

**Emanuele Ruffaldi\***  
**Paolo Tripicchio**  
**Carlo Alberto Avizzano**  
**Massimo Bergamasco**

PERCRO Lab  
Scuola Superiore S. Anna  
Pisa 56100, Italy

# Haptic Rendering of Juggling with Encountered Type Interfaces

---

## Abstract

Haptic interaction in a virtual world can be tool mediated or direct; and, among direct interactions, the encountered haptic interfaces provide physical contact only when there is contact with a virtual object. This paper deals with the haptic rendering of the catching and throwing of objects by means of this type of interface. A general model for the rendering of the impact is discussed with the associated formalism for managing multiple objects and multiple devices. Next, a key parameter for simulating the impact is selected by means of a psychophysical test. Finally, a working system is presented with the application of the rendering strategy to the case of haptic juggling, showing the possibility of effectively performing basic juggling patterns with two balls.

## I Introduction

Over the centuries, the sense of touch has contributed to the growth of human civilization and technology, giving human beings the ability to interact and modify the surrounding world. The relevance of this sense is reflected in the interest in haptic perception and feedback in numerous fields of research and especially in the ones connected to virtual environments. The objective of research in this domain is to combine an understanding of human perception with technological advancement aimed at providing realistic feedback in the exploration of virtual worlds. A large part of this research deals with the kinesthetic interaction of contact that is highly connected with the type of haptic interface employed.

Haptic interaction in virtual environments is usually mediated by a tool handle that is used to display contacts with virtual objects. In tool-based interfaces, users hold a stylus, a knob, or other simple object that is represented in the virtual environment by a functionally equivalent virtual shape. By means of this handle, users interact and perceive haptic information about the shape and texture of virtual objects present in the virtual world. In this case, the perception of contact with the virtual object is mediated by the tool resulting in an indirect contact between the user and the virtual object. The opposite of tool-based mediation is direct contact, which happens when there is no mediation by the tool and the user touches a physical interface that transmits a contact with the virtual object.

The mediation of the interaction is a property of several types of interfaces. In particular, directness can be provided both by multipoint exoskeletons (Bergamasco, 1997), and by encountered type interfaces. This work focuses on the latter type because they are transparent by definition.

An encountered-type device is not in contact with the user all the time, but instead it tracks selected user body parts and provides contact only when there is an interaction with a virtual object. This kind of device has been proposed by McNeely (1993) and Tachi, Maeda, Hirata, and Hoshino (1994) independently. In the literature, this interface type has been used to render limited accessible portions of single or multiple objects (Gruenbaum & McNeely, 1997) as well as to display a large object approximating local regions with surface patches (Tachi, Maeda, Hirata, & Hoshino, 1995). In particular, this approach allows the system to provide a real touch sensation and it frees the user from coupling constraints with the haptic device, making the experience, specifically in application domains, more immediate and realistic.

Multipoint and customized end-effectors have been presented that can produce the correct sensation at the encounter stage, but only for fixed or slowly moving objects. An interesting investigation can be made on the interaction with objects having dynamic properties, and indeed the specific objective of the research discussed in this paper is on haptic rendering of the catching and tossing of dynamic objects. This type of interaction poses questions about the integration of encountered devices with dynamic simulation, the realism of the impacts, and the visuohaptic coherence of the objects.

This paper contributes to the domain of encountered interfaces by first introducing a theoretical control strategy for dealing with the rendering of impacts, based on the evaluation of impact perception produced with the haptic interface. Then, the theoretical control strategy is applied in a case study for a two-ball juggling simulation implemented by means of a dual point 3-DOF haptic interface co-located with a graphical display. Finally, the paper presents the resulting working system with an evaluation using four subjects.

The next section introduces an overview of the relevant work in this domain. Then the general model

of encountered interfaces is discussed, followed by a model of the impact. This theoretical model has been tested with a perceptual evaluation of the impact by means of the haptic interface. Later, the juggling case is introduced, describing first the interaction paradigm, the system setup, and detailed control logic. Finally, conclusions and future directions are presented.

## 2 Related Work

In this section, related work is presented in the domain of haptic interaction and a synthetic comparison of the features characteristic of encountered haptic interfaces by first addressing the original works on encountered haptic interfaces and then examples about more recent fingertip haptics. This paper is put in context of this research by discussing the haptic display of dynamic objects.

Haptic rendering of contact is a key aspect for interaction with objects. In particular, research has focused on rendering stable contact with objects during surface exploration and manipulation, while the first contact with objects is still an open issue. In tool-mediated interfaces, for example, users perceive virtual tool inertia when the tool is moved in free space under the limits of the gravity and inertia compensation of the interface. Instead, when the virtual tool impacts a virtual object, a force has to be rendered based on the contacting virtual surfaces and material properties. In the standard haptic rendering algorithms, this force is proportional to the depth of penetration by means of virtual stiffness inducing an impact force that is quite low. A solution to the rendering of first contact has been proposed by the event-based approach (Fiene, Kuchenbecker, & Niemeyer, 2006) in which, during the first milliseconds of contact, the force is generated in an open loop following a force profile that has been recorded for the specific material. This solution is effective for tool-mediated interfaces because it allows the rendering of the impulse of surface contact.

A totally different approach for contact rendering was pursued by the encountered-type haptic interfaces. Since their introduction (McNeely, 1993; Tachi et al.,

**Table 1.** *Encountered and Hand-held Interfaces*

Feature	Encountered	Hand-held
Surface	Specific shape rendering	Any surface (also volumes)
Interaction	Impact and contact	Any force (e.g., guidance)
Control	Requires external tracking	Force and position from device
Transparency	Almost complete	Limited by mechanics and control
Dynamics	Limited by device	Limited by coupling with user
Multi-users	Device sharing	One device per user
Multi-devices	Involves path planning	Limited by device encumbrance

1994) the distinctive features of this type of interface have been to be totally transparent when the user is not interacting with virtual objects, and, at the same time, to realistically render first contact with objects' surfaces. The interaction with the user is provided by a robotic end-effector with a given shape, such as plates for finger and palm contacts, or such as knobs for grasping (Yokokohji, Hollis, & Kanade, 1996). Furukawa, Inoue, Takubo, and Arai (2007) proposed a haptic display constituted of two robotic arms and a flexible sheet to represent virtual objects with different shapes. In these systems, the number of end-effectors is limited to the number of virtual objects and, for this reason, it is necessary to make the end-effector model different virtual objects. This is obtained by precise knowledge of the task or by tracking or predicting the user's motion. In this way, each end-effector is moved into the position of the virtual object in real space ready for contact with the user's hands or fingers. The presence of multiple end-effectors can introduce collisions among them or with the user's body, an issue that Yokokohji, Yoshikawa, and Kinoshita (2001) addressed by means of motion planning.

Table 1 shows, in a synthetic way, the major differences between encountered and conventional haptic interfaces.

A variation of encountered haptics that has to be mentioned is the one for fingertip haptics, in which the focus is the interaction between the fingertip and the virtual object. In Solazzi, Frisoli, Salsedo, and Bergamasco (2006) the device has a plate end-effector that is put in

contact with the fingertip with an angle that depends on the surface of the virtual object. Multiple finger interaction has been proposed by Yokokohji, Muramori, Sato, and Yoshikawa (2005), adopting a patch surface for every finger.

In the encountered interfaces presented so far, the end-effector represents a static virtual object: the end-effector is moved to the correct location before the user's contact and then it is ready for entering in contact or being grasped by the user. Dynamic objects have not been taken into account by research due to mechanical limitations and the limited presence of fast dynamic objects in many haptic contexts such as surgery and virtual prototyping. In the context of tool-mediated haptics, the issue of a dynamic object has been addressed for rendering deformable objects and for gaming purposes. For example, Knoerlein, Székely, and Hardens (2007) presented a system for augmented reality Ping-Pong using a tool-based interface. Adopting the encountered paradigm, haptic devices have to fulfill some requirements to properly display to the user the correct perceptual information and to make the user feel the contact is as realistic as possible. An important issue regards the correct display of contact orientation during the overall interaction. It is then necessary to solve possible discontinuities that arise due to the orientation's changes of the mesh normals. The interacting surface should be capable of rendering local curvatures within the objects, leaving the interaction as transparent as possible, so the choice of interacting surface is an important aspect in the interface design phase.

**Table 2.** *Effects of Reality Combinations and Visual Medium for Haptic Interaction*

Medium/reality	Virtual reality	Augmented reality
Head-mounted display	No occlusion. Requires tool or hand display	Camouflage needed. Possible slow frame rate
Screen between eyes and hands	Partial occlusion. Requires tool or hand display. Possibly limits workspace	Camouflage needed
Screen behind hands	Occlusion is an issue with co-location. Offset with display of tool and hand an option	Occlusion

The investigation of this work does not deal with the surface properties of static objects but instead with the rendering of dynamic objects, in particular in the domain of catching and tossing objects by means of encountered type interfaces. Catching and tossing can be applied to simulation and training of juggling in which these actions play a fundamental role (Sakaguchi, Masutani, & Miyazaki, 1991). In addition, existing juggling trainers are based purely on motion tracking and lack force feedback for rendering the ball's weight and impact (Marshall, Benford, & Pridmore, 2007).

This work addresses the dynamic aspect of the encountered interfaces by first discussing the problem of catching with a specific focus on the perception of the impact, and then by presenting an integrated system for juggling.

Before entering the specifics of the system, it is worth mentioning an aspect of haptic interaction that emerges with encountered haptics and that is taken into account by this work: visuohaptic coherence. The visual stimulus can be briefly organized around two criteria as summarized by Table 2. The first criterion is related to the amount of the real environment that is displayed to the user following the taxonomy by Milgram and Kishino (1994) that covers both augmented reality and virtual reality setups. The second is the medium that provides the stimulus such as a head-mounted display or a screen. It is relevant to make a distinction between the cases in which the screen is placed between the user's eyes and hands or if the screen is just behind the hands as in a typical monitor display or for projection screens. This

discussion on the visual aspect is relevant because haptic interfaces are physical objects that occupy physical space, possibly interfering with the virtual environment or even with the visual display of the virtual objects they are representing. In the case of virtual reality, there is no occlusion by definition but it is typically necessary to display the tool as being held by the user or by the user's hands. This visual stimulus can be provided with no occlusion issues by means of head-mounted displays and with good results if the screen is placed between the hands and the eyes.

