2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Enhancing Bilateral Teleoperation using Camera-Based Online Virtual Fixtures Generation

Mario Selvaggio, *Student Member, IEEE*, Gennaro Notomista, *Student Member, IEEE*, Fei Chen, *Member, IEEE*, Boyang Gao, Francesco Trapani, Darwin Caldwell, *Fellow, IEEE*[1]

***Abstract*— In this paper we present an interactive system to enhance bilateral teleoperation through online virtual fixtures generation and task switching. This is achieved using a stereo camera system which provides accurate information of the surrounding environment of the robot and of the tasks that have to be performed in it. The use of the proposed approach aims at improving the performances of bilateral teleoperation systems by reducing the human operator workload and increasing both the implementation and the execution efficiency. In fact, using our method virtual guidances do not need to be programmed *a priori* but they can be instead automatically generated and updated making the system suitable for unstructured environments. We strengthen the proposed method using passivity control in order to safely switch between different tasks while teleoperating under active constraints. A series of experiments emulating real industrial scenarios are used to show that the switch between multiple tasks can be passively and safely achieved and handled by the system.**

Keywords: shared control teleoperation, camera-based virtual fixtures generation, multitask handling

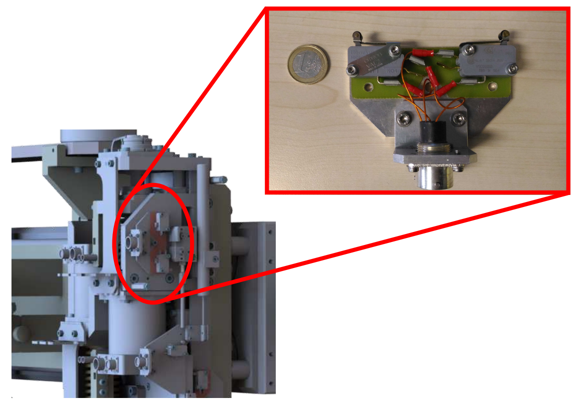


Fig. 1. Two limit switches commonly used in industrial applications: CAD model and related real picture.

I. Introduction

Highly unstructured environments still represent a big challenge for fully autonomous robotic systems. Due to their nature, these scenarios require very accurate estimation of the environment and human-like decision-making which cannot be assured in every circumstance even by state-of-the-art autonomous robots. Human interventions are therefore needed to operate robots under these conditions. Hence, teleoperated robotic systems still represent the only feasible way to accomplish certain tasks in such environments. Moreover, when these tasks require high precision and repeatability, *shared control* can be employed. Shared control teleoperation systems use *virtual fixtures* as human operator perceptual constraints being able to correct the tremor and the drift introduced by the user reducing, in this way, the overall task execution time and increasing the precision and the reliability of the system.

Teleoperation with virtual fixtures has already been successfully used in various robotics fields, such as medical [1], aerial [2] and assistive robotic manipulation [3]. An extensive review about active constraints can be found in [4]. Their usage can help the operator to perform a task following an ideal path while guided by generalized forces acting on the master side of the teleoperation system [5]. However, the design of optimized paths requires the exact knowledge of the environment and accurate description of the task, limiting the use of virtual fixtures mainly to static environments where they can be properly defined offline (i. e. before the task execution starts). In contrast, in case of dynamically changing environments, online generation of virtual fixtures would be required. This can be achieved using information from exteroceptive sensors, e. g. vision systems or laser scanners. Moreover, one of the most critical aspects of the shared control teleoperation is represented by the handling of multiple tasks within the same environment. In case of structured environments a particular strategy can be devised *a priori* and virtual guidances can be programmed accordingly. However, every change in the strategy, e. g., rescheduling of tasks, requires additional effort and can lead to replanning and thus regeneration of corrected fixtures in case of significant changes.

