Technology and Health Care 14 (2006) 123–142
IOS Press

Braccio di Ferro: A new haptic workstation for neuromotor rehabilitation

Maura Casadio[a], Vittorio Sanguineti[a], Pietro G. Morasso[a,*] and Vincenzo Arrichiello[b]
[a]*Neurolab, Department of Informatics, Systems, Telecommunications, University of Genova, Genova, Italy*
[b]*Independent consultant*

Received 17 March 2005
Accepted 6 July 2005

Abstract. This technical note describes a new robotic workstation for neurological rehabilitation, shortly named *Braccio di Ferro.* It has been designed by having in mind the range of forces and the frequency bandwidth that characterize the interaction between a patient and a physical therapist, as well as a number of requirements that we think are essential for allowing a natural haptic interaction: back-driveability, very low friction and inertia, mechanical robustness, the possibility to operate in different planes, and an open software environment, which allows the operator to add new functionalities and design personalized rehabilitation protocols. *Braccio di Ferro* is an open system and, in the spirit of open source design, is intended to foster the dissemination of robot therapy. Moreover, its combination of features is not present in commercially available systems.

1. Introduction

Haptic interfaces are devices that enable human-machine interaction via the kinesthetic and/or tactile sense [2,3]. Typically, a haptic interface is designed as a manipulandum, i.e. a robotic manipulator that is equipped with a handle, which can be grasped by the user. The manipulandum applies forces, which can be programmed to vary with hand position, speed, or acceleration. In virtual reality or telepresence systems, this allows mimicking the dynamic behaviour of virtual or remote environments. Haptic interfaces have been successfully used in various domains, including telesurgery [17], medical and surgical virtual reality systems [12], micro-manipulation [7], and neuromotor rehabilitation.

Although the different application areas share methodological and technological issues, the design constraints can be quite different. In this paper we focus on robot therapy for neuromotor rehabilitation. The best known example in this class of systems is the MIT-MANUS [11], although other systems have been developed or are under development: for instance, MIME [13], KINARM [20], PFM [6] and MEMOS [14]. Such devices can be used to guide or assist the movements of a patient and in close interaction with him, just like a human physical therapist. Among the advantages are lower costs, a better repeatability of the therapeutic exercise, the possibility of a more personalized planning of treatments, and the possibility of precisely monitoring the outcome of therapeutic actions, in the framework of

*Address for correspondence: Pietro G. Morasso, DIST, University of Genova, Via Opera Pia 13, 16145 Genova, Italy. Tel.: +39 0103532749; Fax: +39 0103532154; E-mail: morasso@dist.unige.it.

a specific pathological condition. For instance, the MIT-MANUS system – the first of these devices to be approved by FDA for clinical use – has been demonstrated to accelerate the recovery of stroke patients [10,24]. The rehabilitation protocol consisted of reaching movements, and the robot generated position-dependent forces that helped patients to complete the task. Another possibility is to use the robot to generate perturbations: in patients with spasticity, elbow torque perturbations were shown to increase the elbow range of motion [5]. However, a haptic robotic workstation can do much more than emulating the behavior of a human therapist. By combining haptic with virtual reality technology, it is capable of emulating an extremely rich and adaptable environment that includes visual, auditory, kinesthetic, and/or proprioceptive stimulations. As a consequence, it is possible to study in a quantitative manner the mechanisms of motor learning and sensorimotor adaptation [21] in order to design interacting protocols that are optimal for the individual patient.

Here we report on a new robotic workstation for neurological rehabilitation, shortly named *Braccio di Ferro* (in Italian, it means literally *Iron Arm* but it is also the name of the popular cartoon character *Popeye*), which has been developed from our previous experience in two complementary fields: measurement of the hand mechanical impedance [23], and quantitative assessment of cerebellar ataxia [18]. A preliminary report of the system was presented at the IMEKO International Symposium on the Analysis of Human Functions [19].

In comparison with existing devices, we had several design goals: 1) an extended range of forces, in order to match the natural capabilities of the human motor system; 2) back-driveability of the system, in order to have natural haptic interaction; 3) very low friction and inertia in order to enable experiments near proprioceptive thresholds; 4) mechanical robustness, taking into account the intended clinical usage of the device; 5) the possibility to operate in different planes; 6) an open programming environment, which could allows the user to add new functionalities and design personalized rehabilitation protocols.

2. Design of the robotic manipulandum

The requirement of back-driveability, that mirrors the performance of the human motor system, suggests us to avoid solutions based on linear motor tables that typically use non-reversible gears. For the same reason we avoided the typical robotic design based on kinematic chains with one motor and harmonic gear for each joint. The chosen class of solutions is based on a parallelogram linkage, with direct drive by the motors (Fig. 1): one motor drives the arm of the manipulandum, which is jointed with the forearm, and the other motor drives a crank, which transmits the motion to the forearm by means of a connecting rod. The main features of this arrangement are: 1) good rigidity of the structure; 2) direct drive of the manipulandum, which eliminates any backlash in the force/motion transmission; 3) minimization the overall inertia, because most of the mass is either fixed, or close to the rotation axes.

The next step in the design is to address three basic requirements that are partly conflicting: 1) large workspace, 2) high level of the force generated at the handle, and 3) high degree of manipulability throughout the workspace. In fact, the workspace increases with the size of the manipulandum, whereas the Force to Torque (F/T) ratio, i.e. the force generated at the handle per unit motor torque, decreases. A lower F/T ratio implies the need of more powerful motors in order to guarantee a specified level of hand force. A high and uniform degree of manipulability enhances the back-driveability of the manipulandum, thus simplifying its control. More specifically, having in mind the general requirements of robot therapy, we identified the following design specifications: 1) an elliptical workspace with a major axis of 800 mm (arranged medio-laterally with respect to the user) and a minor axis of 400 mm (aligned with the antero-posterior direction); 2) a F/T ratio better than 2 N/Nm.

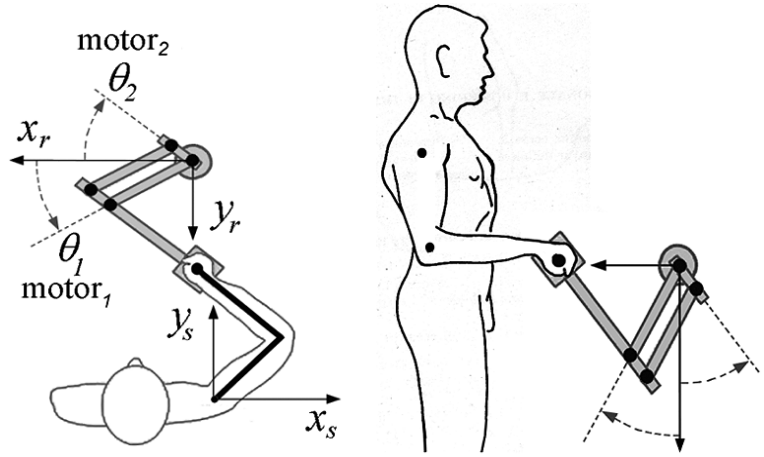

Fig. 1. Mechanical structure of *Braccio di Ferro.* Left panel: horizontal plane arrangement. Right panel: vertical plane arrangement. Two reference systems have to be considered: the robot-based system $[x_r,\ y_r]$ and the subject-related system $[x_s,\ y_s]$. The angular readings of the two motors have opposite signs: one motor determines the rotation of the shoulder ($q_1 = \vartheta_1$) and the other the rotation of the elbow ($q_2 = \pi - \vartheta_2$). The latter angle expresses an "absolute" rotation of the forearm with respect to the reference frame, not a "relative" rotation with respect to the arm link.

2.1. Geometric and kinematic design

In order to determine the manipulandum geometry that gives the best uniformity of performance over the entire workspace, we used the Global Isotropy Index (GII) [22]; this is defined as the ratio between the minimum and the maximum values of the motor torque required to generate a given force at the end-effector, evaluated over the entire workspace. To evaluate the GII value for each candidate configuration, we used a numerical procedure, implemented through the INTELLICAD© and MATHCAD© software packages. The procedure, which operated on a grid of 200 points spanning the workspace, first computed the value of the force/torque ratio and then the corresponding GII value. It was found that by curving the forearm of the manipulandum, as shown in Fig. 2, it is possible to obtain a workspace of the required size and with smaller lengths of the linkage, thus increasing the force/torque ratio. The resulting optimal geometry (in the sense of minimum GII) is characterized by the following parameters: $L_1 = 55$ cm; $L_2 = 19$ cm; $L_3 = 28.3$ cm; $Q_3 = 36.5$ deg; $X_0 = 0$; $Y_0 = 55$ cm (position of the center of the workspace with respect to the origin).

