The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Skill-based telemanipulation by means of intelligent robots

Simon Notheis, Giulio Milighetti, Björn Hein, Heinz Wörn, Jürgen Beyerer

Abstract— **In order to enable robots to execute highly dynamic tasks in dangerous or remote environments, a semi-automatic teleoperation concept has been developed and will be presented in this paper. It relies on a modular software architecture, which allows intuitive control over the robot and compensates latency-based risks by using Augmented Reality techniques together with path prediction and collision avoidance to provide the remote user with visual feedback about the tasks and skills that will be executed. Based on this architecture different skills with high dynamics are integrated in the robot control, so that they can be executed autonomously without the delayed feedback of the user. The skill-based grasping by adherence of smooth or fragile objects during a remote controlled picking and placing task will be exemplary presented.**

I. INTRODUCTION

Both in industrial and in public environments, robots can discharge humans from dangerous, exhausting or even repeatable tasks like rescue missions, handling of heavy parts, palletizing, . . . However, due to the complexity of the task and to the highly dynamic environment, the robot is not always able to achieve the given goal in a fully autonomous way and the presence of a human supervisor on-site cannot be ensured at any time. There are in fact many applications, in which the user cannot or will not physically reach the environment, e. g. underwater missions or maintenance tasks in distant sites. Therefore, it is either convenient or necessary to use teleoperated robots, which are responsible for the execution of the decisions made by a human user. Interesting applications are for example maintenance of drilling platforms in deep sea [9], handling of hazardous materials [10] or even space robotics [7].

A critical point in every teleoperated application is the presence of latency in the control loop between the user's input and the feedback on the robot's current state and environment (see also section III-C). Although model-based approaches have successfully been used to compensate some of the implied problems (see [7], [8]), this delay makes it impossible to react promptly to sudden changes in the environment. Thus, a fully manually controlled robot cannot cope with dynamic scenarios with unpredictable events and disturbances. Therefore, semi-automatic solutions have to be implemented. That means, on the one hand intuitive ways of interaction and an immersive impression of the robot environment have to be provided. On the other hand the robot has to dispose of some relevant basic skills [2], which can be activated by the human and that the robot can perform safely without further external inputs.

An example of such a skill can be the autonomous grasping by adherence of smooth or fragile objects during a remote controlled picking and placing task (see figure 3). The gripper has to exercise the minimal contact force that ensures a stable grasping. That means a force which is on the one hand high enough to avoid slip of the object, on the other hand not so high that it could damage it [1]. A robot can execute such a safe grasping without slip even in case of fragile objects with unknown friction coefficient only if both the slip and the contact force are intelligently controlled. An immediate reaction to external disturbances can only be achieved by means of fast control cycles, which cannot be realized by a manually teleoperated robot. Hence the need for an independent module, which can control the grasping process without the feedback of the human.

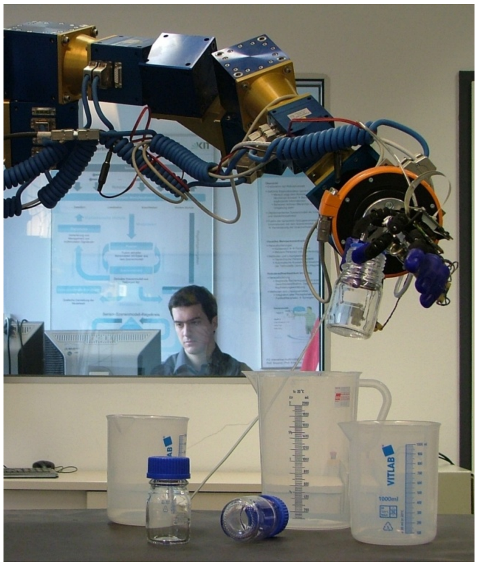

Fig. 1. Precision grasping of fragile objects in hazardous environment.