When the virtual environment is based on the augmented reality concept, there is typically no need to have a virtual representation of the hands; but it is necessary to perform some camouflage of the haptic interface. This technique has been adopted by Yokokohji et al. (1996) with knob-based encountered interfaces using a screen between the hands and the eyes, or more recently by Cosco, Garre, Bruno, Muzzupappa, and Otaduy (2009) with a tool-based interface using a head-mounted display. The adoption of augmented reality techniques has the effect of reducing the frame rate of the visual display down to 25 fps (Cosco et al.).

In the case of a screen behind the hands, typical of projection walls and CAVEs, the haptic interface cannot be easily camouflaged; this induces two possible solutions. The first, and most typical, is the introduction of an offset between the haptic end-effector and the virtual objects on the screen, requiring a visual display of both tools and, in some cases, of the user's hands. The second one is the co-located approach in which the display sys-

tem is precisely calibrated to display virtual objects in a location that is aligned with the physical location of the haptic interface end-effector (Swapp, Pawar, & Loscos, 2006; Viciano-Abad & Reyes-Lecuona, 2008; Bianchi, Knoerlein, Székely, Harders, & Switzerland, 2006). The offset solution is good for tool-based interfaces because the hand is always in contact with the interface, translating the local physical motion into motion on the screen. With encountered interfaces, instead, the hands move in the physical space where the user thinks the object is, under the effect of the offset and the distortion of the visual display. This becomes particularly challenging with dynamic objects; for this reason, this work has adopted a co-located approach.

### 3 Encountered Interfaces for Dynamic Objects

This section presents some terminology for describing the modeling of encountered devices with a general configuration and the fundamental interaction scheme for performing the encountered contact rendering.

A key characteristic of encountered interfaces is their ability to impersonate different objects under the constraints of the mechanical workspace. In the general case of multiple end-effectors, the same object can be impersonated by different effectors at different times. This general configuration can be described by introducing the following notation: end-effectors  $E_i$  of the devices have an interaction shape  $sE_i$ , virtual objects  $V_j$  have shape  $sV_j$ , and the user's body parts involved in the interaction are identified by  $B_k$  with shape  $sB_k$ . In terms of reference frames, the real-world reference frame  $\Omega_R$  and the virtual-world reference frame  $\Omega_V$  are associated by means of an affine transformation.

In this general framework, the objects  $V_j$  have their own dynamics, and they can be paired to an end-effector  $E_i$  depending on the control policy. The pairing policy for the generic encountered task should take into account several aspects such as the prediction of the user's motion, the dynamic collision prediction for avoiding collisions between the end-effectors, and all of the user's body parts.

The control strategy can be represented as a combination of different states with different low-level control approaches. When two entities are paired and no contact with user body parts is occurring, the end-effector  $E_i$  should be controlled in a way to be prepared to enter in contact with the user. In encountered haptics with semi-static objects, the  $E_i$  is placed in the physical position  $x_{V_j}$  corresponding to the position of the virtual object in the real space waiting for contact with the user. In the case of dynamic objects, instead, it is necessary that the end-effector will arrive at the location of impact with the user's  $B_k$  with a given velocity that should be able to provide an impact impulse equivalent to the one of the virtual object at that time. That impulse is  $v_{E_i}(t^*) = kv_{V_j}(t^*)$  when  $x_{E_i}(t^*) = x_{B_k}(t^*)$ . For this reason, at the instant of pairing, the  $E_i$  should move into a resting position that allows the end-effector to acquire sufficient velocity depending on the mechanical and control characteristics. The time of contact  $t^*$  can be predicted based on the dynamic simulation of the objects, the specific task, and the tracking of the user's body motion.

The pairing behaves differently during contact with surfaces or with the grasping of the end-effector because in this case the  $E_i$  should present the same inertial properties of the paired object  $V_j$ . The dynamics of  $V_j$  are affected by the physical interaction of the user with the  $E_i$ . In this way, the user is able to push and move the virtual object around in the virtual world. As discussed later in the control section, Section 6.5, this work focuses on the specific case of two devices and two hands, allowing a simple but effective strategy. Generalized path planning is an open problem that requires further investigation (Shigeta, Sato, & Yokokohji, 2007).

From this overview of the interaction model, the discussion enters the specific case of modeling and rendering the impact with dynamic objects.

### 4 Impact with Encountered Haptics

This section discusses the formulation for the haptic rendering of the impact in encountered interfaces. The proposed formulation is based on balancing the

energy of the real and virtual impacts, resulting in a practical relationship for the control of the end-effector velocity before the impact. The following analysis and the perceptual experiments presented in Section 6 provide a justification for the proposed balancing between the virtual and real energies.

Adding dynamic properties to a virtual environment introduces a new set of non-negligible complexities. Dynamic objects can therefore require additional issues from haptic devices. In particular, there is the need to render a correct representation of the mass/inertia properties. Each object that interacts with the user shows its own mass and inertia that is perceived by the human body as a reflected force required to accelerate the body of the object. The correct representation of this force is fundamental to improve the transparency of the environment. The control of reflected inertia is made possible by means of closed loop feedback on the motor torque, which can partially compensate for or enhance the real inertia of the haptic device. The following notation is adopted:  $M(q)$  is the inertia matrix,  $C(q, \dot{q})$  is the Coriolis matrix,  $\tau$  is the motor torque,  $J(q)$  is the Jacobian of the robotic arm,  $F_{\text{ext}}$  is the exchanged force,  $g(q)$  is the gravity component,  $M_O$  is the mass of the virtual object, and  $x_O$  is the virtual object's position. The first step is to compare the force exchange term  $F_{\text{ext}}$  in the Lagrange formulation (Sciavicco & Siciliano, 2003) of the joint space dynamic model and then apply it to a slow velocity case:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + J^T F_{\text{ext}}, \quad (1)$$

with the dynamics of the virtual object:

$$F_O = M_O \ddot{x}_O. \quad (2)$$

It is then theoretically possible to derive a closed loop control algorithm that represents the desired object mass by equating  $F_{\text{ext}}$  with  $F_O$ :

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) - M_O J^T \ddot{x}_O. \quad (3)$$

Then, replacing the virtual object velocity  $\dot{x}_O$  with the one of the end-effector  $\dot{x}_E = J(q)\dot{q}$ , we obtain:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) - M_O J^T \left( J(q)\ddot{q} + \frac{dJ(q)}{dt} \dot{q} \right). \quad (4)$$

If the velocity is assumed to be small, then the Coriolis component and the derivative of the Jacobian can be neglected:

$$\tau = (M(q) - M_O J^T J)\ddot{q} + g(q). \quad (5)$$

This formulation is relatively easy to be integrated within the control loop algorithms and it allows an impact force  $F_{\text{ext}}$  to be generated that completely stops the virtual object. Unfortunately, two aspects limit the application of this rule. The first is associated with the performance of a haptic interface that is affected by factors such as interface dynamics, sensor quantization, mechanical impedance, and sampling of the controller. An impedance is considered achievable if it satisfies robustness properties such as passivity (Brown & Colgate, 1994).

#### 4.1 Impact Model

Velocity and position are relatively easy to display by enhancing the concepts discussed in the semi-static rendering design and providing an integration of the velocity and position of the objects as part of the reference control of device motion. However, tracking of velocities for both the haptic device and the objects are of high importance for a clean rendering at the instant of contact. At the point of contact, the controller should dissipate onto the human hand the kinetic energy existing in the moving object.

Particular attention should be given to energy transfer during impact when the high frequency of the velocity change may affect the real perception of mass. Two types of impact are likely to happen: bouncing and grasping impacts. Bouncing happens when the virtual object detaches immediately after impact; in this case, the portion of energy loss is proportional to a coefficient. Bouncing impact is not typical if it is considered hand manipulation or even catching; however, velocity and energy estimation can be carried out with an analysis similar to the one of an elastic impact, which is the focus of our discussion.

After impact, the object is grasped by the human hand with a consequent loss of overall energy that has to be dissipated to the user. However, almost all haptic devices

have a mechanical bandwidth that is well below the typical frequencies expressed during impact. Therefore, even if the control loop compensates and exactly transfers the energy that should have been delivered during the impact, this effect would result in unrealistic phenomena, given that the energy would have been transferred with timing consistent with the mechanical bandwidth (typically from 5 to 40 Hz in common haptic interfaces).

Instead of controlling the end-effector by means of the virtual mass, this work proposes to shore up the impact energy by an explicit pre-warping of the haptic velocity. Therefore, by considering a frontal impact of the object with the hand, the virtual energy loss during impact can be defined as:

$$E_V = \frac{1}{2}M_O v_{O^-}^2 + \frac{1}{2}M_H v_{H^-}^2 - \frac{1}{2}(M_O + M_H)v_+^2, \quad (6)$$

where  $M_O$  and  $v_{O^-}$  are the mass and modulus of the velocity of the virtual object before impact, while  $M_H$  and  $v_{H^-}$  are the same for the hand. The variable  $v_+$  is the modulus of the velocity of the combined virtual object and hand after impact. Due to the absence of bouncing,  $v_+$  can be calculated in a straightforward manner as:

$$v_+ = \frac{M_H v_{H^-} + M_O v_{O^-}}{M_H + M_O}. \quad (7)$$

The real energy loss due to the impact of the hand with the haptic interface can be computed as follows, assuming that the post-impact velocity of the haptic interface is the same  $v_+$  as the combined hand and virtual object:

$$E_R = \frac{1}{2}M_{hi}^* v_{hi^-}^2 + \frac{1}{2}M_H v_{H^-}^2 - \frac{1}{2}(M_{hi}^* + M_H)v_+^2, \quad (8)$$

where  $M_{hi}^*$  is the apparent mass at the end-effector expressed in Cartesian space accordingly to Khatib (1986). Assuming no dissipation in the impact, it is possible to equate  $E_R$  and  $E_V$ , and after some algebra we obtain the following equation for the haptic velocity  $v_{hi}$  before impact:

$$v_{hi^-} = v_{O^-} * \sqrt{\left| \frac{M_O}{M_{hi}^*} + \left(1 - \frac{M_O}{M_H^*}\right) \left(\frac{v_+}{v_{O^-}}\right)^2 \right|}. \quad (9)$$