A geometric analysis of the workspace (Fig. 2 bottom panel) was also carried out with the purpose of identifying the range of motion of the different bars of the linkage and the appropriate positions of mechanical/electrical stops, which are necessary for safety reasons. Figure 3 shows the complete frame, which allows the rotation of the arm-motor assembly around the horizontal axis and the regulation of the height.

The following step was to carry out a kinematic analysis of the manipulator (see appendix A) in order to evaluate the degree of manipulability of the chosen configuration. Figure 4 shows the distribution of force ellipses throughout the workspace, which is qualitatively acceptable. From the quantitative point of view we derived from this analysis a number of useful indicators, averaged over the whole workspace. The first one is the F/T ratio:

$$\underline{\text{F/T ratio}}: \ 2.21 \pm 0.19 \ \mathrm{N/Nm},$$

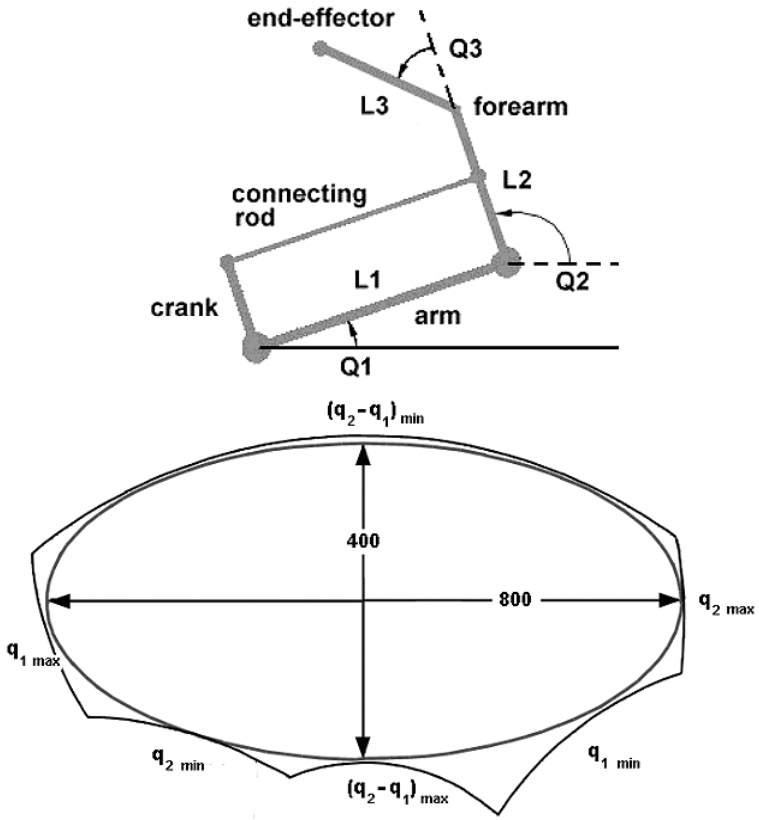

Fig. 2. Top panel: geometric configuration of the arm. Bottom panel: elliptical workspace (in mm), approximated by the curves related to the joint limits.

This is better than the initially chosen target value of 2. The corresponding GII has the following value:

$$\underline{\text{Isotropy index}}: \ f_{\max}/f_{\min} = 1.5370 \pm 0.2809$$

Another index, which is useful for characterizing the kinematics of robotic manipulators is the *manipulability index*, first proposed by Yoshikawa [25]: $w = \sqrt{\det(JJ^T)}$. This index goes to zero in the neighborhood of kinematic singularities. For smooth operation it should vary as little as possible within the workspace. In our case, we found the following value

$$\underline{\text{Manipulability index}}: \ 0.2257 \pm 0.0195 \text{ m}$$

with a coefficient of variation of 8.6%, which is quite acceptable for our purposes. The kinematic analysis, together with human psychophysics, can be used for evaluating the required spatial resolution of the robot. In particular, we may consider the "velocity ellipses" defined in the appendix A, which relate small translations of the robot end-effector with small rotations of the motors (or angular encoders aligned with the motors) and take into account the JND (Just Noticeable Difference) in lateral localization that characterizes the human tactile system [3]: this of the order of 0.15 mm. We expect the spatial resolution of the robot to be significantly better. Considering that a direct drive approach will be adopted, if we suppose that the motor rotation angles are sampled with 17 bits per revolution, the average positional

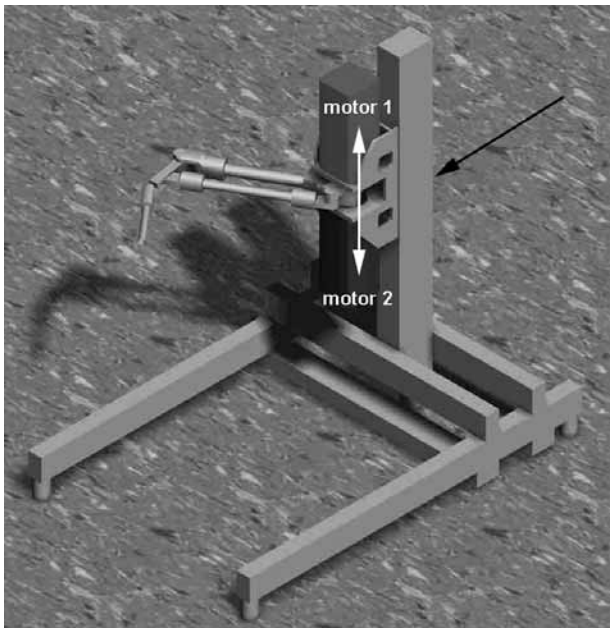

Fig. 3. Complete layout of the system, which show the possibility to regulate the height and the orientation of the workplane (rotation around the horizontal back axis).

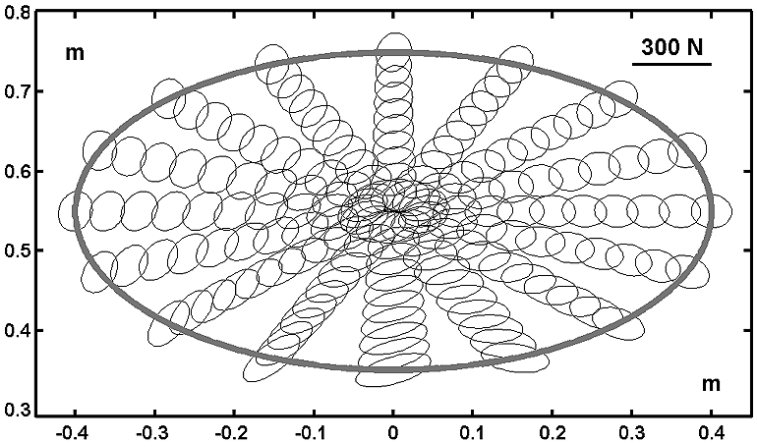

Fig. 4. Distribution of force ellipses in the workspace. The ellipses are generated by constant torque vectors of 50 Nm.

uncertainty can be estimated by analyzing the velocity ellipse over the workspace, thus yielding the positional resolution:

$$\underline{\text{Positionalresolution}}\ 0.024 \pm 0.002\ \text{mm}$$

Such strong resolution in angular measurement is also required for achieving precise estimates of angular speed, acceleration, and jerk by means of numerical derivation techniques. Table 1 summarizes the overall specifications of the manipulandum.

Table 1
Overall features of *Braccio di Ferro*

Braccio di Ferro	
Workspace (elliptical) (cm)	80×40
Arm rod & connecting rod (L1) (mm)	Length 550; tube ∅ 30; thickness 7.5.
Forearm, segment 1 (L2)	Length 187; tube ∅ 25; thickness 3.
Forearm, segment 2 (L3)	Length 283; tube ∅ 25; thickness 3.
Bending angle (between segments 1 &2) (deg)	36.5
Isotropy index ($f_{\max}/f_{\min}$) (a.u.)	1.537 ± 0.2809
Manipulability index (m)	0.2257 ± 0.0195
Angular resolution of the encoders (bit/rev)	17
Positional resolution at the HIP (mm)	0.024 ± 0.002
Continuous force at the HIP (N)	> 50 N
Peak transient force at the HIP (N)	$>$ 200 N
Sampling frequency of the impedance control loop (Hz)	1000

2.2. Dynamic design

First of all, we consider static forces and torques. In particular, as regards the continuously deliverable force at the handle (in all directions and over the whole workspace), we think that a value of about 50 N is appropriate in order to take into account the foreseeable range of forces to be applied to the patients. From this, we may estimate the minimum value of the continuous torques that the motors must provide, taking into account the F/T ratio estimated above. The result is 22.6 Nm.