Through the cooperation of Fraunhofer IOSB and IPR a general skill-based teleoperation concept has been developed and will be presented in the following sections. First an exemplary hardware and software set-up will be described, then the adopted control concepts will be illustrated. In the last section first experimental results will be discussed.

II. EXPERIMENTAL SET-UP

In order to test the proposed concept in different scenarios, two platforms have been developed. After a very

S. Notheis, B. Hein and H. Wörn are with the Institute for Process Control and Robotics (IPR), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany `name.surname@kit.edu`

G. Milighetti and J. Beyerer are with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), 76131 Karlsruhe, Germany `name.surname@iosb.fraunhofer.de`

short introduction of the two systems, the common software architecture will be described in detail.

A. Hardware Architecture

The first hardware set-up consists of two 6-axes industrial robots Kuka KR16. One of them carries a standard firewire camera providing the general scene overview for the remote user, the second robot performs the grasping and manipulation tasks. In an exemplary scenario, the objective is to grasp unordered, soft objects as shown in figure 2.

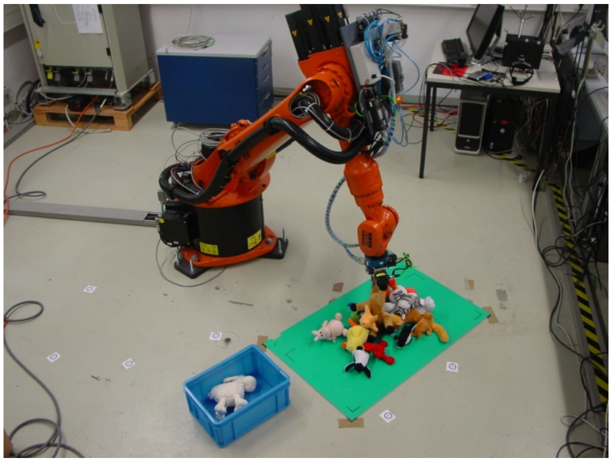

Fig. 2. Exemplary scenario with an 6-axes industrial robot equipped with a gripper, and an unordered pile of soft objects which are to be grasped.

For an alternative scenario (e. g. maintenance task), a mobile robot platform is equipped with a multiple axes arm, which is provided with a gripper. The scene view camera is mounted on an additional pan/tilt unit above the mobile platform covering the workspace of the arm and gripper. Although the hardware integration of the platform and arm is not yet finished, all hardware and software components were tested individually. The pan/tilt unit, for example, may replace the robot-mounted camera in the first scenario.

For the experiments on reactive grasping, a standard PG 70 Gripper of the company Schunk has been used (see figure 3). A tactile sensor array DSA 9205 of the company Weiss Robotics [4] has been integrated in one of its fingers and a slip sensor developed and patented at Fraunhofer IOSB [3] in the second one. The slip sensor uses the same working principle as an optical mouse. A light-emitting diode (LED) lights the surface, while a digital signal processor together with a CMOS sensor are responsible for the extraction of images and the detection of patterns (see figure 4). By comparing two images in sequence, it is possible to calculate how these patterns have changed over time and consequently the sensor displacement. The covered distance is given in counts. The sensor used for the experiments shown in this paper has a measuring rate of 1.500 fps, a nominal resolution of 300 cpi and a size of 23 mm x 60 mm x 6 mm.

The scene reconstruction is made by means of a time-of-flight (TOF) camera with a resoution of 64 x 48 pixels mounted on one side of the gripper. The sensor uses modulated infrared light to generate a 2D depth image at a framerate of 25 fps.