In this paper we address the problem of online virtual fixtures generation and update and we show how it can be tackled using a stereo camera system at the slave side. Moreover, we consider the particular case of task interruptions and/or task-switches during the execution of multiple sequential tasks. This is the case of particular industrial environments in which remotely controlled robotic systems are employed for maintenance operations to reduce personnel exposure to hazards. Specifically, we consider maintenance scenarios in which pressing multiple limit switches (see Fig. 1) is demanded. In this context we show how, using our method, the online reordering of the tasks can be easily and correctly handled.

The work leading to this publication has received funding from the European Union's Seventh Framework Programme under grant agreement no. 608849 EuRoC.

[1]Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. *email:*{name.surname}@iit.it.

II. Related Work

The first author introducing virtual fixtures was Rosemberg in his works [6] and [7]. Interactive generation of active constraints has been previously investigated by Bettini et al. [8]: in their work the authors extended the use of virtual fixtures using computer vision as sensor to provide reference trajectory to the virtual fixtures control algorithm. More recently, many authors have dealt with virtual fixtures use in shared control teleoperation. Ferraguti et al. [9] proved a passivity preserving condition for redirection of virtual fixtures forces in surgical robotics. Comparison between shared control and normal teleoperation in terms of task performances, control effort and cognitive operator workload in a bolt-spanner task can be found in Boessenkool's work [10]. In addition, Vozar et al. [11] analyzed the improvement caused by the use of non-holonomic constraints in time delayed space teleoperation, while Smisek et al. [12] experimentally quantified the effect of the guidance inaccuracy during a peg-in-hole insertion task.

Telerobotic systems have been widely investigated in the past in terms of performances. Transparency, intended as the capability of transferring the task impedance to the human operator located at the master side, is mainly limited by stability issues. The destabilizing effect of the closed loop system increases with communication delay, unavoidable in any teleoperation architecture. Lawrence [13] showed that it is possible to find a trade-off between transparency and stability regardless of the used teleoperation scheme. Passive behavior can be obtained using a passivity controller which prevents the released energy from being greater than the injected one. In this paper we use the two-layer approach from Franken et al. [14].

The contribution of this paper is twofold. First, we show an online camera-based virtual fixtures generation technique suitable for bilateral teleoperation systems when used in unknown or partially known environments. Second, we extend the use of active constraints making them suitable for multiple tasks execution. This leaves the freedom of choosing the sequence of the tasks to the user improving in this way her/his experience using human-in-the-loop systems.

III. Theory

In many teleoperation systems master and slave sides are represented by n-degree-of-freedom impedance/admittance controlled robots which behave as passive decoupled mechanical systems. Once connected by means of a passive bilateral coupling, if the communication delay is negligible the overall system can be considered stable. In order to enhance the performances of a teleoperation system, virtual guidances can be added at the master side. Virtual guidances can help to precisely move the teleoperated slave robot along a desired path while leaving the ultimate control to the user. This results in additional forces imposed on the master robot. For the purpose of proofing system stability in such conditions, it is suitable to model every physical system as a two-port Hamiltonian mechanism. Formally, it can be represented as:

$$\begin{cases} \dot{x} = [J(x) - R(x)] \frac{\partial H}{\partial x} + g(x) u \\ y = g^T(x) \frac{\partial H}{\partial x} \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$ represent the system state $H(x) : \mathbb{R}^n \to \mathbb{R}$ is the Hamiltonian function, namely the sum of system stored energy, $J(x)$ represents the internal coupling, $R(x)$ the internal dissipation, $g(x)$ the input matrix, u the input and y the system response. It can be shown that for a two-port Hamiltonian system the passivity condition with respect to $u^T y$ can be written as:

$$u^T y = \dot{H}(x) + \frac{\partial^T H}{\partial x} R(x) \frac{\partial H}{\partial x} \geq \dot{H}(x) \tag{2}$$

If virtual constraints are passive elements, the effect of the connection to the master side does not influence the passivity condition. In this case the term u in Eq. (1) in presence of virtual fixtures can be written as:

$$u = F_{ext} + F_{fb} + F_{vf} \tag{3}$$

where F_{ext} is the force exchanged with the human operator, F_{fb} the force feedback and F_{vf} represents the additional guidance force of the virtual fixtures. In particular, the term F_{fb} is the force coming from the slave side and depends on the adopted teleoperation scheme, such as position-position or position-force, while F_{vf} is computed by a suitable constraint enforcement method.

