Human-Scale Bimanual Haptic Interface

Thomas Hulin Mikel Sagardia Jordi Artigas Simon Schaetzle Philipp Kremer Carsten Preusche

Institute of Robotics and Mechatronics
German Aerospace Center (DLR), Germany
E-mail: {firstname.lastname}@dlr.de

Abstract

This article presents a haptic system for bimanual haptic feedback that is composed of two light-weight robot arms. The system has a workspace and force capabilities similar to that of two human arms. Sophisticated control strategies are implemented to enable using this system as haptic interface. Depending on the requirements of the task one of three different handles can connect the human hand to the robot. Besides the human-machine-interface, the haptic rendering software is improved such that collisions of two dynamical interaction objects are computed in real-time. The system has been proven to be well suited as haptic interface for multimodal virtual assembly simulations.

1. Introduction

All systems designed to be operated by a human being have to provide an interface for their users, for example a keyboard and a monitor in the case of computers. This interface is called Human-Machine-Interface (HMI), sometimes also named as Human-System-Interface or Man-Machine-Interface. Systems built for applications like Virtual Reality (VR) simulations or teleoperation in which operators must be able to act intuitively require transparent HMIs, i.e. interfaces that display the virtual environment as realistically as possible.

The level of immersion in those environments depends strongly on the quality and quantity of feedback modalities (e.g. visual, acoustic, or haptic) that are provided by HMIs. For some applications like virtual assembly verification or remote maintenance, haptic feedback is crucial for task completion.

Applications like training of mechanics on virtual mock-ups, as required by the aeroplane industry, or assembly verifications for the automotive industry, call for workspaces and applicable forces roughly equal to those of human arms. Because of its specifications, the DLR Light-Weight Robot (LWR) [1] is well suited as kinesthetic-haptic device. One single LWR has a length of about one meter and is able to apply forces / torques of around $150\,N$ / $25\,Nm$ in any valid configuration.

Figure 1: Two-robot system for bimanual haptic feedback.

This paper describes a bimanual HMI, which is based upon two LWRs (Fig. 1). It is equipped with a head-mounted display for stereo-vision and stereo-acoustic feedback. The following section details the technical specifications of the haptic system and introduces three handles that connect the human hand to the LWRs. Section 3 discusses control issues for using the LWRs as haptic device. An evaluation of the system in two VR scenarios is described in section 4, while section 5 summarizes the main results and concludes with future work.

2. System Description

This article presents a haptic HMI for bimanual haptic feedback (shown in Fig. 1). The haptic system is

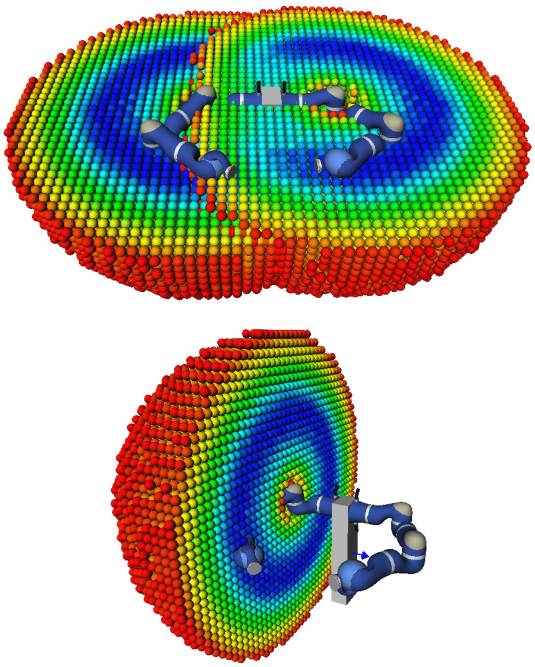

Figure 2: Sectional drawings of the workspace. Blue spheres mark points inside the workspace at which the robot can reach more than 75% of all possible three dimensional orientations, whereas red spheres mark points with less than 8% respectively.

composed of two LWR arms [1] that are horizontally attached at a column.