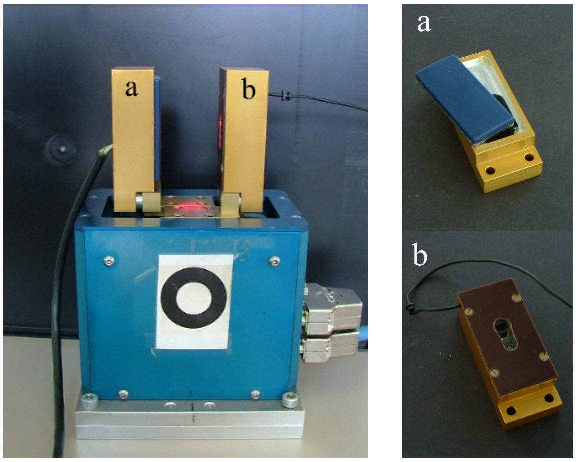

Fig. 3. Gripper with integrated tactile array (a) and slip sensor (b).

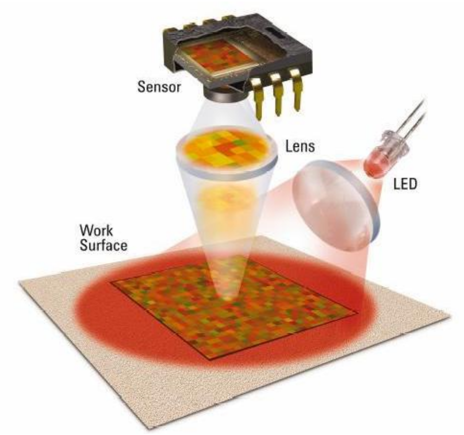

Fig. 4. Functioning principle of the slip sensor [5].

The remote user wears a head-mounted display (HMD) that is USB-powered and uses OLED displays with a resolution of 800 x 600 pixels. The movements of the user's head are tracked using an additional inertial sensor (INS) mounted on the HMD. A standard 6D mouse serves as further input device.

B. Software Architecture

The implemented software architecture provides a modular structure with components for every kind of I/O devices including sensors and robot control, as well as for all algorithmic tasks like processing image or scene data, deriving and validating commands from the user's inputs or path planning. The components can be easily (re-)connected and configured graphically during runtime. To support local as well as distributed components, CORBA (ACE/TAO) is used as underlying middleware. All components share well-defined interfaces and data flow structures to support extensibility and exchangeability.

The generic input and output ports (*DataSinks* and *DataSources*) of each component (*ParentDevice*) are characterized by their communication type (synchronous *Push* or asynchronous *Notify*) and the data structure they are transporting (e. g. *Vector6* for position/orientation information, velocity/acceleration values, force/torque sensor data, etc.). Additional information can be stored in *Configs* for each component and data port. Compatible ports can be connected and disconnected via a graphical representation of the system's structure which can be generated and updated on-line. Figure 5 shows the schematic structure of an exemplary component with variable number of input and output ports, each of them characterized by communication method and transported data structure. Figure 6 shows the visualized structure of all input and output (feedback) components of the proposed scenario that are directly involved in the interaction with the remote user. The components reading and processing sensor data for the scene analysis are not shown in the graph. The properties of each component and their connections can be changed on-line by means of this graphical interface.

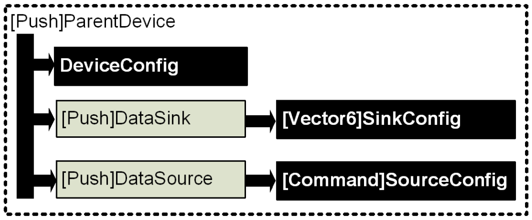

Fig. 5. Schematic structure of each software component in the proposed software framework.

C. 3D Scene Model

Central aspect for the success of a teleoperated task is, of course, the efficient reconstruction of the actual scene, so that the user can access every important information. All stationary objects and obstacles as well as the robots' geometry and their kinematics information are known and represented by CAD models. The robots' pose is updated with the current axis values received from the robot control and is projected in the static scene. The 3D scene model of non-rigid objects or scene parts that may change dynamically is generated on-line via the 3D TOF sensor and represented as a 3D point cloud.