The required level of the peak torque is derived from dynamic analysis. As shown in Appendix B, the manipulandum is characterized by the following dynamic equation

$$T_{\text{motor}} = T_{\text{intrinsic}} + T_{\text{haptic}} \tag{1}$$

which says that the motor torque required for a given motion includes a term that expresses the intrinsic dynamics of the manipulator ($T_{\text{intrinsic}}$) and a term (T_{haptic}) that implements the haptic interaction between the user and the programmed virtual object. The intrinsic torques, which are "parasitic" for the intended application, should be as small as possible. They depend upon the dimensions of the manipulandum, the distribution of masses, and grow more than linearly with the speed of movements. In the appendix we compute the upper bound of the intrinsic torques for the designed manipulandum, considering "very fast" movements, in the context of the envisaged application. The estimated values, for the two motors, are:

$$T_{\text{intrinsic}} = \begin{bmatrix} 34.73 \pm 10.87 \\ 5.63 \pm 1.63 \end{bmatrix} \; Nm \tag{2}$$

Virtual haptic objects

As regards the virtual haptic component T_{haptic}, it can be expressed as follows

$$T_{\text{haptic}} = J(q)^T F_H \tag{3}$$

where $J(q)$ is the Jacobian matrix of the manipulandum andF_H is the external force, applied by the subject to the handle or, with a minus sign, by the handle to the subject. The requirement that the continuous exertable force is better than 50 N allows us to introduce in the virtual haptic object a virtual weight of 5 kg or, considering a spatial resolution of the order of 0.1 mm, a virtual stiffness up to

500 N/mm that is much larger than the minimum stiffness for objects to be perceived as rigid [3], whose order of magnitude is evaluated to be around 25 N/mm.

Strong peak torques are required for emulating inertial loads or elastic impacts with rigid objects. For example, if the subject attempts to move a virtual 5 kg mass with the same "fast" movements already considered for intrinsic torques, the required peak force during haptic interaction is 117.5 N that corresponds, by taking into account the F/T ratio of the manipulandum, to an average peak torque of about 53 Nm.The same order of magnitude of the peak torque can be evaluated in the implementation of virtual elastic impacts between a 1 kg mass and a rigid surface at an impact speed of 1 m/s. The value estimated above for a "typically strong" haptic interaction (of the order of 50 Nm) should be added to the estimated values of intrinsic torques in a "typically fast" movement, given by Eq. (2) (of the order of 35 Nm, in the worst case). Adding a little safety margin, we can say that a peak torque of about 90 Nm is required of the selected motors, in order to support that range of interactions.

Dynamic analysis is one of the relevant steps for dimensioning the links and joints of the manipulandum, taking into account two conflicting requirements: 1) robustness and 2) lightness. The former requirement goes in the direction of heavy and bulky solutions that conflict with the need of perturbing as little as possible the natural movements of the patients. On the other hand, we must take into account that very light frames, even if sufficiently robust to transmit the required range of forces, may be affected by vibrations in the frequency range of human movements that must definitely be avoided for obvious "ergonomic" reasons. In order to meet the requirements we used the following solution, which is a well balanced trade-off between robustness and lightness and was proven to be virtually vibration-free. For all the pieces we used aluminium: larger tubes (diameter: 30 mm; thickness: 7.5 mm) for the bars of the arm and the connecting rod; a smaller tube (diameter: 25 mm; thickness: 3 mm) for the forearm; precision machined joints with tube-clamping sockets; twin angular contact ball bearing hinge joint at the elbow, and spherical bearings at both ends of the connecting rod.

At the other end of the spectrum, we should consider the typical normal forces that people can discriminate when detecting features in tactile exploration: they fall in the range 0.4–1.1 N [3]. This means that the dry friction element in the robot dynamic equation (the term $D(q, \dot{q})$ in Eq. (B.1)) should be smaller than the lower bound (0.4 N) divided by the F/T ratio (2.21): 0.18 Nm. This constraint is difficult to analyse in a quantitative way at the design level, because it depends on both the motor and the linkage. However, it emphasizes the fact that the manipulandum must be machined with very good precision and the motor itself must be characterized by very low rotational friction. An experimental test, after construction, must estimate the residual friction and verify the previous requirement.

A final dynamic requirement is related to the sampling frequency of the control loop of the bi-directional interface: it is known indeed that with a frequency smaller than 1000 Hz, the user may perceive virtual surfaces softer than expected or, even worse, may perceive a vibration instead of a surface [2].

2.3. Actuators and sensors

The kinematic/dynamic analysis performed in the previous section suggests to choose motors with low inertia, low friction, and torque requirements as follows: continuous torque better than 23 Nm and peak torque better that 90 Nm. We chose AC brushless servo drive systems, based on rare earth permanent magnets, because they provide high levels of dynamic performance and torque density, even at very low speed. In particular we selected the Ultract® II series motors (Phase Motor Control spa) because they can be supplied with special windings, which allow specifying the crucial parameters (torque constant, peak torque, and continuous torque) in order to match the application requirements formulated in the previous section. The features of the motors we adopted are summarized in Table 2.

Table 2
Main features of the motors

Phase Motion Control – Motor ULTRACT® II Size 7 mod 726.05.3SM0000	
Nominal torque	35 Nm
Peak torque	97 Nm
Torque constant	9.66 Nm/A
Electrical time constant	17 ms
Dimensions (W × H × L)	145 × 145 × 400 mm
Weight	22.1 kg
Rotor inertia	2.42 10^{-3} kgm^2
Shaft torsional resonance frequency	600 Hz

Each motor is equipped with an incremental digital encoder of the SinCos type, with 4 incremental channels (sine, inverse sine, cosine, inverse cosine: 2048 c/rev) and 2 absolute channels (index and inverse index: 1 c/rev). The absolute signal allows the initial reset at start-up and the incremental signal, through interpolation, provides a resolution of 17 bits/rev.

As shown in Fig. 2, the two motors face each other: their flanges are bolted onto a precision machined aluminium frame with collinear axes; the shafts are interference mounted with shrink rings to the first two bars of the pantographic linkage; the other two bars transmit the motion to the end-effector by means of low-friction ball-bearings.

The Ultract® II motor model 726 is controlled by the DSP-based AX-V 10144 digital motion control system (Phase Motion Control spa). The provided firmware, written in the GPLC programming environment, runs three main real-time tasks: 1) a high-speed current loop (at 16 kHz); 2) a middle-speed position loop (at 4 kHz) which interpolates the SinCos signal, updates the control current, and carries out safety/diagnostic checks; 3) a low-speed task (at 125 Hz) for user-defined functions. The firmware interacts with an external control unit (in our case, a PC) by *sending* a copy of the encoder signals and *receiving* the desired current reference. The frequency of this loop can be as high as 4 kHz, but we chose a lower one (1 kHz) because it is sufficient for our applications.

2.4. Control architecture and haptic rendering scheme

Figure 5 shows the control architecture, which consists of two AX-V 10144 control units (one for each motor) and two personal computers: PC1 is the Configuration and Visualization workstation; PC2 is the Real-Time Control Unit. PC1 communicates with the two control units via a serial RS485 link. This link is only used at system setup, to initialize the firmware of the control units. PC2 manages the real-time control that implements the haptic rendering scheme, with a 1 kHz update rate. It is equipped with an Analog and Digital I/O PCI card (Sensoray, model 626), in which we use the following channels: a) two 14 bit D/A channels for commanding the reference values of the motor currents; b) two 24-bit counters for receiving the repetition signals of the two digital encoders; c) six 16 bit A/D channels for acquiring the readings of the force sensor.