A. Virtual Fixtures

From a geometric point of view, active constraints can be modeled using different primitives such as points or parametric curves, surfaces and volumes. Their geometric description constitutes the basis for the successive computation of attractive or repulsive forces. Suitable geometric primitives that allow to specify both starting and ending points and relative tangents are cubic splines, formally defined as:

$$p(s) = \begin{bmatrix} x(s) \\ y(s) \\ z(s) \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} s^3 + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} s^2 + \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} s + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \tag{4}$$

where $s \in [0, 1]$ is the curve parameter that has been chosen to go from 0 to 1 to simplify the computation of the force associated with the relative virtual fixture (see Sect. III-C). Moreover, it is clear that the use of these path primitives is intrinsically safe when there are no obstacles between the robot and the target region.

B. Path Generation

The generation of the path described by Eq. (4) consists in finding the parameters that describe it, namely $a_x, a_y, a_z, b_x, \ldots, d_y, d_z$. Once the initial and final points and tangents have been defined, the computation of these parameters is straightforward. As regards the starting point, it has been chosen to be coincident with the position of the robot at the same instant at which the path is generated, while the initial tangent is forced to lie along the z axis of the TCP

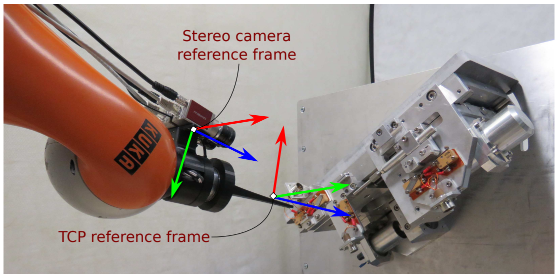


Fig. 2. The slave robot end-effector: the stereo camera and the TCP (Tool Center Point) reference frames.

reference frame (see Fig. 2). As far as the final point and the final tangent are concerned, these are obtained by the aid of a stereo camera system mounted on the slave robot in eye-in-hand configuration, as depicted in Fig. 2. The final points are detected in both images and their 3D coordinates are computed in the stereo camera reference frame making use of triangulation techniques. For a calibrated horizontal stereo camera system the following holds:

$$z_p = \frac{b}{\Delta d} f \tag{5}$$

where z_p is the distance of the detected point, b is the baseline, i. e. the horizontal distance between the cameras, Δd is the horizontal disparity and f is the cameras focal length. The coordinates x_p and y_p are obtained in a similar fashion. Each considered point $p = [x_p, y_p, z_p]$ is then used by the active constraints generator module to provide the path directly in the TCP reference frame. The time required to online generate the virtual fixtures is negligible with respect to the operating cycle time of the system making the considered online active constraints generation approach feasible.

C. Force Generation

The most common constraint enforcement method, widely used in previous literature, generates the force as a function of the shortest distance between the position of a point and the constraint geometry. There are several ways of computing the minimum distance which can be iterative or in closed form depending on the mathematical description of the constraint. If the constraint is specified as in Eq. (4) the shortest distance computation can be performed rapidly by finding the roots of the polynomial representing the first derivative of the distance function:

$$d(s) = \|p - p(s)\| \tag{6}$$

As stated above, the fact that the parameter s belongs to the interval $[0, 1]$ narrows down the roots search to this interval making this procedure very efficient even by using simple techniques such as the well known bisection method. Once the minimum distance has been found, the operator helping force can be calculated as proportional to it, such as:

$$F(p,s) = K_p (p - p(\bar{s})) \qquad \bar{s} = \min_s \{d(s)\} \tag{7}$$

This can be interpreted as a mechanical spring or, equivalently, as a proportional controller (lossless passive systems).