In semi-automatic mode, valid grasp points for randomly positioned and unknown objects can be calculated by using the algorithms presented in [6]. The depth image of the point cloud is preprocessed by closing gaps and smoothing the image, and then used to extract the objects' contours. An extended 2D template matching algorithm determines valid grasp points considering a model of the given gripper's contact surface (see figure 7).

III. USER INTERACTION AND CONTROL CONCEPT

This section will describe how the remote user can interact with the system. We will show how the scene view camera, the robot, and the grasping task are controlled and how the visual feedback for the user is realized.

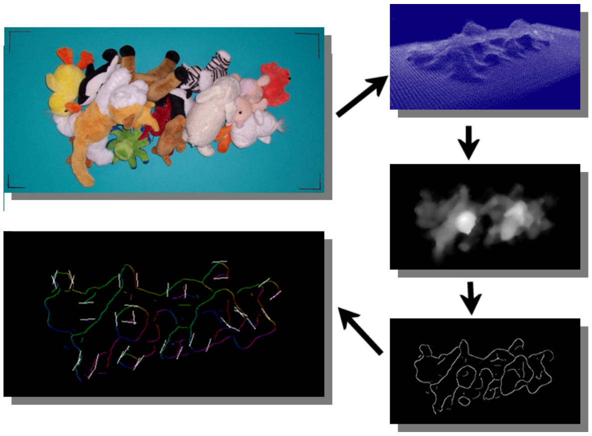

Fig. 7. Image processing steps to extract possible grasp positions from the depth image of the point cloud recorded by the 3D TOF sensor [6].

A. Camera Control

The scene view camera, which is mounted either on the pan/tilt unit or on the first robot, is controlled via the inertial sensor of the HMD providing an intuitive hands-free scene overview. In the first case, the orientation of the camera is directly controlled by the orientation of the user's head. In the second case, the camera mounted on the robot's flange is moved on a spherical path around the target object following the relative movements of the head.

However, this way of controlling the camera has shown to be not very intuitive if it is extended to movements with higher degrees of freedom, i. e. moving the camera on a spherical surface around the target object and allowing at the same time zooming in/out (either by moving the camera towards/away from the target object or digitally zooming the image by setting the camera's region of interest). However, such a full camera control can be realized using the 6D space mouse.

Video streaming has been done by using standard video streaming APIs (VLC, GStreamer) and different codec/bitrate configurations. The location of the video source and its parameters are communicated via correspondent components of the proposed software framework. The images can be pre- or postprocessed on both ends of the video stream, allowing to add virtual information and user interfaces directly into the camera image (see section III-E).

B. Robot Control

The presented system supports both a semi-automatic and a manual control mode for realizing the robot's movement in order to perform the grasping task.

In semi-automatic mode, the remote user is asked to select one of the grasp points determined by the image processing algorithms mentioned in section II-C. The proposed valid grasp points and the corresponding movements of the robot to reach them can be simulated and rendered into the camera image. By controlling the scene view camera, the user can verify the planned grasp. After selecting a grasp point, the

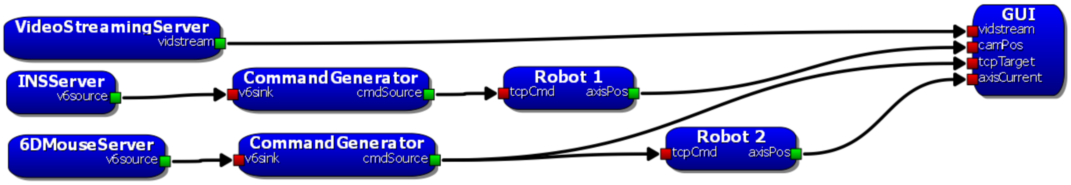

Fig. 6. On-line visualization of the system components involved in the presented scenario.

user can control the execution speed of the task by holding down the 6D mouse in direction of the z-axis.

