)$, and multiplied by a suitable scalar. The absolute values of the force control errors, of course projected onto the internal forces space, are shown in the three plots on the left side; as we can notice, after a short transient

(the duration of which depends upon the *gain* value of the closed-loop force integral (2.107)), the converge and keep very low values during all of the manipulation task.

The efficiency of the force controller is shown by the behaviour the angles (expressed in *degrees*) between the three *overall* force vectors (not projected) applied to the respective contacts, and the three normal versors to the surface; we can notice how these values, plotted on the right side of Fig. 2.7, reduce to low values (a few degrees) right after the end of the respective force error transient.

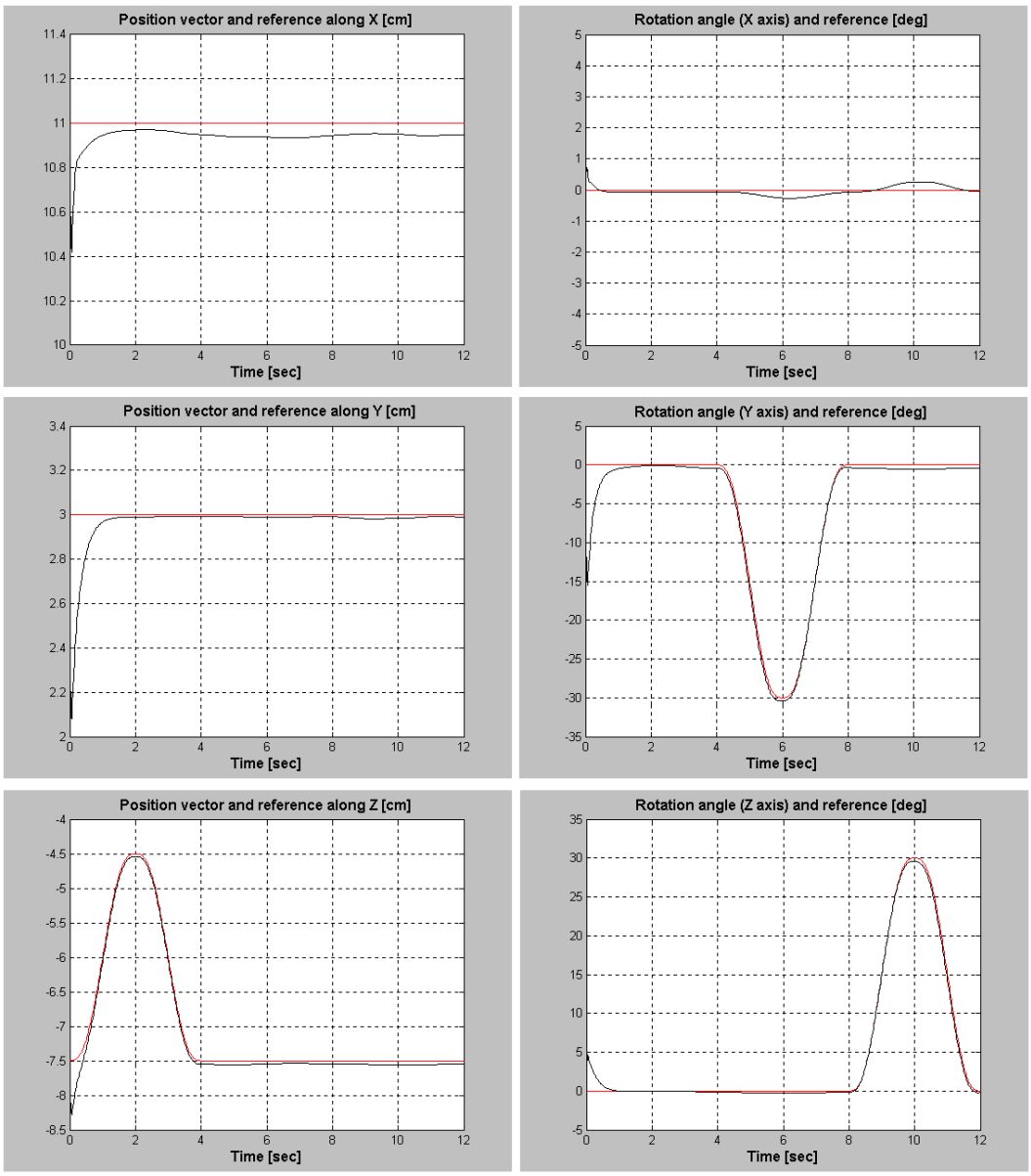

Figure 2.6: Following position and orientation references: desired values (red) and resulting values (black).

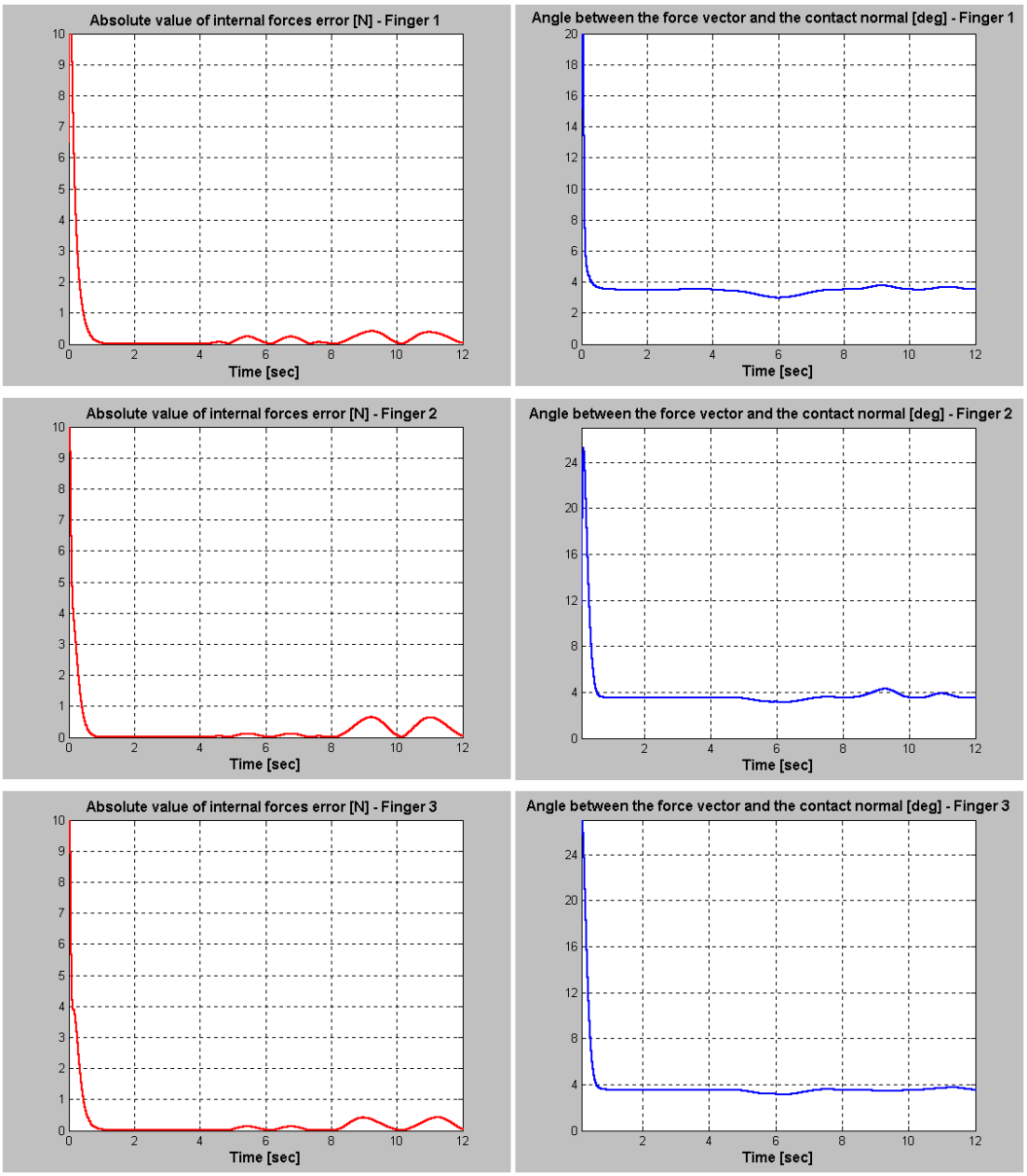

Figure 2.7: Force control signals, present in the simulated system.

Chapter 3

Functional structure and control hardware for the integrated hand-arm system

Concerning the general control schemes for the hand-arm system, we can think of an integrated architecture built in a modular way, and organized in several levels; that is, the control system can be implemented as a set of software *tasks*, operating in a *multitask* and *multiprocessor* environment, hierarchically distributed over the whole computing hardware. Each one of them may send commands to the lower level modules, and receive from them feedback signals.

Such a modular structure is coherent with the goals given in the Introduction, and it is almost consequential with the theoretical developments of the algorithmic control approach that will be hereafter described; moreover, it shows quite obvious implementation advantages, that will be made clear during its description.

3.1 Functional structure of the control system

Starting from the *software* viewpoint we have, as already mentioned, a *multi-level* structure that may be represented as in Fig. 3.1. The meaning of each level and each module will be explained in more detail in the algorithmic description of the hand-arm control approach; however, we can already present a brief overview of such a functional architecture in a "bottom-up" way, differently from what will be given when describing the specific control algorithms.

Referring to Fig. 3.1, we can first of all identify a "very low" control level (VLLC), made up of modules (or *tasks*) that directly interact with the respective physical system (in our case, the fingers of the robot hand, and the supporting arm); at this level, each module implement a joint *velocity* control for each physical entity inside the system; their task is to actuate, through the electrical actuators and the force transmission systems, the joints of the respective entity, in order to follow "at best" the velocity references from the upper level modules; such control is obtained

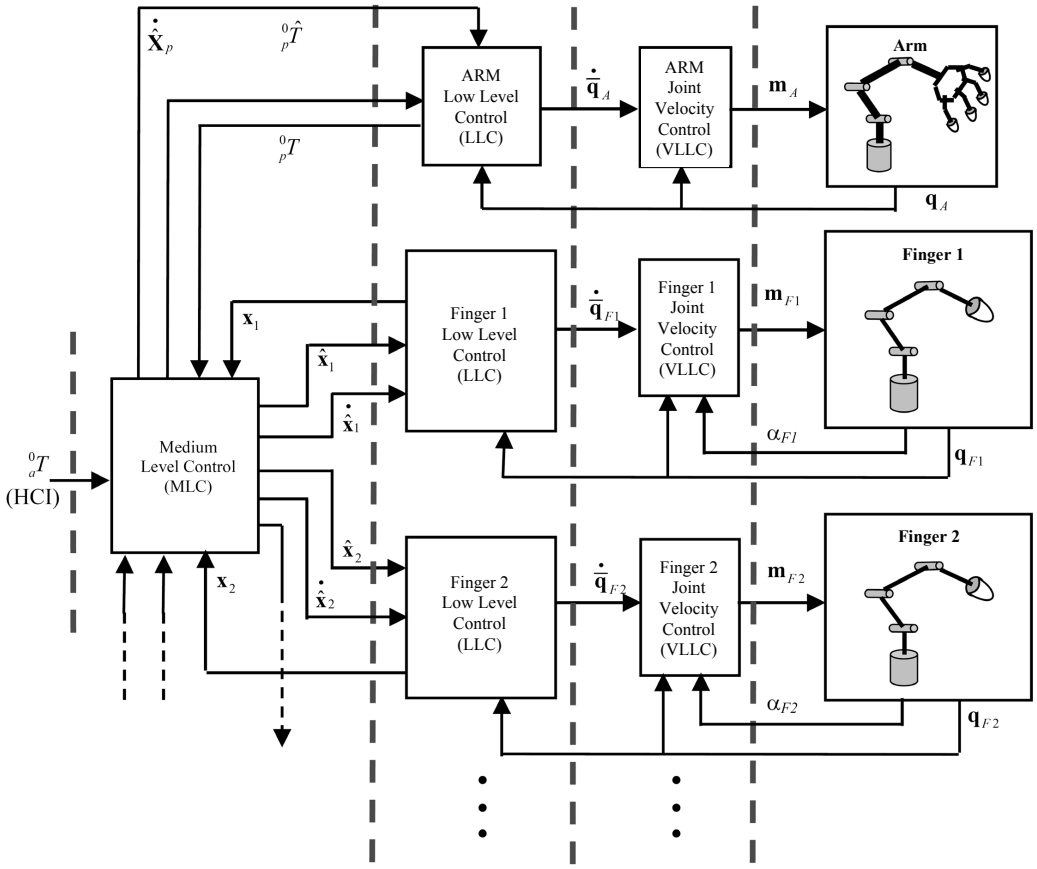

Figure 3.1: Functional architecture of the control system.

through a suitable closed-loop scheme, by measuring the necessary feedback signals directly from sensors located on the system joints.

The VLLC modules are independent one another, and their implementation is strictly related to the physical sensory and actuation system of the respective robotic entity, as well as to the specific mechanical force transmission system. In particular, the robot hand fingers are actuated by means of a structure composed of *tendons*, and this implies particular requirements while realizing the respective VLLC modules, as it will be made clear in the following.

By going to the immediately upper level, we find a layer of "low level" controllers (LLC), made up of modules which have a particular meaning in the context of the functional control structure; in particular, the role of each LLC module is to realize a cartesian control on a purely *kinematic* basis, of each physical entity; that is, each LLC module receives position commands (as well as *feed-forward* velocity signals) for the robot end-effector, in cartesian space, and it has the task of asymptotically

follow these signals through joint velocity references, to be sent to the lower level (VLLC), while receiving the joint position feedback from the respective physical entity.

On this basis, we can design each LLC module in a uniform way, by taking care of making explicit all of the necessary quantities for executing the above mentioned cartesian task, and including all of the computing structures which depend on the robot kinematics, so that only a given part of the module will be modified, while going from one robot structure to another; that implies that each LLC module corresponds to a computing system which depends on the kinematic structure of the robot (*robot-dependent*), but not on the specific task to be performed (*task-independent*), since the execution of such task, and its decomposition into local cartesian sub-tasks, is deferred to an upper level, which we denote as "middle level" of control (MLC).

The MLC level consists of the *global* task controller, and it is composed of a single module, able to compute the cartesian position and velocity signals, that have to be followed by the respective LLC modules; in order to allow the upper level to evaluate the fulfilment of the global task, each LLC module makes available, as output, the cartesian position of the respective end-effector (tool frame). Such a module, as we already mentioned, operates exclusively in cartesian space, since it does not have any need to know the kinematic structure of the controlled robot parts; in other words, it is a *robot-independent* and *task-dependent* module.

Finally, we can notice by referring again to Fig. 3.1, the fact that the middle level global task has control and scheduling inputs, given by a possible upper level, for example a *Human-Computer Interface* (HCI); such an interface is driven by a human operator, or by any "intelligent" controller, which provides the possibility of assigning to the general system controller a more complex sequence of operations (for example, assembly, coordinated transport, etc.).

3.2 The hardware control environment

Concerning the *Hw/Sw* environment, proposed in order to realize the control schemes that will be described in more detail in the next Section, we can state that, due to the modular and multi-level architecture before described, we reach in a natural way a real-time environment, which is *multi-tasking* and *multi-processor*, effectively implementing the control schemes. In particular, in Fig. 3.2 we represent the hardware structure, where the control boards employed in the system are shown.

As we can easily deduce, this architecture leads to the separation of the MLC control level from the LLC and VLLC, distributed in parallel computing boards; in particular, we have a Master board, where the MLC task runs, and two Slave boards, for each LLC and VLLC module of the robot hand fingers, as well as one for the arm. This separation can be obtained by means of suitable communication channels, for exchanging the different input and output signals of the modules, by using a VME bus architecture, which is part of the "rack" contaning the control

boards.

The hardware instruments that have been used in order to realize the different control schemes are therefore: the rack, where the computing boards are mounted, some of which directly interacting with the robot, and on which the control *tasks* are executed; one common Workstation PC, connected to the rack by means of local Ethernet connection, onto which all of the development and implementation facilities are installed, such as MATLAB, Simulink, RDS, Real-Time Workshop, and that also hosts the management software for the real-time operating system Tornado/VxWorks.

In particular the control boards, communicating one another via VME Bus, are of two types:

- a Motorola *MVME167* board, which hosts the task corresponding to the MLC module, and also operates as network *gateway* between the host PC and the other Slave boards;
- several *MVME162-22* boards, onto which the physical I/O devices are mounted, that directly interact with the robot through *flat cable* connectors; on these boards, the tasks related to the LLC/VLLC control level modules operate, and exchange interface signals with the MLC module of the Master board;

Onto the 162 boards, several interfaces for managing the *piggy-back* modules, onto which the analog-digital I/O devices necessary for controlling the physical system operated by the VLLC tasks.

The advantages of the above mentioned realization are, mainly: a) the overall reduction of the computing time for the individual board, due to the distribution of the software on more processors; b) the possibility of choosing different real-time sampling frequencies for each task, acting on the same board or on different boards; c) the *modularity* of the hardware, which allows to substitute a robot architecture with a different one, or to add a new one, simply by substituting the respective VLLC modules and, whenever needed, the related I/O systems, as well as the partial modification of the LLC modules (but only concerning the kinematic peculiarities of the given robot).

It must be noticed how the differentiation of the real-time sampling times of the tasks allows to obtain good performances for the whole control system, if we assign a higher frequency to the VLLC tasks (since they implement, in fact, velocity servo-controls on the system actuators, and therefore they are realized with relatively simpler structures, from the algorithmic/functional point of view); and, at the same time, we can choose lower sampling frequencies for the LLC modules, that contain more complex algorithmic structures (concerning the kinematic structure of the system), possibly subdividing them further in several control tasks with different sampling times. Even more the MLC control level, which is hosted by a parallel Master board, will need higher real-time computing times, and possibly computing boards with higher performances, given the overall complexity of the computations to be performed, related to the global control task.

3.3 Experimental setup: the DIST-Hand robot

3.3.1 Kinematic structure

The DIST-Hand is an anthropomorphic robot hand, designed and currently commercialized by the spin-off company *GRAALTECH* of the *GRAAL* lab of the University of Genoa.

The configuration of the 4 fingers, perfectly identical in their structure (see Fig. 3.3), and similar to the human skeleton, has 3 fingers on the upper part, and an opposed thumb in the lower part; each finger (including the thumb) has 4 degrees of freedom, obtained through rotational joints, of which one (the first) has axis orthogonal to the "palm", while the others have 3 parallel axes, which are orthogonal to the first one (see Fig. 3.4).

Moreover, on each joints are mounted several *pulleys*, which host the tendons that are used for actuating the structure.

In Fig. 3.5 we can notice the organization of the tendons transmission system on a single finger, described in the Section concerning the system actuation. For the moment, we mention first of all that on each joint we can find different types of pulleys:

- Free pulleys: they are meant uniquely for the *routing* of tendons; they are free to rotate, and used uniquely for routing the tendon (ideally without friction) towards the joint that has to be actuated;
- Actuation pulleys: they are solidal with the joint; the tendon is anchored on them, and it ends there.

From now on, we will denote with $\mathbf{q}$ the joint angles of the hand, expressed in radians.

It is also necessary to estabilsh the "zero" position of the hand, which corresponds to $\mathbf{q} = \mathbf{0}$; afterwards, we will be able to calibrate the joint position sensors, as well as to position the different frames at the base of each finger.

In such a position, it will be important to have the joints more or less in the middle of their limits, in order to guarantee a maximal motion freedom in both rotational directions.

At the same time, the zero position should be such that all fingers are in a position convenient for grasping; therefore, we adopted the following positioning:

- Joint 1: for each finger, apart the thumb, we position the joint such that $q_{1,j} = 0$ (where $q_{i,j}$ is the $i-th$ joint of finger j) corresponds to link 1 aligned with the support on which it is mounted; for finger 1, instead, we decided to adopt $q_{1,1} = 0$ corresponding to link 1, rotated by $-45°$ about the axis of joint 1, again with respect to its support; therefore finger 1, in its zero position, will be reasonably close to the other fingers;

- Joint 2: for fingers 2,3 and 4, we have a rotation of $+45°$ about the respective axis, between link 2 and link 1, with respect to their alignment; for finger 1, links 1 and 2 are aligned, corresponding to $q_{2,1} = 0$;
- Joints 3 and 4: for all fingers, we have a rotation of $+45°$, as we did for joint 2.

3.3.2 Actuation

In order to generate the necessary forces for the actuation, we used dc motors *Hitec HS-81MG*, whose main features are resumed in the following table:

Power supply	$5V$
Speed at $4.8V$	$0.15\frac{\text{sec}}{60°}$
Torque at $4.8V$	$2.2Kg \times cm$
Weight	$17.5Kg$
Dimensions	$28 \times 13.7 \times 28mm$

Each motor, in order to develop a suffucient torque for moving the hand, is equipped with a gear reduction mechanism, that reduces the speed at the slow transmission shaft.