Their workspace is similar to that of human arms. Two sectional drawings of the workspace are shown in Fig. 2. The spheres represent possible end-effector positions in the overlapping workspaces of the two LWRs. At each position, three dimensional orientations of the end-effector are checked for reachability. As result, sphere colors are indicating the reachability index at each point [2].

2.1. Light-Weight Robots (LWR)

The LWR is a light-weight, flexible, revolute joint robot, which by its overall sensory equipment is especially suited for working in the sensitive area of human interaction [1]. The robot's size, power and manipulation capabilities as well as its workspace are fairly similar to that of a human arm and they turn the LWR into a well suited HMI although it was not explicitly designed for this purpose. With its seven serially linked joints, the robot possesses a redundant kinematics that allows for null-space movement, which is valuable for avoiding collisions and optimizing the robot's configuration.

The LWR is equipped with very light gears, powerful

Dynamic Mass	2 x $14\,kg$
Max. Payload	2 x $14\,kg$
Maximum Span	2 x $936\,mm$
Nr. of Joints	2 x 7
Sensors on each wrist	6-DoF Force-Torque Sensor
Sensors in each Joint	2 Position, 1 Torque Sensor
Sampling Rates	$40\,kHz$ current control $3\,kHz$ joint internal $1\,kHz$ cartesian
Motors	DLR-Robodrive
Gears	Harmonic Drive

Table 1: Specifications of the haptic system.

motors and weight optimized brakes. As safety feature, these brakes need power supply to be released and they are activated as soon as the power is off. The electronics is integrated in each joint, including the power converters. The robot arms are able to handle loads up to $14\,kg$ in the whole workspace, while having a total weight of also $14\,kg$. Each of the LWRs joints has a motor position sensor and a sensor for joint position and joint torque. Thus, the robot can be operated for position, velocity and torque, being controlled at an update rate of $1\,kHz$, which allows for a highly dynamic behavior. An additional 6-Degree-of-Freedom (DoF) force-torque sensor is mounted on the wrist of each robot. This sensor can measure very precisely external forces, e.g. applied by a human operator.

For the use of the LWR as HMI, a main research topic focuses on safe human-robot interaction. Therefore, a biomechanical evaluation with crash tests has been carried out [3]. Furthermore, a thorough research on safety issues with respect to control strategies has been performed recently [4].

2.2. Handles for Haptic Interaction

A small manual flange allows changing fast the handle that connects the robot to the human hand. Three different handles are currently in use with the haptic system: a magnetic clutch, a grip-force interface and a joystick.

Magnetic clutch: the human hand is attached to a bracket in such a way that fingers are free to move (Fig. 3, left). Therefore, this interface can be used in combination with a tactile finger feedback device [5] or a finger-tracking device, e.g. the CyberGlove® [6], whose data can be used for visualizing a virtual hand in order to increase immersion, or even to control a multi-DoF device like the DLR Hand II [7]. This kind of hand attachment supersedes the visual tracking of the

Figure 3: Magnetic clutch (left), grip-force interface (middle) and joystick handle (right).

hand pose, because it can be calculated from the forward kinematics of the robot. The bracket itself is magnetically coupled to the robot flange. The geometry of the clutch, the arrangement of the magnets and their strength define the attaching forces and torques. If the applied forces or torques exceed this maximum force of the clutch, the user is detached from the robot and the integrated dead-man switch disconnects the power supply, which activates the brakes and stops immediately the robot.

Grip-force interface: a one-DoF grip-force feedback device with force feedback to the forefinger (Fig. 3, middle). This additional DoF can be used to grasp objects in a virtual reality simulation, or to explore their properties. Furthermore, this device can be used for telemanipulation tasks with force-feedback, e.g. closing a gripper with a certain force.

Joystick: a joystick handle equipped with a mini-joystick, a switch and several buttons, including a dead-man button (Fig. 3, right). This handle is especially suited for interactive tasks in virtual environments. Thus, the human user is able to change online control parameters of the robot, parameters of the virtual reality simulation, or adjust the visualization.