The relationship between the velocity of the virtual object and the end-effector can be expressed as

$v_{hi^-} = k v_{O^-}$ , where the  $k$  term defined below can be considered as a correction term for adapting the mass of the virtual object to the mass of the haptic interface:

$$k = \sqrt{\left| \frac{M_O}{M_{hi}^*} + \left(1 - \frac{M_O}{M_H^*}\right) \left(\frac{v_+}{v_{O^-}}\right)^2 \right|}. \quad (10)$$

The  $k$  factor is a function of all the involved masses and also of the relationship between the resulting velocity  $v_+$  and the initial velocity  $v_{O^-}$  of the virtual object. It is useful to highlight two singular cases of Equation 10. Figure 1 represents the values of  $k$  when varying the mass ratio and the velocity ratio. In the contour plot, two conditions with identical  $k$  factors equal 1: when the mass of the haptic device equals the mass of the virtual object, and when the velocity after the impact is the same of the object. The zone with zero values represents the inversion condition that delimits impacts in the same direction from impacts from opposite directions and it is expressed by the following equation:

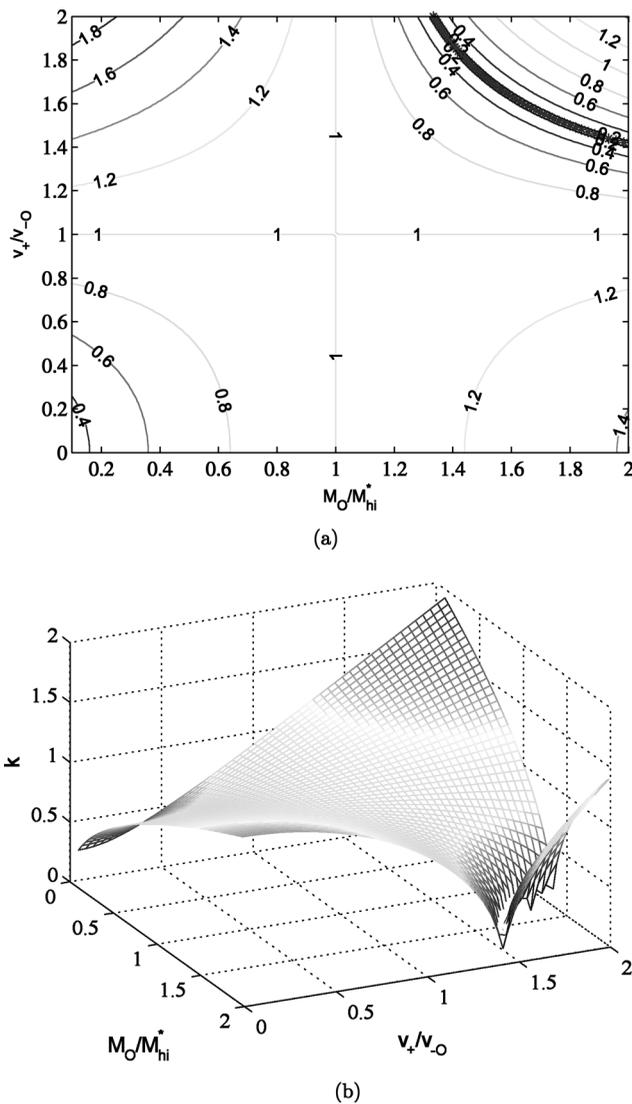
$$\alpha = \frac{\beta^2}{\beta^2 - 1}, \quad (11)$$

where  $\alpha = \frac{M_O}{M_{hi}^*}$  and  $\beta = \frac{v_+}{v_{O^-}}$ .

Finally, in the specific case of an impact producing the halt of the hand  $v_+ \approx 0$ , the  $k$  factor becomes independent of the incoming velocity and depends only on the ratio between the masses. Using a tennis ball with mass (58 g) and a haptic interface with apparent mass (223 g) a theoretical ratio  $k_{dev}$  of 0.51 is obtained. This factor is a numeric estimate of the ratio between velocities that should be able to produce a transfer of kinetic energy from the haptic interface to the hand equivalent to the one of the ball. Since this is a numeric estimate, it is important to know whether the introduction of the  $k$  factor helps the user perceive the impact as equivalent to a real object. For this reason, a perceptual evaluation is performed as discussed in Section 5.

## 5 Perceptual Evaluation

A two-phase perceptual evaluation was designed for calibrating the device impact force to the perceived impact intensity. The first phase has the purpose of esti-



**Figure 1.** Values of factor  $k$  with respect to the velocity ratio  $v_+/v_{O-}$  and the mass ratio  $M_O/M_{hi}^*$ . The factor is 1 in two occurrences, when masses are the same and when velocities are the same. The factor becomes zero at the point of velocity inversion shown in gray. (a) Contour plot of the  $k$  factor based on ratio of the velocities and masses. (b) Surface plot of the  $k$  factor equivalent to the contour above.

mating the relationship between the real impact and the perceived impact intensity. The second phase has the objective of investigating the relationship between perceived impacts generated by real and virtual objects rendered by means of a haptic interface. Specifically, the second phase offers the possibility of understand-

ing whether the perceived difference between real and virtual objects follows the relationship between impact velocities  $v_{hi-}/v_{O-}$  introduced in Equation 10.

In the literature, some studies on weight perception give results for a specific testing condition and reference weights, but not for impact forces (Kreifeldt & Chuang, 1979; Ross, Brodie, & Benson, 1986). For this reason, in the context of this work, we decided to perform a test to investigate the perception threshold of users with real tennis balls.

The relation between the physical intensity of a stimulus and the corresponding intensity perceived by a user can be described by a psychometric function. This function can be obtained by means of a psychometric procedure in which subjects are observed by looking at their responses to a sequence of stimuli. Thresholds that specify the psychometric function's location along the stimulus axis are then usually compared to analyze different stimulus conditions (Hill, 2001). In a forced-choice design of psychometric experiments, a pair of stimuli are presented to the user and one of them is a reference stimulus to be identified. In particular, this work adopted a two-alternative forced-choice (2-AFC) design choosing the reference stimulus to be a ball falling from a reference height controlled by means of a reflective motion capture device. In every step of the test, a real ball was released from two different heights and the user was asked which one gave the feeling of strongest impact. Several preliminary trials were necessary to identify the correct positioning of the subject and the placement of the hand on the table, in particular asking the subject to not have the hand rigid and to not have additional collateral impacts on the table. In 2-AFC tests, the comparison alternately uses a reference height against a height computed at every step using an algorithm. The algorithm used to perform the generation of the second stimulus and the evaluation is QUEST as proposed by Watson and Pelli (1983). QUEST is a Bayesian adaptive method that evaluates thresholds of psychometric functions by means of sequences of steps. The chosen psychometric function is the Weibull function  $W_T(x) = 1 - (1 - \gamma)exp(-10^{(x-\alpha)(\beta/20)})$ , where  $\beta$  represents the slope of the psychometric function depending on the conditions,  $\alpha$  is introduced to

make zero the ideal test point, and  $\gamma$  expresses the probability of success at zero intensity. In the 2-AFC,  $\gamma$  is assigned a value of 0.5. In particular, QUEST uses a probability density function representing the initial guess about the location of the threshold and then it uses the Bayes theorem to update the algorithm after each response, choosing an optimal stimulus to be presented to the user. The algorithm has a termination rule based on the confidence interval of the threshold and the number of steps (50 in our case). This work employed the MATLAB implementation by Pelli.

In the first test, a reference height was used of 15 cm with one user (male, age 25, left-handed) and two repetitions, requiring about 30 steps of QUEST. The results of our first test session give the parameters of the psychometric function in addition to the threshold. In particular, the threshold shows that the subject could recognize the presented reference stimuli with an error rate of about 30%.

In the second phase of the evaluation, the objective was to establish the difference in perception between real impact forces of objects and simulated impact forces of objects based on the velocity correction scheme introduced in Section 4.1. In this work, we opted for the 2-AFC design, this time having, for every step, a real ball falling from a variable height and a ball attached to the end-effector of the GRAB haptic interface (see Section 6 for a description of the device) programmed for simulating the impact of a virtual ball by computing the impact velocity from the virtual ball height. This is peculiar to this type of perceptual evaluation because it mixes two different domains: the real domain and the virtual one. For managing the real and virtual duality, it was decided to adopt a protocol in which two separate QUEST procedures were interleaved. In the first procedure,  $QR_{\text{fix}}$ , the reference stimulus is produced by a real ball falling from a height of 15 cm with respect to the user's hand while the comparison stimulus is produced by the haptic device using the height computed by the QUEST algorithm. The next procedure  $QV_{\text{fix}}$  has the opposite role: the reference stimulus is produced by the haptic interface with a height of 15 cm, while the comparison stimulus is provided by a real ball. These