This mechanism is embedded inside the motor package, in order to take the minimum volume possible.

The motors are controlled by the input voltage, they show a good frequency response, and have a smaller voume and cost.

Each finger is equipped with 5 motors: the first 4 actuate a single tendon, through a pulley which is solidal with the slow shaft, whereas the last motor mounts 2 tendons onto its pulley.

In Fig. 3.6 we can see the spatial arrangement of motors, pulleys and tendons corresponding to the internal part of the "motor box"; the naming convention is the following one:

$M1, ..., M5$ = motors

$P1, ..., P5$ = motor pulleys

$T1, ..., T6$ = tendons

The correspondence between motors-pulleys-tendons is the following one:

Motors	Pulleys	Tendons
M_1	P_1	T_1
M_2	P_4	T_4
M_3	P_3	T_3
M_4	P_2	T_2
M_5	P_5	T_5, T_6

Since the pulleys have a larger diameter with respect to the motor dimension, in order to reduce the occupancy, the latter are placed with pulleys alternately to each

side ot the base where they are placed; in this way, motors can still be mounted adjacent to each other, without any contact between their pulleys.

Between motors and finger joints, it has been designed a system of tendons (6 per finger) made up of polyhetilene, with a well-defined path through the finger pulleys (Fig. 3.5).

In order to develop an arbitrary set of joint torques, the minimum number of tendons to be employed should be $n + 1$, where n represents the number of joints per finger; in our case, therefore, we should use at least $4 + 1 = 5$ tendons.

However, with only 5 tendons the *routing* path would have been awkward and, in fact, less functional; therefore, we decided to organize the actuation system in the following way:

- for joint 1 we have 2 tendons, directly connected to it (antagonist), and actuated by motor 5;
- for the other 3 joints, we have an actuating structure that respects the above mentioned requirement: that is, we have $3+1 = 4$ tendons, realizing the given joint torques; these tendons are in turn actuated by motors $1, 2, 3$ and 4.

It is worth to mention, at this point, that the forces that can be developed onto the tendons by the motors are not all independent; in fact, this is true only for the first 4 tendons, separately actuated.

For the last two (T_5, T_6), instead, we have only one motor and a common pulley; therefore, if the shaft exerts a torque in a given rotational direction, the force on the other one will be zero.

A last point to be mentioned is that, between motors and finger, the tendons are running inside sheats, suitably lubricated, in order to reduce as much as possible the sliding friction.

Briefly talking about the arm-related part of the integrated system, we adopted a common industrial manipulator *PUMA 260*; the device has 6 rotational *dof*, and the last 3 joints give an Euler angles *wrist*; it is acuated by 6 dc motors, and the positional feedback is provided to the control system through standard optical *encoders*, in order to read the joint angular position.

Finally, the complete system is shown in the photo of Fig. (3.7).

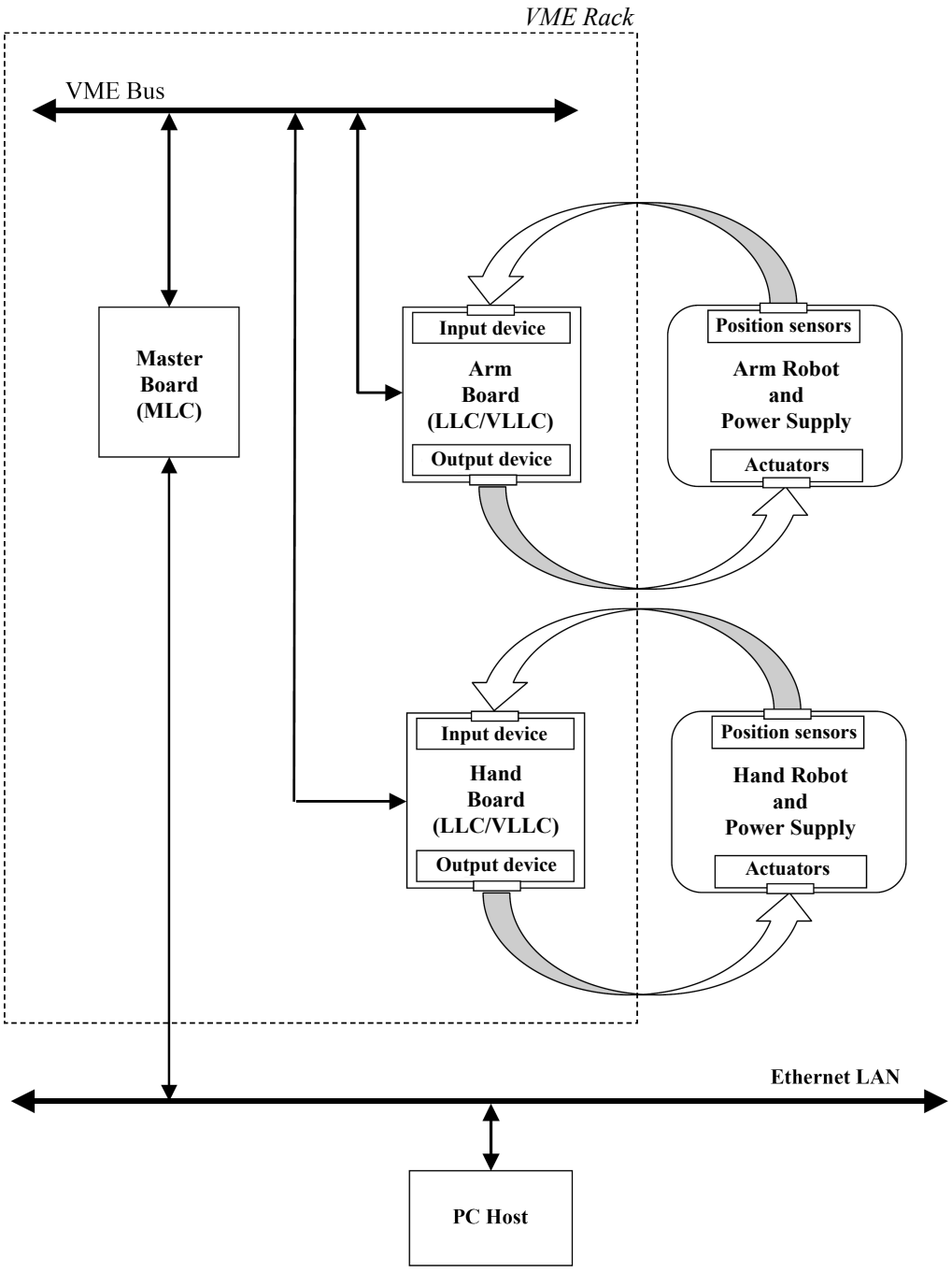

Figure 3.2: Hardware structure of the control system.

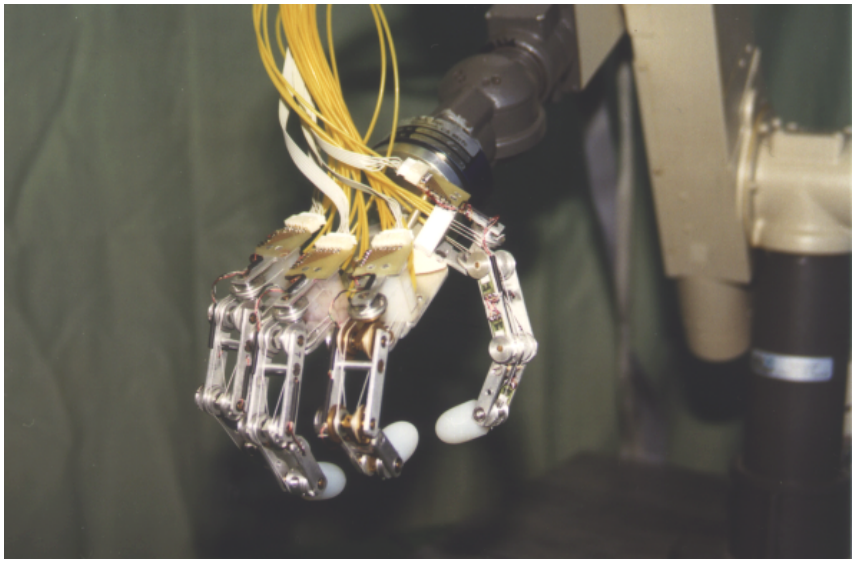

Figure 3.3: The DIST-Hand.

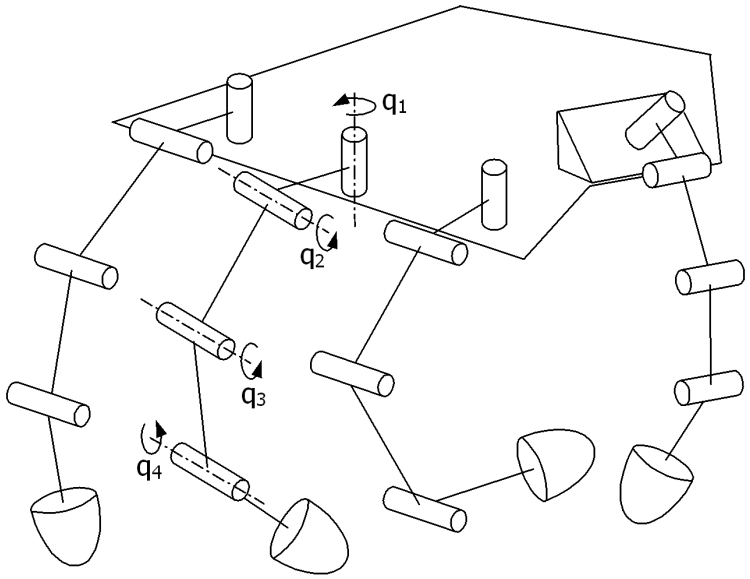

Figure 3.4: Schematics of the hand.

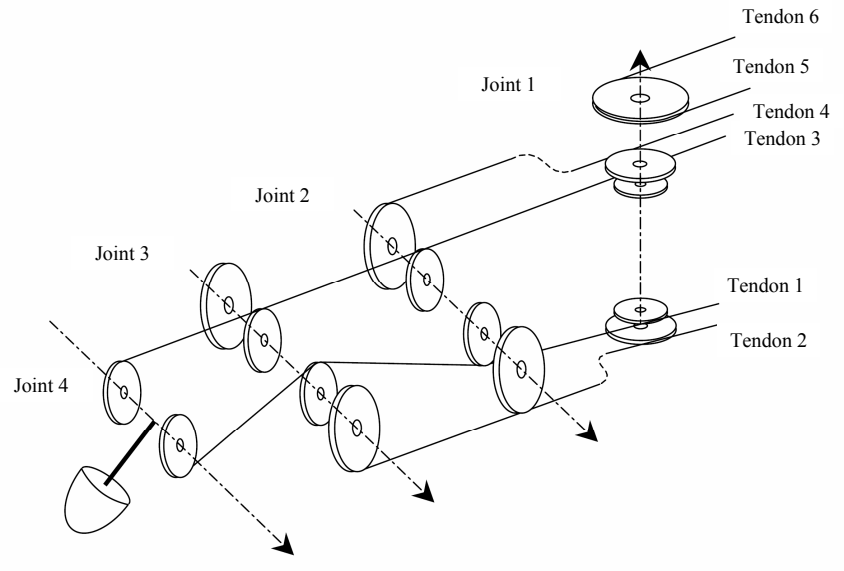

Figure 3.5: Actuation structure for a single finger.

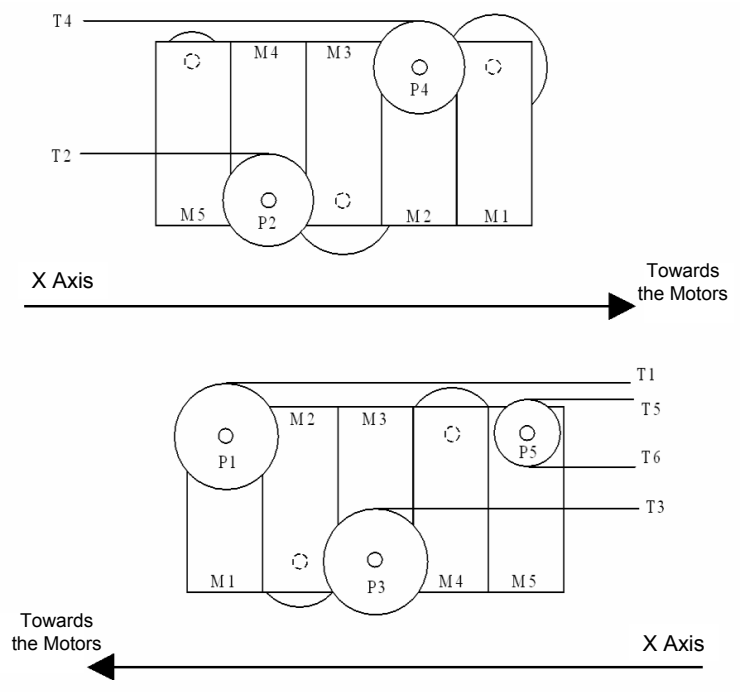

Figure 3.6: Motor box for a single finger (side views).

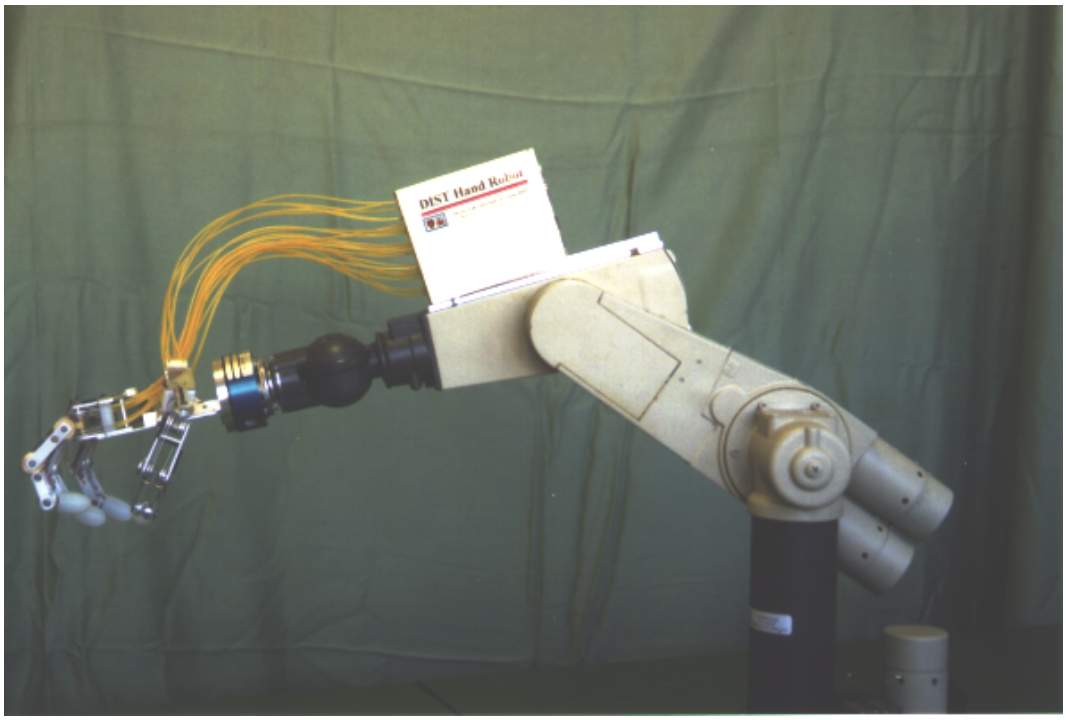

Figure 3.7: The integrated hand-arm system.

Chapter 4

Algorithmic control architecture for the integrated hand-arm system

In order to describe the algorithms used for the coordinated hand-arm control, it is necessary to do some preliminary considerations, useful for developing the global system control schemes. By referring to Fig. 4.1, we consider the problem of following a given trajectory in cartesian space with the grasped object, controlled by the hand-arm system; as we can notice, we put into evidence the different reference frames, with the respective rotation matrices, so that we can formulate the problem in terms of position and orientation errors of the different parts.

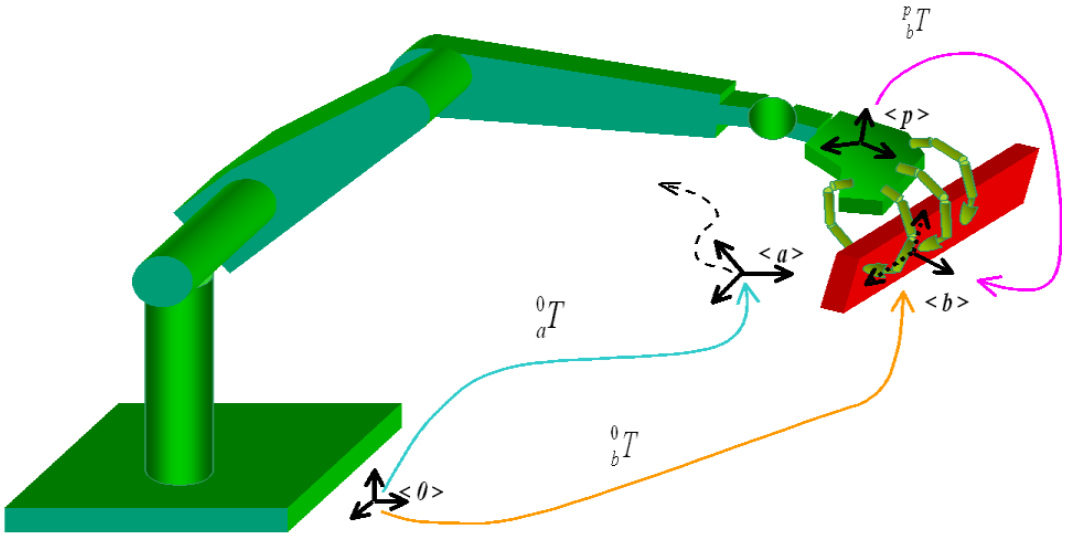

Figure 4.1: Schematic view of fhe integrated hand-arm system.

Assuming a stable grasp of the object from the hand (assumption that can be fulfilled by control laws such as the one hereafter proposed), we can see the possibility of obtaining the motion of the body in cartesian space, by using at the same time the degrees of freedom available from the hand and the arm, in a coordinated way. This is in agreement with what we expect to obtain from an anthropomorphic system, that should perform in a natural way manipulation tasks that are, at the same time, of a large extension (with respect to a given workspace) and of a general nature (with respect to the shape and dimension of the manipulated object); all of this, moreover, ensuring a good level of "dexterity" of the complete system.

Given the above mentioned goals, we can first of all develop the control algorithms by decomposing the task in two distinct parts, one related to the hand and the other to the arm, and afterwards join (and complete) the so obtained results in a single middle-level control scheme (MLC), able to satisfy the given specifications.

By considering the complete system, the problem of following a trajectory can be formulated, in analogy with what said in Chapter 2, as the problem of reaching an asymptotic coincidence of the goal frame $< a >$ with the body frame $< b >$, by zeroing the position and orientation error $\mathbf{e}$.

If we assume all quantities to be referred to an inertial base frame $< 0 >$, concerning the computation of the error $\mathbf{e}$ (given by the already developed relationships in Chapter 2) it can be immediately noticed that we need to know the *absolute* pose of the object (that is, the transformation matrix ${}^0_bT$); this quantity depends upon all of the positional variables of the system, both the ones related to the arm, and the ones of the hand, as it will be clear in what follows.

Going back to the general control algorithm, we can see from Eq. (2.29) that a purely kinematic control of the system allows an exponential convergence to zero of the cartesian error, provided that the velocity $\dot{\bar{\mathbf{q}}}_b$ is realized "at best" by the hand-arm system.

At this point, we consider in a first attempt the idea of realizing the given velocity (2.29) by splitting the controller in two distinct parts, conceptually related to the hand and the arm subsystem (from now on, we will denote with the index H all quantities related to the hand, and with A the ones related to the arm).

4.1 Description and implementation of the MLC module

In the present Section, we will describe the algorithmic structure of the Middle-Level control module (MLC); this description, and the consequent implementation, will be subdivided in a first time between the two main system parts (hand and arm), which will be the starting point for the integrated control approach of the whole system.

4.1.1 The MLC controller for the hand, and the implementation of grasping

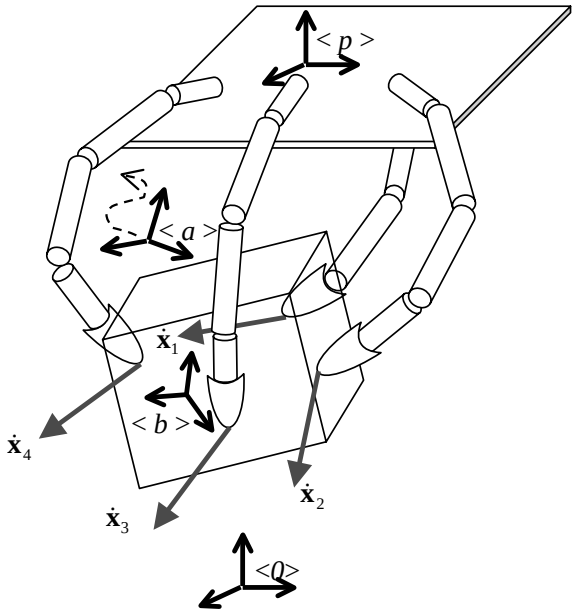

Figure 4.2: Cartesian velocities of manipulation.

