The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# 1kHz Is Not Enough — How to Achieve Higher Update Rates with a Bilateral Teleoperation System Based on Commercial Hardware

Daniel Kubus, Ingo Weidauer, and Friedrich M. Wahl

***Abstract*— Teleoperation has a long history in the robotics community and numerous bilateral teleoperation systems employing manipulators have been proposed in the literature. On the one hand, systems have been designed which employ commercial hardware and hence generally suffer from low update rates and high delays due to restrictions of commercial manipulator controllers and haptic device controllers. On the other hand, bilateral teleoperation systems designed by research institutions often provide only few degrees of freedom. Our 6DoF bilateral teleoperation system, however, combines the amenities of commercial hardware with a high performance distributed control architecture which enables us to achieve update rates of more than $2kHz$ and delays in the range of only $100\mu s$. This paper focuses on the architecture of our system and demonstrates how to achieve this performance using commercial hardware. Moreover, we show why update rates of more than $1kHz$ are essential for certain teleoperation tasks. Especially with high approach velocities and stiff environments, high update rates and low delays are key requirements for stability and thus for realistic haptic perception. We present experimental results demonstrating the influence of the update rate on system stability. These results not only highlight the benefits of high update rates but also give hints on how to estimate the update rate necessary to achieve stable teleoperation for a given environment stiffness.**

## I. Introduction

Teleoperation is a comparatively old – yet fruitful – field of robotics. Numerous bilateral teleoperation systems and control approaches for bilateral teleoperation have been proposed during the last five decades. Recently, Hokayem and Spong published a comprehensive historical review of the field [1] mainly focusing on control approaches. Regarding bilateral teleoperation systems for undelayed teleoperation, on the one hand, systems have been designed using commercial hardware, which generally suffer from low update rates and high delays due to restrictions of commercial manipulator controllers and haptic device controllers. On the other hand, bilateral teleoperation systems designed by research institutions often provide a restricted number of degrees of freedom but feature high update rates and low delays of the feedback loop.

The following subsection addresses the problem of *the optimal update rate* in bilateral teleoperation and haptic rendering. Subsection I-B motivates why certain application areas require significantly higher update rates than those common today. In Subsection I-C the most important bottlenecks in teleoperator design are addressed and in Subsection I-D the structure of this paper is outlined.

The authors are with the Institut für Robotik und Prozessinformatik, Technische Universität Braunschweig, 38106 Braunschweig, Germany. {d.kubus,i.weidauer, f.wahl}@tu-bs.de

### A. The Optimal Update Rate

A key requirement of teleoperation systems is stability while at the same time providing maximum transparency [1]. These contradictory goals are hard to combine. Concerning realistic haptic perception, *perceptual stability*, which is closely related to control stability, proves to be another key requirement according to Choi et al. [2]. Perceptual instability refers to any unrealistic perceived sensations which cannot be attributed to the physical properties of an object in the environment or the end-effector.

Apart from control stability, human perceptual capabilities are to be considered in teleoperator design. In [3]–[5] O'Malley et al. relate hardware design parameters of haptic interfaces to human perception capabilities examining size discrimination tasks in virtual environments. Obviously, force control bandwidth of a haptic device must exceed human force control bandwidth, which is clearly below $30Hz$ [6]. However, humans are able to *perceive* vibrotactile stimuli of approx. $1kHz$ [6].

Generally, control stability and thus perceptual stability improve with increasing update rate. In contrast to *bilateral teleoperation*, the achievable update rate in *haptic rendering* primarily depends on the haptic device and the available computing power. On the one hand, a recent study by Booth et al. [7] shows that the minimum subjectively acceptable haptic rendering update rate is lower than the de facto standard rate of $1kHz$ – i.e., approx. $550Hz - 600Hz$. Moreover, they argue that the refresh rate does not depend on surface stiffness. However, merely comparatively low surface stiffness values ($0.2\frac{N}{mm}$ to $1.2\frac{N}{mm}$) have been examined. The experimental results contained in [2], [8] support this claim in the case of similar stiffness values but, on the other hand, they clearly show that higher update rates improve the perception of stiffer environments. In [9] Křenek claims that rendering rates of up to $5kHz$ improve perception quality when rendering stiff smoothly curved surfaces [10].The required update rate depends on several factors, i.e., mechanical and electrical properties of the haptic device, the employed control approach, the environment stiffness, etc. These interdependencies cause the rates mentioned above to differ by an order of magnitude. Thus, there is no *optimal update rate* for *haptic rendering* or *bilateral teleoperation* but only a minimum rate that is required regarding a specific system, control approach, and environment. Diolaiti et. al. relate the maximum displayable object stiffness to characteristics of the control loop, e.g. update rate and delay, and mechanical characteristics of the

device [11] – thus providing a theoretical understanding of the underlying interdependencies.

But which rate is actually required to guarantee stability for a given environment, hardware setup, and control approach? Our experimental results give hints on how to determine this rate and hence the necessary computing power.

### B. Teleoperation Scenarios Requiring High Update Rates

As already suggested, certain teleoperation scenarios require high update rates (or at least elaborate control approaches) to guarantee perceptual stability. Especially in highly-dynamic teleoperation tasks with stiff environments, high update rates and low update delays are essential for perceptual stability. In the realm of teleoperation, numerous scenarios exist in which both achievable update rates and delays are clearly limited due to the restricted capabilities of available communication channels (remote maintenance, remote surgery, etc.) or due to the physical limitation of signal propagation speed (deep space teleoperation). In several other application scenarios – robotic surgery, micromanipulation, handling of hazardous goods, etc. – master and slave may be coupled by high-speed communication systems allowing real-time capability and hence high update rates and low delays. Needless to say that – apart from the update rate – the employed control approach has a crucial effect on perceptual stability, control stability, and transparency.