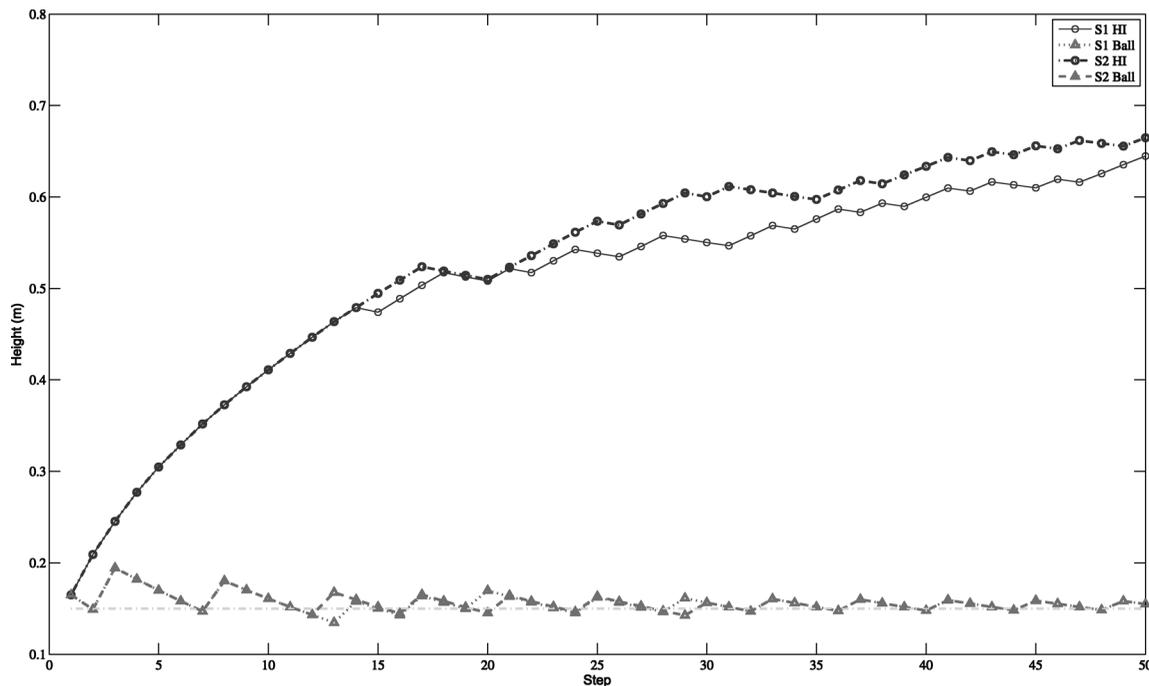
two tests were randomly interleaved within the same trial in order to prevent strategy formation. In both cases, the haptic interface is velocity controlled, producing an impact velocity based on the rules of ballistics without any correction. In this work, this test was performed with only one subject, using two repetitions, each terminating after 50 steps of QUEST. The profiles of the two interleaved estimations  $QV_{\text{fix}}$  and  $QR_{\text{fix}}$  are shown in Figure 2. The horizontal axis shows the step of the QUEST, while the vertical axis presents the stimulus recognized by the subject as stronger. The stimulus obtained in the last steps provides the ratio between the perceived value and the reference.

Two thresholds were obtained from the QUEST algorithm averaged across the two repetitions: in  $QV_{\text{fix}}$  a real stimulus matches the virtual reference of 0.15 m at a real height of 0.6548 m ( $SD = 0.0140$  m), while in  $QR_{\text{fix}}$  the threshold is 0.1548 m ( $SD = 0$ ). From  $QV_{\text{fix}}$  we obtain that the virtual height  $h_V$  is perceived to be 4.2 times smaller than  $h_R$ . The result of  $QR_{\text{fix}}$  can be explained by the one of  $QV_{\text{fix}}$  and by the behavior of QUEST: If the variable stimulus is much smaller than the reference, then the subject always chooses the reference, causing the threshold to converge to the reference.

By means of the equation  $v = \sqrt{2gh}$ , the ratio between velocities at the moment of impact  $k_{\text{exp}} = v_{hi}/v_0$  is determined as 0.49. This value can be compared to  $k_{\text{dev}} = 0.51$  computed in Section 4.1.

From this value we decided to fix the correction factor  $k$  at 0.50 and then to again perform the interleaved psychometric evaluation for assessing the resulting perceptual difference between the real and virtual objects. Figure 3 shows how the application of a  $k$  factor of 0.50 improves the perceived difference with respect to real and virtual objects in the evaluation performed with one subject in two repetitions. In this case, both  $QV_{\text{fix}}$  and  $QR_{\text{fix}}$  converge to the same value of 19.5 cm, that is, 30% of the real reference stimulus. This is aligned with the literature threshold on perceived impact height of real objects.

Now that both estimated and measured  $k$  produce a good result of perceived impact it can be applied to a case study for haptic juggling as discussed in Sections 6 and 7.



**Figure 2.** Execution of the psychometric procedure based on the QUEST algorithm comparing haptic and real stimulus in two interleaved tests. The figure shows the two tests,  $QV_{fix}$  and  $QR_{fix}$ , with one subject in two repeated sessions with a reference stimulus of 15 cm.  $QR_{fix}$  converges to the reference as explained in the text, while  $QV_{fix}$  expresses the reduced impact explained by the  $k_{dev}$  factor introduced in the theoretical section.

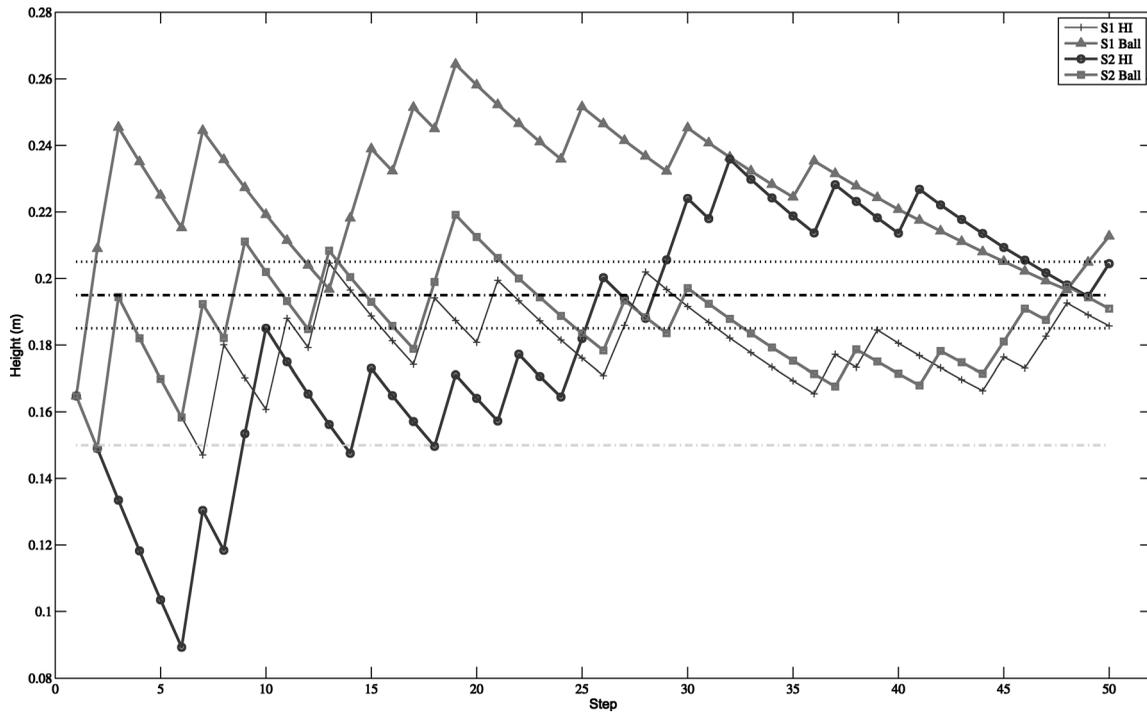
## 6 Haptic Juggling

After having discussed the model of catch and the perceptual adjustment for impact rendering, it is possible to apply these results to a juggling simulation based on the encountered haptic. In this section, the characteristics of juggling in general are discussed, including the proposed haptic rendering mechanism based on the general model presented above and the perceptual result for the correction of the velocity at impact. The specific system used for demonstrating the result is presented in terms of hardware and software architecture. This section concludes with a discussion of the control loop.

### 6.1 Juggling Dynamics

At this point, the above considerations about perception of the catch in the specific task of juggling are applied, trying to identify the necessary control requirements for designing a juggling simulator. Indeed, catch

and toss is not sufficient and, as illustrated by Beek and Lewbel (1995), there are other aspects to be considered to understand the juggling skill. The key factor in juggling is the tendency of the two human limbs to move at the same frequency in sync. The particular type of coordination displayed by juggling hands depends on the type of juggling pattern. In the cascade pattern, to cross the balls between the hands effectively requires that one hand catches at the same rate that the other hand tosses. The fountain pattern, in contrast, can be stably performed in two ways: by tossing and catching the balls simultaneously with both hands or by tossing a ball with one hand and catching one with the other at the same time. Juggling requires the hands of the user to move along more or less elliptical trajectories. In a good performance, the balls should be released at the inside of the ellipses and caught at the outside. Correct timing and sequencing of the balls' trajectories impose a consistent rhythm on the juggler, so that frequency and phase locking between the hands become fundamentals



**Figure 3.** Execution of the psychometric procedure in which the haptic interface uses a  $k$  corrected velocity. This test is performed by the same subject in Figure 2 with two repetitions. In this case the perceived difference between the impact of the real ball and the virtual ball is highly reduced. The dashed line in the bottom shows the reference real height (15 cm) of the real ball, while the dashed midline is the computed perceived height (19.5 cm).

to accomplish the task. Location, direction, and velocity of tosses are fundamental for global coordination, and the balls' trajectories get smoother as expertise develops.

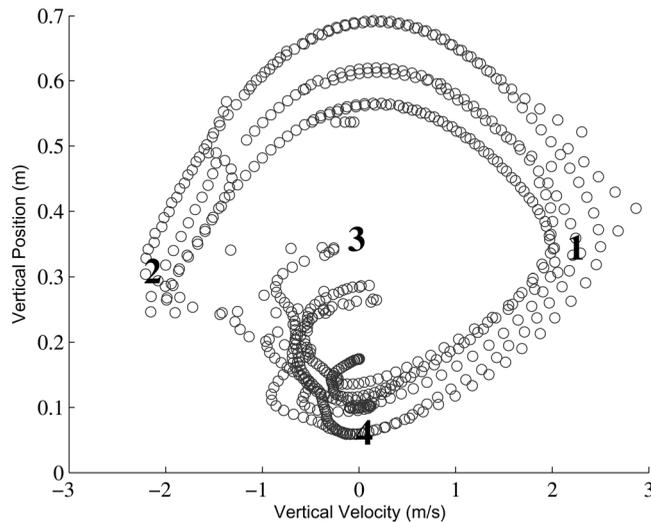
A juggler has to accommodate rather severe task constraints in order to juggle successfully. Time constraints and sequencing are formulated by the Shannon theorem of juggling (Shannon, 1993):

$$(F + D)H = (V + D)N, \quad (12)$$

where  $F$  is the time that a ball spends in air (flight),  $D$  the time that it spends in a hand (dwell),  $V$  the time a hand is vacant (vacant),  $N$  represents the number of balls, and  $H$  the number of hands. To perform a good juggling performance, it is then desirable that every ballistic trajectory of the ball is almost the same and easy to catch. In fact, to catch a ball, a juggler must adapt his or her own movements to its position, moving speed, and moving direction as soon as possible. In particular, it is

desirable to predict the expected time and arrival position and consequently to modify his or her own motion pattern to match these constraints.