In particular, let us consider the subsystem made up of the only robot hand, that is depicted in Fig. 4.2; starting with the global error $\mathbf{e}$, we can obtain a control law related to the robot hand only, as already done in Chapter 2, by simply imposing a cartesian velocity to the body of the form

$$\dot{\hat{\mathbf{q}}}_{b,H} \triangleq -\gamma_H \mathbf{e};\ \gamma_H > 0 \tag{4.1}$$

with γ_H a suitable gain factor (for sake of simplicity, a scalar); while defining the reference $\dot{\hat{\mathbf{q}}}_{b,H}$, we choose to neglect the feed-forward velocity term $\dot{\mathbf{q}}_a$ for sake of simplicity, however without loss of generality for the method.

Afterwards, by considering the relation

$$\dot{\hat{\mathbf{X}}}_H \equiv J_b \dot{\hat{\mathbf{q}}}_{b,H} \tag{4.2}$$

where $\dot{\hat{\mathbf{X}}}_H \triangleq col(\dot{\hat{\mathbf{x}}}_{1,H}, ..., \dot{\hat{\mathbf{x}}}_{h,H})$, we obtain the absolute cartesian velocities (i.e. related to $< 0 >$) $\dot{\hat{\mathbf{x}}}_{i,H}$, to be imposed to the fingers. We also notice how the law formulated in (4.2), referred to the inertial frame $< 0 >$, can be now considered in relationship with the geometric space solidal with the rigid palm, given by the frame $< p >$. In fact, a simple projection of the geometric and kinematic quantities onto

the palm frame, through the rotation matrix ${}^0_pR^T$, allows to formulate the kinematic control law in the following way:

$$\dot{\hat{\mathbf{X}}}_{H,m} = -\,{}^0_pR^{(h)T}(J_b\gamma_H\mathbf{e}) \tag{4.3}$$

where ${}^0_pR^{(h)}$ is the block-diagonal organization (of order h) of the matrix ${}^0_pR$, and after renaming $\dot{\hat{\mathbf{X}}}_{H,m}$ with the index m, in order to denote the overall velocity reference signal for the hand, for the object motion.

We notice how the rotation matrix ${}^0_pR$ depends on the only arm subsystem, and it can be computed starting from the posture of the latter, that is from the joint angles $\mathbf{q}_A$; we also notice that the subdivision of the control between the two subsystems leads us to consider such quantity, related to the "hand" entity, as an external input coming from the complimentary "arm" entity.

Still concerning the part related to the hand, we can see that the matrix J_b, which is necessary in order to transfer the cartesian velocities from the object to the fingertips, depends on the absolute pose of the object (since the vectors $\mathbf{s}_i$ appearing in its definition (2.6) depend on it), and therefore, for its computation we need the global transformation matrix ${}^0_bT$, already needed for the computation of the global error $\mathbf{e}$; at this point, we can think of obtaining this quantity by composing two distinct and independent transformations: the first one, related to the arm, given by the already mentioned ${}^0_pT$, and the second, related to the hand, given by the matrix ${}^p_bT$.

In particular concerning the latter, the estimation of the object *pose* $< b >$, referred to the palm $< p >$ is an operation that can be implemented with the method described afterwards, given the position of fingertips on the object, and projected onto the palm ${}^p\mathbf{s}_i(\mathbf{q}_{i,H}); i = 1, 2, ..., h$; this computation will be therefore part of the hand subsystem, together with the *grasping* controller hereafter described.

As far as the object grasp is concerned, we need to develop a control law that, in a middle-level context, should be able to guarantee the persistence of contact in the predefined points, during the whole coordinated manipulation task.

Even though the force control laws developed in Chapter 2 (and, in particular, the closed-loop law given by (2.107)) have been shown to be adequate for a generic multi-finger manipulation system, in absence of suitable force sensors on the fingertips, it is necessary to develop a control methodology that does not make use of force feedback, but that is nevertheless able to obtain, in principle, a stable grasp of the manipulated object.

On the basis of some considerations, in principle of a purely geometric nature (but in fact dynamically valid), we can formulate a control strategy for the grasping as the one hereafter described.

Consider Fig. 4.3, where the hand system is represented; in order to keep the fingertips in contact with the object, the described idea simply consists in realizing a cartesian control of mutual distances, where we impose, for each fingertip, to keep a fixed distance with each other, in a closed-loop scheme. In this way, in fact, we assign kinematic constraints to the hand system, such as to avoid the drift of each

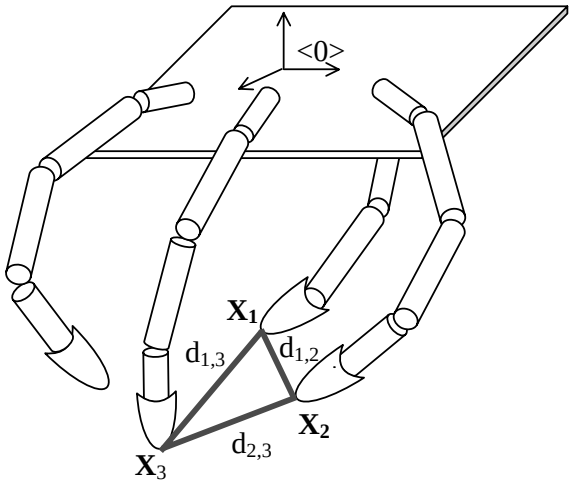

Figure 4.3: Control of fingertip distances.

part, and maintain the "grasp configuration" also in absence of the object, as it will be clear from the algorithmic description.

If we assume of having selected the grasp configuration for the object (given by the vectors $\sigma_i, i = 1, 2, ..., h$), we can therefore assign the mutual distance references, by computing the following quantities (constant in time)

$$d_{i,j} \triangleq \|\sigma_j - \sigma_i\| \, ; i \neq j \tag{4.4}$$

Given these quantities for each pair of fingertips, we will be therefore able to define a distance error vector, in the following way: let d be the reference distance between any two fingertips, and define with

$$\mathbf{e}_d \triangleq \mathbf{v} - d\mathbf{w} \tag{4.5}$$

the error, where $\mathbf{v} \triangleq (\mathbf{x}_j - \mathbf{x}_i)$ is the vector joining two fingertips, and $\mathbf{w} \triangleq \frac{\mathbf{v}}{\|\mathbf{v}\|}$ the respective unit versor. By introducing a suitable Lyapunov function, given by

$$V_d \triangleq \frac{1}{2} \|\mathbf{e}_d\|^2 = \frac{1}{2} \mathbf{e}_d^T \mathbf{e}_d \tag{4.6}$$

we obtain, by taking its temporal derivative,

$$\dot{V}_d = \mathbf{e}_d^T \dot{\mathbf{e}}_d = (\mathbf{v}^T - d\mathbf{w}^T)(\dot{\mathbf{v}} - d\dot{\mathbf{w}}) \tag{4.7}$$

Since $\mathbf{v}, \mathbf{w} \perp \dot{\mathbf{w}}$, where $\mathbf{w}$ is a unit vector, we will have $\dot{V}_d = \mathbf{e}_d^T \dot{\mathbf{v}}$; now, by choosing

$$\dot{\mathbf{v}} \equiv -\gamma_d \mathbf{e}_d; \gamma_d > 0 \tag{4.8}$$

we obtain, as desired, $\dot{V}_d < 0$.

This, in turn, requires that $\dot{\mathbf{x}}_i = \gamma_d(\mathbf{v} - d\mathbf{w})$, being $\mathbf{x}_j$ a reference point for fingertip i. Finally, by repeating the same reasoning for all fingertips $\mathbf{x}_j; j \neq i$, and summing up all kinematic reference signals for fingertip i, we obtain the following control law:

$$\dot{\hat{\mathbf{x}}}_{i,d} \triangleq \gamma_d \sum_{j \neq i} (\mathbf{v}_{i,j} - d_{i,j}\mathbf{w}_{i,j}); i = 1, 2, ..., h \tag{4.9}$$

that results in a cartesian velocity reference, as it is the one computed for manipulating the object in (4.3); finally, we can collect in a single vector $\dot{\hat{\mathbf{X}}}_{H,d} \triangleq col(\dot{\hat{\mathbf{x}}}_{1,d}, \dot{\hat{\mathbf{x}}}_{2,d}, ..., \dot{\hat{\mathbf{x}}}_{h,d})$ the overall reference signal for keeping mutual distances.

At this point, by superimposing to the motion control of the object given in (4.3) the distance control of (4.9), we obtain the overall velocity reference signal given by

$$\dot{\hat{\mathbf{X}}}_H \triangleq \dot{\hat{\mathbf{X}}}_{H,m} + \dot{\hat{\mathbf{X}}}_{H,d} \tag{4.10}$$

The so obtained kinematic control law (4.10), constitutes therefore the hand subsystem of the middle-level cartesian control, to which we will now add the complimentary arm subsystem. In particular, we notice how the resulting signal (4.10) is a pure cartesian velocity reference system for the underlying level (LLC); in fact, since the LLC modules accept both position and velocity signals (in cartesian space), we need to supply an additional position signal, that can be obtained in the following way.

Consider the position

$$\hat{\mathbf{x}}_i \triangleq {}^p_b T \overline{\sigma}_i; \; i = 1, 2, ..., h \tag{4.11}$$

(expressed in *homogeneous* coordinates), to be intended as the instantaneous position of the $i - th$ contact point (estimated) on the object; if we consider the latter as a further reference for the corresponding fingertip, we trivially obtain with (4.11) the additional position signal for the fingertip, to be sent to the respective LLC module.

In Fig. 4.4 we show the middle-level module related to the robot hand, by letting open, for the moment, the arm subsection; as we can notice, the different signals that are present in the system correspond, in fact, to what up to now developed for the control of manipulation through the only hand; in Fig. 4.4 we also indicate the LLC modules for the fingers, with the respective input and output signals, that will be described in the next Section. For sake of clarity, in the diagram of Fig. 4.4 we do not put into evidence the VLLC modules of the fingers, that are obviously connected with the respective LLC modules (as it is instead shown in Fig. 3.1); the description and particularization of the LLC/VLLC modules will be given later on in more detail. In the next paragraph, we will now describe an algorithm related to the estimation of position and orientation of the rigid body, using the informations related to the posture of the hand system.

4.1.2 Estimating the object pose

In order to compute the rigid object pose $< b >$ with respect to the base frame $< 0 >$, actually coincident with the palm $< p >$, we propose a simple estimation

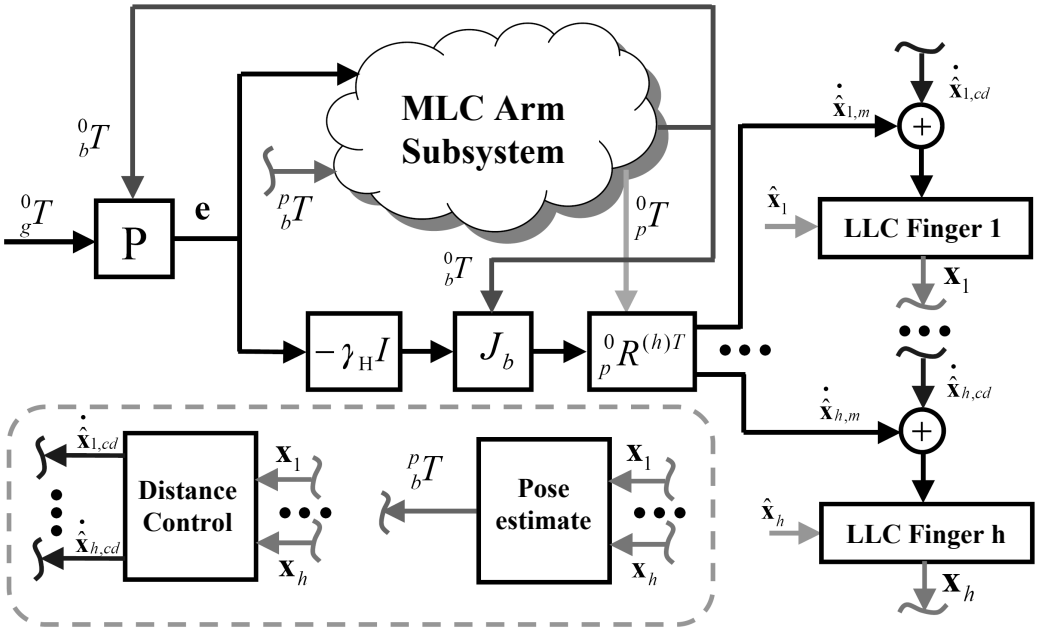

Figure 4.4: The MLC subsystem related to the robot hand.

method, based on the knowledge of contact points on the object that, by assumption, are constant in time (as in the contact model of Eq. (2.11).

Referring to Fig. 4.5, where $< 0 > \equiv < p >$, we wish to determine the position ${}^{p}\mathbf{x}$ and orientation ${}^{p}_{b}R$ of frame $< b >$. First of all, we need to make two preliminary assumptions: a) we assume to know, on-line, the vectors $\{{}^{p}\mathbf{P}_1, {}^{p}\mathbf{P}_2, ..., {}^{p}\mathbf{P}_h\}$, which define the instantaneous position of the contact points, with respect to frame $< p >$; this information is simply given by the position of the fingertips in cartesian space, that can be therefore easily computed from the measurement of joint angles $\mathbf{q}_i$ for each finger; moreover, b) we know *off-line* the constant vectors $\left\{{}^{b}\mathbf{P}_1, {}^{b}\mathbf{P}_2, ..., {}^{b}\mathbf{P}_h\right\} \triangleq \{\sigma_1, \sigma_2, ..., \sigma_h\}$, that give the position of the same contact points with respect to frame $< b >$; this information is given by the prior choice of the grasp configuration for the object.

We immediately notice how, in order to perform the estimation, it may be sufficient using only 3 of the h contact points, for example $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$.

Then, we define an intermediate frame $< b' >$ which is solidal to the rigid body, whose origin is located in one of the contact points (e.g. $\mathbf{P}_1$), and whose orthogonal axes are hereafter specified.

Consider the plane given by the 3 contact points $\{\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3\}$; let $\mathbf{n}$ be the normal versor to the plane, and $\mathbf{t}$ the direction $(\mathbf{P}_2 - \mathbf{P}_1)$, lying on the plane; let finally $\mathbf{w}$ be the versor that completes the other two to a right-hand orthonormal frame;

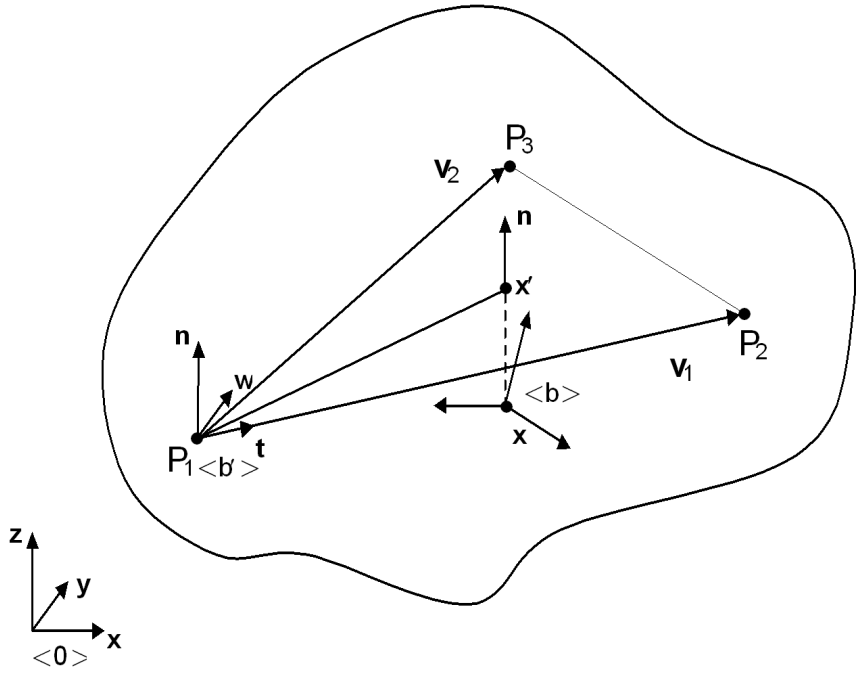

Figure 4.5: Definition of cartesian frames for pose estimation.

therefore we have:

$$< b^{'} > \triangleq \{\mathbf{t}, \mathbf{w}, \mathbf{n}\} \tag{4.12}$$

Moreover, define

$$\mathbf{v}_1 \triangleq (\mathbf{P}_2 - \mathbf{P}_1)\,;\ \mathbf{v}_2 \triangleq (\mathbf{P}_3 - \mathbf{P}_1) \tag{4.13}$$

Then, we can recover the orientation matrix ${}^{b'}_{b}R$, by taking into account that

$$\begin{cases} {}^{b}\mathbf{n} \triangleq \frac{{}^{b}(\mathbf{v}_1 \wedge \mathbf{v}_2)}{\|\mathbf{v}_1 \wedge \mathbf{v}_2\|} \\ {}^{b}\mathbf{t} \triangleq \frac{{}^{b}\mathbf{v}_1}{\|\mathbf{v}_1\|} \\ {}^{b}\mathbf{w} \triangleq {}^{b}(\mathbf{n} \wedge \mathbf{t}) \end{cases} \tag{4.14}$$

indeed, we simply have

$${}^{b}_{b'}R = \left[{}^{b}\mathbf{t} \left| {}^{b}\mathbf{w} \right| {}^{b}\mathbf{n} \right]^{T} \tag{4.15}$$

Afterwards, we define with $\mathbf{x}^{'}$ the point obtained by projecting the origin $\mathbf{x}$ of frame $< b >$ onto the plane $\{\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3\}$. We notice how the vector $\left(\mathbf{x}^{'} - \mathbf{P}_1\right)$, lying on the plane, can also be expressed as

$$\left(\mathbf{x}^{'} - \mathbf{P}_1\right) = -\sigma_1 + k\mathbf{n} \tag{4.16}$$

being

$$\begin{cases} \sigma_1 \triangleq (\mathbf{P}_1 - \mathbf{x}) \\ k \triangleq (\mathbf{P}_1 - \mathbf{x})^T \mathbf{n} = \sigma_1^T \mathbf{n} \end{cases} \tag{4.17}$$

But, since the plane onto which $\left(\mathbf{x}^{'} - \mathbf{P}_1\right)$ is lying, is also generated by the versors $\{\mathbf{t}, \mathbf{w}\}$, we have

$$\left(\mathbf{x}^{'} - \mathbf{P}_1\right) = a\mathbf{t} + b\mathbf{w} \tag{4.18}$$

being

$$\begin{cases} a = (-\sigma_1 + k\mathbf{n})\mathbf{t} \\ b = (-\sigma_1 + k\mathbf{n})\mathbf{w} \end{cases} \tag{4.19}$$

Therefore, the coefficients k, a, b can be obtained from the available vectors σ_1,${}^b\mathbf{n}$,${}^b\mathbf{t}$,${}^b\mathbf{w}$ (already computed for frame $< b' >$), and, finally, we can obtain the position of ${}^0\mathbf{x}$, which is the origin of $< b >$:

$${}^p\mathbf{x} = {}^p\mathbf{x}^{'} - k{}^p\mathbf{n} = {}^p\mathbf{P}_1 + a{}^p\mathbf{t} + b{}^p\mathbf{w} - k{}^p\mathbf{n} \tag{4.20}$$

In a fully similar way to the computation of matrix ${}^b_{b'}R$, we can also compute the orientation matrix ${}^p_{b'}R$, being:

$$\begin{cases} {}^p\mathbf{n} \triangleq \frac{{}^p(\mathbf{v}_1 \wedge \mathbf{v}_2)}{\|\mathbf{v}_1 \wedge \mathbf{v}_2\|} \\ {}^p\mathbf{t} \triangleq \frac{{}^p\mathbf{v}_1}{\|\mathbf{v}_1\|} \\ {}^p\mathbf{w} \triangleq {}^p(\mathbf{n} \wedge \mathbf{t}) \end{cases} \tag{4.21}$$

from which we have

$${}^p_bR = {}^p_{b'}R\,{}^b_{b'}R^T \tag{4.22}$$

By concluding, once we compute the matrix ${}^p_bR$ through (4.22) and (4.21), and the point ${}^p\mathbf{x}$ through (4.20), we can compute the desired transformation matrix

$${}^p_bT = \begin{bmatrix} {}^p_bR & {}^p\mathbf{x} \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.23}$$

4.1.3 The MLC control of the arm

We will now describe the MLC sub-module related to the arm, starting from a control approach based on the cartesian task of position-orientation, as already done for the hand sub-system.