In manual mode, the remote user provides relative movement commands for the TCP in cartesian coordinates via the 6D mouse. The TCP's orientation may be provided by the user as well, or can be determined by the system depending on the geometry of the gripper and the objects to grasp.

In both cases, as well as during camera control via the HMD, the difficulties introduced by the latency have been considered and some solutions have been adopted that will be described in the next section.

C. Latency compensation

When transmitting video or input device data over unreliable channels like internet connections, latency is alway a problem that the user has to face. In interactive scenarios like remote controlling of a robot that has to perform tasks which require high positional precision and fast reaction time, this is not acceptable.

Currently three different approaches have been implemented to counteract latency: visual feedback of the input commands, path prediction combined with on-line collision avoidance and activation of autonomous basic skills.

All inputs from the user resulting in camera or robot movements are translated into an absolute reference frame before being sent to the robot. That means, that only absolute target positions instead of relative movement commands are sent to the robot and therefore all commands generated by the user are independent of the robot's current state which may not be known exactly due to the connection latency. In this way the user can directly be provided with visual feedback about the corresponding commands and the associated target positions. This information is augmented directly into the scene view image, so that the user can always see in advance what the robot is going to do and react promptly in case of wrong trajectories, even in presence of big latencies.

In addition, the robot's actual movement towards the provided target position is predicted on-line and a collision test is made within the current scene model. Since this is done on the robot's end of the communication channel, the given input commands can be validated or rejected before reaching critical positions.

Last, the user disposes of different basic skills, which can be executed autonomously by the robot. One very important skill, which is always required for the execution of almost every task, is for example the safe grasping of parts. The control algorithm used by the robot in order to realize such a crucial skill is described in the next section.

D. Grasp control

In order to ensure a stable grasping without damaging the object, a slip and force control has been developed. Once this mode has been activated by the user, the gripper reacts immediately to a minimal slip of the object (for the presented experimental set-up 1 count represents approximately 0.085 mm) by closing very quickly. If no slip is detected for a certain period, the gripper opens slowly again. This human-like behavior leads to a rapid response to external disturbances and at the same time to a gradual return to steady-state conditions once the disturbance disappears.

Modifying the time constant which affects the reaction of the gripper, the priority of the control algorithm can be given to the minimization either of the contact forces or of the allowed slip. Moreover the control parameters can be optimized depending on the friction between finger and object. The friction coefficient is estimated at the beginning of every grasping process by opening very slowly the gripper till the object starts slipping. In this moment the relation (1) is verified.

$$\mu = \frac{F_{\text{Weight}}}{F_{\text{Grasp}}} \tag{1}$$

The weight of the object F_{Weight} is measured by means of a force-torque-sensor in the robot wrist and the contact force F_{Grasp} by means of the tactile array in the finger. It has been demonstrated by numerous experiments, that a classification of the friction coefficient in one of just three different classes (small, medium and big friction) is enough in order to select optimal control parameters. Some experimental results will be briefly discussed in section IV.

E. Graphical User Interface

As already mentioned, predicted robot movements and target positions derived from the user's input are rendered into the camera image. Further information calculated from the scene model (like distances, etc.) or addition virtual cameras can easily be added. The intrinsic and extrinsic parameters of the actual camera are transfered to the virtual camera in the 3D scene. All stationary parts of the scene including the robots are then only rendered into the graphics system's z-buffer and the camera image is undistorted and projected into the background of the scene. In this way the visual information of the original camera image and the 3D depth information of the scene model are combined. Any additional information (like grasp points, distance informations and planned robot movements visualized by an additional semi-transparent robot model) can thus be rendered correctly considering depth information and occlusion.

Moreover, a simple menu is added into the scene, so that the user can select some options and confirm commands. This menu is represented by a minimal XML-based description (*MiniUI*) that was developed at the IPR. By using this compact abstract description, the structure and entries of the user interface can be easily generated or reconfigured during runtime. The MiniUI can be rendered in different GUI environments (2D or 3D OpenGL placement within the scene, 2D window system GUIs, Java Applets/Midlets, etc.) and can be transmitted via CORBA, standard network sockets or bluetooth.