Figure 6 shows the software architecture: 1) Planning and application design software on PC1; 2) Real-time control on PC2. PC1 runs the RT-Lab® software package (Opal-RT Technologies Inc) and operates as a *Command station*. RT-Lab, among other functions, is a graphical user interface, which supports the user in the design of Simulink® (Mathworks Inc) models in order to implement specific real-time control applications. It also automatically translates Simulink models into C programs, that are then compiled into real-time processes on a remote target computer (the QNX machine on PC2); the QNX machine carries out the actual haptic rendering scheme and other real-time functions. PC1 and

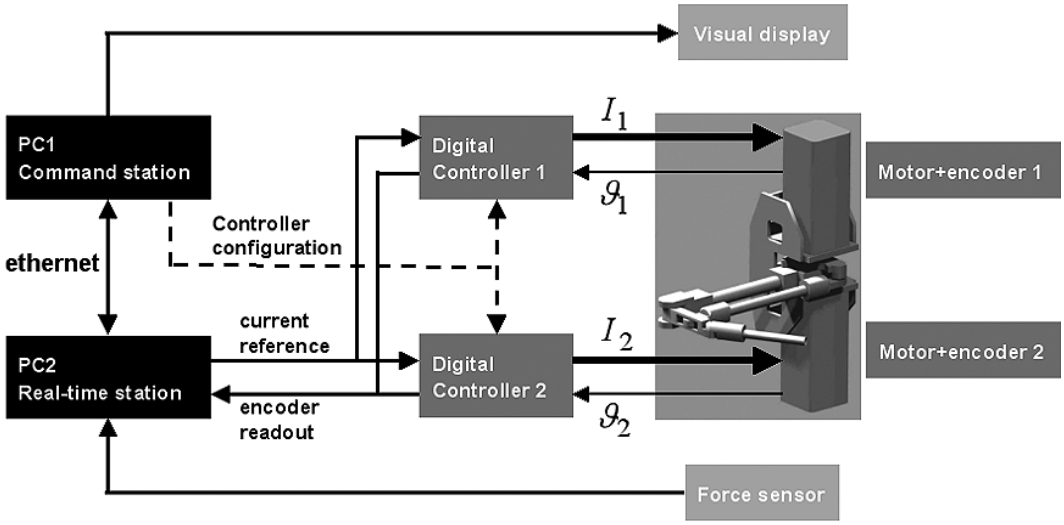

Fig. 5. Control architecture. PC1: Configuration and Command workstation; PC2: Real-Time workstation. The current reference is updated by PC2 at 1 kHz and the actual current is delivered to the motors by each digital controller at 4 kHz. The encoder reading is transmitted to the digital controller at 4 kHz and repeated to PC2 at 1 kHz. PC2 digitizes the output of the force sensor and PC1 refreshes the visual trace of the Virtual Environment of the visual display.

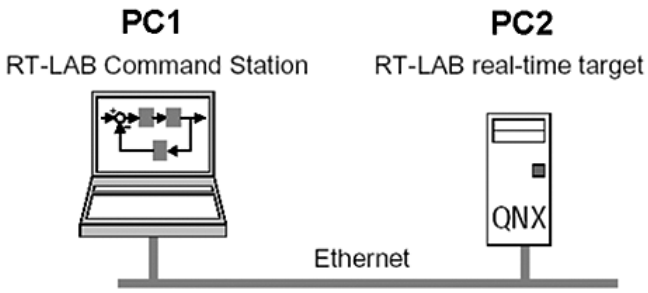

Fig. 6. Software architecture: PC1 operates as a command station and runs RT-LAB®. PC2 operates as a real-time station and runs RT-LAB real-time target, as a QNX® machine. PC1 and PC2 interact over a TCP/IP link.

PC2 communicate via Ethernet according to the TCP/IP protocol. RT-Lab requires that the Simulink model is subdivided into two subsystems: 1) a *Console* subsystem, which runs on the *Command* station and operates the visualization blocks (Scope, Display, XY Graph, etc.) and the user-interaction blocks (Switch, Slider Gain, etc.); 2) a *Master* subsystem, which runs on the real-time target.

PC2 is a QNX machine. It runs under the multi-thread, multi-tasking real-time operating system QNX® (QNX Software System Ltd). The basic features of QNX are: 1) Microkernel architecture; 2) Message-based interprocess communication; 3) Process scheduling with priority assignment. The microkernel is smaller than 8 kB and carries out only two tasks: message passing and scheduling. Applications are represented under QNX as interacting real-time processes, which may communicate with external devices. Inter-process communication is carried out by means of byte packets. Each process has a priority level (32 levels are possible) and the process scheduling algorithm can use one of three methods: 1) FIFO; 2) round-robin; 3) adaptive.

From the implementation point of view, the human's bilateral interaction with a virtual environment can be achieved either by manipulating the interaction force given the movement or by manipulating the

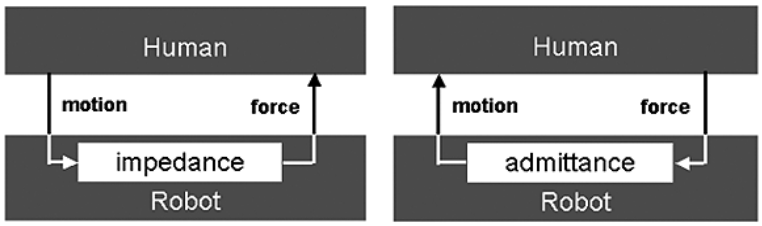

Fig. 7. A (left panel): impedance control scheme. B (right panel): admittance control scheme.

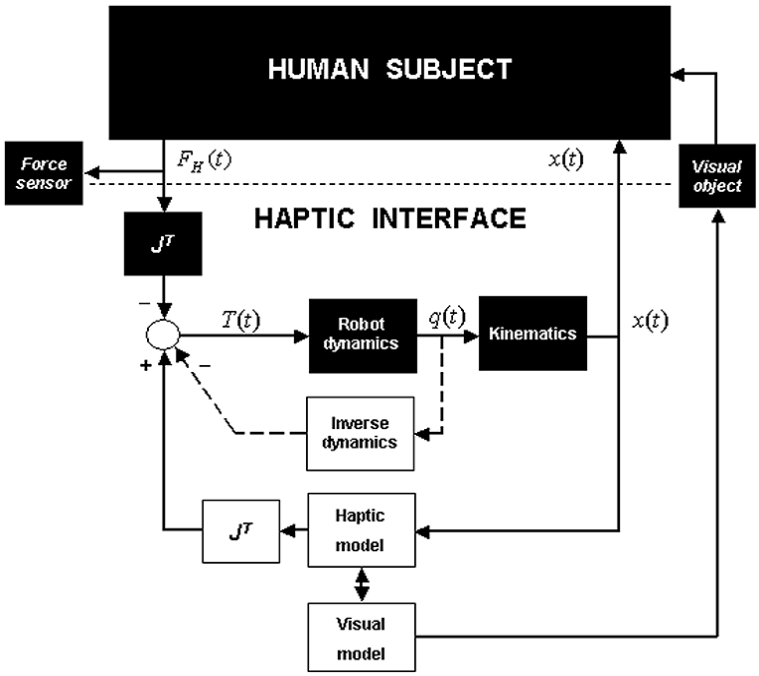

Fig. 8. Haptic rendering scheme: detailed diagram of the impedance control mechanism. The filled boxes correspond to transformations that occur in the real world. The empty boxes are actual computations that occur in the virtual world and are carried out by the haptic architecture. The dashed line corresponds to a computation that is carried out off-line. T is the commanded torque; q is the vector of rotation angles, x is the position of the handle; F_H is the interaction force; J is the Jacobian matrix of the manipulandum; Inverse dynamics is implemented by Eq. (B.1).

movement given the force. In the former approach, also known as *impedance control mode*, the robot acts as a mechanical impedance and the human as an admittance (Fig. 7, left panel). In other words, the robot reads the human's motion input and produces a force output according to the impedance of the emulated environment or virtual haptic object. In the alternative *admittance control mode* (Fig. 7, right panel), the force is measured and the motion is commanded, i.e. the robot acts as admittance and the human as impedance. The choice between the two schemes depends on the relative importance of the intrinsic dynamics and the frictional forces of the robot (with respect to the virtual dynamics) as well as on the accuracy that can be obtained in practice for the models of these disturbance sources. In a direct-drive, back-driveable systems, like *Braccio di Ferro*, the intrinsic dynamics and the frictional forces are small and can be modeled with good accuracy, as is demonstrated in test section. In this case, there is a strong correlation between the force reflected on the handle by the commanded motor

torques and the force applied to the handle by the human user: thus, the impedance control scheme is the natural solution. It should be noted that, in the admittance control mode, accuracy and performance are critically dependent on the accuracy of the force sensor, whereas in the impedance control mode they depend on the accuracy of the rotation sensor. The latter, which typically consists of a digital encoder, is very precise and the resolution is virtually unlimited; in contrast, force sensors, which typically use strain gauges, are intrinsically much less precise, have a limited range of measurement, are mechanically delicate, and their resolution is limited by the background noise. Moreover, impedance control is more robust and has less stability problems than admittance control.