From the functional point of view, we already said (see Fig. 3.1) that the MLC module implements a purely cartesian control of the whole system, and in particular, referring to the arm subsystem, we have an algorithmic structure that allows the computation of cartesian velocities related to the end-effector frame of the arm, by exchanging the related signals (*feedback* and *feed-forward*).

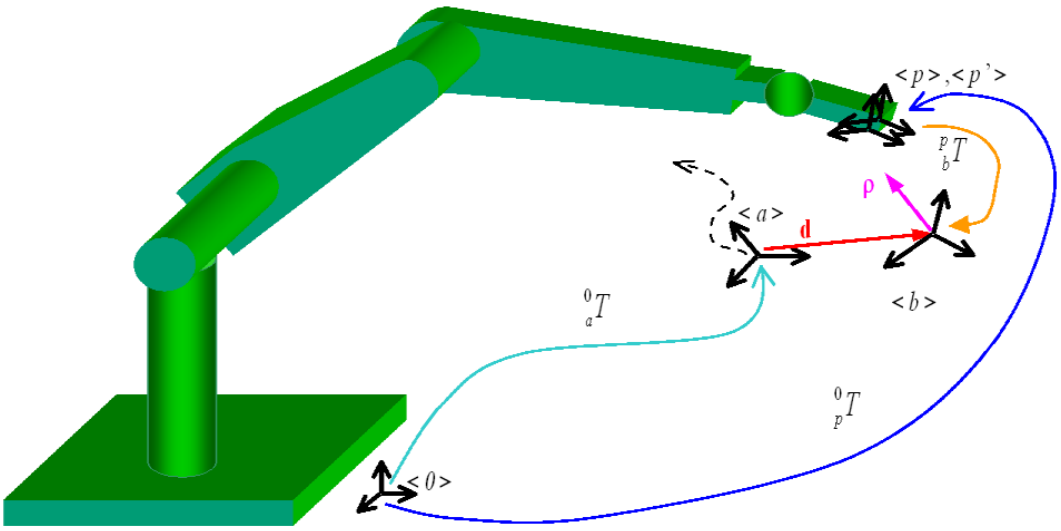

Figure 4.6: The arm sub-system.

Starting again from the computation of the overall error $\mathbf{e}$ of position and orientation with respect to the base $< 0 >$, we can say, as already done for the hand, that we need to provide a suitable *feed-forward* cartesian velocity signal, given by the similar formula

$$\hat{\dot{\mathbf{q}}}_{b,A} = -\gamma_A \mathbf{e};\ \gamma_A > 0 \tag{4.24}$$

where γ_A is another positive gain factor, and where the feed-forward velocity term $\dot{\mathbf{q}}_a$ has again been neglected.

Now, assume to impose a linear and angular velocity $\hat{\dot{\mathbf{q}}}_{b,A}$ to the body, by using the only arm subsystem, and keeping the hand grasp fixed to a given posture; this will clearly imply to have already a given *constant* position of frame $< b >$ with respect to the palm $< p >$. But, in that case, imposing the cartesian velocity $\hat{\dot{\mathbf{q}}}_{b,A}$ to the body $< b >$ is equivalent to impose a cartesian velocity, denoted with $\hat{\dot{\mathbf{X}}}_p$, to the hand palm $< p >$, which is solidal with the last link of the arm; this velocity will be given by

$$\hat{\dot{\mathbf{X}}}_p = S_{p,b} \hat{\dot{\mathbf{q}}}_{b,A} \tag{4.25}$$

being $\hat{\dot{\mathbf{X}}}_p \triangleq col(\hat{\mathbf{v}}_p, \hat{\omega}_p)$, and having defined with

$$S_{p,b} \triangleq \begin{bmatrix} I_3 & [\mathbf{r}_{p,b} \wedge] \\ 0 & I_3 \end{bmatrix};\quad \mathbf{r}_{p,b} \triangleq (\mathbf{O}_b - \mathbf{O}_p) \tag{4.26}$$

the matrix that "transfers" cartesian velocities from a frame to another, on the same link (in this case, the end link of the arm); moreover, we assume to have chosen a

suitable value for the proportional gain γ_A. In order to realize the desired velocity, it is necessary to impose the signal (4.25) to the arm, as a feed-forward velocity signal, to be sent to the respective LLC module.

For reasons that will become clear in the following, instead of sending a pure velocity command, we chose, at MLC level, to compute an equivalent positional (linear and angular) signal, obtained through an *integration* procedure of (4.25) with suitable initial conditions, and to send both the positional and the feed-forward signals to the LLC module.

This idea leads to the definition, as positional reference for the arm system, an auxiliary "goal" frame $< p' >$ given by the transformation matrix

$$ {}^{0}_{p'}T(t) \triangleq \left[\begin{array}{c|c} {}^{0}_{p'}R & \mathbf{x}_{p'} \\ \hline \mathbf{0} & 1 \end{array}\right] \tag{4.27} $$

such that:

$$ \begin{cases} {}^{0}_{p'}T(0) \equiv {}^{0}_{p}T(0) \\ \\ \dot{\mathbf{X}}_{p'} \equiv \dot{\widehat{\mathbf{X}}}_{p} \end{cases} \tag{4.28} $$

In other words, we defined $< p' > \equiv < p >$ at time $(t = 0)$, and set the velocity $\dot{\mathbf{X}}_{p'}$ of frame $< p' >$ given by $(\mathbf{v}_{p'}, \omega_{p'}) \equiv (\widehat{\mathbf{v}}_p, \widehat{\omega}_p)$, previously computed.

In order to define the temporal evolution of the goal $< p' >$, the resulting signal is simply given by numerical *integration* formulae, analogous to the ones given at the beginning of Chapter 2:

$$ \begin{cases} \dot{\mathbf{x}}_{p'} = \widehat{\mathbf{v}}_p \\ \\ {}^{0}_{p'}\dot{R} = [\widehat{\omega}_p \wedge] \; {}^{0}_{p'}R \end{cases} \tag{4.29} $$

By resuming, as a consequence of the "integration" process (4.29) we obtained the definition of an auxiliary frame $< p' >$, initially assumed to be coincident with $< p >$, and subsequently considered as a cartesian position reference, of which we know (and use) also the information related to its *velocity* $\dot{\mathbf{X}}_{p'}$.

The arm MLC subsystem is finally depicted in Fig. 4.7, where we put into evidence the integratio process with a block "$\frac{1}{s}$", with a slight notation abuse (since it is not an integration process in the common sense of the term, as it is clear from (4.29)); to the arm LLC module, as it is evident from the picture, will be sent a complete position and velocity signal of the *goal* frame $< p' >$, that must be realized "at best" by the control system LLC/VLLC, as it will be explained in the related Section.

We also notice, concerning the arm, that under the assumption of a perfect temporal coincidence of the two frames, i.e. $< p > \equiv < p' >$, we also have $\dot{\mathbf{X}}_p \equiv \dot{\mathbf{X}}_{p'}$, and therefore, $\dot{\mathbf{q}}_{b,A} \equiv \dot{\widehat{\mathbf{q}}}_{b,A} = -\gamma_A \mathbf{e}$.

4.1.4 The hand-arm integration

At this point, in order to complete the description of the MLC control level, we can look back at the results obtained separately in the two previous Sections, so that

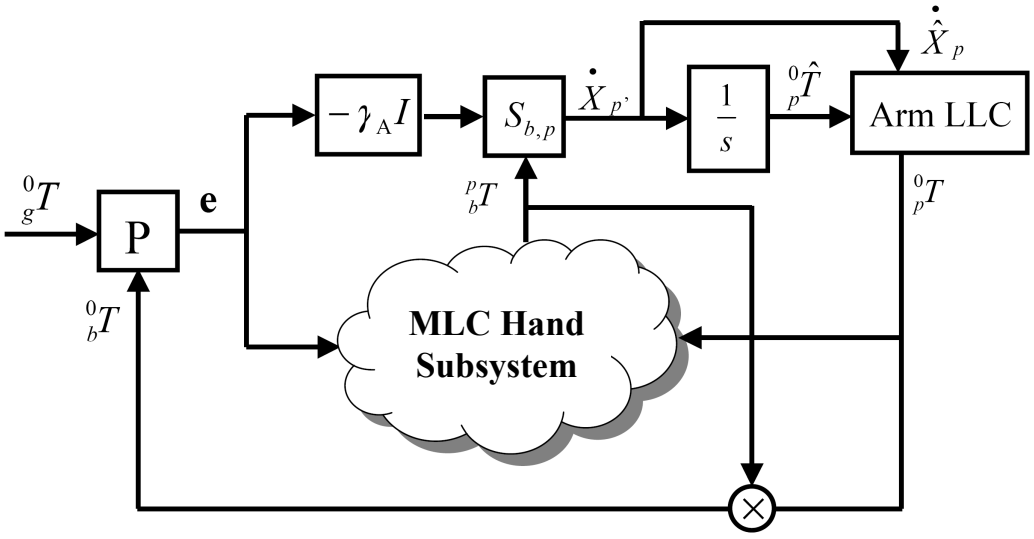

Figure 4.7: The arm MLC subsystem.

the meaning of the integration of the complete hand-arm system will be evident; in particular, the meaningful quantities for tracking the position reference with the given object are provided by the two reference signals, of cartesian position and velocity, defined by the equations (4.10), (4.11) concerning the hand part of the MLC, and by (4.25), (4.27), concerning the arm subsystem.

In the previous Sections we put into evidence, therefore, how the general task can be completed by considering the two robotic entities separately; that means:

- we developed a kinematic control law (cartesian) related to the only hand part, by considering the whole arm subsystem as nothing more that a fixed support, to the end-effector of which (i.e. the rigid hand palm) is connected the other subsystem, and acting as a single entity which is controlled for executing the task (Eq. (4.10), (4.11));
- afterwards, we obtained a control law related to the arm subsystem, also such as to perform the given task, without making use of the hand subsystem, which is considered as a rigid gripper, able to keep a fixed position of the object with respect to its own support (the palm), as it is evident from Eq. (4.25), (4.27)

Since the two references contribute separately and simultaneously to the execution of the given task (i.e. zeroing the global position and orientation error $\mathbf{e}$ of the object), it is clear that we can proceed to *superimpose* the two control laws, as it is also evident from Fig. (4.7) and (4.4), if considered as a whole; hereafter, we resume the definitions related to the reference frames, plus the *feed-forward* signals

generated by the complete MLC cartesian control system; in particular, we have:

$$\begin{cases} {}^{0}_{p}\widehat{T} \triangleq {}^{0}_{p'}T \\ \\ \dot{\hat{\mathbf{X}}}_p \triangleq -\gamma_A S_{p,b}\mathbf{e} \equiv \dot{\mathbf{X}}_{p'} \end{cases} \tag{4.30}$$

concerning the arm (see again Eq. (4.29) for computing ${}^{0}_{p'}T$), and

$$\begin{cases} \widehat{\mathbf{x}}_{H,i} \triangleq {}^{p}_{b}T\overline{\sigma}_i; i = 1, 2, ..., h \\ \\ \dot{\hat{\mathbf{X}}}_H \equiv -\gamma_H J_b \mathbf{e} + \dot{\hat{\mathbf{X}}}_{H,d} \end{cases} \tag{4.31}$$

concerning the hand, being $\dot{\hat{\mathbf{X}}}_{H,d}$ the grasp correction term (mutual distances) given by (4.9).

Now we notice that the term $S_{p,b}$ appearing inside (4.30), that allows to transfer the velocity references for the object $< b >$ onto the palm $< p >$, is not anymore a constant matrix, but a *time-varying* term according to the instantaneous *pose* (estimated) that the object assumes relative to the palm, due to the manipulation operated by the hand subsystem; in order to estimate the object pose, we use the method previously described.

Moreover, we notice that the correction term $\dot{\hat{\mathbf{X}}}_{H,d}$ inside (4.31) can be considered a *transient* term of a small entity, since we suppose that the grasp configuration is constant, during all of the manipulation phase.

By concluding, we finally notice how all of the position/velocity references that the system assumes, are *jointly* leading to the convergence to zero of the global position/orientation error $\mathbf{e}$, according to

$$\dot{\mathbf{q}}_b = \dot{\hat{\mathbf{q}}}_{b,H} + \dot{\hat{\mathbf{q}}}_{b,A} = -(\gamma_A + \gamma_H)\mathbf{e} \tag{4.32}$$

A suitable tuning of the gain parameters γ_H, γ_A allows to assign a higher or lower dexterity to one subsystem with respect to the other; their value and evolution in time may play a crucial role while performing the desired task, leaving a wide choice for choosing the operation modality of the integrated hand-arm system.

At this point, we see how inside the control law for the arm, given by (4.30), together with the integration process (4.29), we chose to use the additional reference $< p' >$ since, in absence of operation of the arm (i.e. for a gain value $\gamma_A = 0$), it is necessary to have *anyway* a closed-loop position control scheme (where the reference position is in this case constant), such as to react to possible disturbances or positional drifts acting on the system; this means that the signal coming from the integration process ${}^{0}_{p'}T$ has to be kept by a control loop that will take place, as we will see in the following, in the respective LLC module. As far as the hand is concerned, due to the negligible masses, the presence of drifts can instead be neglected, and the pure velocity control, with references given by the second of (4.31), is sufficient to ensure the system stability, also in absence of the gain γ_H (notice how inside (4.31), in fact, the position reference terms are *redundant* quantities for the system, but set only for sake of completeness of the whole position/velocity control system, which will be realized by the respective LLC modules).

4.2 Description of LLC modules

We will consider now the LLC control modules, related to the different system parts which, in the present case, are given by the arm subsystem, and the individual fingers of the hand subsystem.

As it will be clear from the next description, the LLC modules are entities which partly depend upon the kinematic structure of the respective robotic subsystem (that is, partly *robot-dependent*), but, in their general algorithmic structure, they are built in a uniform way, according to a common logical scheme; such a scheme can, in fact, be replicated for each additional or substitutive subsystem, that can be present in the general control context.

Let us first of all consider the arm subsystem, and the related LLC control module, depicted in Fig. 4.7 within the context of the Middle Level controller, and refer to the general diagram of Fig. 3.1 at the beginning of the Chapter, with the respective input and output signals shown.

In this context, as already mentioned, the task of the LLC module is to ensure the tracking of a cartesian reference signal for the *end-effector* of the subsystem, in the most general case with position and orientation, plus the additional feed-forward velocity signals, also given in cartesian space, as a linear and angular velocity. Such task has to be translated and performed within the context of a purely kinematic control, i.e. through the specification of suitable *joint velocity* reference signals for the given subsystem, whose realization will be deferred to the VLLC module at the lowest level.

When considering the tracking of a goal frame, generally denoted by $< g >$, by means of the end-effector, whose frame will be here denoted by $< e >$, we can define a generalized error vector $\mathbf{e} \triangleq col(\mathbf{d}, \rho)$ (for sake of clarity, dropping for the moment the indices), as already done previously concerning the overall manipulation task; moreover, assume to have the *feed-forward* cartesian velocity signal $\dot{\mathbf{q}}_g \triangleq col(\omega_g, \mathbf{v}_g)$; by defining a suitable Lyapunov function, in a fully analogous way to our previous developments, we can define cartesian control law

$$\dot{\overline{\mathbf{X}}} = -\gamma \mathbf{e} + \dot{\mathbf{q}}_g; \gamma > 0 \tag{4.33}$$

having indicated with γ a generic gain factor, related to the arm LLC module, and with $\dot{\overline{\mathbf{X}}}$ the cartesian velocity reference for the end-effector.

Now, we consider the general kinematic law that relates the *joint* velocities of a general robotic system with the *linear* and *angular* velocity in cartesian space of the end-effector solidal frame $< e >$. As it is well known, such a relationship is given by

$$^0\dot{\mathbf{X}}_{e/0} = \ ^0J_{e/0}(\mathbf{q})\dot{\mathbf{q}} \tag{4.34}$$

being $^0\dot{\mathbf{X}}_{e/0} \triangleq col(^0\omega_{e/0}, ^0\mathbf{v}_{e/0})$ the generalized end-effector velocity $< e >$ with respect to the base frame $< 0 >$, $^0J_{e/0}(\mathbf{q})$ the related Jacobian matrix, and where all of the quantities are projected to the $< 0 >$ frame.

When talking about the LLC modules, it is now important to examime the law (4.34), by considering the possibility of realizing the cartesian velocity reference $\dot{\mathbf{X}}$ in (4.33) through a kinematic (therefore always in the joint velocities) control, while keeping into account the presence of possible singular configurations for the arm(obviously extended, subsequently, to the hand fingers).

4.2.1 Handling the kinematic singularities for the LLC control

Standard manipulators are often designed with no more thab 6 *dof*, which is the minimum necessary number for performing complete position and orientation tasks in cartesian space. Although 6 dof are sufficient for a conventional design, this also may limit the potential application field for the manipulator.

When a robot follows a given trajectory with the end-effector and needs, for example, to avoid collisions with an obstacle, more than 6 *dof* are required. Moreover, due to the non-linearity of the kinematics, there are particular configurations where the robot kinematics degenerates, and becomes in fact equivalent to less than 6 *dof*. These configurations are called *singular points* (or *singularities*).

Inverse kinematics is one of the necessary functionalities for the control system that so far we have been talking about. We define, in general, the position and orientation of the end-effector $\mathbf{X}$, described as a nonlinear function of the joint variables $\mathbf{q} \in \Re^n$, as:

$$\mathbf{X} = f(\mathbf{q}) \tag{4.35}$$

The differential relationship is described by the following linear equation in the Jacobian matrix $J(\mathbf{q}) \triangleq \frac{\partial f}{\partial \mathbf{q}}$:

$$\dot{\mathbf{X}} = J(\mathbf{q})\dot{\mathbf{q}} \tag{4.36}$$

In order to compute the inverse kinematics we solve, whenever possible, Eq. (4.36) w.r.t. $\dot{\mathbf{q}}$. When $\mathbf{X}$ and $\mathbf{q}$ have the same dimension, computing the inverse kinematics is the same as finding the inverse of both the nonlinear function (4.35) and the linear one (4.36). In both cases, the conditions for existence of the inverse function are $\det J(\mathbf{q}) \neq 0$. Here $\mathbf{q}$ is called a singular point, if $\det J(\mathbf{q}) = 0$.

If the number of joints is bigger than the dimension of the manipulation variable, a manipulator is called *redundant*. Such a manipulator is characterized by the fact that there exist infinite solutions for the inverse kinematics.

For kinematically redundant manipulators, singular points are defined as the $\mathbf{q}$ that make $J(\mathbf{q})$ rank-deficient.

Singular points are not only found at the borders of the workspace, but also in the middle, and they limit the trajectories of the end-effector that the manipulator could follow.

The most serious problem concerning singularities is not given by the singular points themselves, which have a zero volume in the workspace, but rather by all postures in the *neighborhood* (where the definition for "neighborhood" will be made clear in the following) of singular points, within a finite volume. Near to the singularities, to a small change in $\mathbf{X}$ it corresponds a large variation in $\mathbf{q}$, therefore

producing a large joint velocity signal that cannot be realized by the limited actuator forces and velocities.

In order to identify the velocities to be assigned as joint references, in the most general case, we can try do compute the ones with minimum norm $\|\dot{\mathbf{q}}\|$ with respect to a suitable "weight" matrix W, hereafter defined.