Figure 8 shows the current scene model while the system waits for the user to choose one of the provided grasp positions. The left robot carries the camera, the right one will perform the grasping task. The semi-transparent model is used to visualize predicted robot movements. Figures 9 and 10 show the view of the virtual and real camera respectively. Figure 11 shows the merged view in which the original camera image is augmented with all possible grasp points, the planned movement of the robot towards the grasp point currently selected by the user, and the MiniUI to control the grasping task. Besides the MiniUI, additional virtual and real camera views together with miscellaneous status information can be rendered into the camera image as well.

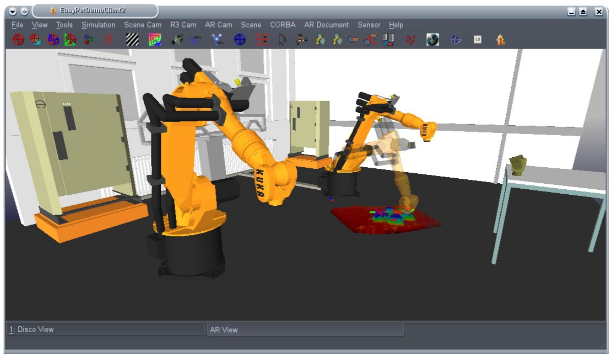

Fig. 8. Current scene model: The left robot carries the scene view camera, the right one performs the grasping task.

IV. EXPERIMENTAL RESULTS AND FUTURE WORK

In order to validate the proposed concept, a remote user has been asked to grasp unordered, soft objects in the scenario already shown in figures 2 and 8. Most of the experiments have been carried out using an internet connection over the 3G UMTS cellphone network. Using the techniques described in section III-C, it was possible to compensate the latency issues between the user's control input and the visual feedback (typically 0.5 to 2.0 seconds depending on the video encoding and the internet connection type and quality). However, it might be interesting to evalueate alternate protocols (e.g. BTP [11]) to support optimized data transmission between distributed software components.

Camera control via head movements is not always very intuitive for a remote user (e.g. when controlling more than

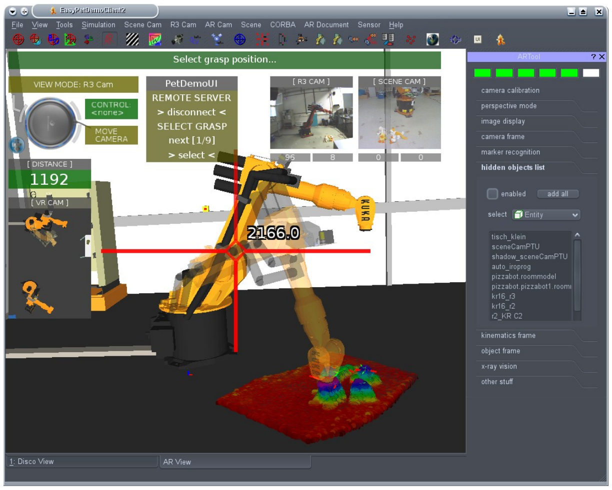

Fig. 9. Scene view of the virtual camera.

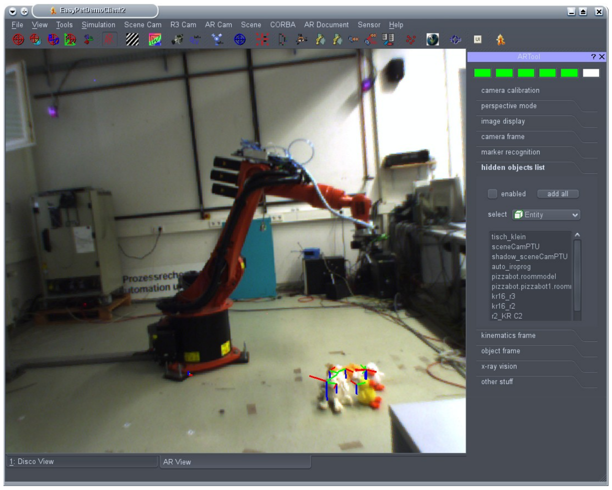

Fig. 10. Scene view of the real camera.