The impedance control scheme adopted for *Braccio di Ferro* does not require a force sensor on the handle, but we mounted one for calibration purposes. It employs strain gauges to reconstruct 3 force components and 3 torque components (Gamma SI-130-10, ATI Industrial Automation Inc); for our purposes we are only interested in two of the six components (X and Y force components) and for these components the force range is $\pm$ 130 N, with a resolution of about 1N. In a following section we show that such force sensor readings can be reconstructed with great accuracy from the motor drive and the kinematic/dynamic model of the manipulandum.

Safety features must be integrated in the control in a secure way. The control unit of each motor has input lines that control the emergency stops, independently of the control software. We designed a simple electronic interface that can activate these lines either manually, by means of an emergency pushbutton, or automatically, when the robot arm reaches one of the four mechanical stops, which mechanically constrain the movement of the linkages to the designed workspace.

3. Implementation of Virtual Haptic Objects (VHOs)

In general, the goal of haptic rendering is to simulate the interaction of the user with a virtual haptic object. This interaction occurs at an ideal point, designated as Haptic Interaction Point (HIP), as if the user were touching the object though a short stick. The HIP corresponds to the position of the grasped handle and the contact forces felt by the user are reflected on the HIP according to the haptic rendering scheme described in the previous section, i.e. impedance control. The spatio-temporal pattern of the reflected forces depends on the attributes of the virtual object, which may emulate familiar physical objects, in agreement with classical mechanics, or may implement unfamiliar force fields, specifically designed for inducing motor learning and/or monitoring the modifications of the human mechanical impedance as learning proceeds.

The virtual haptic object is emulated by calculating the interaction forces, which are consistent with the actual movement: $F_{\mathrm{Haptic}}(t) = f(x_{HIP}(t))$. For example, suppose that the virtual object is a ball of radius R_B, mass m_B and elastic coefficient K_B and let us designate with x_B the position of its center. The corresponding haptic force can be programmed as follows:

get x_{HIP}
if $|x_{HIP} - x_B| > R_B$ then $F_{\mathrm{Haptic}} = 0$ else
  $F_{\mathrm{Haptic}} = K_B(x_{HIP} - x_B)$;
end

and this force is reflected to the HIP by activating the motors according to the following equation:

$$T_{\mathrm{motor}}(t) = J^T(q) F_{\mathrm{Haptic}}(t) \tag{4}$$

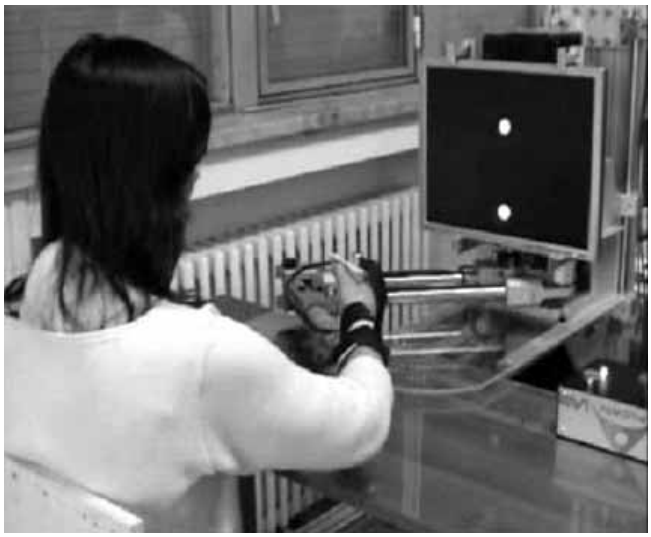

Fig. 9. Implemented prototype of *Braccio di Ferro*.

Together, the functions above can be implemented as a Simulink® impedance control block to be run on the target machine. Moreover, the same force, with a minus sign, will appear in the dynamics of the ball: $F_{\mathrm{Ball}} = -F_{\mathrm{Haptic}} = m_B \ddot{x}_B$. The numerical integration of this equation (typically by means of another Simulink® block) allows to update in real-time the motion of the virtual ball on the screen.

In general, a VHO is a complex reactive system, in which we may distinguish two basic components: 1) a haptic component; 2) a visual (or, more generally, audio-visual) component. The latter one may include motion/deformation of the virtual object, motion of the user's HIP, visualization of the interaction force, etc. Visual objects are represented by using the Matlab® Virtual Reality Toolbox, which is based on the Virtual Reality Modeling Language (VRML) [8].

The logical structure of a protocol of robot therapy, which includes the interaction between the haptic and the visual part of a VHO, is described by a finite state machine, i.e. an event-driven (reactive) system, implemented by means of another Matlab® tool, namely Stateflow®. This tool, which uses a variant of the finite state machine notation established by Harel [9], is a graphical representation of a finite state machine, where states and transitions form the basic building blocks of the system. Stateflow provides a block that can be included in a Simulink model. The collection of Stateflow blocks in a Simulink model is the Stateflow machine. Additionally, Stateflow enables the representation of hierarchy, parallelism, and history. Figure 8 shows a block diagram of the overall virtual haptic system.

4. Experiments and tests

After construction of the prototype system described in the previous sections (Fig. 9) we carried out a number of experiments and tests in order to verify the design assumptions.

4.1. Positional and torque calibration

The positional accuracy of the kinematic model of the manipulandum was evaluated by positioning the end-effector according to a 1 cm grid covering the workspace and comparing the nominal position

Table 3
Estimated parameters of the dynamical model of *Braccio di Ferro*

α_{11} [kgm^2]	α_{22} [kgm^2]	β_1 [kg]	β_2 [kg]	γ_1 [Nms]	γ_2 [Nms]	δ_1 [Nm]	δ_2 [Nm]
0.96 ± 0.01	0.117 ± 0.001	0.246 ± 0.01	0.038 ± 0.005	0.15 ± 0.09	0.009 ± 0.06	0.02 ± 0.01	0.03 ± 0.0071

with the position predicted by the kinematic model. The average positional error, which depends on the angular discretization induced by the encoders, the uncertainty of the reference angular values of the motors, the mechanical imprecisions and uncertainties related to the mechanical linkage, was estimated to be 0.37 ± 0.27 mm over the whole workspace.

The next step was to calibrate the current-torque conversion factor, also indicated as torque constant K_T. For the selected motors this factor has a nominal value of 9.66 Nm/A $\pm$ 10%. In order to carry out the calibration, we mounted the force sensor on the robot handle and we clamped it, while driving the motor currents I_m according to the following equation:

$$T = K_T I_m = J(q)^T F \tag{5}$$

The force vector F was generated with growing intensity (from 0 to 10 N) in 32 equally spaced directions and we compared the planned forces (which depend on K_T) with the forces measured by the force sensor. A linear regression was carried out with K_T as unknown. The result of this calibration was that the two constants had to be decreased by 7.1%.

4.2. Identification of the dynamical model

Equation (B.1) describes the intrinsic dynamics of the robot. The inertial part, which includes the acceleration-dependent term and the Coriolis term, depends linearly on the four parameters indicated in equations B.2 and B.3: α_{11}, α_{22}, β_1, β_2. For simplicity, we eliminated gravity by operating on the horizontal plane. The friction part, which in Eq. (B.1) is indicated as $D(q, \dot{q})$, can be decomposed into two terms, viscous friction and dry friction:

$$D(q, \dot{q}) = \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \begin{bmatrix} \mathrm{sign}(\dot{q}_1) \\ \mathrm{sign}(\dot{q}_2) \end{bmatrix} \tag{6}$$

All together, we have 8 unknown constants that identify the dynamical model of the manipulandum. In order to estimate the constants, we applied pseudo-random torques to the free (ungrasped) manipulandum: filtered white noise plus an elastic component, which kept the handle in the neighborhood of the selected workspace point. We carried out the numerical differentiation of the digital encoder readings, using the Savitzky-Golay filter method [16], and finally performed a linear regression of the generated torques on the unknown parameters. Table 3 shows the values of the estimated parameters, averaged over different parts of the workspace. By using these parameters it is possible to reconstruct the motor torques, according to the model of intrinsic dynamics:

$$T_{\mathrm{intrinsic}} = f(q, \dot{q}, \ddot{q}; \alpha_{11}, \alpha_{22}, \beta_1, \beta_2, \gamma_1, \gamma_2, \delta_1, \delta_2) \tag{7}$$

Figure 10 compares the torque reconstructed according to the model above and the commanded torque and it shows that the error is very small. The correlation coefficient is 0.95 for one motor and 0.98 for the other. The estimated α and β parameters of Table 3 are rather close to the values calculated from the blueprints of the mechanical parts, shown in Appendix B. Moreover, the δ parameters, which express the global dry friction, are compatible with the upper limit of 0.18 Nm formulated in Section 2.2 for matching psychophysical data.