For the moment, let us assume that the J matrix is full rank, for a given configuration $\mathbf{q}$; therefore, we wish to solve the following constrained minimization problem:

$$\begin{cases} \min_{\dot{\mathbf{q}}} \|\dot{\mathbf{q}}\|_W^2 \\ \dot{\mathbf{X}} = J\dot{\mathbf{q}} \end{cases} \tag{4.37}$$

being $\|\dot{\mathbf{q}}\|_W^2 \triangleq (\dot{\mathbf{q}}^T W \dot{\mathbf{q}})$ the minimum (squared) norm of the vector $\dot{\mathbf{q}}$, *weighted* by the matrix W. Since W is positive definite, we can compute its *square root* Σ, through which we can re-write the problem (4.37) as

$$\begin{cases} \min_{\dot{\mathbf{q}}} (\dot{\mathbf{q}}^T \Sigma^T \Sigma \dot{\mathbf{q}}) \\ \dot{\mathbf{X}} = J\dot{\mathbf{q}} \end{cases} \tag{4.38}$$

Now, by substituting the variables $\dot{\mathbf{z}} \triangleq \Sigma\dot{\mathbf{q}}$, from which it follows $\dot{\mathbf{q}} \triangleq \Sigma^{-1}\dot{\mathbf{z}}$, and by indicating with $\overline{J} \triangleq J\Sigma^{-1}$ the "transformed" Jacobian matrix, we can reformulate the problem (4.38) in the standard least-squares form

$$\begin{cases} \min_{\dot{\mathbf{z}}} \|\dot{\mathbf{z}}\|^2 \\ \dot{\mathbf{X}} = \overline{J}\dot{\mathbf{z}} \end{cases} \tag{4.39}$$

that, as we can verify by using the Lagrange multipliers, it admits a solution given by

$$\dot{\mathbf{z}} = \overline{J}^{\#}\dot{\mathbf{X}} \triangleq \overline{J}^T(\overline{J}\overline{J}^T)^{-1}\dot{\mathbf{X}} \tag{4.40}$$

being $\overline{J}^{\#}$ the *pseudo-inverse* of the matrix $\overline{J}$; after simple algebraic manipulation, we have

$$\dot{\mathbf{z}} = (\Sigma^{-1})^T J^T (JW^{-1}J^T)^{-1}\dot{\mathbf{X}} \tag{4.41}$$

By reformulating the obtained result in terms of $\dot{\mathbf{q}}$, we find the solution of the original problem

$$\dot{\mathbf{q}} = W^{-1}J^T(JW^{-1}J^T)^{-1}\dot{\mathbf{X}} \tag{4.42}$$

While inverting the matrix $\overline{J}$, in the general case we can face the problem of losing rank, and, in correspondence of postures $\mathbf{q}$ near to singular points, very high joint reference velocities will be generated (tending to infinity into the singularities). If we desire to obtain anyway limited velocity references, also in presence of these situations, it is then possible to add a *regularizing* factor, given by the following formula

$$\overline{J}^* \triangleq \overline{J}^T(\overline{J}\,\overline{J}^T + K)^{-1};\ K > 0 \tag{4.43}$$

where K is a suitable matrix (positive semi-definite) that will be subsequently characterized, and where $\overline{J}^*$ is the pseudo-inverse, regularized by K, of the (weighted)

Jacobian matrix $\overline{J}$. By explicitly writing $\overline{J}$ in terms of the initial J matrix, we obtain

$$\overline{J}^* = (\Sigma^{-1})^T J^T (JW^{-1}J^T + K)^{-1} \tag{4.44}$$

The implementation of the computing procedure for $\overline{J}^*$ in (4.43) requires the choice of K, possibly variable with the configuration $\mathbf{q}$, that, as it will be clear in the following, allows to obtain limited joint velocities (and for sure finite) also in the neighborhood of singular configurations.

Let us consider therefore the singular values decomposition (SVD) of a generic matrix $\overline{J}$, in our case with dimension $(6 \times n)$, through which we can always write the matrix $\overline{J}$ as the product of 3 matrices

$$\overline{J} = U \cdot S \cdot V^T \tag{4.45}$$

where:

- S is a $(6 \times n)$ matrix uniquely defined with, in its main diagonal, the 6 *singular values* of $\overline{J}$, defined as the squared roots of the eigenvalues $s_i \geq 0$ of the matrix $(\overline{J}\ \overline{J}^T)$, positive semi-definite, and with null off-diagonal elements;
- U and V are two square, orthogonal matrices, with dimension (6×6) e $(n \times n)$, respectively; therefore, $U^T U = I_6$ and $V^T V = I_n$;

From the (4.45) decomposition, we conclude that

$$\overline{J}^T = V \cdot S^T \cdot U^T$$

and, based on the properties expressed by the so obtained SVD matrices, the following developments are possible:

$$\begin{aligned}
\overline{J}^* &= V \cdot S^T \cdot U^T (U \cdot S \cdot V^T \cdot V \cdot S^T \cdot U^T + K)^{-1} = \qquad (4.46)\\
&= V \cdot S^T \cdot U^T (U \cdot S \cdot S^T \cdot U^T + U \cdot U^T \cdot K \cdot U \cdot U^T)^{-1} = \\
&= V \cdot S^T \cdot U^T \cdot (U \cdot S \cdot S^T \cdot U^T + U \cdot K^* \cdot U^T)^{-1} = \\
&= V \cdot S^T \cdot (S \cdot S^T + K^*)^{-1} \cdot U^T
\end{aligned}$$

after having defined

$$K^* \triangleq U^T \cdot K \cdot U \Rightarrow K^* \geq 0 \tag{4.47}$$

Now, by choosing

$$K^* \equiv diag(k_1, k_2, ..., k_n), k_i \geq 0 \tag{4.48}$$

and consequently structuring the matrix $K \equiv U \cdot K^* \cdot U^T$, based on the definition of S we have

$$S^T \cdot (S \cdot S^T + K^*)^{-1} = \begin{bmatrix} \frac{s_1}{s_1^2+k_1} & 0 & ... & 0 \\ 0 & \frac{s_2}{s_2^2+k_2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \frac{s_6}{s_6^2+k_6} \\ ... & ... & ... & ... \\ 0 & 0 & ... & 0 \end{bmatrix} \triangleq S^* \tag{4.49}$$

where the matrix S^* contains the "regularized" singular values of $\overline{J}^*$, therefore obtaining

$$\overline{J}^* = V \cdot S^* \cdot U^T \tag{4.50}$$

Now, in place of the exact solution (4.40) to the problem (4.39), that makes use of the exact pseudoinverse $\overline{J}^{\#}$, we use instead the approximate solution

$$\dot{\mathbf{z}} = \overline{J}^* \dot{\mathbf{X}} \tag{4.51}$$

by recalling also that $\dot{\mathbf{q}} = \Sigma^{-1}\dot{\mathbf{z}}$, finally, the velocity signal that will be sent to the joints will be given by

$$\dot{\mathbf{q}} = \Sigma^{-1}\overline{J}^* \dot{\mathbf{X}} \tag{4.52}$$

At this point, we can consider the problem of the regularizing effects of the term K, by analyzing the modified singular values given by (4.49); it is clear that, in absence of one of the k_i factors, the corresponding singular value in (4.49) is simply $\frac{1}{s_i}$, that therefore takes back the non-regularized value, that we find in the exact matrix $\overline{J}^{\#}$; nevertheless, in presence of singularities, at least one of the values s_i tends to 0, and therefore it is necessary to introduce a corresponding value k_i, such that the regularized value will tend to $s_i^* = 0$ instead of infinity.

So, it is clear that we need in some way to *tune* the value assumed by the k_i coefficients, based on the distance from the singularity; therefore we can think of defining, for each k_i, a smooth function $k_i(s_i)$, that varies between a maximum value $k_M \triangleq k_i(0)$, and a non-zero minimum value beyond a given threshold, s_M; for example, a simple "raised cosine" function can be effective for this purpose (see Fig. 4.8).

We notice therefore that the problem of kinematic singularities can be solved, by defining two suitable values for the regularizing function k_i:

- The maximum value k_M, taking place in the singularity position;
- The threshold s_M, below which the regularization starts

The computation of $k_i(s_i)$, moreover, can be done *independently* for each singular value $s_i; i = 1, 2, ..., 6$.

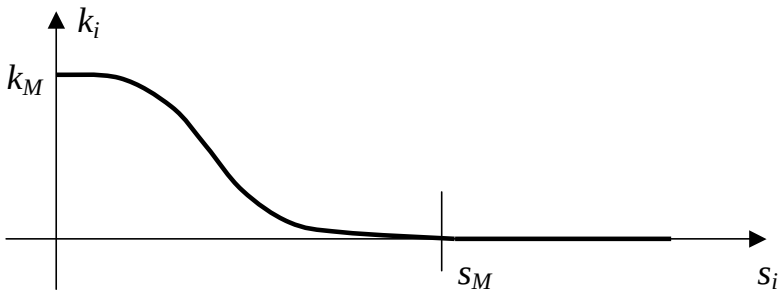

Figure 4.8: Adaptation of the regularization factors for the SVD.

4.2.2 The joint kinematic control law

From the relationship (4.34) we deduce, based on the developments of the previous Section, that a possible joint vector velocity realizing the law (4.33) is given by

$$\dot{\bar{\mathbf{q}}} = J^*(\mathbf{q})(-\gamma\mathbf{e} + \dot{\mathbf{q}}_g) \tag{4.53}$$

where we indicate with J^* the regularized pseudoinverse of the end-effector Jacobian matrix $J(\mathbf{q})$ (omitting for sake of simplicity the weight factors W in (4.37), and therefore setting $W = I$); the latter is computed with (4.50), through a suitable SVD decomposition algorithm, and the regularizing factors k_i in (4.49).

Assuming to have a cartesian error measure $\mathbf{e}$, we finally have the generic joint control law, given by (4.53).

In the case of the arm, let us consider the frame $< e > \equiv < p >$, i.e. define the end-effector frame coincident with the palm frame, solidal with the lask link of the wrist; moreover consider, as position reference, the frame $< g > \equiv < p' >$ coming from the upper MLC level, for which the position/velocity cartesian references are given by the formulae (4.30) of the previous Section.

As far as it concerns the computation of the error $\mathbf{e}_{p',p}$, having also the two transformation matrices ${}^0_{p'}T$ (from the first of the (4.30)) and ${}^0_pT(\mathbf{q}_A)$, the latter evaluated as cartesian feedback from the arm posture $\mathbf{q}_A$, we proceed as for the MLC task, through the versors' Lemma formula of (2.22); moreover, concerning the feed-forward velocity signal $\dot{\mathbf{X}}_{p'} \triangleq col(\omega_{p'}, \mathbf{v}_{p'})$, the latter is given by the second of the (4.30), as it is also evident from Fig. 4.7.

By resuming, in order to realize the tracking of the goal frame $< p' >$, it will be necessary to impose to the palm $< p >$ a reference velocity given by

$$\dot{\bar{\mathbf{X}}}_p = -\gamma_{A,llc}\mathbf{e}_{p',p} + \dot{\mathbf{X}}_{p'} \tag{4.54}$$

and the law (4.54) has to be realized through the pseudo-inversion of the Jacobian matrix ${}^0J_{p/0}(\mathbf{q}_A)$.

At this point it is clear that the *robot-dependent* part of the LLC control scheme is obtained by computing the 2 matrices ${}^0J_{p/0}(\mathbf{q}_A)$ e ${}^0_pT(\mathbf{q}_A)$, both known analytic functions of the posture $\mathbf{q}_A$ of the arm subsystem.

In order to complete the LLC module related to the arm, it is then useful to add the possibility of having a palm frame whose position, with respect to a fixed frame solidal to the last link, can be specified on-line; in other words, it will be useful to compute the matrices $J(\mathbf{q}_A)$ e $T(\mathbf{q}_A)$ (indices omitted for sake of simplicity) with respect to a frame chosen in advance, to which the palm is solidal, in order to be able to decide the relative location of the desired frame $< p >$, at the time of task implementation and execution.

The latter reasoning leads to define a further auxiliary frame $< t >$ on the end-effector, and the related location ${}^t_p\overline{T}$ of the palm; it will be therefore

$$ {}^0_pT(\mathbf{q}_A) = {}^0_tT(\mathbf{q}_A)\ {}^t_p\overline{T} \tag{4.55} $$

where the required analytic real-time computation now concerns the matrix ${}^0_tT(\mathbf{q}_A)$. Moreover, the computation of the Jacobian ${}^0J_{p/0}$ can be performed by using the one of the frame $< t >$, ${}^0J_{t/0}$, given by

$$ {}^0J_{p/0} = \overline{S}_{t,p}\ {}^0J_{t/0} \tag{4.56} $$

being $\overline{S}_{t,p}$ defined and computed analogously to (4.26), and finally

$$ \dot{\tilde{\mathbf{q}}}_A = {}^0J^{\#}_{p/0}(\mathbf{q}_A)\ \dot{\overline{\mathbf{X}}}_p \tag{4.57} $$

(with possible regularization of the pseudoinverse, ${}^0J^{*}_{p/0}$) where, this time, computation of the Jacobian matrix now concerns ${}^0J_{t/0}(\mathbf{q}_A)$.

By concluding, the algorithmic structure of the LLC control module for the arm is depicted in Fig. 4.9, where the different input and output signals for the module are put into evidence, as well as the computation of the pseudoinverse.

As far as the LLC level for the hand is concerned, we can first of all notice from Fig. 3.1 that it is made up of h independent modules, one per finger. In particular, we can consider the LLC control related to the generic finger i, being such modules structurally identical for all fingers.

Therefore, by recalling the Middle Level control law related to the hand subsystem, we can first refer to Fig. 4.4, and to eq. (4.31), that expresses the cartesian position/velocity references that have been processed by the MLC module; by recalling that such references have to be then realized by the underlying module, we can think of an organization completely similar to the one developed for the arm subsystem, but now taking into account the fact that we deal with purely *linear* (hence not including information about the *orientation* of the involved parts).

This statement is given by the fact that, referring to the multi-finger control law mentioned several times, in order to obtain the object manipulation with the only hand it is necessary to control the only linear position/velocity of the fingertips, as

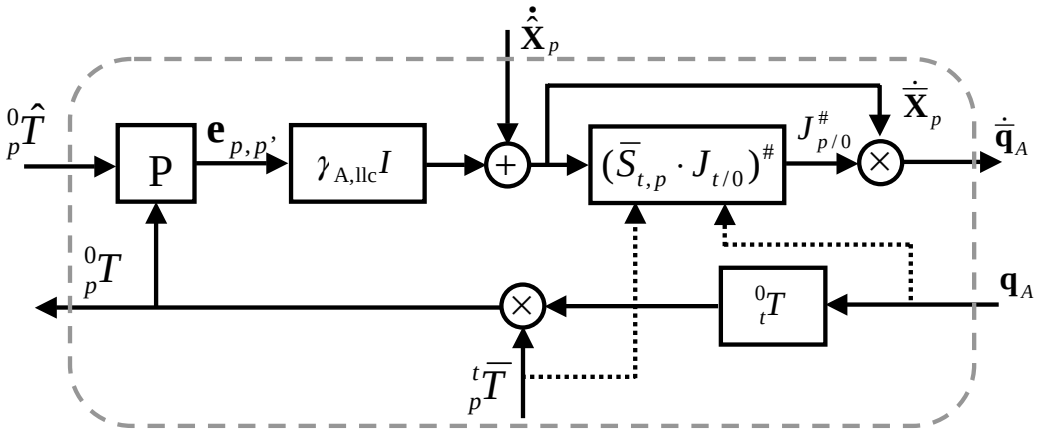

Figure 4.9: The LLC module of the arm.

it is clear from (4.31), and the related equations (4.9), (4.3) (velocity references) and (4.11) (position references).

Therefore, the LLC module accepts as input from the MLC module a position reference $\widehat{\mathbf{x}}_{i,H}$ given by (4.11), plus a feed-forward velocity reference $\dot{\widehat{\mathbf{x}}}_{i,H}$ given by (4.9), (4.3), and returns in output, as cartesian feedback for the MLC module, the current cartesian position $\mathbf{x}_{i,H}$ of the fingertip; in order to obtain the latter, the LLC module makes use of the transformation matrix ${}^{p}_{e_i}T(\mathbf{q}_{i,H})$ which, given the joint posture $\mathbf{q}_{i,H}$ of the respective fingertip, allows to obtain the position of the origin of an end-effector frame $< e_i >$, solidal with the lask link of the finger (i.e. the tip) and with origin on the *contact point* $\mathbf{P}_i$.

The cartesian control implemented by these modules can be in fact considered a *subset* of the complete position and orientation control that has been implemented for the LLC module related to the arm; we can therefore state the same control laws previously developed but restricted, in this case, to the only *linear* part of the above mentioned quantities; in particular, the low-level kinematic control law for the hand is represented by the following equation

$$\dot{\bar{\mathbf{q}}}_{i,H} \triangleq J^{\#}_{L,\mathbf{P}_i}(\mathbf{q}_{i,H})[\dot{\widehat{\mathbf{x}}}_{i,H} + \gamma_{H,llc}(\widehat{\mathbf{x}}_{i,H} - \mathbf{x}_{i,H})]; i = 1, 2, ..., h \tag{4.58}$$

being $\dot{\bar{\mathbf{q}}}_{i,H}$ the reference joint velocities of the single finger, that will be assigned to the underlying VLLC controller, $\gamma_{H,llc}$ a suitable gain parameter (constant), and $J_{L,\mathbf{P}_i}$ the linear part of the related Jacobian matrix in $\mathbf{P}_i$, of which we evaluate the pseudoinverse (again, with possible regularization $J^{*}_{L,\mathbf{P}_i}$, analogous to the arm case).

At this point we must immediately notice how, as also evident from the MLC control law, all of the cartesian quantities of the finger LLC have to be considered projected on frame $< p >$, related to the hand palm; that means, we have a control law represented in the cartesian space of an observer which is *solidal* with the hand

palm, and not incluing in any way the drift effects (rotational and translational) of the mentioned $< p >$ frame. This fact is coherent with the idea of control developed in the Section related to the MLC algorithm of the integrated hand-arm system, for which the two velocity contributions to the object $< b >$ given by the kinematic controllers of the arm and the hand are superimposed, and lead, separately and simultaneously, to the accomplishment of the global positioning task.

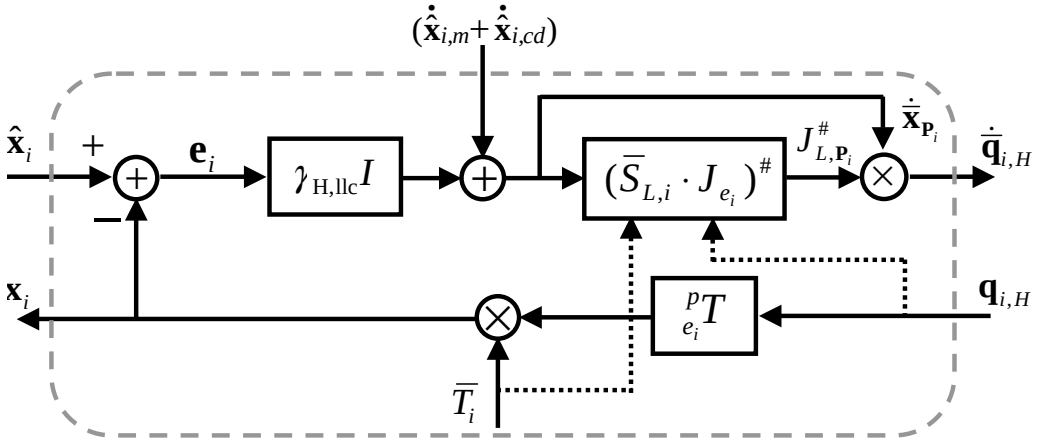

Figure 4.10: The LLC module for a single finger.

It is now possible to visualize the LLC control scheme for the single finger, given in Fig. 4.10; in particular, we notice from the picture how, as it had been put into evidence for the arm, the cartesian control law (4.58) can be obtained by evaluating the cartesian quantities related to a frame $< e_i >$ solidal to the fingertip, but with origin *not necessarily* coincident with the contact point $\mathbf{P}_i$, that has been used in order to define the position and velocity references; this is obtained, on one side, through the translation of the origin $< e_i >$ into the contact point $\mathbf{P}_i$ by means of a given transformation matrix $\overline{T}_i$, and, on the other side, by considering the computation of the *linear* Jacobian in $\mathbf{P}_i$ from the *complete* Jacobian ${}^{p}J_{e_i}(\mathbf{q}_H)$,

$$J_{L,\mathbf{P}_i} = \overline{S}_{L,i}\ J_{e_i};\ \overline{S}_{L,i} \triangleq \left[\ I_3 \quad [\mathbf{r}_{e_i,\mathbf{P}_i}\wedge]\ \right] \tag{4.59}$$

being $\overline{S}_{L,i}$ defined in a similar way to (4.26), here for the linear case.

4.3 The VLLC modules of the hand and related tendons control

In this Section we will examine the algorithmic structure for the joint velocity control of the robot hand, by considering first of all a linear model of a tendon-controlled system, and afterwards obtaining the corresponding approach for the velocity control, with compensation of the instantaneous variations of tension [6], [31], [7].

In the present control approach, we must consider several problems concerning an independent joint actuation for the single finger; in fact, as it will be made clear in the Section, concerning the implementation of the control on the experimental setup of the hand-arm system, it exists a non-negligible coupling between the tendon tensions, when actuating the structure, and the consequent joint motion resulting from such actuation.

Let us consider for example the figure (3.5) of Chapter 3, which exemplifies the sliding system used for the DIST-Hand: we can notice the presence of a complex *routing* system of the tendons onto the different pulleys of the finger, which imposes non-negligible torques on the intermediate pulleys, necessarily present in order to guide the tendon from the entry point in the finger to the last pulley, to which is anchored. Moreover, from the motor side, providing the tension to the anchored tendons, a similar routing system may be present.