D. Switch Between Multiple Tasks

When multiple tasks have to be accomplished the operator needs, sometimes, to efficiently switch among them, changing, for instance, the order in which they must be carried on. As already stated above, when the operator moves away from a predefined virtual fixture she/he feels a force proportional to the distance from the nearest point of the path (see Eq. (7)). In order to make the switch happen, a threshold value on the distance or on the force can be considered. In our implementation we compute, at each time step, the vector containing the minimum distances between the current robot position and all the present virtual constraints. The minimum value in this vector represents the only distance contributing to the attractive force. This approach turns out to be the most suitable for our case for two reasons: when the virtual fixtures are close to each other, the switch can easily happen without applying too much force, while when they are far, the safety of the task execution is guaranteed if the operator has to necessarily deviate from the current path.

The system behavior during the switch depends on the used passivity controller. In our case we have an active constraints generator connected through force exchange to the master robot. In terms of system model this can be seen as two interconnected two-port Hamiltonian systems (see Eq. (1)). In order to preserve the passivity condition during the switch, we attach an energy tank to the interconnected systems which monitors the exchanged energy and applies passivity correction if the energy level drops below a desired value. This passivity action is explicated through damping injection on the master device allowing the tank to get replenished. This can be seen in Fig. 3 where an example experiment is reported. In the considered experiment the user switch between two 1D virtual fixtures located at $y = 0.05$

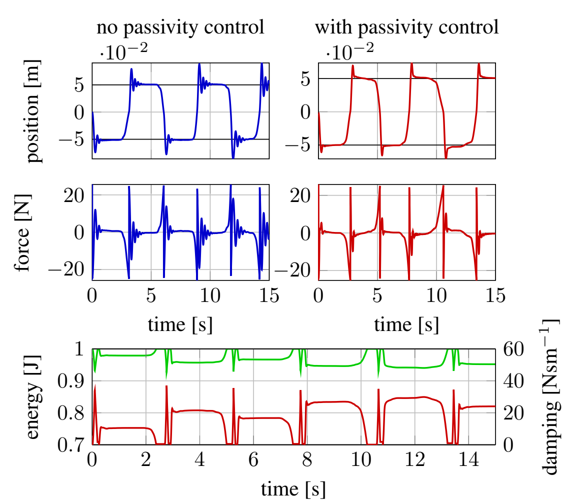


Fig. 3. Comparison between virtual fixture switch procedures with and without passivity control. The graph at the bottom depicts the trends of energy (green) and damping (red) in case of controlled switch.

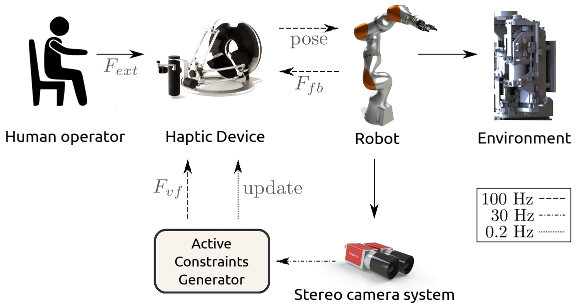


Fig. 4. Block diagram of the overall teleoperation system with virtual fixtures at the master side. The active constraints generator block computes and updates virtual constraints using feature detection algorithms in the images provided by the stereo camera system. Different lines types are intended to show different working frequencies.

and $y = -0.05$ along y axis. As it can be noticed, the passivity control allows to modulate the switch in terms of fluctuation and overshot keeping the system stable. The used control parameters are: $K_p = 500$ Nm^{-1}, $H_{init} = 1.0$ J, $H_d = 1.0$ J, $H_{max} = 1.0$ J and $\alpha = 500$ sm^{-2} (for further details refer to Franken et al. [14]).