For better understanding the simple, but interesting, dynamics, it is possible to analyze data obtained from tossing one ball in the vertical direction and capturing the motion with an external optical tracker at 200 Hz (Figure 4). The phase plane reports the vertical position of the ball versus its velocity. There are four relevant phases in the diagram: a flying phase (1–2), described by the upper parabola, a subsequent impact and grasping phase (2–3) which rapidly adjust the velocity of the ball toward that of the hand, a toss preparation phase (3–4) during which the hand moves down, and finally the subsequent toss (4–1). Hand and ball share the phase diagram during the catch stages (3–4–1) in which the position and the velocities of both hand and ball are rigidly constrained.



**Figure 4.** Phase plane of captured juggling data with the states superimposed.

## 6.2 Interaction

The proposed interaction scheme for haptic juggling is the following. Two 3-DOF haptic arms have their end-effectors  $E_{1,2}$  replaced by two tennis balls. The user sees the virtual balls  $V_1$  and  $V_2$  on the screen and when he or she moves his or her hand to catch the ball falling, there is a real ball with the correct impact perception. Then the user keeps the ball in the hand and throws it in the air. The pairing policy and the control strategy is presented at the end of Section 6.2.

The kinematic constraints introduced in this scenario by the physical presence of the robots (workspace and interference) have been solved with the following solution.

- The robot location has been organized such that each robot covers a proper amount of each hand workspace.
- During flight, the position of the robot-attached balls and the virtual balls may differ.
- At grasping, the robot that is closer to the grasping hand outputs the perceptual representation.

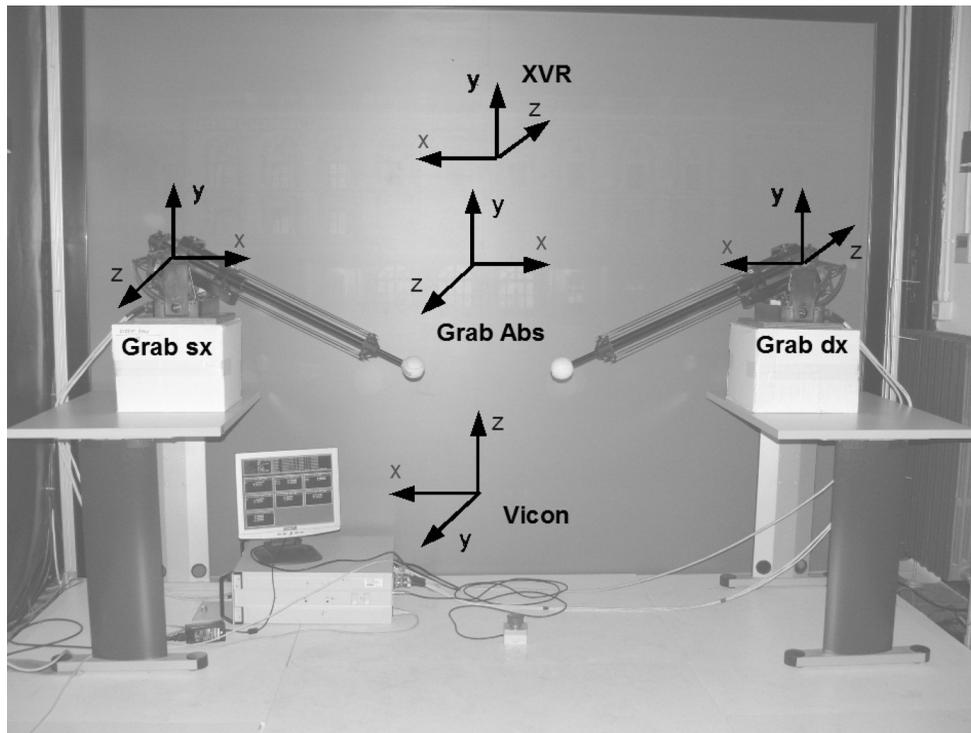
To grant good performance, the closed loop control needs to observe the following three basic rules.

- Reduce device speed and wash out its position whenever the hand moves away from it.
- Maintain the position of the end-effector close to the user's hand, avoiding collision and controlling the relative position along the direction more likely to impact with a closer object.
- In proximity of an impact, control the velocity of the end-effector to match the proper position and impact conditions.

## 6.3 Hardware Architecture

The robotic device used for the interaction is the GRAB Haptic Interface (Bergamasco, Avizzano, Frisoli, Ruffaldi, & Marcheschi, 2006), a large workspace 3-DOF haptic interface that has previously been used for single or multiple finger interaction. In particular, the device has a box-shaped workspace of 400 mm depth, 400 mm height, and 600 mm width. This haptic device is able to generate continuous forces at the end-effector of 4 N in the worst condition while the peak forces are up to 20 N. The device has a position accuracy at zero load of less than 1%, that is, 1 mm over 100 mm. In its standard configuration, the GRAB device has a thimble as its end-effector, allowing the user to directly interact with virtual objects. In this system, the thimble has been replaced by a standard tennis ball (3.5 cm radius). The size and the shape of the contact element were selected to provide a good interaction with the whole hand that would be satisfactory for the training application.

The system was composed of two GRAB robotic arms (reaching a workspace of 1200 mm wide) that were placed inside an L-shaped 3D projection environment, composed of two large projection screens (2.00 m  $\times$  2.70 m wide), one back projected in front of the user and the other direct projected on the floor. Stereo vision was provided by two projectors for every screen with resolution 1440  $\times$  1050 and refresh rate of 60 Hz. Passive filters and glasses by INFITEC (Ulm, DE) allowed decoupling of images with reduced ghosting effects. The environment was equipped with a VICON MX-20+ (OMG PLC, UK) infrared motion capture system, configured with seven cameras each having a resolution of 2 megapixels. The VICON system uses infrared strobes mounted around the cameras to



**Figure 5.** Coordinate reference systems of the setup. The haptic devices are shown on the side, with their local reference systems. The visualization (XVR) and common device reference system (Grab Abs) have been calibrated with the motion capture reference. In the photo it is possible to see the frontal and floor projection screens.

track the position of retroreflective 6-mm markers running at 200 Hz with a resulting position resolution less than 0.5 mm. This capture system is used both as a head-tracking system, for obtaining correct stereo projection to the user, and as a user hand tracker, for evaluating hand motion during the task. Markers were attached to the dorsal part of the user's hand and on top of the stereo-vision glasses. In this way, the projection system can accomplish co-location between the virtual environment and the user/haptic interface, resulting in a complete mixed reality application. Figure 8 shows a user juggling with the present system.

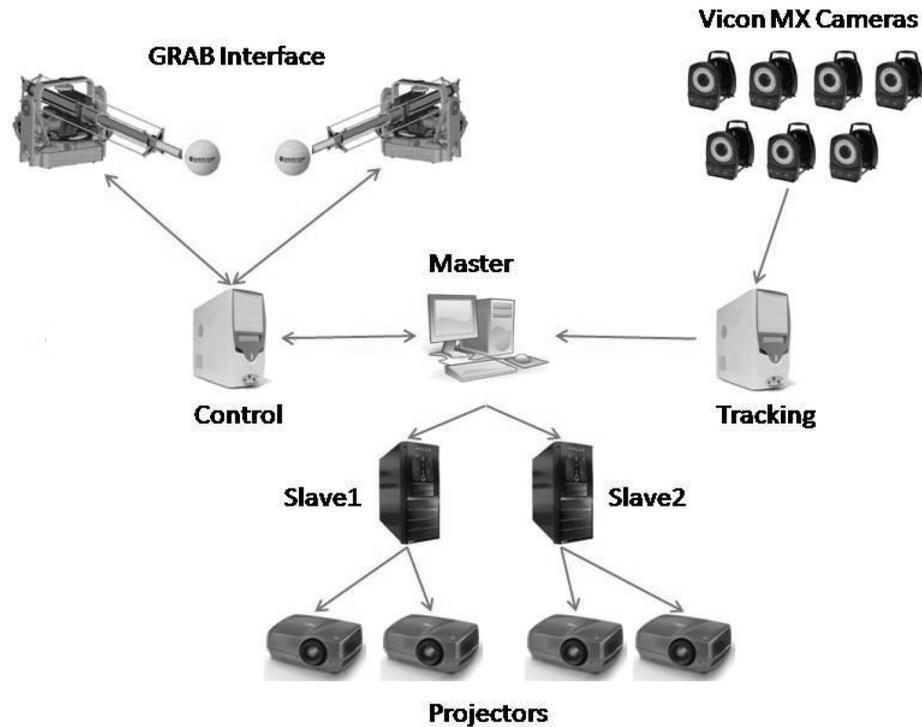
To perform this exact co-location, several transformations were applied between the different coordinate reference systems. In particular, user motion was utilized, tracked in the space of the VICON. User interaction with the GRAB device performed in a reference system first relative to every end-effector and then relative to the center of the two devices. Finally, there is a

metric visualization reference system that is centered on the screen. Haptic interface calibration was performed by placing a four-marker shape over the GRAB device end-effector and aligning it with the measurements from the VICON system. The visualization system was calibrated by displaying a cross exactly placed over the display screen and aligning it using a motion capture system. The coordinate systems involved are shown in Figure 5.

The overall system (see Figure 6) is composed of five computers: the first one dedicated to haptic control, and the second one for motion tracking. The last three computers act as a cluster with a master and two slaves generating the stereo images for the four projectors.

#### 6.4 Software Architecture

The graphic rendering was generated using the eXtreme Virtual Reality (XVR) framework (Carrozzino,



**Figure 6.** Architectural view of the system. On the top left the two haptic interfaces are connected to the control machine, that communicates the status of the interface to the master computer. In the top right is the VICON capturing system, and in the bottom right is the visualization cluster.