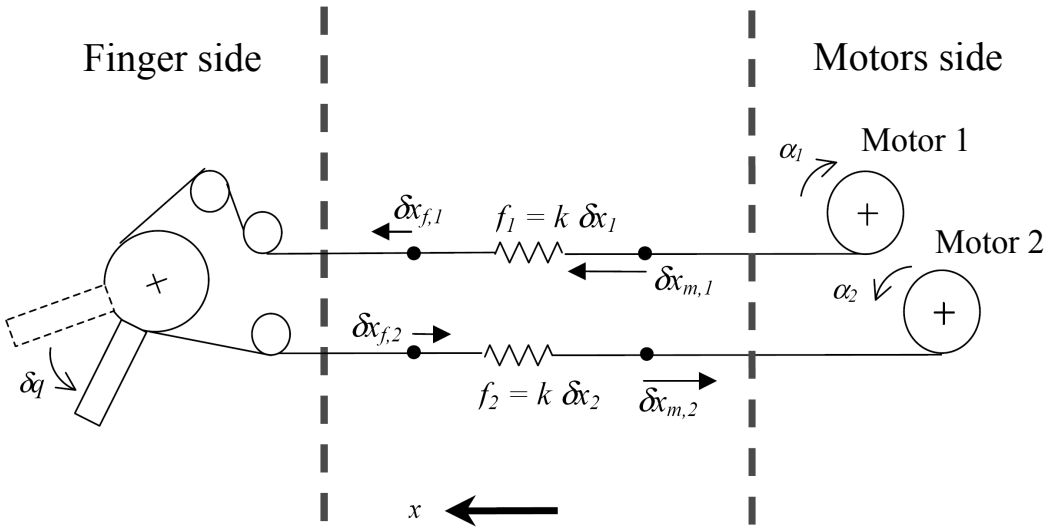

Figure 4.11: Example of a system actuated by tendons.

For sake of clarity, consider now the figure 4.11, exemplifying a simple tendon actuation system, through which we will then introduce the reasoning for the most general case, and afterwards apply the so obtained results to our experimental setup. First of all, it will be necessary to examine the following relationships:

- The relationship that exists between tendon displacements, measured on a common horizontal axis **x**, and the displacements of the joints **q** (given, in the case shown in figure, by a single element), expressed in *radians*, around their axes, being all of the above quantities referred to an initial *zero* position, to be specified in advance;

- The relationship between the same linear/angular displacements as seen from the motors' side, the latter denote with α;
- The relationship between the forces exerted by the tendons, and the respective torques on the actuation pulleys (solidal with the respective link), both on the finger and the motors' side;

We notice first of all as we consider *elastic* tendons; that means, we assume that a *linear* relationship exists between the tendon tension and the relative elongation with respect to the rest condition; moreover, we assume that the tendons are *inextensible* from both the finger and the motors' side; therefore, once two arbitrary points are specified on both sides of a tendon, with *null* tension, respectively given by the vectors $\mathbf{x}_f^0, \mathbf{x}_m^0$, the elasticity will be given by a linear relationship between the relative displacements of the two points, and it will be distributed along an "intermediate" zone of the tendon, hence, by calling $\mathbf{f}$ the tendons tension, we will assume the validity of the following relationship

$$\mathbf{f} = k\delta\mathbf{x} = k(\delta\mathbf{x}_f - \delta\mathbf{x}_m);\ \mathbf{f} > \mathbf{0} \tag{4.60}$$

being $\mathbf{x}_f$ and $\mathbf{x}_m$ the position vectors of the chosen points on the tendons, respectively on the finger and the motors' side, and where the vector $\delta\mathbf{x}_f \triangleq \mathbf{x}_f - \mathbf{x}_f^0$ expressed the displacement from the initial position of the points (on the finger side), and similarly for the motors' side.

Finally, we will assume the fundamental hypothesis of working always in a condition of non-zero tension on all tendons, therefore the elongations of the "springs" $\delta\mathbf{x}$, that exemplify the system of Fig. 4.11, and the related tensions $\mathbf{f}$, will always be strictly *positive*. We also notice here, that the positions of the points $\mathbf{x}_f^0, \mathbf{x}_m^0$ have been chosen arbitrarily, but such that the tendons are assumed without tension ($\mathbf{f} = \mathbf{0}$), in the initial condition; this assumption will be afterwards substituted with another one, more general, of an initial *pre-tensioning* of the system, as it will be made clear by the following descriptions.

Starting from the finger side, with the above mentioned hypotheses, the following holds

$$\delta\mathbf{x}_f = A\delta\mathbf{q} \tag{4.61}$$

where the constant matrix A, assumed to be full rank, contains the radii of the pulleys (in our case, only one), with signs dependent on the routing of the respective tendon (i.e. on one or the other side of the given pulley); such relationship expressed therefore the translation along the x axis of the specified points on the tendons $\delta\mathbf{x}_f$, on the *finger* side, as a consequence of a variation of the finger posture $\delta\mathbf{q}$ (recalling that the tendons are inexensible on both sides).

By repeating the same reasoning concerning the motors' side, we can write a similar relationship

$$\delta\mathbf{x}_m = B\delta\alpha \tag{4.62}$$

being in this case $\delta\mathbf{x}_m$ the displacement of the specified points of the tendons on the motors' side, $\delta\alpha$ the angular motor displacement, and B another full-rank matrix

that contains the radii of the motor pulleys, defined in a fully similar way to the matrix A.

By considering now the relationship between the tendon tensions $\mathbf{f}$ and the joint torques τ_f, τ_m, on the finger and motors' side respectively, we can easily show, based on the energy conservation principle, that the following hold:

$$\tau_f = A^T\mathbf{f};\ \tau_m = B^T\mathbf{f}; \tag{4.63}$$

where A and B are the same matrices above defined.

We can show that, if the number of tendons t is at least one unity higher than the number of actuated joints n, due to the Charatheodory condition it exists a strict subset of the null-space $\ker(A^T)$, that we denote by $\ker^+(A^T)$, of tendon tensions such that

$$\mathbf{f}_0 \in \ker^+(A^T) \Longleftrightarrow \left\{\mathbf{f}_0 \in \ker(A^T); \mathbf{f}_0 > \mathbf{0}\right\} \tag{4.64}$$

therefore, from the first of (4.63), we conclude that there exist tendon tensions $\mathbf{f}_0 > \mathbf{0}$ such that no joint torques are generated, i.e. $\tau_f = \mathbf{0}$.

At this point, it is clear that we can choose the initial positions of the tendon points (or, equivalently, of the zero *motor* angles α, when the zero positions of the joint angles $\mathbf{q}$ have already been fixed), such as to impose a *non-null* tension to the tendons at rest, therefore relaxing the initial assumption. By denoting with $\mathbf{f}_0$ such a pre-tensioning term, we can express the overall tension as

$$\begin{cases} \mathbf{f} = k(A\mathbf{q} - B\alpha) + \mathbf{f}_0; \\ \\ A^T\mathbf{f}_0 = \mathbf{0};\ \mathbf{f}_0 > \mathbf{0} \end{cases} \tag{4.65}$$

where the term $(A\mathbf{q} - B\alpha)$, obtained from the (4.61), (4.62), provides the additional elongation of the tendons, consequent to the displacements of the motors and the joints with respect to their zero position ($\mathbf{q} = \mathbf{0}, \alpha = \mathbf{0}$).

If we apply to the finger joints a vector $\mathbf{c}$ of external torques (assuming that we can neglect the masses of finger links), at equilibrium the following relationship will hold

$$\tau_f = \mathbf{c} \tag{4.66}$$

ovvero, a causa delle (4.65), (4.63)

$$kA^T(A\mathbf{q} - B\alpha) = \mathbf{c} \tag{4.67}$$

that implies

$$\mathbf{q} = A^+B\alpha + \frac{1}{k}(A^TA)^{-1}\mathbf{c} \tag{4.68}$$

being A^+ the left pseudoinverse of the matrix A

$$A^+ \triangleq (A^TA)^{-1}A^T \tag{4.69}$$

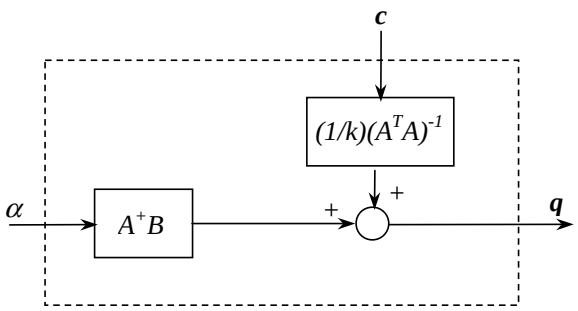

Figure 4.12: Schematics of the motors-joints system.

The expression (4.67) can be commented as follows: we can consider the vector α as a system input, while $\mathbf{c}$ can be considered a "disturbance", as it follows from the scheme in Fig. 4.12.

Now, we can ask first of all how can we control such a system, starting from considerations concerning the position control of the finger joints, and afterwards consider the velocity control as a natural consequence of the subsequent developments. Assume then to have a vector $\overline{\mathbf{q}}$ of reference joint positions, to be realized through a suitable motor position signal $\overline{\alpha}$; if we assume to have zero (or at least negligible) external torques $\mathbf{c}$, this control can be realized in open loop, be obtaining $\overline{\alpha}$ as one of the solutions to the equation

$$\overline{\mathbf{q}} = A^{+} B\alpha \tag{4.70}$$

or, equivalently,

$$(A^T A)\overline{\mathbf{q}} = A^T B\alpha \Longleftrightarrow A^T(A\overline{\mathbf{q}} - B\alpha) = \mathbf{0} \tag{4.71}$$

Therefore, in order to impose the previous condition, we can proceed in two distinct ways:

1. By considering equation (4.71), as it is written on its left-hand side, we can think of obtaining a vector α with *minimum norm*, by taking the preudoinverse of the matrix $A^T B$; in this way, we have

   $$\overline{\alpha} = (A^T B)^{\#}(A^T A)\overline{\mathbf{q}} \tag{4.72}$$

   that means,

   $$\overline{\alpha} = B^T A(A^T B B^T A)^{-1}(A^T A)\overline{\mathbf{q}} \tag{4.73}$$

   Such a solution, in general, leads however to an additional tendons elongation given by the quantity $(A\overline{\mathbf{q}} - B\overline{\alpha})$, that here amounts to

   $$\delta\overline{\mathbf{x}} = [I - BB^T A(A^T B B^T A)^{-1} A^T]A\overline{\mathbf{q}} \neq \mathbf{0} \tag{4.74}$$

In fact, this "unnecessary" elongation $\delta\overline{\mathbf{x}}$, and the consequent additional tension $k\delta\overline{\mathbf{x}}$, does not generate any joint motion, since $\delta\overline{\mathbf{x}} \in \ker(A^T)$; indeed:

$$A^T\delta\overline{\mathbf{x}} = A^T(A\overline{\mathbf{q}} - B\overline{\alpha}) = \mathbf{0} \tag{4.75}$$

2. Instead of proceeding as before, we can reconsider the relationship (4.71), as it is written on its right-hand side, thus obtaining another solution, not with minimum norm, but such as to avoid any additional elongation. Such a solution can be obtained by simply zeroing the second factor in (4.71), written on the right-hand side,
$$\overline{\alpha} = B^+A\overline{\mathbf{q}} = (B^TB)^{-1}B^TA\overline{\mathbf{q}} \tag{4.76}$$

At this point, we notice how the assumption of the latter case, i.e. $A\overline{\mathbf{q}} = B\overline{\alpha}$, furtherly implies a relationship between the velocities given by

$$A\dot{\overline{\mathbf{q}}} = B\dot{\overline{\alpha}} \tag{4.77}$$

and, therefore, if we desire to impose a predefined velocity to the joints $\dot{\overline{\mathbf{q}}}$, we can proceed by imposing to the motors a corresponding angular velocity given by

$$\dot{\overline{\alpha}}_m = B^+A\dot{\overline{\mathbf{q}}} \tag{4.78}$$

that constitutes the starting point for building the desired VLLC module, being $\dot{\overline{\alpha}}_m$ the velocity term necessary to impose the desired movement to the finger.

The control law (4.78), to be intended as *ideal* reference for the motor velocity, however, it is not *per se* sufficient for guaranteeing the velocity control of the finger, because of different possible reasons.

First of all, imposing a velocity reference requires the presence of a control loop for the single motor, with sufficiently good performance in terms of precision, rapidity of following the input signal, and immunity to disturbances; this requires the synthesis of a suitable regulator, as we will also see in the following, but we can already state that, in presence of an error while following the motor velocities, some elongation or relaxation effects on the tendons, at least transient, will be induced.

This fact leads to the presence of "spurious tems" of tensioning or relaxation of the tendons, which have to be in any case avoided; wherever we assume to have a *positive* additional tension, belonging to the already mentioned subspace $\ker^+(A^T)$ (and therefore such as not to generate joint torques) no relevant problems will be present, since the additional tension is by itself a transient term, that will be relaxed during the motion of the structure (of course, we always assume not to reach the *saturation* of the tendon elongation, since in that case we will loose the linear elastic model, that allowed to perform the control in the so far described way).

Instead the case of relaxed tendons, up to a null tension, is a definitely more critical problem, since not only it invalidates the adopted linear model, but it makes loose control on the tendon itself, which can even slide out of its support guide.

Possible causes for these phenomena of tensioning or relaxation of tendons are manifold: for example, the presence of an obstacle on the finger, that means of a

non-negligible external torque $\mathbf{c}$; or, as already mentioned, the slow following of the reference velocity signal by the motors; or even friction phenomena (both static and dynamic ones) concerning the sliding of tendons inside their guides, as well as on the pulleys of the whole structure.

In order to cope, in particular, with the tendon relaxation cases, it is therefore necessary to the define an additional control law for the tendon, such as to avoid perturbing, at least in principle, the system actuation. This leads to define a further tensioning term, and therefore a positive elongation, denoted with

$$\mathbf{f}_p \triangleq k\delta\mathbf{x}_p;\ \delta\mathbf{x}_p \in \ker^+(A^T) \tag{4.79}$$

At this point, it is necessary to observe that the elongation term $\delta\mathbf{x}_p$ in (4.79) must be obtained through a *motor* actuation, and therefore it must necessarily belong to the space of displacements or, which is the same, of velocities that can be attained by means of the actuation structure; we furthermore notice, for sake of clarity, that the overall elongation $\delta\mathbf{x}_p$ results in an additional linear displacement of the points on the tendon, from the *motors* side only, that can be indicated by $\delta\mathbf{x}_{m,p}$, since it does not have to impose any *additional* joint movement, and therefore to the tendons on the finger side ($\delta\mathbf{x}_{f,p} \equiv \mathbf{0}$). But, from (4.62) we also conclude that any vector $\delta\mathbf{x}_m$ of linear displacements on the motors' side, that we wish to obtain through rotations (or angular velocities) of the motors $\delta\alpha$, must also belong to the image $span(B)$ of the B matrix previously defined.

The two above mentioned requirements can be resumed as follows: it must exist an additional elongation $\delta\mathbf{x}_{m,p}$ of the tendons, obtained as only displacement of the points on the motors' side, such that

$$\delta\mathbf{x}_{m,p} \in \ker(A^T);\ \delta\mathbf{x}_{m,p} \in Span(B);\ \delta\mathbf{x}_{m,p} > \mathbf{0} \tag{4.80}$$

that means, the elongation vector must belong to the intersection, obiously assumed to be non-empty, of the two spaces in (4.80), and it must have strictly *positive* components. Therefore, if we define with $\Upsilon^+ \triangleq \left\{\ker^+(A^T) \frown Span(B)\right\}$ the intersection of the two sets, the condition (4.80) is equivalent to the following one

$$\delta\mathbf{x}_{m,p} \in \Upsilon^+ \subset \Re^t \tag{4.81}$$

under the assumption $\Upsilon^+ \neq \emptyset$.

At this point, assume to have found a suitable vector $\delta\overline{\mathbf{x}}_p$ satisfying (4.81); it is clear that we can obtain an arbitrarily large positive elongation, by simply multiplying $\delta\overline{\mathbf{x}}_p$ for any scalar value $k_p > 0$ (by using for sake of simplicity a monodimensional elongation space, although in general $\dim(\Upsilon^+) \geq 1$); moreover, since $\delta\overline{\mathbf{x}}_p \in Span(B)$, the equation

$$B\delta\alpha = k_p\delta\overline{\mathbf{x}}_p \tag{4.82}$$

has for sure a solution, that can be obtained as

$$\delta\alpha_p = k_p(B^+\delta\overline{\mathbf{x}}_p) \tag{4.83}$$

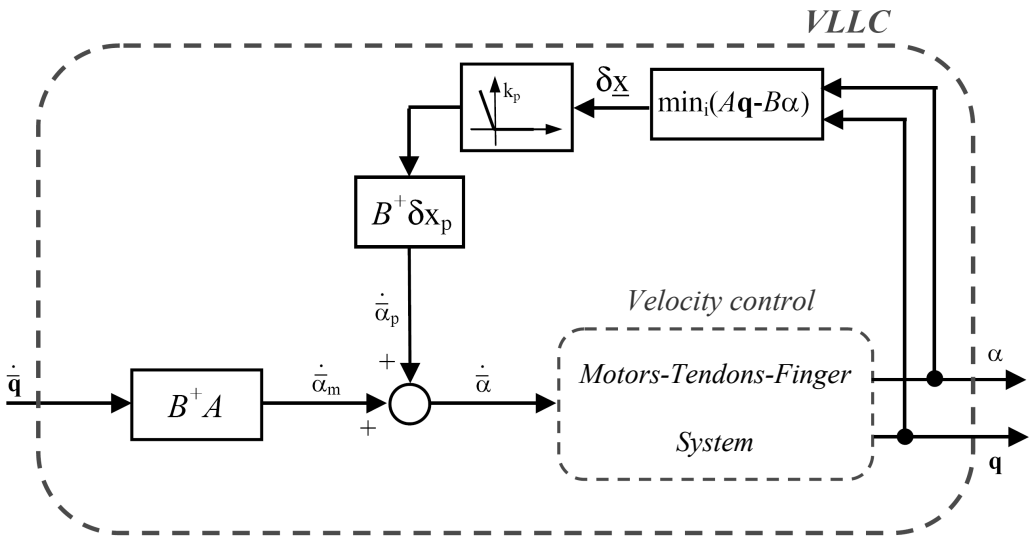

Figure 4.13: VLLC control system for a single finger.

and this is a vector of motor displacements that allows to give the desired, positive elongation.

The reasoning done so far can be easily extended to the case of velocity control, by imposing a pre-tensioning velocity to the motors of the same form, $\dot{\bar{\alpha}}_p$.

In order to proceed with the tendons control, an idea is therefore the following:

- we first of all evaluate the *relative* elongation vector of single tendons, with respect to their initial condition (where the tendons already have a non-zero tension), given by the formula $\delta\mathbf{x} = (A\mathbf{q} - B\alpha)$;
- we compute the elongation of the less "tense" tendon, given by $\delta\underline{x} = \min_i(\delta x_i)$;
- if $\delta\underline{x} > 0$, we do not add any pre-tensioning; otherwise, we add to the velocity reference $\dot{\bar{\alpha}}_m$ a term given by

  $$\dot{\bar{\alpha}}_p \triangleq k_p(B^+\delta\overline{\mathbf{x}}_p)$$

  being k_p in turn a variable term, given by

  $$k_p = \lambda\,|\delta\underline{x}|$$

  where, finally, $\lambda > 0$ is a suitable gain value.

At this point, the complete VLLC control scheme can be represented as in Fig. 4.13, where we can notice the sum of the two complimentary terms of velocity reference for the motors, that provides the overall reference given by

$$\dot{\bar{\alpha}} = \dot{\bar{\alpha}}_m + \dot{\bar{\alpha}}_p \tag{4.84}$$

Such reference will be sent to the closed-loop velocity control system for the motors, equipped with a suitable PID regulator.

Chapter 5

Simulations and experimental results

We will now show first of all some experimental results on the only hand subsystem, for which we will give a preliminary description; afterwards, we will show results of a simulation, concerning a manipulation task with the complete hand-arm system model, made up of an industrial manipulator PUMA260, and the DIST-Hand, with which the experimental results of the hand subsystem have been obtained.

5.1 Experimental results on the DIST-Hand setup

Before obtaining tasks for the hand subsystem, some control schemes have been implementes, and a long simulative phase has been performed, where also some design errors have been fixed.

The following experimental results are referred to the only hand subsystem, which is the most complex part of the system.

In order to obtain an almost ideal VLLC, a motor velocity regulator has been designed, such as to obtain very good performances, both static and dynamic, thus allowing an almost perfect tracking of the reference signal; at the same time, the tendon control devised in Chapter 4 (VLLC Section) hase been realized and calibrated.

The performed experiments concerned the position/orientation control of a light and easily "graspable" object. The position references have been given as signals, evolving as a *ramp*, always expressed with respect to the reference $< p >$.

Concerning the position references, from Figures 5.2, 5.3 it is evident that, during the rising (falling) phase of the ramp, we have a small error that keeps constant, but is zeroed at the end of the ramp, that is, when the reference becomes *constant* (this is due to the missing application of the *feed-forward* velocity signal $\dot{\mathbf{q}}_a$, that has been omitted, as in the previous Chapter, for sake of simplicity, and also in order to show the effectiveness of the control scheme also in computationally simplified situations).