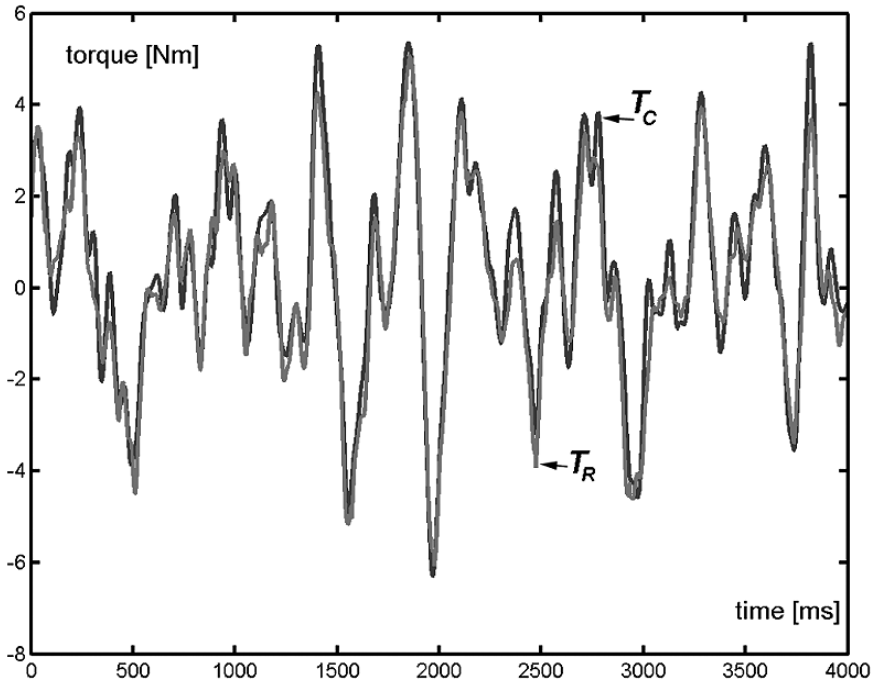

Fig. 10. Identification of the dynamical model of the robotic arm. T_C: commanded torque (filtered random noise + attractive force field). T_R: reconstructed torque after identification of the dynamical model. Only one motor is shown.

4.3. Use of the dynamical model for approximating the force sensor readings

The model identified in the previous section can be used for estimating the actual force at the HIP during a generic haptic interaction, without any need of the force sensor mounted on the handle. We can re-write Eq. (B.1) in the following form

$$T_{motor} = T_{\text{intrinsic}} + J^T F_H \tag{8}$$

from which we can recover the force exchanged at the HIP

$$F_H = \left[J^T(q)\right]^{-1} \left(T_{\text{motor}} - T_{\text{intrinsic}}\right) \tag{9}$$

In order to validate the accuracy of the procedure we performed different tests in which the subject was asked to interact with a VHO, while applying strong and quick shaking movements. During the experiments, we recorded the kinematic variables (q, $\dot{q}$, $\ddot{q}$) and the readings of the force sensor F_{sensor}; then we estimated the HIP force by means of Eq. (9). For a typical experiment, in which the VHO was a simple viscous-elastic system and the subject was asked to perform random shaking movements inside a 10 cm circle, Fig. 11 shows the time course of the measured sensor force, the reconstructed force and the haptic force, programmed by the VHO. As expected, the measured and reconstructed forces are very similar: correlation coefficient = 0.99. In conclusion, we have demonstrated that the readings of the force sensor can be approximated with very good precision by means of Eq. (9): the sensor is not necessary in normal operation and its role can be limited to the calibration phase. Getting rid of the sensor is also useful because its mass is comparable to that of the manipulandum.

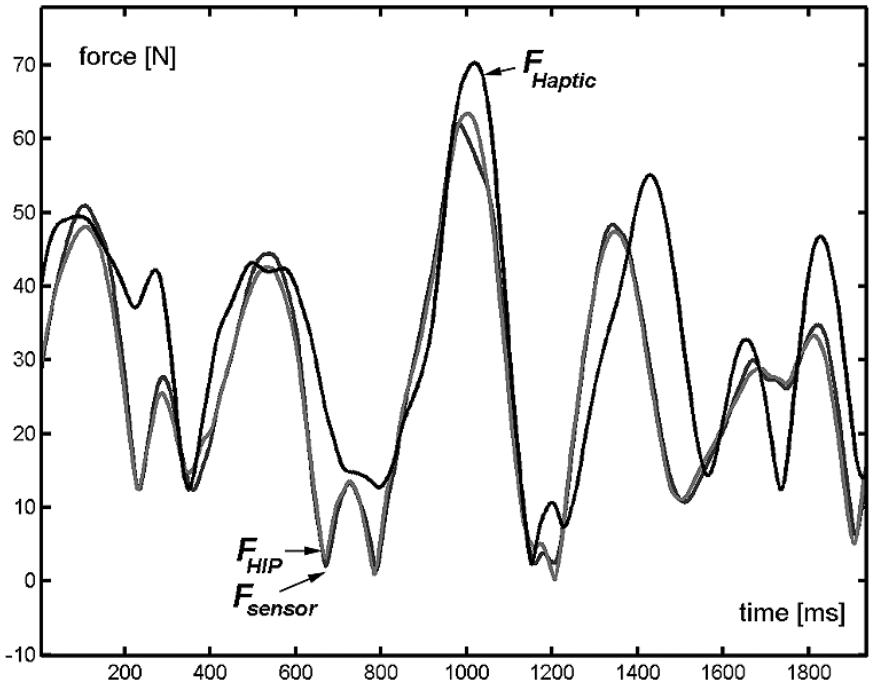

Fig. 11. Reconstruction of the interaction forces. F_{sensor}: actual interaction force measured by the force sensor; F_{Haptic}: force computed according to the virtual haptic model; F_{HIP}: reconstructed interaction force. F_{sensor}, F_{Haptic}, F_{HIP} are the magnitudes of the corresponding force vectors. Correlation coefficient between F_{sensor} and $F_{HIP} = 0.99$. The difference between F_{HIP} and F_{Haptic} is due to the fact that the intrinsic dynamics of the manipulandum subtracts a part from the commanded torque before being reflected at the HIP.

4.4. On-line compensation of intrinsic dynamics

The intrinsic dynamics of the manipulandum can be compensated on-line, at least partially, by using Eq. (9), which was introduced for off-line reconstruction of the interaction forces. In the on-line compensation the equation is applied by adding to the commanded torque, determined by the VHO, the torque calculated from the model of intrinsic dynamics, multiplied by a gain ρ that must be smaller than 1:

$$T_{\mathrm{motor}}(t) = J^T(q)F_H(t) + \rho\, T_{\mathrm{intrinsic}}(q, \dot{q}, \ddot{q}) \tag{10}$$

Ideally, the gain should be equal to 1 in order to have a perfect compensation, but in practice it is advisable to use a smaller value for avoiding the danger of instability of the control scheme. In fact, the torque computed according to the dynamical model is subtracted from the torque determined by the intrinsic dynamics. Due to the unavoidable inaccuracies of the model, the subtraction might be equivalent to a residual transfer function with poles in the right-hand part of the complex plane or very close to it, thus making the control system unstable. In order to understand how to choose the value of the gain ρ, we performed several experiments using the same paradigm described in the previous section and changing the value of the gain from 0 to 1. For each experiment we calculated the correlation between the force sensor reading and the designed interaction force, as shown in Table 4. As expected, the correlation coefficient increases with the gain up to a point ($\rho = 0.8$) beyond which the performance tends to deteriorate because the overall control is approaching a condition of instability. In any case, it should

Table 4
Correlation coefficient between the magnitude of the force sensor and the magnitude of the force output of the virtual object, as a function of the gain of the dynamic compensation control

Gain	Correlation coefficient
0	0.77
0.5	0.92
0.8	0.96
0.9	0.94
1	0.93

be observed that the optimal value of the gain (as regards the correlation coefficient) is likely to depend on the task and thus it is impossible to find a universal value. In practice, on-line compensation is not really necessary if the manipulandum has small inertia and small friction because the slight "distortion" of the virtual object is probably below the perceptual threshold, according to the literature on haptic psychophysics quoted above.