IV. System Description

The overall teleoperation system including the active constraints generator is depicted as a block diagram in Fig. 4. The absolute input to the system is represented by the human operator who guides the haptic device in its cartesian space. The change in position and orientation of the haptic device, opportunely scaled and mapped, is sent to the robot as a cartesian space desired position. In order to make the connection appear natural to the user when teleoperating, the slave desired pose is expressed with respect to the camera reference frame. The active constraints generator calculates goal poses every 5 s using feature detection algorithms in the images coming from the stereo camera system mounted on the slave robot. These poses are then used to compute and update virtual fixtures as explained in Sect. III-B and summarized in Algorithm 1. The output of this subsystem is directly the force that is rendered by the haptic device.

Algorithm 1 Virtual Fixtures Update and Rendering

Require: *image*, *elapsedTime*
  if *elapsedTime* $\geq 5s$ **then**
    $n \leftarrow$ limitSwitchesDetection(*image*)
    $vf \leftarrow$ updateVirtualFixtures(n)
  else
    for i **to** n **do**
      $d \leftarrow$ calculateMinDistance(vf_i)
    end for
    $d_{min} \leftarrow \min(d)$
    $F_{vf} \leftarrow$ computeForce(d_{min})
    $F_{vf} \leftarrow$ passivityControl(F_{vf})
    $F \leftarrow F + F_{vf}$
  end if

A. Hardware

The system hardware consists of a KUKA LWR 4+ slave robot teleoperated by means of a Force Dimension Omega 7 master device. The stereo camera system is composed of two Allied Vision Mako G-125C cameras which provide 1.2 Megapixel (1292×964) frames at 30 fps. An external force-torque sensor is mounted on the robot end-effector. It is sampled at 1 kHz, and a low pass filter with cutoff frequency $f_c = 100$ Hz is used to filter the force signal. Two computers running Linux Ubuntu 14.04 are used: one of these is connected to the two cameras and it is receiving signals through the Ethernet port; the other one is connected to the Omega via USB and to the robot with an Ethernet cable. These two laptops communicate between each other by means of a Wi-Fi socket.

B. Software

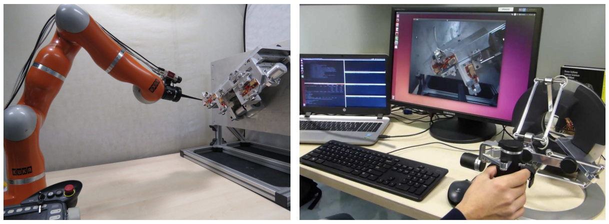
The KUKA LWR 4+ and one computer are connected through Fast Research Interface provided by the robot. The Fast Research Interface Library [15] runs on a remote computer which is connected to the KRC (KUKA Robot Controller) via an Ethernet connection. This subsystem runs at 5 ms for testing purposes, even though it can be operated up to a frequency of 1 kHz. The robot is commanded in cartesian impedance mode with following values for the stiffness $K = [2500\ 2500\ 2500\ 300\ 300\ 300]$ and damping ratio $D = [0.7\ 0.7\ 0.7\ 0.7\ 0.7\ 0.7]$, along and about its cartesian axes. The vision node has double function: it communicates with cameras via Ethernet using the Vimba SDK and exploits OpenCV library [16], which contains basic image processing functions, in order to detect and localize limit switches in the images. The accuracy of the localization algorithm is below 5 mm. As regards the virtual fixtures, the magnitude of the tangents at the starting and ending points has been set to 0.01. The value used for K_p of Eq. (7) is 500 Nm^{-1}. The Omega 7 device is interfaced to the system by means of the Force Dimension SDK. It provides high frequency signals of haptic device pose, velocity and force. In addition, the energy tank based control has been implemented to stabilize the force feedback loop between master and slave sides.