To increase immersion and to obtain a more intuitive impression of the virtual scenario, all the mentioned interfaces can be used in combination with a vibro-tactile feedback device [8] for haptic feedback to the forearm.

3. Control Issues

The control of the robot arms is challenging. On one side, due to its seven joints, the LWR has a redundant DoF that has to be controlled. On the other side, the robot's inertia must be scaled down in order to improve free-space movement behavior. And additionally, since the workspaces of the two robots overlap, collision detection and avoidance must be implemented.

This section assumes the robot being impedance controlled, i.e. torques are commanded to the robot and a backdrivable behavior is enabled in every joint.

3.1. Null-space Motion

As mentioned above, due to its seven joints, the LWR has a redundant DoF. This means that it can maintain a fixed pose with its end-effector, while moving freely its elbow. This redundant DoF can be used for two purposes: for optimizing the robot's configuration and as safety measure by featuring a compliant behavior.

The compliant behavior is inherent in the LWR, if it is impedance controlled, which makes possible pushing the elbow away.

The approach used for optimizing the configuration is described below. It allows for commanding a force at the elbow without disturbing the human operator. This desired force F_d can be set as parameter, for example such that it pushes the elbow of the right robot to the right, and vice versa for the left robot. Given a desired force, the corresponding desired torques T_d of the first four joints result as product of the transposed partial Jacobian matrix from the base (0) to the elbow (joint 4) with the six dimensional force vector F_d

$$T_d = {}^0J_4^T F_d. \tag{1}$$

The resulting force at the end-effector caused by T_d is determined by

$$F_E = ({}^0J_7^T)^+ \cdot \begin{bmatrix} T_d \\ \text{- - -} \\ 0 \end{bmatrix}, \tag{2}$$

where $({}^0J_7^T)^+$ is the pseudoinverse of the transposed full Jacobian matrix. Given that the elbow motion should not have any effect on the end-effector, the force F_E must be compensated. The required joint

torque T_{comp} to compensate the force F_E at the end-effector is

$$T_{comp} = -^0 J_7^T \cdot F_E. \quad (3)$$

Therefore, for the commanded torques it yields

$$T = \begin{bmatrix} T_d \\ - - - \\ 0 \end{bmatrix} + T_{comp}. \quad (4)$$

The torques T push the elbow in the direction of F_d, while not affecting the end-effector. Note that in singular configurations T vanishes.

Due to the fact of non-ideal actuators (friction, ripple, dynamics), the commanded torques do not correspond exactly to the performed torques, and the elbow motion is slightly perceivable.

3.2. Feedforward

As stated above, the mass of the each robot is of about $14\,kg$. Although the robots' gravitation is compensated, a human operator would soon get tired during a haptic simulation, due to the inertia. To avoid this, the perceived inertia must be reduced. This requires measuring the external forces at the end-effector.

As the robot's mass matrix is known, these forces can be determined from the torques of the seven joints. Yet, two problems arise with this approach. First, the calculated force at the end-effector is not accurate, because the model parameters of the robot are not perfect. And second, in singular configurations not all six values of the end-effector's force vector can be determined.

On account of this, a force-torque sensor is mounted on each robot's wrist. This sensor measures very accurately the applied forces, which can be used for scaling down the inertia of the robots.

For the LWR a feedforward compensation is applied, such as described in [9]. Different gains are used for translation and rotation, in order to obtain a stronger reduction of the inertia for rotations around the human's hand. With this feedforward compensation the perceived translational inertia is reduced to 33%, and the rotational inertia to 25% of their original values.

3.3. Collision detection

In contrast to many other haptic systems, which are mechanically designed such that collisions are impossible, the presented haptic system can collide. On the one hand, as described before, the workspaces of the two LWRs intersect. On the other hand, the robots can collide against the table on which they are mounted.