Apart from the stability considerations, the on-line compensation is intrinsically more inaccurate than the off-line algorithm, described in the previous section, because it cannot use the non-causal methods (like the Savitzky-Golay filter) for estimating the first two time derivatives of the motor angles and this introduces noise and delay particularly in the estimate of the acceleration.

4.5. Estimating the sensitivity of the haptic interface at very low levels of force

In order to evaluate how well the system behaves when very small interaction forces are involved, we implemented a "fragile membrane touching" VHO, which simulates a fragile elastic flat surface: the task for the user is to touch the membrane and slide over it, without loosing contact but without breaking it. The membrane is represented, visually, as a line on the visual display and, haptically, as an elastic contact force, linearly related to the distance between the membrane and the actual HIP-position when the latter is beyond the former one. The position of the HIP is visualized as a moving circle and after contact the circle color is changed, keeping the new color as long as contact is maintained. Moreover, the shape of the circle is flattened as a function of the contact force and fragility is implemented as a force threshold: when the force goes beyond such threshold the virtual membrane is "broken". The task of the subject is to reach the membrane and then slide along it without breaking. Remarkably, the contact force could not be measured with the force sensor because its range of measure is insufficient; we recovered the contact force from the commanded torque according to the previously described algorithm. Figure 12 shows an example of performance. The task was repeated with decreasing values of the force threshold and we found that it could be carried out in a reliable way, without breaking the membrane, for values as low as 1–1.5 N. Below that level the subjects could sense the membrane and the "breaking event" but were unable to carry out the task. Considering that frictional forces of the manipulandum were evaluated to be well below 1 N we may conclude that the force resolution of the system is sufficient to provide an interaction down to the psychophysical haptic threshold [4].

5. Concluding remarks

At the conclusion of the design, construction, and test of *Braccio di Ferro*, we are confident that its operational range is consistent with the typical human motions of the upper limb and thus is fit for robot

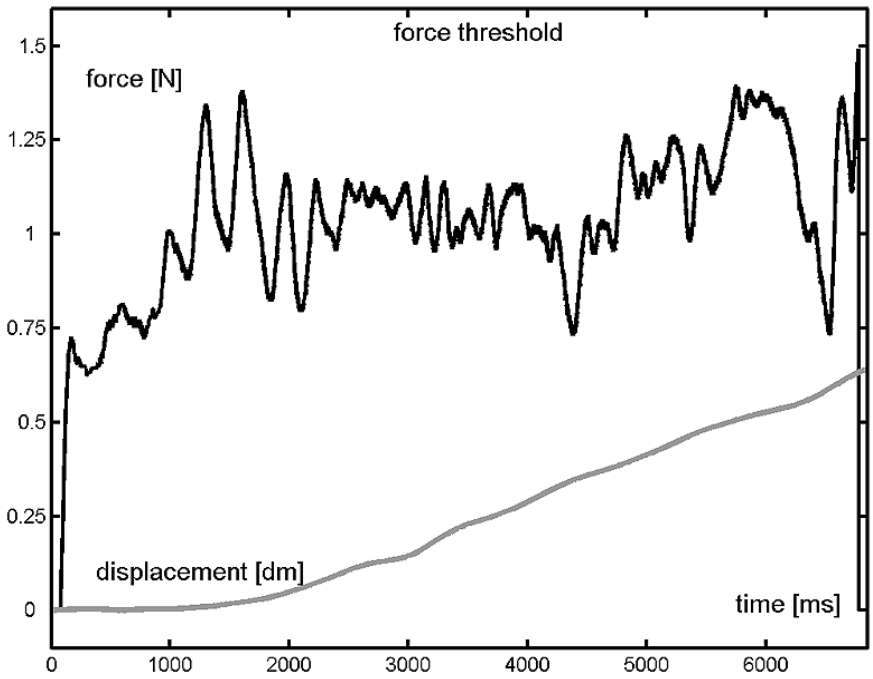

Fig. 12. Example of performance in the "fragile membrane touching task". Force threshold: breaking force of the membrane; actual force at the HIP; lateral displacement along the membrane. In the plotted performance, the membrane is broken in the end.

therapy. From this we are going to plan motor learning studies and design motor rehabilitation protocols that can exploit the features of the system. *Braccio di Ferro* is an open system and, in the spirit of open source design, it is intended to foster the dissemination of robot therapy. We are eager to share experience with others who are willing to purse this line of applied research.

Acknowledgments

This work was supported by a FIRB2001 grant awarded by the Ministry of University to P. Morasso.

Appendix A: Kinematic analysis of the manipulator

The kinematics of the robot is defined by the following equations, which give the coordinates of the end-effector, in the robot coordinate system, as a function of the motor angles:

$$\begin{cases} x_r = L_1 \cos q_1 + L_2 \cos q_2 + L_3 \cos(q_2 + q_3) = L_1 \cos q_1 + L_a \cos q_2 - L_b \sin q_2 \\ y_r = L_1 \sin q_1 + L_2 \sin q_2 + L_3 \sin(q_2 + q_3) = L_1 \sin q_1 + L_b \sin q_2 + L_b \cos q_2 \end{cases} \tag{A.1}$$

where $L_a = L_2 + L_3 \cos q_3$ and $L_b = L_3 \sin q_3$. From this it is immediate to calculate the Jacobian matrix of the kinematic transformation:

$$J = \begin{bmatrix} -L_1 \sin q_1 & -L_a \sin q_2 - L_b \cos q_2 \\ +L_1 \cos q_1 & +L_a \cos q_2 - L_b \sin q_2 \end{bmatrix} \tag{A.2}$$

which allows expressing the following relationships: between rotation speed of the motors and hand speed

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \tag{A.3}$$

and between the hand force and motor torque:

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = J^T \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{A.4}$$

In particular, it is of interest to consider, for each point of the workspace, the size and orientation of the "force ellipse", which depicts the end-effector force determined by a torque vector of constant magnitude (e.g. the maximum torque):

$$\underline{\text{Force ellipse}}: \; T^T T \leqslant T^2_{\max}; \; F^T [J J^T] F \leqslant T^2_{\max} \tag{A.5}$$

This ellipse is used for evaluating the degree isotropy of the robot in the workspace. It is also of interest to consider the "velocity ellipse", which plots the end-effector velocity determined by a motor rotational velocity of constant magnitude:

$$\underline{\text{Velocity ellipse}}: \; dq^T dq \leqslant dq^2_{\min}; dx^T [J J^T]^{-1} dx \leqslant dq^2_{\min} \tag{A.6}$$

There is a duality between the force and velocity ellipses: they have the same shape (ratio of the eigenvalues) but are rotated by 90 deg, one with respect to the other. Since the same ellipse describes the relation between small displacements instead of velocities, we may apply it to the smallest displacements of interest in order to have an estimate of spatial resolution.

Appendix B: Dynamic analysis of the manipulator

The manipulandum is characterized by the following dynamic equations

$$T_{\text{motor}} = I(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + D(q,\dot{q}) + J(q)^T F_H \tag{B.1}$$

where T_{motor} is the 2D vector of motor torques; $I(q)$ is the inertia matrix of the arm; $C(q,\dot{q})$ is a matrix related to Coriolis torques; $G(q)$ is the term due to gravity; $D(q,\dot{q})$ is a dissipative friction term, and F_H is the external force, applied by the subject to the handle or, with a minus sign, by the handle to the subject. The first 4 terms on the right-hand side of the equation correspond to the intrinsic dynamics of the manipulator ($T_{\text{intrinsic}}$). The last term corresponds to the haptic interaction (T_{haptic}). Thus, the equation above can also be written as $T_{\text{motor}} = T_{\text{intrinsic}} + T_{\text{haptic}}$.