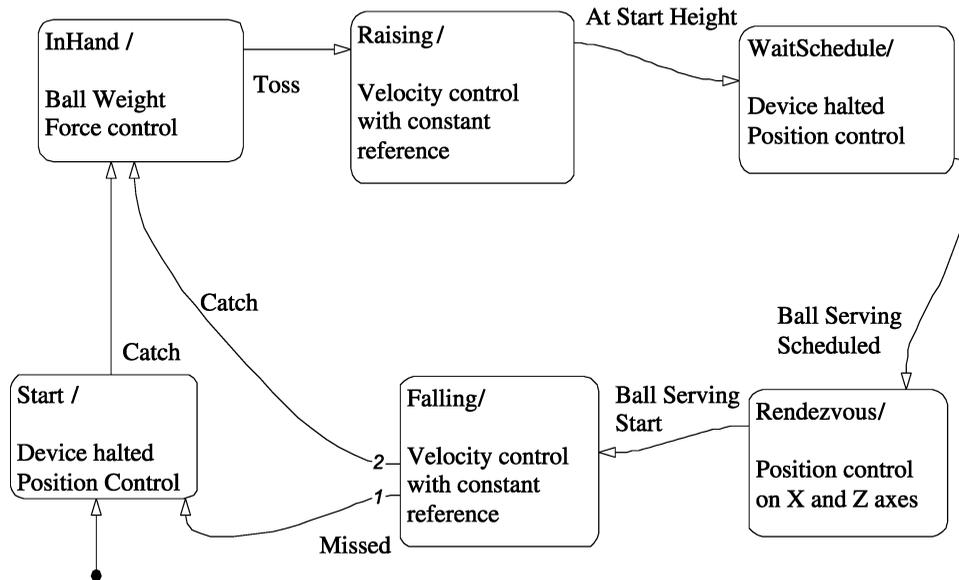
Tecchia, Bacinelli, Cappelletti, & Bergamasco, 2005) that affords rapid prototyping of virtual environments. The immersive visualization adopted for this work makes use of the XVR Network Renderer (Marino, Tecchia, & Bergamasco, 2007) that distributes the visualization on the visualization cluster. The juggling simulation and device control were implemented using Simulink (Mathworks, Natick, MA) running at 1 kHz on the control computer. The control application is interfaced with the motion capture system by means of a network connection for obtaining information about the user's head and hand position and orientation. The information from the simulation of the balls and the position and orientation of the user's head are sent to the XVR 3D graphic application respectively for visualizing the balls and correcting the 3D projection.

### 6.5 Control

This section explains the control strategy adopted for the implementation of the interaction paradigm dis-

cussed above. The control scheme is based on finite state machines that manage the simulation of the balls  $V_j$ , the pairing with end-effectors  $E_i$ , and their control. The simulation of the balls is based on the simple dynamics of ballistics and it is possible to easily track and anticipate the trajectory of the virtual object from the starting position and velocity at the moment of the throw. The balls  $V_j$  follow their own dynamics except when they are attached to the end-effector, a condition that happens when they are paired and in contact with the user. In the case of juggling, this condition happens between the catch and throwing phases. The simulation is performed on the common reference system of the workspace that is shared with the haptic interface.

The pairing policy is based on the specific task and on the layout of the two devices. For this reason, the space is partitioned along the  $x$  axis of the robot absolute reference system Grab Abs (see Figure 5) leaving the left side managed by Grab Sx and the right side by Grab Dx. If more than one ball is falling on the same parti-



**Figure 7.** Diagram of the finite state machine of control. Each end-effector  $E_i$  is in charge of one hand  $B_k$  and at every cycle it is paired with a specific virtual ball  $V_j$  depending on the scheduled time of arrival. The type of low-level control (force, position, or velocity) is reported inside every state box. In the Rendezvous state the  $x$  and  $z$  axes are in the haptic common reference system in which  $x$  is along the horizontal and  $z$  is vertical.

tion, the selected end-effector must decide which ball to serve. The scheduling is based on the time of flight or more precisely by the time it takes to reach the space in which the hands can typically be found for catching the ball. By adopting a shortest-time first served (STFS) algorithm, it is possible to grant all the falling balls to be displayed to the user if the balls are juggled with a temporal schedule similar to the typical one of juggling.

The STFS policy requires an estimation of the time of impact between the virtual ball and the user's hand. In the setup discussed here, the position of the hands is not used in control, and instead the catch is triggered by using a vertical threshold along the  $z$  axis. This means that the time  $t^*$  is obtained from the equation  $z_{V_j}(t^*) = Z^*$ . In the general case, an estimate of a hand's trajectory can be obtained from motion capture eventually supported by a Kalman filter. At a given instant  $t_0$  it is possible to obtain a trajectory estimation  $P_{t_0}(t)$  from which the impact time  $t^*$  can be obtained by minimizing the modulus distance between the trajectory of the hand and the one of the ball.

The following section discusses the state machine controlling the end-effector.

## 6.6 Interface Control Finite State Machine

The device arm control is based on a finite state machine that switches control modalities at the precise moment of the occurrence of events. In particular, in our design, six events and six possible states for the arms (see Figure 7) are recognized. The simulation starts with a condition in which the end-effector  $E_i$  is halted in the air at a given position waiting to be caught by the user. This new state is the *InHand* state, in which the user perceives the end-effector as the virtual object. The *Raising* state is activated by the tossing event triggered by means of a threshold over the velocity of the end-effector. This event marks the loss of pairing between the ball and the end-effector. In the *Raising* state, the  $E_i$  moves with a constant velocity into a resting position for not interfering with user's switching to *WaitSchedule*. The  $E_i$  keeps *WaitSchedule* until the pairing scheduler



**Figure 8.** Photo of usage of the haptic juggling system. The user is looking in front of himself. This photo refers to the cascade examples in which the hands were not tracked.

allocates the end-effector to a new ball served for the same hand.

At the moment of pairing, the impact is estimated occurring at instant  $t^*$  with position  $P_{V_j}(t^*)$  and velocity  $V_{V_j}(t^*)$ . From the encountering approach discussed so far, at time  $t^*$ , the paired end-effector should be at the same position but with a velocity scaled by the  $k$  factor. The satisfaction of both the velocity and position constraint at the time of impact can be obtained by a hybrid position-velocity controller. In this implementation, the problem was simplified by splitting it into two states: the first is *Rendezvous* during which the end-effector position matches the position of the paired ball; and the second is *Falling*, during which the  $E_i$  is velocity-controlled with the scaled target velocity  $kV_{V_j}(t^*)$ . The *Falling* state is activated a few instants before impact time  $t^*$ . When the user catches the ball during *Falling*, the cycle is closed, and the  $E_i$  enters the *InHand* state. Alternatively, if the user does not catch the ball, the end-effector returns to the starting state ready to restart the cycle.

In terms of low-level control, the end-effector is controlled with different controlling schemes depending on the state. In *InHand*, impedance control is adopted, generating a force that applies to the user the weight and inertia of the ball. During the *Falling* and *Raising* states, the end-effector is controlled in velocity, while in the *Start* and *Rendezvous* states, position control is adopted.

In this work, it has been assumed that the catch is good in the sense that there is no bouncing and the user, after the catch, grasps the object. A more detailed state machine could be employed for alternating between *Falling* and *InHand* phases.

## 7 Evaluation of the System

As detailed in the interaction paradigm section, a juggling training system should allow the practitioner to perform at least the basic juggling patterns. In particular, it is essential to give the possibility of performing

both fountain and cascade. This grants the possibility of juggling balls with a single hand or of crossing the trajectories and performing juggling with both hands. The implemented system allows both the patterns to be executed. Figure 9 shows two execution plots of a non-expert practitioner trying to perform the single launch typical of the fountain and cascade patterns.

The rest of this section employs two data sets. First, a recording of three juggling trials from a novice juggler, in which the motion capture system tracks both hands and the head by means of clusters of reflective markers. The second evaluation session was performed with four subjects.

The dynamics of the juggling simulator can be analyzed using a phase plane plot similar to the one obtained with real data (see Figure 4). Figure 10(a) shows the motion of one arm and one of the balls in the  $y$  axis and the associated velocity, during the execution of a two-ball cascade-like sequence. The complete interaction scheme can be analyzed looking at the two arms and the two balls as shown in Figure 10(b). In this case, the instant in which each arm is paired with the ball must be identified, and also the way in which, during cascade, the two aims exchange their roles with respect to each ball.