Although the robots are operated impedance controlled — they can be pushed away at each segment — a collision detection between the robotic arms is required in order to prevent damage of the robot arms.

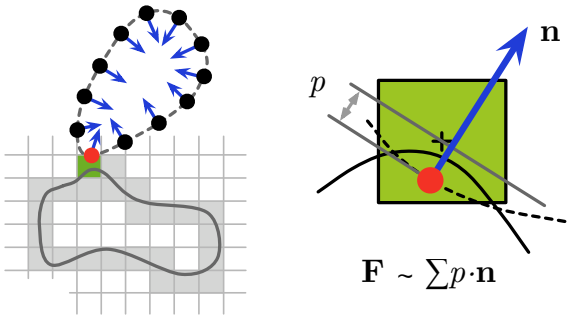

Figure 4: Schematic description of the Voxmap-PointShell® Algorithm. Left: voxmap and pointshell structures are colliding; right: single force computation in a colliding surface-voxel.

The implemented collision avoidance algorithm is based on the robot's partial forward kinematics, i.e. transformation matrices from the robot base to each joint, which specify the pose of each joint. As soon as the distance between a joint and the other robot or the table is below a certain radius, a spring-like force pushes against the imminent collision. In addition to that of the joints, collision avoidance for the three introduced handles is implemented in a similar way.

4. Virtual Reality (VR) Applications

The HMI is used with two different VR scenarios described in this section. For both, two virtual and moveable objects are coupled to the robots. By the use of a tracking system for the HMD, the operator is able to look around in the virtual environment. The virtual scene is rendered in the haptic and visual domain. Both are displayed to the human operator using the two robots of the HMI and the HMD respectively. Depending on the requirements of the task, one of the three introduced handles can be used for each robot.

This section is divided into two subsections: the first one explains the haptic rendering algorithm used to compute collisions and forces in the virtual world, whereas the second subsection describes the performed tests.

4.1. Voxmap-Pointshell® Algorithm

The haptic rendering algorithm used to compute collision forces and torques for the haptic feedback is an adapted version of the Voxmap-PointShell® (VPS) Algorithm [10]. The VPS algorithm enables haptic rendering of virtual objects moving in almost arbitrarily complex virtual environments at an update rate of $1\,kHz$.

Two data-structures are used to compute collision responses: *voxmaps* and *pointshells*, as shown in Fig. 4. Voxmaps are voxelized volume structures arranged for simulating the static properties of the objects. Pointshells, on the other side, represent dynamic or moving objects through clouds of points; each point is located on the surface of the polygonal model and possesses a normal vector pointing inwards the object.

Both data-structures are generated offline, and the algorithms to obtain them are explained in detail in [11]; a short description is given in the following lines, though.

The voxelizing algorithm navigates fast in the gridded bounding box of each triangle of the polygonal model and performs collision tests between candidate surface-voxels (Fig. 4 left, grey shadowed voxels) and triangles using the *Separating Axis Theorem* (SAT); if a voxel is colliding with a triangle, it is marked as a surface-voxel. The SAT simplifies the collision check problem to one dimension stating that two convex shapes do not intersect with each other if and only if no axis exists such that the projections of the shapes on it overlap. The axes to test are $\mathbf{a} \in \{\mathbf{e_i}, \mathbf{f_j}, \mathbf{e_i} \times \mathbf{f_j}, \forall i, j \in \{1, 2, 3\}\}$, being $\mathbf{e_i}$ the cartesian coordinate axes set in the center of the voxel and $\mathbf{f_j}$ the edges of the triangles. After obtaining all the surface-voxels, the voxmap is layered generating a distance-field inwards and outwards the object.

The pointshell generator uses the previously computed voxmap structure of the model and projects surface-voxel centers on the triangles. This is achieved minimizing the square distance function $Q = \|\mathbf{T} - c\|^2$, where $\mathbf{T}$ is a vectorial expression of the triangle and c the center of the surface-voxel to be projected. Once the points are obtained, the normals are computed analyzing the voxmap neighborhood of the point.