V. Experiments and Results

In this section we present both the setup of the experiments conducted on the mock-up of an industrial environment and the related outcomes. In section V-A we briefly describe the scenario in which the experiments are carried out, while in section V-B we extensively analyze the recorded experimental data.

A. Experimental setup

The specific evaluation procedure adopted in this work to demonstrate the validity of the proposed system consists in pressing 5 limit switches (see Fig. 1) in different order. The experimental setup is shown in Fig. 5. In the startup phase the offset between the robot and the haptic device is computed, allowing to start from generic initial conditions. The operator points the cameras towards the limit switches so that their



(a) KUKA LWR 4+ and mock-up of the industrial environment. (b) Teleoperator workplace with haptic and visual feedback.

Fig. 5. Setup of the experiments.

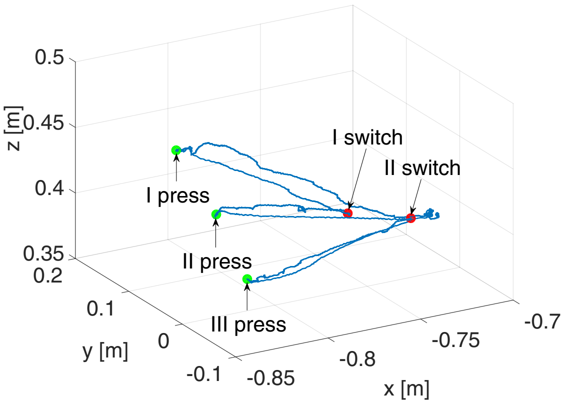
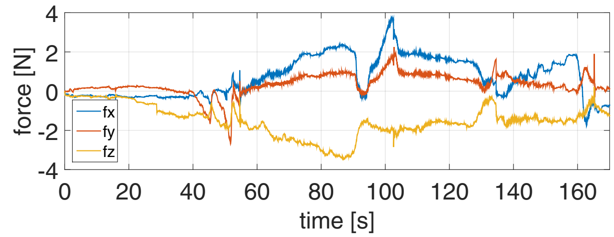
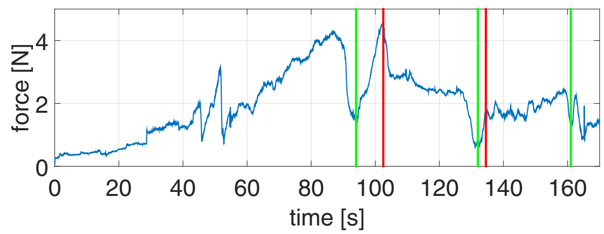


Fig. 6. Robot position (expressed in the robot base frame) recorded during the execution of the considered experimental test with 3 limit switches pressed. Green dots represent the positions of the robot end-effector when limit switches are pressed, while red dots denote the positions in which the switches between two virtual fixtures occur.



(a) Components of the force rendered by the haptic device due to the virtual fixtures (except for the scale factor K_p, these components coincide with those of the error between the commanded position and the desired path).



(b) Magnitude of the force due to virtual fixtures. Timestamps marked in green represent the time instances at which the limit switches are pressed, red lines indicate the switch between two virtual fixtures.

Fig. 7. Recorded force data during the considered experiment.

poses in the stereo camera reference frame can be calculated. These poses are used to generate the virtual fixtures required by the system to operate as previously described.