Afterwards, concerning the orientation references, in Figures 5.4, 5.5 we can observe the 3 components of the orientation error $\rho_{b/p}$ of the object, compared with the reference vector $\rho_{a/p}$, and the related errors (we can also notice that, since the orientation references are given by *R-P-Y* angles, the resulting orientation vector shows an apparently less clear behaviour; this is a consequence of changing representation). Also in this case, as we can see, the ramp behaviour produces small errors, which are zeroed at the end of the ramp, for the above mentioned reasons.

Moreover, we kept into account the dynamic interaction between fingers and object, by adopting the algorithmic structure of the previous Chapter, in particular concerning the *distance* control between fingertips. As we can verify from Figures 5.6, 4.5, apart from a short initial transient, the three distances stabilize and keep constant, thus ensuring that the grasp control has been done accurately.

We can nevertheless verify the quality of the experimental results on the DIST-Hand from a video sequence, from which we selected some frames, shown in Fig. 5.1.

5.2 Simulations on the complete hand-arm system

The complete hand-arm system model that has been considered for performing coordinated manipulation tasks, as described in the previous Chapters, corresponds in fact to the system given in Fig. 3.7; for this system, provided a complete set of structural data, we built a simulator with graphical visualization based on the OpenGL library.

We can first of all describe the simulated task, that can be seen in Fig. 5.8, corresponding to the following Middle-Level (MLC) commands:

- Starting from a given pose, the hand-harm system is first controlled by splitting the two Cartesian positioning tasks; that means, in the approach phase to the target, i.e. the object in a previously measured position, we control the arm in such a way that the palm frame $< p >$ reaches an optimal position for grasping, while at the same time keeping the hand in an "open" posture;

- Subsequently, while keeping the reached arm posture, the hand is closed, by controlling in Cartesian space the individual fingertips, bringing them to the desired contact points;

- Once that the object graps is obtained, the controller commutes to the coordinated task of hand-arm manipulation, as it has been represented and described in the previous Chapters; by suitable modulating the middle-level gains, related to the hand and arm parts (i.e. the two scalars γ_H, γ_A), we can therefore see as it is possible to obtain the desired task of cartesian motion of the object, by means of the *simultaneous* motion of the two interplaying parts, at the same time avoiding loosing the object, that is kept by the hand subsystem through the *constant distances* task. We notice how the selected frames in Fig. 5.8

underline the joint role of the hand and the arm, when we ask the system to perform the exemplified task of vertical motion and subsequent rotation "on air" of the grasped object.

- Afterwards, we ask the system to position the object onto a "virtual" support, given by the blue-colored block in the animation; this task is, once again, performed jointly by the two parts, that brings the object to the goal position, onto the box;
- Finally, in order to release the object and return to the rest position, the task is again split between arm and hand, so that the fingers are free to move, release the grasp, and then bring the arm and hand in the initial position.

In Fig. 5.9 we show the result of following position and orientation references, during one phase of coordinated manipulation (object movements "on air"). Moreover, in Fig. 5.10 we can see how the joint velocity references given to the arm and the fingers are simultanously present, in order to perform the desired coordinated control task.

5.3 Conclusions

Among the goals of the so far presented work, we already mentioned, as an essential component of the whole development, the need of obtaining a result which is coherent with the preliminary considerations given in the Introduction; that means, we put into evidence and satisfy the need of having an *integrated* robot system, where the integration is given both at the *functional* and the *algorithmic* level, and, in the particular case dealt with, it concerned an *anthropomorphic* manipulation system, made up of a multi-finger robot hand, and an articulated arm.

However, we notice how the proposed functional and algorithmic architecture is fully general, and can be applied to several complex systems, of which nowadays many examples exist. In the special edition of the Journal [54], already mentioned in the Introduction, we can see how, in recent research works in the field, already many efforts have been made in order to achieve the following goals:

- A more or less complete formalization of several control architectures for complex systems;
- The adoption of software *modules*, "connected" one another by tools/editors (graphical or not);
- The structuring of the above mentioned modules into *multi-level* hierarchical functional architectures, where each level corresponds to a "layer" of abstraction, aiming to solve the global control problem;

For each case, a *flexible* and efficient framework has been developed, for the organizazion and integration of the main software components. However, in many

cases only a few demonstrations and formal proofs have been done, concerning the correctness of the resulting global controller.

The work developed in the framework of the present Thesis, where we tried to obtain, as a further main goal, some clarity about the validity of the algorithms and mathematical formalisms at all control levels, puts therefore the basis for future developments concerning manipulation and grasping for the hand-arm systems already available, and the possible integration of further components, always acting in a coordinated way.

Referring to the particular case of the considered hand-arm system, it is also important to notice how, based on the results so far obtained, it will be possible to obtain in an next future more and more accurate and, at the same time, *natural*, tasks, in the context of *humanoid* robotic structures, that nowadays are subject of a large amount of study and experiments.

Moreover, future developments on the present system will lead to implement control schemes suitable for performing higher level tasks, such as: identifying the optimal grasp for a given object, *visual servoing* tasks (using computer vision methodologies for object localization), and an "intelligent" scheduling of a complete manipulation sequence, such as the one previously described.

Concerning the present hand-arm system, for the next future it has been also planned the introduction of several kinds of sensors ([7], [6]): for example, "extrinsic" tactile sensors, based on conductive rubber, provide the possibility of evaluating the exact position of the contact point on the fingertip, and "intrinsic" force sensors, able to compute the 3 components of the instantaneous force applied to the fingertip. By applying these sensors to each fingertip, we can obtain an integrated sensory system with which the *force* control systems for stable grasping and manipulation of objects can be developed, as we already indicated in the present work, where we proposed a possible implementation that resulted in satisfactory simulated results.

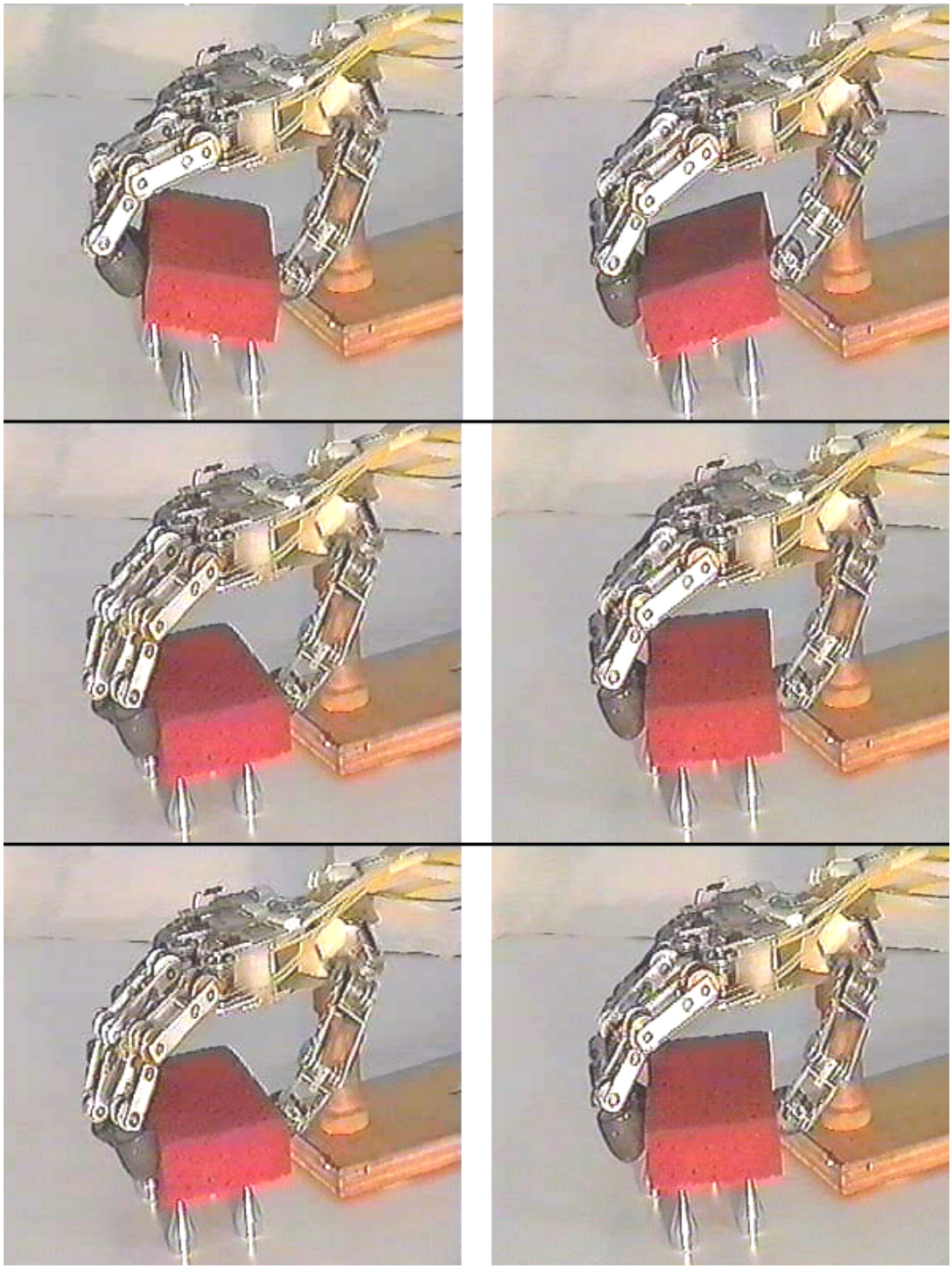

Figure 5.1: Snapshots of an object manipulation task.

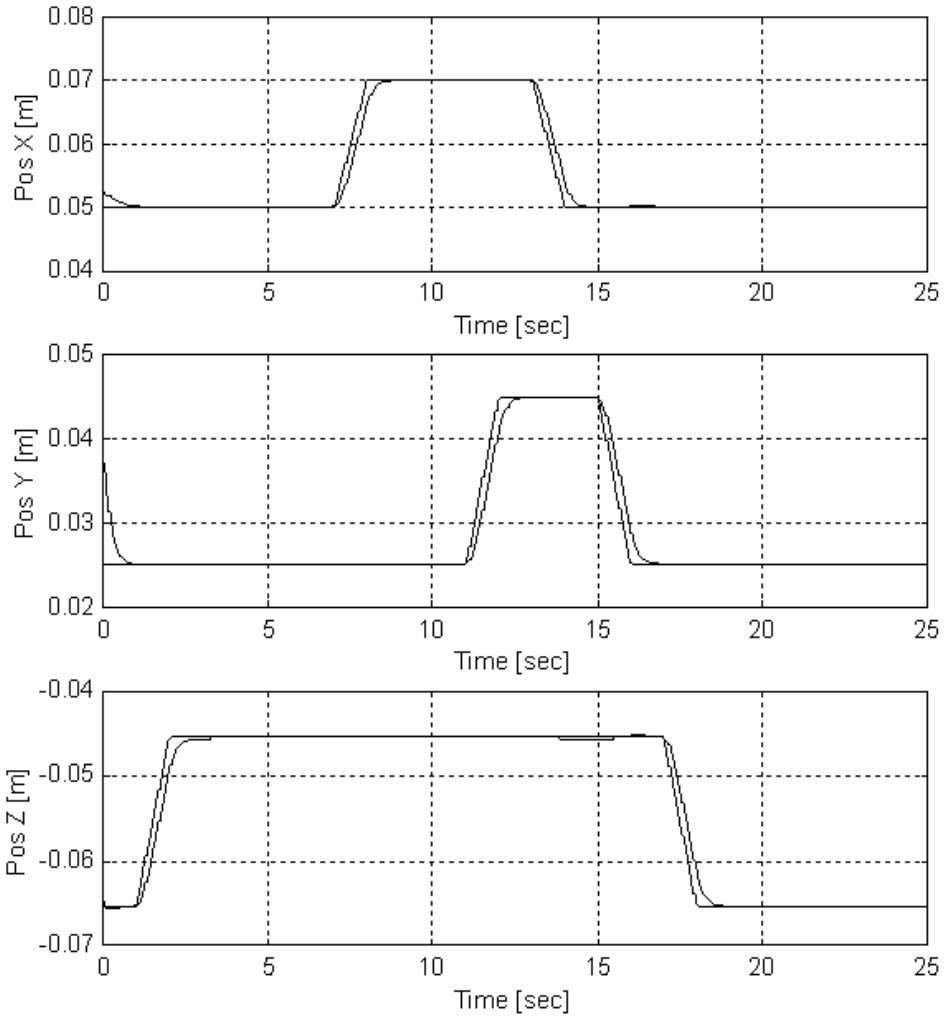

Figure 5.2: Tracking the position references.

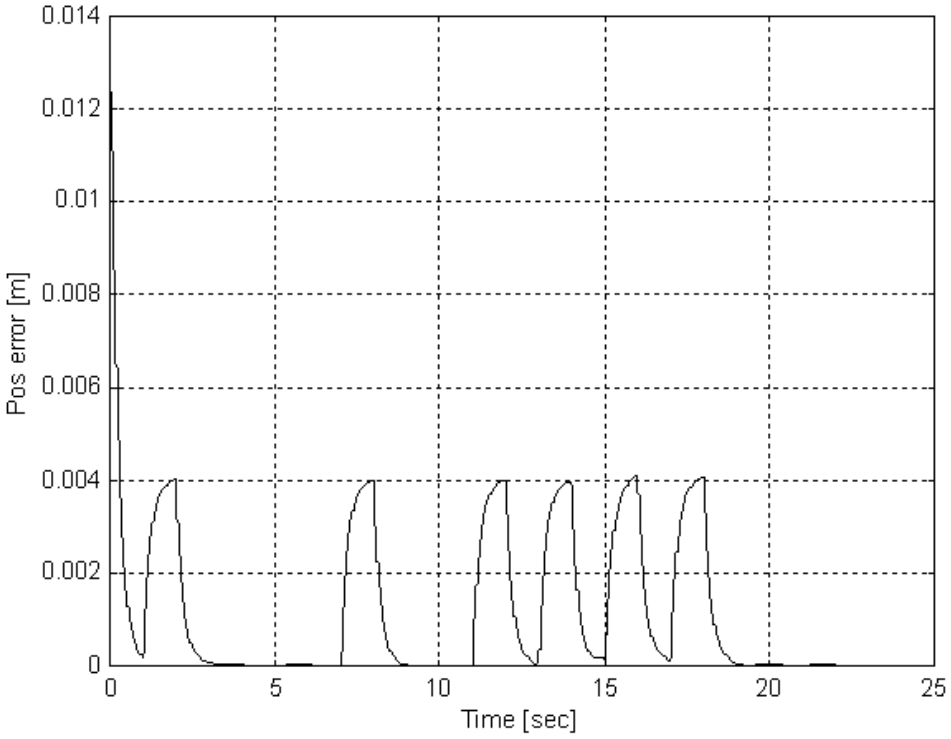

Figure 5.3: Position error.

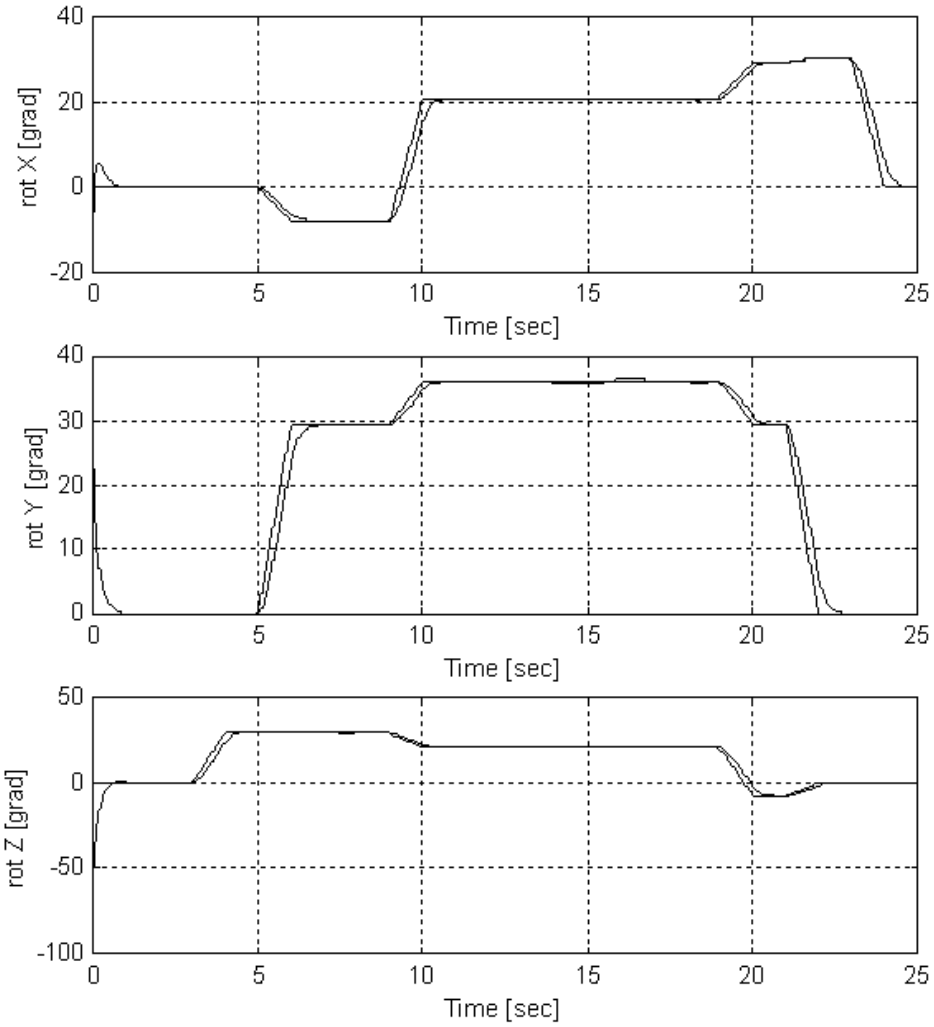

Figure 5.4: Tracking the orientation references.

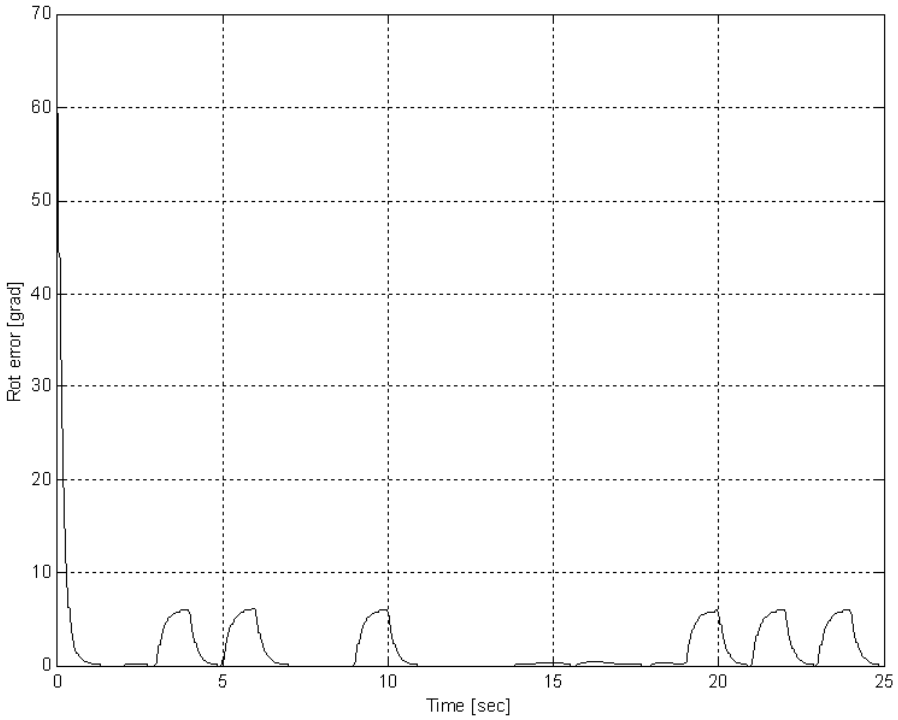

Figure 5.5: Orientation error.