During the haptic simulation, collision detection and force computation are performed every $1\,ms$ in the original VPS algorithm [12], traversing all the pointshell-points that are in the scene. Every time a point is inside a surface-voxel, a collision is detected. The penetration is calculated measuring the distance from the point to the normal plane that goes through the center of the voxel. As shown in Fig. 4, during each haptic cycle, the penetration and the normal of colliding point k yield a single collision force $\mathbf{F_k} = p \cdot \mathbf{n}$, and all the collision forces summed together yield the total repulsion force. More detailed explanations concerning the collision detection and force computation are given in [12, 13].

4.2. Performed experiments

Regarding the system configuration of the robots, two dynamic or moving objects have been considered in the simulations, each one controlled by one of the robots. Every haptic cycle three pairs of models must be checked for collision: (i) the right object controlled by the right robot against the static scene, (ii) the left object controlled by the left robot against the scene, and (iii) the right object against the left object.

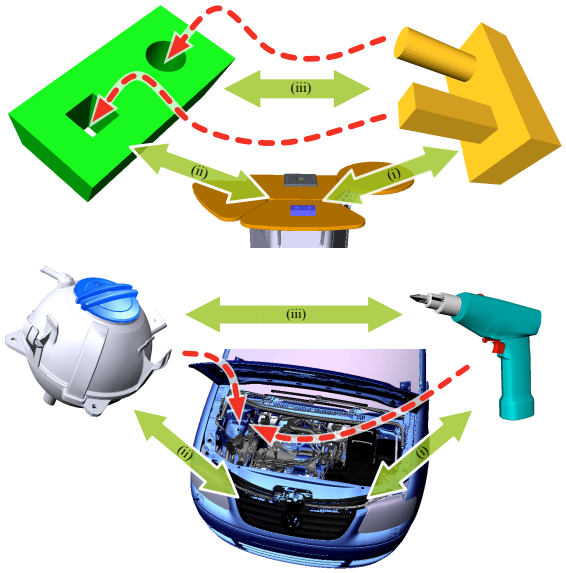

Figure 5: Simulations performed using VR. Top: *Peg-in-hole* test; bottom: assembly test. Dashed arrows represent movement paths, whereas solid arrows show the three pairs of objects to be checked for collision.

Two virtual reality simulations have been performed, both shown in Fig. 5. The first one consists of a *pins*-object coupled to the right robot and a *holes*-object moved by the left robot, constituting the classical *Peg-in-hole* benchmark set in a virtual static scenario containing a table.

The aim of this simulation is to show that the system stays stable during the simulation. In fact, although simple geometries are used, inserting a *pins*-object into a *holes*-object represents a challenging task in VR, given that typically it has been difficult to provide a stable and realistic haptic feedback coping with all the contacts that occur in the simulation. In addition, notice that in the particular moment where the *pins*-object is being inserted into the *holes*-object, a special situation comes up: the robots are coupled in such a way that their relative movement is only allowed in the direction established by the axis of the holes. This situation is handled successfully while maintaining stability through all the simulation.

The second simulation consists in assembling a coolant tank inside the engine hood of a VW Touran.

A remarkable feature of this simulation is the complexity of the objects in the virtual environment. The left robot is coupled with a virtual model of the coolant tank (25,263 triangles), while the other robot is coupled to a virtual electric drill (35,545 triangles). Both are within a scene occupied by the VW Touran model (3,364,266 triangles). Due to the large workspace of the robots, there is no need to scale the motion, i.e. moving a robot one meter will cause the corresponding virtual object to move one meter, too.

The goal of this scenario is to show that it is possible to check the suitability of virtual models in a very early design stage of product development, i.e. without building real mock-ups it can be verified whether the objects can be assembled and maintained. Therefore, possible designing errors can be easily detected and the development process of new cars can be sped up.

Moreover, since the system can be used intuitively, people that are not familiar with robots can also work with the system nearly without training. A practical example would be mechanics that can provide their knowledge and experience directly in the design process by checking assembly tasks on the virtual models.