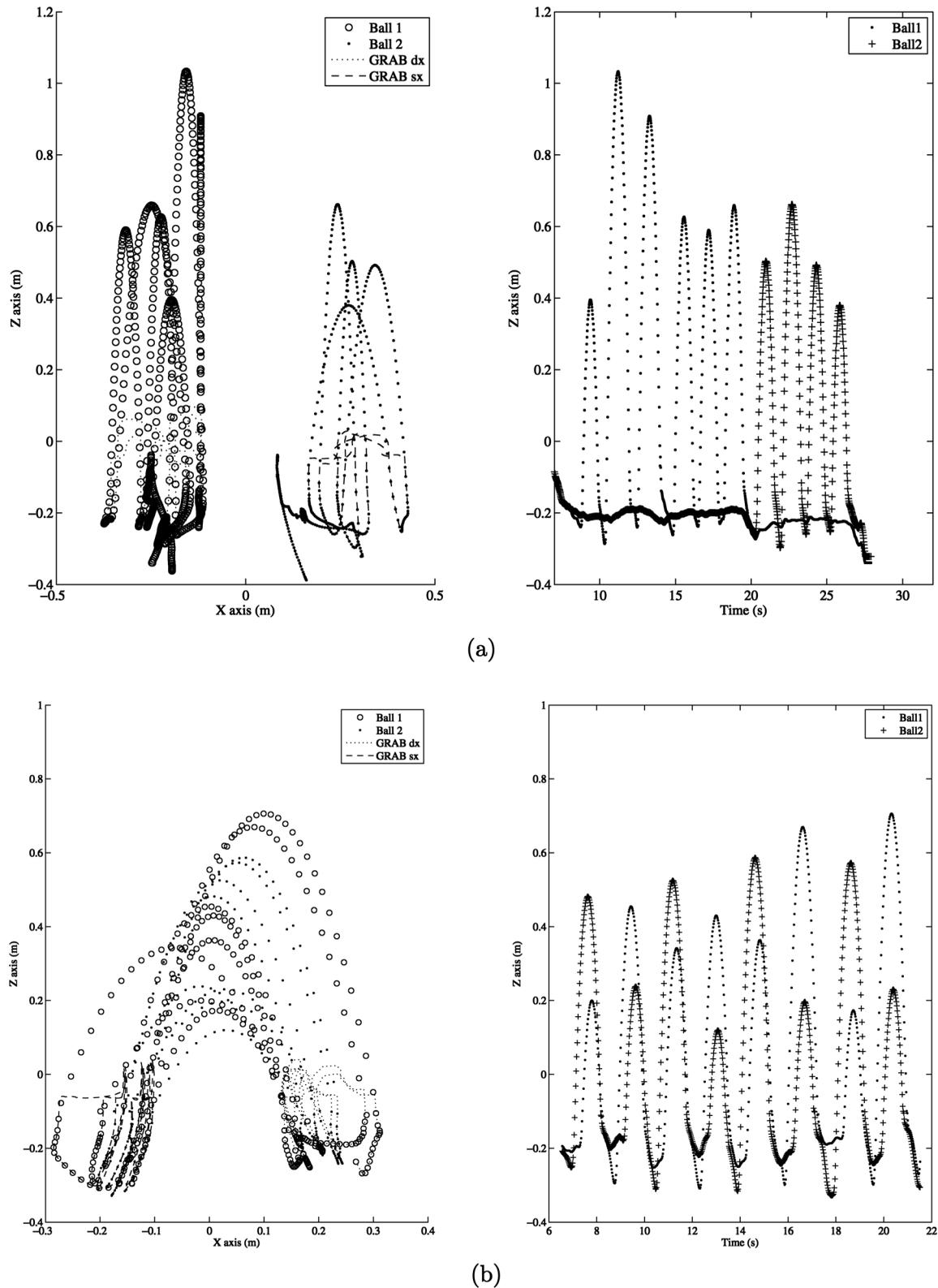
Motion capture of the user's head and hands allows the system to evaluate the characteristics of the juggling pattern for assessment, giving space not only to system assessment but future training with the system. First, the motion of the hands and then the rotation of the head are investigated. Figure 10(c) presents the evolution of the vertical axis showing the motion of the left hand (dot-dash) with the toss and catch points highlighted respectively by squares and diamonds. The continuous line is the left robotic arm that follows the control strategy described above, and in particular it is almost coincident with the hand during grasping of the ball attached to the end-effector. The visible displacement between the hand and the end-effector position is motivated by the radius of the ball and the varying positioning of the reflective markers placed on the wrist of the user. As in Figure 10(a) the dashed and dotted lines correspond to two virtual balls following the cascade pattern.

The other information that can be used for estimating the quality of the interaction is the rotation of the head with respect to the horizontal  $x$  axis. Expert jugglers do not focus their attention on the hands, but instead look in front of them at midair. Figure 11 reports the statistics of head rotation along the  $x$  axis, with negative values when looking downward, for 30 s of juggling in the three trials. The values stay in the range between  $-15^\circ$  and  $0^\circ$  which is quite satisfactory for this configuration.

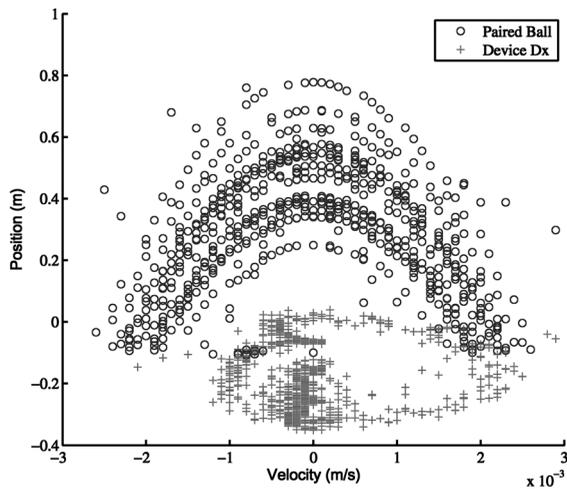
A characteristic of the system presented here is the possibility of modifying the gravitational acceleration parameter that could be employed in a training protocol for slowing down the balls in the early stages of training. In this way, the user could be able to concentrate on timing and trajectory planning. In this first evaluation, the gravity factor was changed from  $9.81 \text{ m/s}^2$  to  $4.0 \text{ m/s}^2$  in order to measure experimental data with reduced acceleration at the end-effector, consequently reducing nonlinear phenomena generated by saturation of motor torques (maximum force is about 8 N for the given test position). The maximum acceleration, with minimal torque saturation for the device, was found to be approximately  $25 \text{ m/s}^2$ . Given the formula  $h_{\max} = v^2/a$ , a reciprocal space-saving in the workspace may be achieved with the reduction of the gravity parameter.

In particular, four novice subjects were tested (aged 26–30 years) for the purpose of evaluating the effect of the gravity factor on their performance. The subjects were asked to perform a two-ball pattern with balls crossing in the air. After a practicing phase, each subject performed three trials of 120 s with three different randomized levels of acceleration (2, 4, and  $6 \text{ m/s}^2$ ). The general performance measures were the ratio of lost balls against the number of tosses, and the mean length of sequences with successful throw pairs.

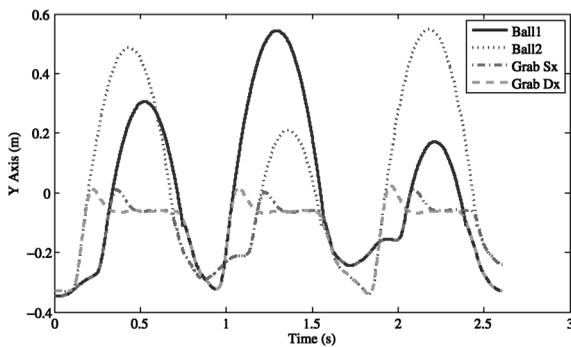
From this brief evaluation, a strong effect of the gravity factor on the effectiveness of the system emerges. In particular, the lowest gravity factor increases the number of errors due to the more difficult estimate of ball behavior, an aspect that is also motivated by the anticipation of throwing the second ball with respect to the peak of the first. As expected, the faster behavior due to an



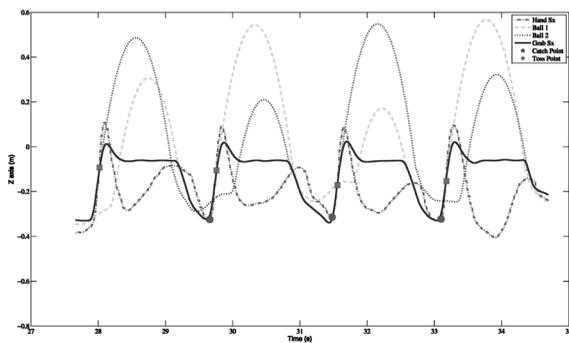
**Figure 9.** X – Y plots of the (a) single launch and (b) cascade patterns generated by a non-expert juggler during a session with two virtual balls. In the single launch pattern, similar to the classic fountain, each ball is paired with the same end-effector, while in the cascade they are exchanged. These trials have been performed with gravity  $4.0 \text{ m/s}^2$ . (a) Example of the single launch pattern. (b) Example of the cascade pattern.



(a) Phase plane ( $z$  axis) of one of the end-effectors and one of the virtual balls.

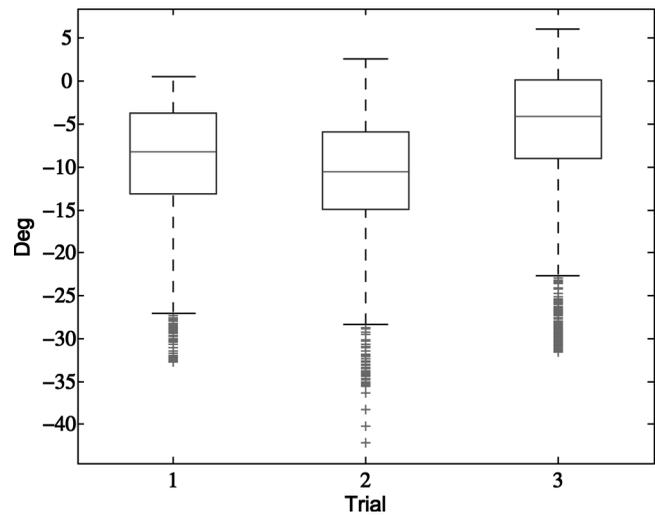


(b) Trajectory along the  $z$  axis of two end-effectors and the two balls.



(c) Position of the hand with respect to the balls and the device arm.

**Figure 10.** (a) The phase plane of the ball (circle) and the paired device (plus sign). (b) The trajectory of both balls and arms along the  $y$  axis. (c) Hand position with respect to the balls and the device arm.