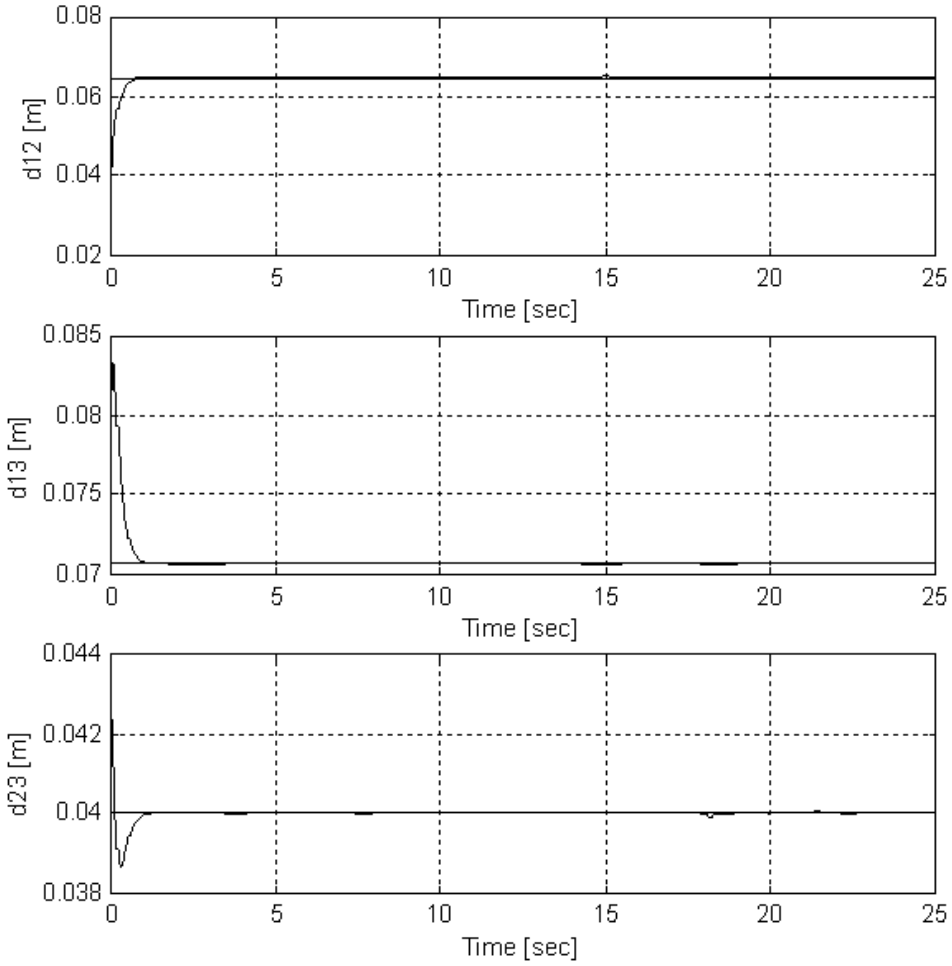

Figure 5.6: Keeping a constant distance between fingers.

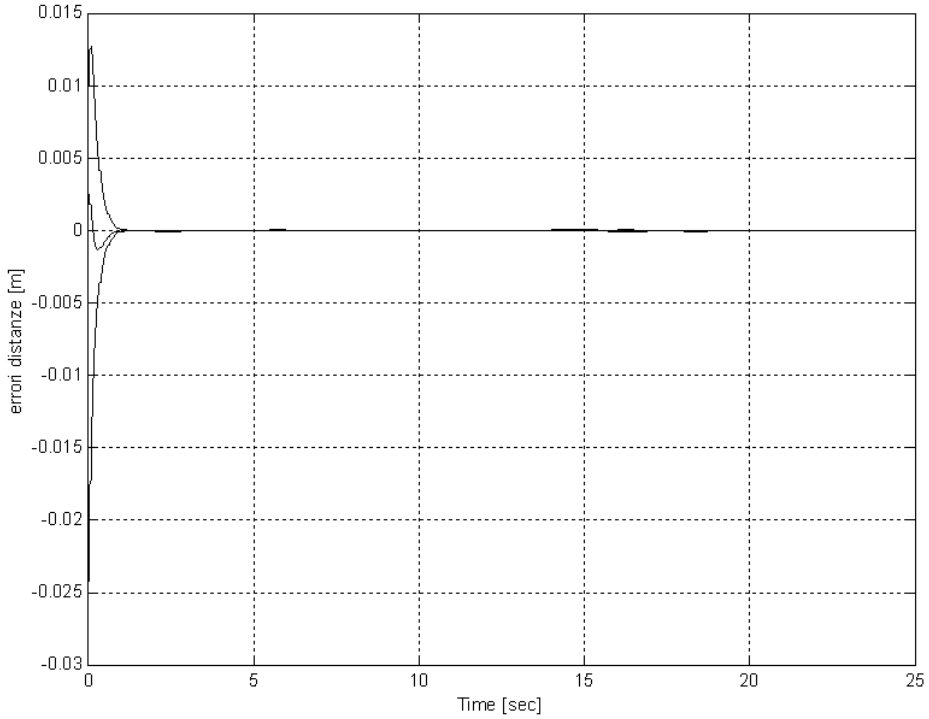

Figure 5.7: Distance errors between fingers.

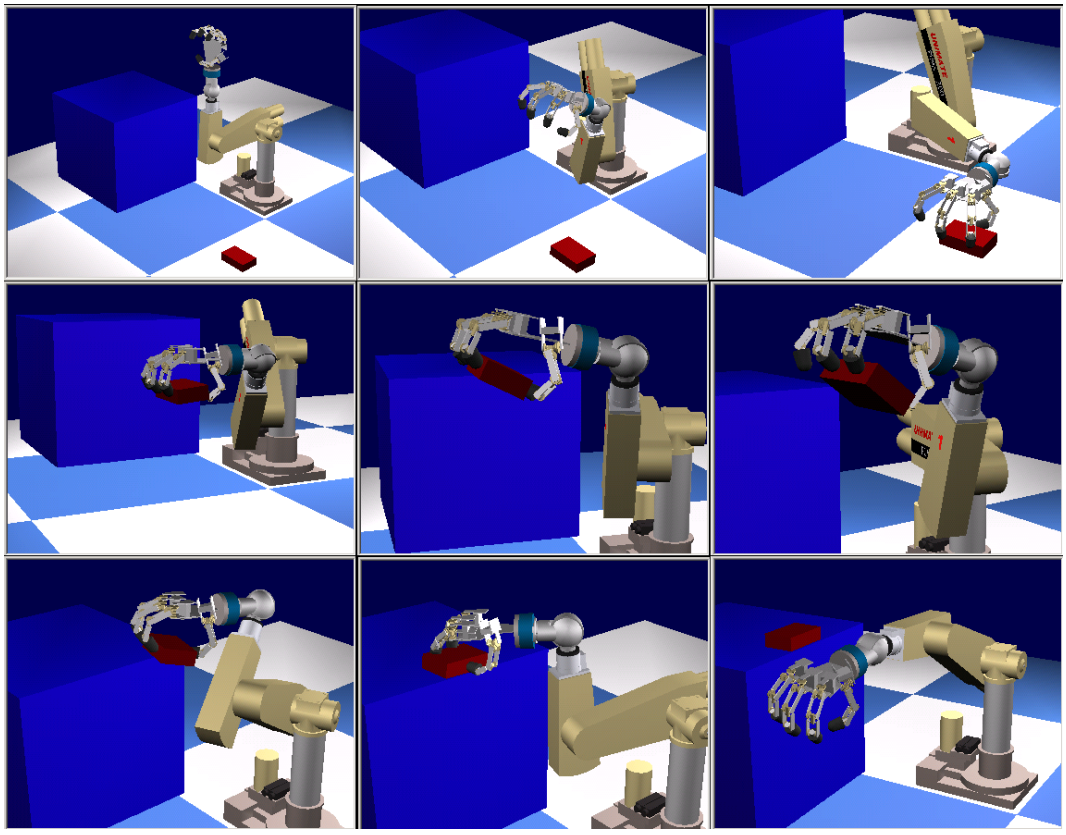

Figure 5.8: Graphical simulation of the complete, coordinated manipulation task.

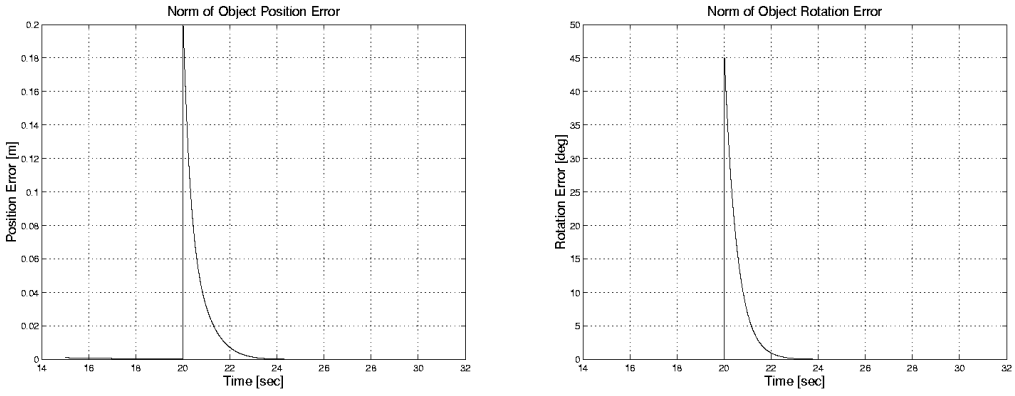

Figure 5.9: Position and orientation errors for the object.

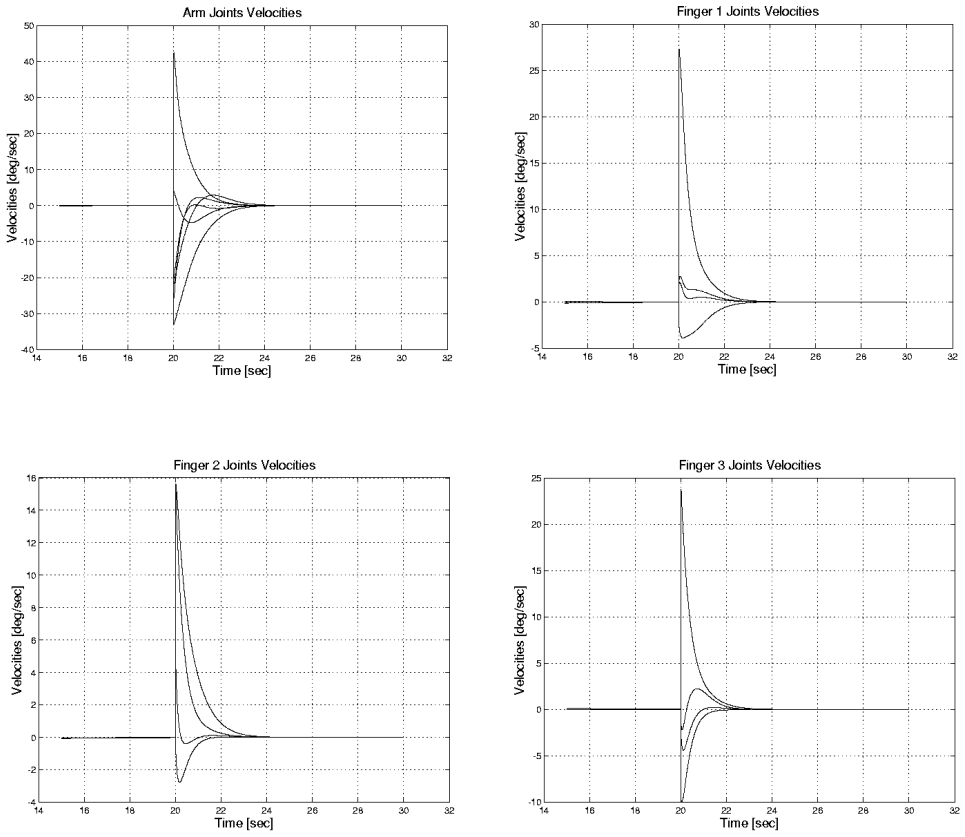

Figure 5.10: Joint velocities of the arm and hand, during the coordinated task.

Bibliography

[1] M. Aicardi, A. Caiti, G. Cannata, G. Casalino, "*Stability and robustness analysis of a two layered hierarchical architecture for the closed loop control of robots in the operational space"*, Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya (Japan), 1995

[2] M. Aicardi, A. Caiti, G. Cannata, G. Casalino, "*Hierarchical architecture for the closed loop control of robots in the operational space: stability and robustness analysis"*, Proc. Int. Conf. on Advanced Robotics, S. Felix de Guixol, Barcelona (Spain), 1995

[3] M. Aicardi, G. Cannata, G. Casalino, "*Contact Force Canonical Decomposition and the Role of Internal Forces in Robust Grasp Planning Problems"*, Int. Journal of Robotics Research, vol. 15 n. 4, Aug. 1996

[4] "*A New Design for a Dextrous Robotic Hand Mechanism*" IEEE Control Systems Magazine, Vol 12, August 1992, pp 35-38.

[5] S. Bernieri, A. Caffaz, G. Cannata, G. Casalino "*The DIST-Hand Robot*", IROS '97 Conf. Video Proceedings, Grenoble (France), September 1997

[6] A. Caffaz, G. Panin, "*Control Strategies for Tendon-driven Devices*", ICMT '99, Pusan (Korea), October 1999.

[7] A. Caffaz, G. Casalino, G. Cannata, G. Panin, E. Massucco, "*DIST-Hand, an Anthropomorphic, Fully Sensorized Dextrous Gripper*", IEEE Int. Conf. Humanoids 2000, Boston, Settembre 2000

[8] G. Casalino, G. Cannata, G. Panin, A. Caffaz, "*On a two-level Hierarchical Structure for the Dynamic Control of Multifingered Manipulation*", IEEE ICRA 2001, Seoul (Korea), May 2001

[9] G. Casalino, G. Panin, F. Giorgi, A. Caffaz, "*Dexterous Object Manipulation via Integrated Hand-Arm Systems*", IEEE Int. Conf. Humanoids 2001, Tokyo (Japan), Nov. 2001

[10] A. Cole, J. Hauser, S. Sastry, "*Kinematics and Control of a Multifingered Robot Hand with Rolling Contact"*, IEEE Trans. on Automatic Control, 34(4), 1989

[11] J. Crisman, C. Kanojia, Z. Ibrahim. "*Graspar: A Flexible, Easily Controllable Robotic Hand*" IEEE Robotics & Automation Magazine, June 1996.

[12] M. R. Cutkosky, P. K. Wright (1986a): *"Friction, stability and the design of robotic fingers"*, Int. Journal of Robotics Research, 5(4): 20-37.

[13] M. R. Cutkosky, P. K. Wright (1986b): *"Modeling manufactoring grips and correlations with the design of robotic hands"*, Proc. of 1986 IEEE Int. Conf. on Robotics and Automation, San Francisco, CA, 1533-1539.

[14] Engelberger "*Robotics in practice: management and application of industrial robots*", AMACOM, division of America Management Association, Inc., 1980

[15] Fu, Gonzalez, Lee 1989, "*Robotica*", ed. McGraw-Hill ed. italiana 1989

[16] L. Han, J. Trinkle, "*Dextrous Manipulation by Rolling and Finger Gaiting"*, Proc. IEEE Int. Conf. on Robotics and Automation, May 1998

[17] H. Hanafusa, H. Asada (1977a): *"A robot hand with elastic fingers and its application to assembly process"*, Proc. IFAC Symp. on Information and Control Problems in Manufactoring Technology, Tokyo, 127-138.

[18] H. Hanafusa, H. Asada (1977b): *"Stable prehension by a robot hand with elastic fingers"*, Proc. 7th International Symp. on Industrial Robots, Tokyo, 311-318.

[19] G. Hirzinger, B. Brunner, J Dietrich, J Heindl, "*Sensor Based Space Robotics-Rotex and Its Telerobotic Features*", Trans. on Robotics and Autom.,Vol. 9, n.5, October 1993.

[20] J. Hollerbach. "*Workshop on the Design and Control of Dexterous Hands*" MIT Artificial Intelligence Laboratory. A.I. Memo No. 661. April 1982.

[21] J. Hong, G. Lafferriere, B. Mishra, X. Tan, "*Fine Manipulation with Multifinger Hands"*, Proc. IEEE Int. Conf. on Robotics and Automation, 1990

[22] S.C. Jacobsen, J.E. Wood, D.F. Knutti, K.B. Biggers. "*The Utah/MIT Dextrous Hand Work In Progress.*" Center for Biomedical Design, University of Utah.

[23] Z. Ji (1987): *"Dexterous hands: Optimizing grasp by design and planning"*, PhD dissertation, Dept. of Mechanical Engineering, Stanford University, Stanford, CA.

[24] J. Kerr, "*An Analysis of multifingered Hands"*, PhD thesis, Dept. of Mechanical Engineering, Stanford University, 1984

[25] O. Khatib, "*A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation*", IEEE Journal on Robotics and Automation, Feb. 1987

[26] Kerr, Roth 1986, "*Analysis of multifingered hands*", Int. Conf. on Robotics Research, Vol.4, No.4, 1986.

[27] Klafer, Chmielewshi, Negin 1989, "*Robotic engineering: an integrated approach*", Prentice-Hall 1989.

[28] Z. Li, S. Sastry (1988): *"Task oriented optimal grasping by multifingered robot hands"*, IEEE Int. Journal of Robotics and Automation, 4(1): 32-44.

[29] Z. Li, P. Hsu, S Sastry, "*Grasping and Coordinated Manipulation by a Multifingered Robot Hand"*, Int. Journal of Robotics Research, Aug. 1989

[30] L. Ljung, "*System Identification-Theory for the User"*, Prentice-Hall, Englewood-cliffs, N.J., 1987

[31] E. Massucco, A. Caffaz, "*Control Design for Tendon Driven Devices*" , IASTED Int. Conf. MIC 2000, Innsbruck, Febbraio 2000

[32] Y. Matsuoka. "*Embodiment and Manipulation Learning Process for a Humanoid Hand*" S.M. Thesis. M.I.T. Department of Electrical Engineering and Computer Science. 1995

[33] B. Mishra, J. T. Schwartz and M. Sharir, "*On the existence and synthesis of multifingered positive grips"*, Algorithmica, 2:541-558, 1987

[34] D. J. Montana, "*The kinematics of contact and grasp"*, Int. Journal of Robotics Research, 7(3), 1988

[35] D. J. Montana, "*The kinematics of contact with compliance"*, IEEE Int. Conf. on Robotics and Automation, 770-775, 1989

[36] R. M. Murray, S. Sastry, "*Grasping and Manipulation using Multifingered Robot Hands"*, Robotics: Proc. Symp. on Applied Mathematics, vol. 41, 1990

[37] R. M. Murray, Z. Li, S. Sastry, "*A Mathematical Introduction to Robotic Manipulation*", CRC Press 1993

[38] Y. Nakamura, K. Nagai, T. Yoshikawa (1987), *"Mechanics of coordinative manipulation by multiple roboric mechanisms",* Proc. of 1987 IEEE Int. Conf. on Robotics and Automation, Raleigh, NC, 991-998.

[39] Y. Nakamura, K. Nagai, T. Yoshikawa, "*Dynamics and stability in coordination of multiple robotic mechanisms"*, Int. Journal of Robotics Research, 8(2), 1989

[40] V. D. Nguyen, "*Constructing force-closure grasps"*, Int. Journal of Robotics Research, 7(3), 1988

[41] D. Prattichizzo, A. Bicchi, "*Dynamic analysis on mobility and grasp stability of general manipulation systems"*, IEEE Trans. on Robotics and Automation, 14(2): 241-258, 1998

[42] M. Rosheim. "*ROBOT EVOLUTION, The Development of Anthrobotics.*" John Wiley & Sons. 1994.

[43] J.K. Salisbury and C. Rouff. "*The Design and Control of a Dexterous Mechanical Hand.*" Proceedings of the ASME Computing Conference. 1981

[44] J.K.Salisbury, M.T. Mason, "*Robot Hands and the Mechanics of Manipulation*", Cambridge: MIT Press, 1985

[45] J. K. Salisbury, M. T. Mason, "*Robot Hands and the Mechanics of Manipulation*", Cambridge: MIT Press, 1985

[46] J. K. Salisbury, "*Kinematic and Force Analysis of Articulated Hands*", PhD thesis, Dept. of Mechanical Engineering, Stanford University, 1985

[47] Salisbury J.K., "*Whole-arm manipulation*", Proc. of the 4-th International Symposium of Research, Santa Cruz, CA., MIT Press, 1987

[48] Sander 1991, "*Robotics: designing the mechanisms for automated machinery*", Prentice-Hall 1991.

[49] G. Schlesinger (1919): "*Der Mechanische Aufbau der Kunstlichen Glieder*". In: M. Borchardt *et al* (Edts.), "*Ersatzglieder und Arbeitshilfen fur Kriegsbeschadigte und Unfallverletzte*". Berlin: Springer, 321-699.

[50] Shimoga, Goldenberg 1996, "*Soft robotic fingertips*", The International Journal of Robotics and Automation, Vol.13, No.1, February 1997.

[51] D.B. Stewart, E. Schmitz, P.K. Khosla, "*The chimera II Real -Time Operating System for Advanced Sensor-Based Control Applications*", Ieee Trans. on Systems Man and Cybern., Vol. 22, n.6, November/December 1992, pag. 1282-1295.

[52] Taylor, C.L. and Schwarz, R.J. (1955): *"The anatomy and mechanics of the human hand"*, Artificial Limbs, 2: 22-35

[53] S. T. Venkataraman, T. Iberall, "*Dextrous robot hands*", Springer-Verlag New-York 1990.

[54] The International Journal of Robotics research, vol. 17 n. 4, April 1998