Inertial terms The first two terms on the right-hand side of the equation represent the inertial part of the intrinsic dynamics and the two matrices have the following structures [22]

$$I(q) = \begin{bmatrix} \alpha_{11} & \beta_1 \cos q_{12} + \beta_2 \sin q_{12} \\ \beta_1 \cos q_{12} + \beta_2 \sin q_{12} & \alpha_{22} \end{bmatrix} \tag{B.2}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & \dot{q}_2(\beta_1 \sin q_{12} - \beta_2 \cos q_{12}) \\ \dot{q}_1(-\beta_1 \sin q_{12} + \beta_2 \cos q_{12}) & 0 \end{bmatrix} \quad \text{(B.3)}$$

where $q_{12} = q_1 - q_2$; the α's and β's are parameters which have the dimensions of an inertia moment (kgm^2) and depend upon the geometry of the linkage (defined in the kinematic analysis considered in the previous section) and the distribution of masses. From the blueprints of the mechanical parts we estimated the following approximate values for the parameters in Eqs (B.2) and (B.3): $\alpha_{11} = 0.89$ kgm^2, $\alpha_{22} = 0.12$ kgm^2, $\beta_1 = 0.25$ kgm^2, $\beta_2 = 0.05$ kgm^2. The experimentally estimated values of the same parameters are shown in Table 3.

In order to estimate the range of inertial torques, we simulated typical "fast" movements in different parts of the workspace: straight movements, with an amplitude of 15 cm, a bell-shaped velocity profile [15] and a duration of 200 ms. The corresponding peak velocity is 1.5 m/s and the peak acceleration is 23.5 m/s^2, which determine the following ranges of angular speed and acceleration:

$$\dot{q}_1 : \ 1.91 \ \pm \ 0.04 \ rad/s; \ \ \dot{q}_2 : \ \ 2.33 \ \pm \ 0.05 \ rad/s$$
$$\ddot{q}_1 : \ 27.74 \ \pm \ 0.61 \, rad/s^2; \ \ \ddot{q}_2 : \ \ 33.55 \ \ \pm \ 0.72 \ rad/s^2$$

and inertial forces:

$$T_1^{\text{inertial}} : \ 34.82 \pm \ 9.6; \ T_1^{\text{Coriolis}} : \ 3.26 \ \pm 1.12 \ \ \rightarrow \ \ T_1 = 34.73 \pm 10.87 \text{ Nm}$$
$$T_2^{\text{inertial}} : \ 5.64 \pm 1.64 \ ; \ \ T_2^{\text{Coriolis}} : 2.06 \pm 0.73 \ \ \rightarrow \ \ T_2 = 5.63 \pm 1.63 \text{ Nm}$$

Foot notes

[1]The considered "fast" movements have an amplitude of 15 cm and a duration of 200 ms. The peak acceleration is 23.5 m/s^2 and the corresponding peak force, for a virtual mass of 5 kg, will be $5 \cdot 23.5 = 117.5$ N.

[2]For slow movements the torques predicted by the dynamical model are negligible and thus the application of the force reconstruction method is trivial.

[3]$G(q)$, which is a function of the robot joint angles, is multiplied by the cosine of the angle between the vertical vector and the normal vector to the operational plane. Thus, it is null when the operational plane is horizontal.

References

[1] H. Asada and J.J.E. Slotine, *Robot Analysis and Control*, New York, John Wiley and Sons, 1986.

[2] C. Basdogan and M.A. Srinivasan, in: *Haptic Interfaces. Handbook of Virtual Environments*, K. Stanney, ed., Chapter 6, 2002, 117–134, London, Lawrence Earlbaum.

[3] J. Biggs and M.A. Srinivasan, in: *Haptic Interfaces, Handbook of Virtual Environments*, K. Stanney, ed., Chapter 5, 2002, pp. 93–116, London, Lawrence Earlbaum.

[4] K.R. Boeff and I. Kaufman, eds, *Handbook of Human Perception and Human Performance*, Chapter 12, vol. 1, Cutaneous sensitivity (C.H. Sherrick and R.W. Cholewiak); Chapter 13, vol. 1, Kinesthesia (F.J. Clark and K.W. Horch); Chapter 31, vol. 1, Tactile perception (J.M. Loomis and S.M. Lederman). New York: J. Wiley and Sons.

[5] J.A. Cozens, Robotic assistance of an active upper limb exercise in neurologically impaired patients, *IEEE Transactions on Rehabilitation Engineering* **7** (1999), 254–256.

[6] H. Gomi and M. Kawato, Human arm stiffness and equilibrium point trajectory during multijoint movement, *Biological Cybernetics* **76** (1997), 163–171.

[7] S. Grange, F. Conti, P. Helmer, P. Rouiller and C. Baur, The Delta Haptic Device as a nanomanipulator, in: *SPIE Microrobotics and Microassembly III*, Boston, MA, (2001).
[8] J. Hartman and J. Wernecke, *The VRML 2.0 Handbook, Building Moving Worlds on the Web*, Addison Wesley, Reading (MA), 1996.
[9] D. Harel, Statecharts: A Visual Formalism for Complex Systems, *Science of Computer Programming* **8** (1987), 231–274.
[10] H.I. Krebs, N. Hogan, B.T. Volpe, M.L. Aisen, L. Edelstein and C. Diels, Overview of clinical trials with MIT-MANUS: a robot-aided neuro-rehabilitation facility, *Technology and Health Care* **7** (1999), 419–423.
[11] H. Krebs, N. Hogan and M. Aisen, Volpe BRobot-aided neurorehabilitation, *IEEE Trans. Rehabilitation Engineering* **6** (1998), 75–87.
[12] U. Kühnapfel, H.K.C. Cakmak and H. Maas, Endoscopic surgery training using virtual reality and deformable tissue simulation, *Computer & Graphics* **24** (2000), 671–682.
[13] P. Lum, S. Lehman and D. Reikensmeyer, *The use of a robotic device for post-stroke movement therapy*, Proc. of the International Conference on Rehabilitation Robotics, ICORR'97, 1997.
[14] S. Micera, M.C. Carrozza, E. Guglielmelli, L. Barboni, C. Freschi and P. Dario, *On the use of robotic devices for neurorehabilitation after stroke*, Proc. of 11th International Conference on Advanced Robotics, Coimbra, 2003, 185–190.
[15] P. Morasso, Spatial control of arm movements, *Experimental Brain Research* **42** (1981), 223–227.
[16] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press.
[17] J. Rosen, B. Hannaford, P. MacFarlane and M.N. Sinanan, Force controlled and teleoperated endoscopic grasper for minimally invasive surgery – experimental performance evaluation, *IEEE Transactions on Biomedical Engineering* **46** (1999), 1212–1221.
[18] V. Sanguineti, P. Morasso, L. Baratto, G. Brichetto, G.L. Mancardi and C. Solaro, Cerebellar ataxia: Quantitative assessment and cybernetic interpretation, *Human Movement Science* **22** (2003), 189–205.
[19] V. Sanguineti, M. Casadio and P. Morasso, *Haptic interface for neurorehabilitation*, Proceedings of the 2nd International Symposium on Measurement, Analysis ans Modeling of Human Functions (IMEKO TC 18), Genoa, Italy, June 14–16, 2004, 167–170.
[20] S.H. Scott, Apparatus for measuring and perturbing shoulder and elbow joint positions and torques during reaching, *Journal of Neuroscience Methods* **89** (1999), 119–127.
[21] R. Shadmehr and F.A. Mussa-Ivaldi, Adaptive representation of dynamics during learning of a motor task, *Journal of Neuroscience* **14** (1994), 3208–3224.
[22] L. Stocco, S.E. Salcudean and F. Sassani, Fast constrained global minimax optimization of robot parameters, *Robotica* **16** (1998), 595–605.
[23] T. Tsuji, P. Morasso, K. Goto and K. Ito, Human hand impedance characteristics during maintained posture, *Biological Cybernetics* **72** (1995), 475–485.
[24] B.T. Volpe, H.I. Krebs, N. Hogan, L. Edelsteinn, C.M. Diels and M.L. Aisen, Robot training enhanced motor outcome in patients with stroke maintained over 3 years, *Neurology* **53** (1999), 1874–1876.
[25] T. Yoshikawa, Manipulability of robotic mechanism, *The International Journal of Robotics Research* **4** (1985), 3–9.