B. Outcomes

For the sake of clarity we refer to a generic performed test. In this test 3 limit switches are pressed. Fig. 6 depicts the path followed by the slave robot end-effector. Green dots represent robot positions when limit switches are pressed, while red dots the positions in which the switches between virtual fixtures take place. In Fig. 7 the values of the components and the magnitude of the force due to the virtual fixtures are reported. As it can be noticed, in the initial phase, around 50 seconds, the force is oscillating because the operator looks for the desired virtual fixture. Between 50 and 90 seconds the magnitude of the force increases because the teleoperator is subject to the guidance force generated by the virtual fixture. At 94 seconds the operator presses the first limit switch. The force attains a minimum value since the operator reaches the end of the virtual fixture and uses only the rotational degrees of freedom to accomplish the task. Between 94 and 102 seconds, instead, the operator moves backwards and applies a force towards a different virtual fixture in order to switch to it. The same sequence is, then, repeated starting at 134 seconds when the other limit switch is pressed. Finally, the third limit switch is pressed around 161 seconds, allowing the completion of the overall task in approximately 2 minutes.

Considering that the force required to press the limit switch is equal to 4.125 N on average, the presence of the virtual fixtures does not deteriorate the transparency of the teleoperation system. In fact, during the pressing phase, the force coming from the active constraints never exceeds the 40% of the external force coming from the environment. It has to be pointed out that the force values in the previous diagrams represent the force after the passivity controller. Finally, the sequence of images in Fig. 8 shows the working conditions of the presented system from the teleoperator point of view. In the first row there are 4 frames extracted from the video stream that is shown to the human operator during the execution of the previously considered experimental test. More in detail, the sequence of images represents the first phase of the experiment in which the operator is scanning the environment. The system searches for the limit switches and computes the virtual fixtures that are instantaneously superimposed onto the video frames. Additionally, on the second row the detected limit switches are plotted in the 3D stereo camera reference frame together with the computed virtual fixtures.

VI. Conclusions

In this paper we presented how to improve bilateral teleoperation systems by means of camera-based virtual fixtures generation. Using this method virtual fixtures do not need to be programmed a priori but they can be generated and updated online making the system suitable for multiple task executions in unstructured environments. In addition, we showed how to passively switch between different tasks during shared controlled teleoperation. This