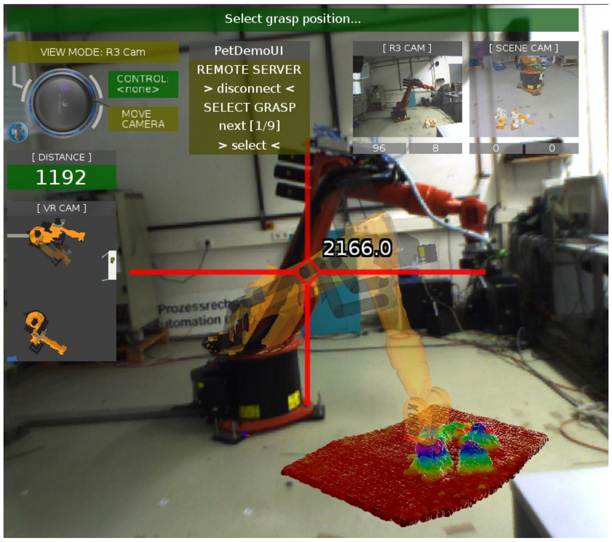

Fig. 11. User's view showing the original camera image augmented with all possible grasp points, the planned movement of the robot, the MiniUI to control the grasping task, and miscellaneous additional information to support the user.

two degrees of freedom, as already mentioned in section III-A). Future experiments will include wrist-worn light-weight devices including inertial sensors and some buttons.

In case of the scene view camera mounted on the second robot and moving on a spherical surface, absolute target position visualization is possible but due to the limited field of view not as flexible as in case of the pan/tilt camera where the pivot point for the camera movement coincides with the view position.

A possible solution to such problems could be to reproduce in advance the desired movements of the real camera on the virtual camera (in the same way the robot movements are predicted based on the input commands), but this would result in a lack of camera image information at the border of the user's view and perspective errors for neraby objects until the corresponding video stream data arrives.

Perspective errors in the predicted image could be reduced by projecting the camera image as a texture directly onto the 3D scene model instead of placing it in the background of the scene. Wide-angle-lenses might help to cover a larger field of view than actually needed for the interaction. The user would only be presented with the central part of the camera image, but the additional image information may be used to reconstruct the user's view from the camera's predicted target position. The challenge will be to ambiently blend the predicted image data with the updated actual image data from the camera with minimal negative perception effects for the user.

Transmitting the yet uncompressed point cloud retrieved from the TOF sensor usually takes up to 5 seconds when using a UMTS internet connection. Due to multi-threading and textual status information, the user's GUI stays fully responsive and the user is kept informed about the current steps taken by the system. Once the point cloud and possible grasp points are transmitted, switching, previewing, and selecting grasp positions by the user is not effected by the latency, because most of the calculations are done directly on the user's side of the connection.

Once the desired position has been validated by the user and reached by the robot, the grasp skill can be activated and performed autonoumsly. Figure 12 shows the results of an exemplary grasp process. The blue curve represents the contact force, the red one the slip rate and the green one the gripper aperture. After a first phase during which the part is stably grasped, an external disturbance is applied at a time t = 60 seconds by trying to remove the grasped object. The gripper reacts strongly with a fast closure. Once the perturbation disappears, the normal steady-state is reached again. During this last phase, as well as during the first one, a constant slow opening alternated with a fast closing can be clearly observed as soon as a slip is detected.