5. Conclusions

This article introduced a human-scale system composed of two LWRs for bimanual haptic feedback. Control aspects like scaling down the robot's inertia, controlling the LWR's redundant degree of freedom for null-space motion, and avoiding collision were discussed.

With its workspace similar to that of both human arms, the high feedback forces and the control loop with an update rate of $1\,kHz$, this system suits very well as a haptic feedback interface, e.g. to perform virtual assembly tasks or to explore complex virtual scenarios without the need of scaling movements or forces. With the three described end-effectors the hardware setup can be adapted to the respective application.

A virtual assembly simulation of a VW Touran showed that even in complex scenarios the Voxmap-PointShell® Algorithm, which has been enhanced at the DLR [10, 11], is able to generate collision responses within $1\,ms$, and therefore it is suited as haptic rendering algorithm for the bimanual haptic system. Also classical benchmarks for haptic rendering — such as *Peg-in-hole*, in which a big number of contacts occur, without giving rise to high collision forces —, are stable with the system.

In future applications, this system may be used for telemanipulation, e.g. to control another two-robot system while feeding back haptic information.

Acknowledgment

The authors of this work would like to express their gratitude to the VRLab of the Volkswagen AG for supporting this research and for providing the models of the virtual car. This work has been also supported by the EU Government within the project "Multimodal Interfaces for Capturing and Transfer of Skill", project number FP6-IST-2005-035005-IP.

References

[1] G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci, and M. Schedl. Dlr's torque-controlled light weight robot iii - are we reaching the technological limits now? In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1710–1716, Washington D.C., USA, 2002.

[2] F. Zacharias, Ch. Borst, and G. Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3229–3236, San Diego, California, Oct. 2007.

[3] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. Safe Physical Human-Robot Interaction: Measurements, Analysis & New Insights. In *International Symposium on Robotics Research (ISRR2007)*, pages 439–450, Hiroshima, Japan, Nov. 2007.

[4] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. Collision Detection & Reaction: A Contribution to Safe Physical Human-Robot Interaction. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3356–3363, Nice, France, Sep. 2008.

[5] T. Hulin, P. Kremer, R. Scheibe, S. Schätzle, and C. Preusche. Evaluating two novel tactile feedback devices. In *Enactive/07*, Grenoble, France, Nov. 2007.

[6] http://www.immersion.com, 2008.

[7] C. Borst, M. Fischer, S. Haidacher, H. Liu, and Hirzinger G. Dlr hand ii: Experiments and experiences with an anthropomorphic hand. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Taipei, Taiwan, Sep. 2003.

[8] S. Schätzle, T. Hulin, C. Preusche, and G. Hirzinger. Evalution of vibro-tactile feedback to the human arm. In *EuroHaptics 2006*, pages 557–560, Paris, France, July 2006.

[9] J.J. Gil and E. Sánchez. Control algorithms for haptic interaction and modifying the dynamical behavior of the interface. In *2nd International Conference on Enactive Interfaces*, Genoa, Italy, Nov. 2005.

[10] M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, and G. Hirzinger. Stable Haptic Interaction with Virtual Environments using an Adapted Voxmap-Pointshell Algorithm. In *Proc. of Eurohaptics 2001*, Birmingham, UK, July 2001.

[11] M. Sagardia, T. Hulin, C. Preusche, and G. Hirzinger. Improvements of the voxmap-pointshell algorithm – fast generation of haptic data-structures. In *53rd Internationales Wissenschaftliches Kolloquium*, Ilmenau, Sep. 2008.

[12] W.A. McNeely, K.D. Puterbaugh, and J.J. Troy. Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling. In *Proc. of SIGGRAPH*, pages 401–408, Los Angeles, California, Aug. 1999.

[13] W.A. McNeely, K.D. Puterbaugh, and J.J. Troy. Voxel-based 6-dof haptic rendering improvements. *Haptics-e*, 3(7), Jan. 2006.