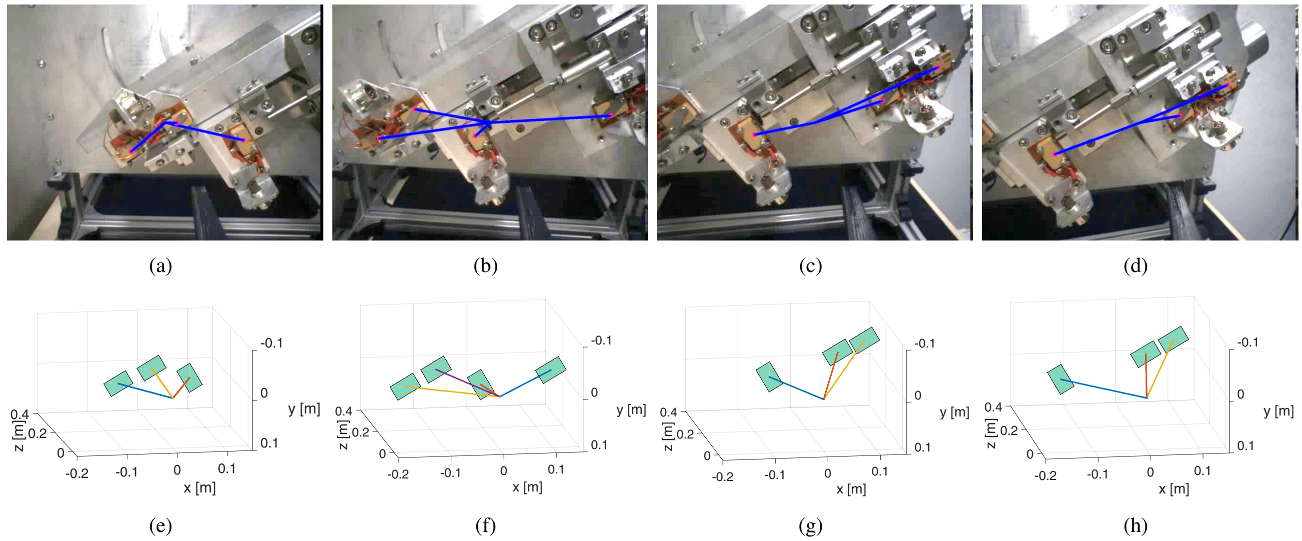


Fig. 8. Key frames taken during the considered experimental test by the stereo camera system located at the slave side (first row) and relative limit switches poses in the stereo camera reference frame (second row). From left to right the operator scans the entire mock-up specifying the robot pose with the haptic device; the virtual fixtures are dynamically computed, consequently updated ((e), (f), (g) and (h)) and shown in the visual feedback to the human operator ((a), (b), (c) and (d)).

allows the operator to easily, rapidly and safely move among various virtual fixtures making the system more efficient with respect to both the implementation and the task execution time. The proposed system has been proven to be a valid alternative with respect to the current consolidated methods based on offline active constraints generation. Furthermore, its reliability has been demonstrated not to be decreased by the online update procedure of the virtual fixtures. The performed experiments confirm what has been stated above and provide us with guidelines towards further developments.

REFERENCES

[1] G. D. Hager, A. M. Okamura, P. Kazanzides, L. L. Whitcomb, G. Fichtinger, and R. H. Taylor, "Surgical and interventional robotics: part iii [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 15, no. 4, pp. 84–93, 2008.

[2] J. Cacace, A. Finzi, and V. Lippiello, "A mixed-initiative control system for an aerial service vehicle supported by force feedback," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1230–1235.

[3] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research*. Springer, 2007, pp. 49–64.

[4] S. Bowyer, B. Davies, and F. Rodriguez y Baena, "Active constraints/virtual fixtures: A survey," *Robotics, IEEE Transactions on*, vol. 30, no. 1, pp. 138–157, Feb 2014.

[5] G. Niemeyer, C. Preusche, and G. Hirzinger, "Telerobotics," in *Springer handbook of robotics*. Springer, 2008, pp. 741–757.

[6] L. B. Rosenberg, "The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments." DTIC Document, Tech. Rep., 1992.

[7] ——, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 76–82.

[8] A. Bettini, S. Lang, A. M. Okamura, and G. D. Hager, "Vision assisted control for manipulation using virtual fixtures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2001: Expanding the Societal Role of Robotics in the the Next Millennium, Maui, HI, USA, October 29 - November 3, 2001*, 2001, pp. 1171–1176. [Online]. Available: http://dx.doi.org/10.1109/IROS.2001.976327

[9] F. Ferraguti, N. Preda, M. Bonfè, and C. Secchi, "Bilateral teleoperation of a dual arms surgical robot with passive virtual fixtures generation," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, September 2015.

[10] H. Boessenkool, D. A. Abbink, C. Heemskerk, F. C. T. van der Helm, and J. G. W. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *IEEE T. Haptics*, vol. 6, no. 1, pp. 2–12, 2013. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TOH.2012.22

[11] S. Vozar, Z. Chen, P. Kazanzides, and L. L. Whitcomb, "Preliminary study of virtual nonholonomic constraints for time-delayed teleoperation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, 2015, pp. 4244–4250. [Online]. Available: http://dx.doi.org/10.1109/IROS.2015.7353978

[12] J. Smisek, M. van Paassen, and A. Schiele, "Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy," in *World Haptics Conference (WHC), 2015 IEEE*, June 2015, pp. 500–505.

[13] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, Oct 1993.

[14] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011. [Online]. Available: http://dx.doi.org/10.1109/TRO.2011.2142430

[15] Torsten Kroeger, "Fast Research Interface Library," http://cs.stanford.edu/people/tkr/fri/html/index.html, 2014.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] N. Turro, O. Khatib, and E. Coste-Maniere, "Haptically augmented teleoperation," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 386–392.

[18] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 440–447.

[19] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4948–4953.

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.