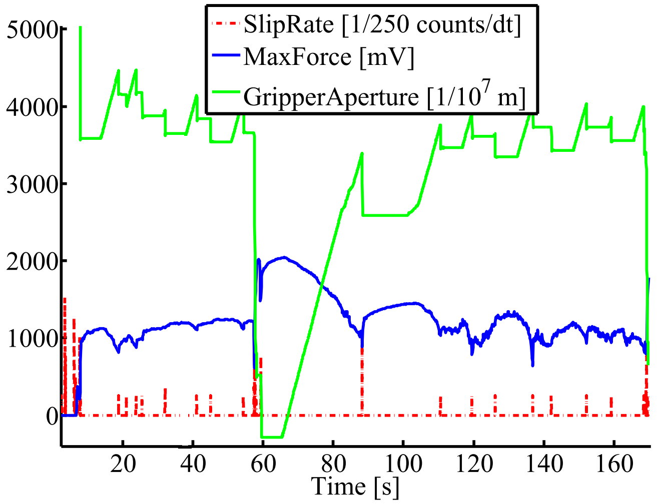

Fig. 12. Experimental results: steady-state and reaction to disturbances.

V. ACKNOWLEDGMENTS

The provided results are part of the cooperation between the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) and the Institute for Process Control and Robotics (IPR) within the framework of the research projects Sonderforschungsbereich (SFB) No. 588 Humanoid robots and ImRoNet. The SFB is supported by the Deutsche Forschungsgemeinschaft (DFG), ImRoNet is supported by the BMWi/DLR.

REFERENCES

[1] Giulio Milighetti and Helge-Björn Kuntze, "On the robot based surface finishing of moving unknown parts by means of a new slip and force control concept", *in IEEE International Conference on Robotics and Automation* (ICRA), Rome, Italy, 2007.

[2] Giulio Milighetti and Helge-Björn Kuntze, "Fuzzy based decision making for the discrete-continuous control of humanoid robots", *in IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), San Diego, CA, USA, 2007.

[3] Marc Bächlin, Jürgen Beyerer, Helge-Björn Kuntze and Giulio Milighetti, "Vorrichtung zum Schlupfüberwachten, kraftschlüssigen Ergreifen, Halten und Manipulieren eines Objektes mittels einer Greiferanordnung", P-2005-02 German Patent 10 2005 032 502.5, 2005.

[4] Weiss Robotics, Innovation in tactile sensing, `http://www.weiss-robotics.de`

[5] Agilent Technologies, Inc., Understanding Optical Mice, 2005.

[6] Benedikt Kaiser, "Bildverarbeitung für ein generisches Entladesystem", PhD Thesis, Karlsruhe Institute of Technology, 2009, `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000015610`

[7] Woo-Keun Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda and T. Doi, "Model-based space robot teleoperation of ETS-VII manipulator", *in IEEE Transactions on Robotics and Automation*, Vol. 20, pp. 602-612, 2004

[8] Probal Mitra, Diana Gentry and Gunter Niemeyer, "User Perception and Preference in Model Mediated Telemanipulation", *in Proceedings of the 2nd Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 268-273, 2007

[9] M. Bengel, K. Pfeiffer, B. Graf, A. Bubeck and A. Verl, "Mobile robots for offshore inspection and manipulation", *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3317-3322, 2009

[10] H. Ohroku and K. Nonami, "Omni-directional Vision and 3D Animation Based Teleoperation of Hydraulically Actuated Hexapod Robot COMET-IV", *in ICGST International Journal on Automatic Control and System Engineering ACSE*, pp. 17–24, 2009

[11] R. Wirz, R. Marin, M. Ferre, J. Barrio, J. M. Claver, J. Ortego, "Bidirectional Transport Protocol for Teleoperated Robots", *in IEEE Transactions on Industrial Electronics*, Vol. 56, no. 9, pp. 3772-3781, Sept 2009