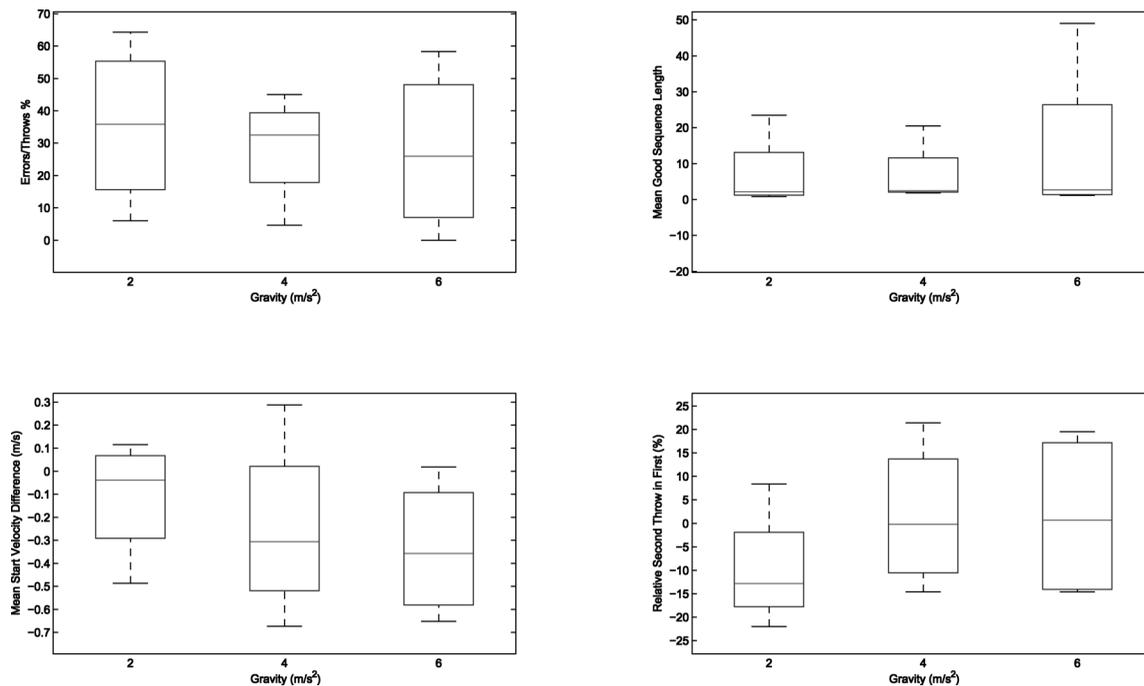


**Figure 11.** The statistics of head rotation along the horizontal  $x$  axis with a negative value when the user looks downward as obtained from three trials of 30s with a novice user. Data is tracked by means of the motion capture system.

increased gravity factor moves the second throw nearer to the optimum of the peak of the first throw. Finally, anecdotal comments from users indicated that the device is quite usable although some rough points concerning the tracking of the hands could be improved.

## 8 Conclusions and Future Work

The haptic rendering methodology introduced in this work gives interesting new opportunities for encountered haptic interfaces because it allows the system to model the interaction with virtual objects that have their own dynamics. This approach is in line with the evolution of interaction techniques and haptics that tend to be more portable or available only when needed. In addition to the specific case of juggling, this technique could be applied to other domains, such as advanced telepresence or in the promising field of sport training with robotics. Moreover, the simulation of impacts and dynamic interaction could be provided not only by means of haptic interfaces, but also by humanoid robots, provided that they have enough mechanical bandwidth and update rate.



**Figure 12.** Evaluation with four subjects under three different gravity conditions. Four different measures have been employed for highlighting the behavior of subjects on the system: the number of lost balls in percentage (top left), the mean good sequences (top right), the difference in the peaks of same pair (bottom left), and the relative timing of the second throw with respect to the first peak.

There are several directions that can be investigated starting from this work. The first is a general question in terms of theory of control of multiple devices, in particular, exploiting different strategies for the scheduling of the pairing when more than two arms are involved.

Other questions arise in terms of haptic juggling evaluation and training. The system is usable for basic patterns but there is a fundamental question to be answered about its capability to be used as a training platform, partially associated with the maximum velocity that can be obtained during the interaction. For the specific case of juggling, mathematics comes in handy to help in solving this multiconstraint problem. From the previously stated Shannon theorem, it is possible to exactly calculate the timing and the needs of a multiple object and device setup depending on the selected juggling pattern. The introduction of a multi-ball dynamic system composed of several devices is an argument that can be discussed in a future evolution of the system, in addition to assessment of its use for juggling training.

Finally, the tracking of hands and integration of the tracking in the control scheme are important aspects not only for improving the fidelity of the catch velocity, but also for the safety of the system. A training protocol based on the proposed system would be able to induce a motor pattern similar to the one of real juggling, supporting the learning process by means of the task simplification provided by gravity scaling. Training by practice could be also supported by visual cues displayed on the screen.

Video material about this paper can be found at the website <http://www.percro.org/papers/encjug/> and in the supplemental material accompanying the online version of this article.

## Acknowledgments

The authors would like to thank the IST-2006-035005-SKILLS EU Integrated Project, that deals with the multi-modal acquisition, modeling, and rendering of data signals for

the transfer of skills in different application fields. We thank also the anonymous reviewers and the Editor for helping us improve this paper.

## References

- Beek, P., & Lewbel, A. (1995). The science of juggling. *Scientific American*, 273 (5), 92–97.
- Bergamasco, M. (1997). Force replication to the human operator: The development of arm and hand exoskeletons as haptic interfaces. In *Proceedings of 7th International Symposium of Robotics Research*, 173–182.
- Bergamasco, M., Avizzano, C., Frisoli, A., Ruffaldi, E., & Marcheschi, S. (2006). Design and validation of a complete haptic system for manipulative tasks. *Advanced Robotics*, 20(3), 367–389.
- Bianchi, G., Knoerlein, B., Székely, G., Harders, M., & Switzerland, E. (2006). High precision augmented reality haptics. *Proceedings of Eurohaptics*, Vol. 6 (pp. 169–178).
- Brown, J., & Colgate, J. (1994). Factors affecting the z-width of a haptic display. In *Proceedings of the IEEE ICRA Conference*, 3205–3210.
- Carrozzino, M., Tecchia, F., Bacinelli, S., Cappelletti, C., & Bergamasco, M. (2005). Lowering the development time of multimodal interactive application: The real-life experience of the XVR project. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 270–273.
- Cosco, F., Garre, C., Bruno, F., Muzzupappa, M., & Otaduy, M. (2009). Augmented touch without visual obtrusion. *The 8th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2009*, 99–102.
- Fiene, J., Kuchenbecker, K. J., & Niemeyer, G. (2006). Event-based haptics with grip force compensation. *Proceedings of the Haptic Symposium*, 117–123, <http://www.seas.upenn.edu/~kuchenbe/pub/pdf/Fiene06-HS-Grip.pdf>
- Furukawa, T., Inoue, K., Takubo, T., & Arai, T. (2007). Encountered-type visual haptic display using flexible sheet. *The 2007 IEEE International Conference on Robotics and Automation*, 479–484.
- Gruenbaum, P., & McNeely, W. (1997). Implementation of dynamic robotic graphics for a virtual control panel. *Presence: Teleoperators and Virtual Environments*, 6(1), 118–126.
- Hill, N. J. (2001). *Testing hypotheses about psychometric functions*. Ph.D. Thesis, University of Oxford, UK.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1), 90.
- Knoerlein, B., Székely, G., & Harders, M. (2007). Visuo-haptic collaborative augmented reality ping-pong. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology*, 91–94.
- Kreifeldt, J., & Chuang, M. (1979). Moment of inertia: Psychophysical study of an overlooked sensation. *Science* 206(4418), 588–590. <http://www.sciencemag.org/cgi/content/abstract/206/4418/588>
- Marino, G., Tecchia, F., & Bergamasco, M. (2007). Cluster-based rendering of complex virtual environments. *Intuition 2007: Proceedings of the 4th International Intuition Conference on Virtual Reality and Virtual Environments*.
- Marshall, J., Benford, S., & Pridmore, T. (2007). Eye-balls: Juggling with the virtual. In *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, 265–266.
- McNeely, W. (1993). Robotic graphics: A new approach to force feedback for virtual reality. *Virtual Reality Annual International Symposium*, 336–341.
- Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, Vol. E77-D (pp. 1321–1329).
- Ross, H., Brodie, E., & Benson, A. (1986). Mass-discrimination in weightlessness and readaptation to earth's gravity. *Experimental Brain Research*, 64(2), 358–366.
- Sakaguchi, T., Masutani, Y., & Miyazaki, F. (1991). A study on juggling tasks. *Proceedings of the IEEE/RSJ International Workshop on Intelligence for Mechanical Systems, Intelligent Robots and Systems '91*, 1418–1423.
- Sciavicco, L., & Siciliano, B. (2003). *Modelling and control of robot manipulators*. Berlin: Springer.
- Shannon, C. (1993). *Claude Elwood Shannon: Collected papers* (p. 850). Piscataway, NJ: IEEE Press.
- Shigeta, K., Sato, Y., & Yokokohji, Y. (2007). Motion planning of encountered-type haptic device for multiple fingertips based on minimum distance point information. *Second Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 188–193.
- Solazzi, M., Frisoli, A., Salsedo, F., & Bergamasco, M. (2006). An innovative portable fingertip haptic device. In *Proceedings of the Third Enactive International Conference*, 229.

- Swapp, D., Pawar, V., & Loscos, C. (2006). Interaction with co-located haptic feedback in virtual reality. *Virtual Reality, 10*(1), 24–30.
- Tachi, S., Maeda, T., Hirata, R., & Hoshino, H. (1994). A construction method of virtual haptic space. *Proceedings of the 4th International Conference on Artificial Reality and Tele-Existence, ICAT'94*, 131–138.
- Tachi, S., Maeda, T., Hirata, R., & Hoshino, H. (1995). A machine that generates virtual haptic space. *VRAIS'95 Video Proceedings*.
- Viciano-Abad, R., & Reyes-Lecuona, A. (2008). Effects of collocation and crossmodal interaction between haptic, auditory and visual cues in presence. *Lecture Notes in Computer Science, Vol. 5024* (p. 832). Berlin: Springer-Verlag.
- Watson, A. B., & Pelli, D. G. (1983). Quest: A Bayesian adaptive psychometric method. *Percept Psychophys, 33*(2), 113–120. Available from <http://view.ncbi.nlm.nih.gov/pubmed/6844102>
- Yokokohji, Y., Hollis, R. L., & Kanade, T. (1996). What you can see is what you can feel—Development of visual/haptic interface to virtual environment. *Proceedings of the 1996 Virtual Reality Annual International Symposium, VRAIS '96*, 46.
- Yokokohji, Y., Muramori, N., Sato, Y., & Yoshikawa, T. (2005). Designing an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behaviors. *The International Journal of Robotics Research, 24*(9), 717.
- Yokokohji, Y., Yoshikawa, T., & Kinoshita, J. (2001). Path planning for encountered-type haptic devices that render multiple objects in 3D space. *Proceedings of the IEEE Virtual Reality Conference*, 271.