The experimental system developed by us, is mainly intended for exploring human strategies in assembly operations, i.e., it is employed in partially highly-dynamic teleoperation tasks with possibly stiff environments.

### C. Bottlenecks

During the last decades numerous teleoperation systems have been developed. Many research institutions have developed experimental teleoperation that offer only few DoFs but provide high update rates or, on the contrary, use commercial hardware and very limited update rates. Few institutions were actually able to develop hardware solutions that enable 6DoF teleoperation systems with high update rates in the range of $1kHz$ – for instance the German Aerospace Center [12]. Nowadays, however, available computing power and bus data rates are no longer the bottleneck with regard to high update rates and low delays respectively. In various teleoperation applications, for example, microsurgery and handling of hazardous goods, the distance between master and slave is negligible and consequently – one may think – communication delays are short and update rates are high. Most teleoperation systems based on commercial hardware, however, do not provide low delays and high update rates. Except for a few companies, commercial robot manufacturers simply do not offer real-time capable high-rate low-level interfaces and most haptic device controllers are limited to update rates $\leq 2kHz$. Our system overcomes these restrictions as described in Section IV-B.

Instead of increasing the update rate or applying sophisticated control approaches to gain stability, mechanical and electrical characteristics of the system may be optimized as well. For instance, Kawai et. al. use an analog circuit which works as a spring-damper system for low-level control of their haptic device to reduce the influence of the sampled-data system – and thus improve stability [13].

### D. Structure

The remainder of this paper is structured as follows: Section II briefly reviews the control approach currently employed in our bilateral teleoperation system. In Section III, a problem that several haptic devices have in common is addressed: noise in velocity and acceleration signals derived from encoder measurements. Section IV introduces the architecture of our bilateral teleoperation system. Moreover, peculiarities of our driver for Phantom Premium devices and the QNX operating system are presented. Practical hints on how to implement such a teleoperation system using commercial hardware are also addressed. Experimental results demonstrating the advantages of high update rates are presented in Section V. Section VI gives an outlook and concludes the paper.

## II. Brief Review of the Control Approach

Our teleoperation system is currently based on a position-force architecture [14], i.e. the pose of a haptic device (master) with 6 *active* DoFs is used to control the pose of a 6DoF slave manipulator and the forces measured by a wrist-mounted 6D force-torque sensor are fed back to the haptic device.

This brief review is not intended to give a formal in-depth discussion of the employed control approach but a basic understanding of the structure of our system. Fig. 1 shows a structural overview of the system.

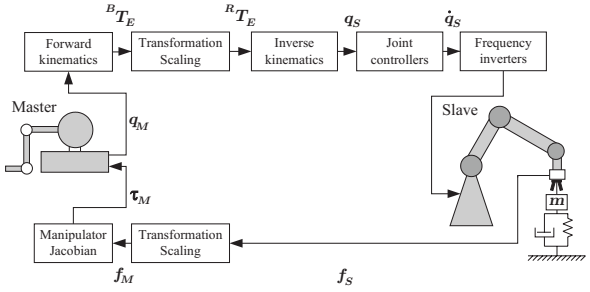


Fig. 1. Structural overview of the teleoperation system. The environment is illustrated by a mass-spring-damper system.

Starting at the master, the current joint angles of the master $\boldsymbol{q_M}$ are forwarded to the forward kinematics yielding the homogeneous transformation ${}^{B}\boldsymbol{T_E}$ relating the end-effector (stylus) to the base of the master. Before calculating the inverse kinematics of the slave manipulator, the translational and the rotational motion of the master may be scaled if desired. For orientation scaling quaternions are employed. The resulting transformation ${}^{R}\boldsymbol{T_E}$ relates the end-effector to the base of the slave. The joint angle setpoints $\boldsymbol{q_S}$ calculated by the inverse kinematics are then forwarded to the joint controllers. Joint velocities $\dot{\boldsymbol{q}}_{\boldsymbol{S}}$ calculated by these controllers

serve as input for the frequency inverters of the slave. To let the user experience the forces and torques that are exerted onto the robot end-effector, the forces and torques $\boldsymbol{f_S}$ that are measured by the wrist-mounted force-torque sensor have to be transformed to the end-effector frame of the master device and scaled, if desired. Using the manipulator Jacobian $\boldsymbol{J_M}$ of the master, the joint torque setpoints $\boldsymbol{\tau_M}$ for the master are calculated based on $\boldsymbol{f_M}$, i.e., $\boldsymbol{\tau_M} = \boldsymbol{J_M^T f_M}$.

## III. How to Reduce Noise in Velocity and Acceleration Signals

As Çavuşoğlu already mentioned in [15], the velocity output of the Phantom device is averaged which leads to a significant lag of approx. $50ms$. Unfortunately, no acceleration output is provided at all. Using more advanced filtering techniques like a linear Kalman filter [16] or a velocity estimator [17], the lag of the velocity signal can be drastically reduced while at the same time reducing the noise in the joint angle measurements $\bar{q}_{i,k}$ of the master and the slave and thus increasing system stability. Note, however, that the computational costs of these 12 linear Kalman filters (6 for the master and 6 for the slave) are *not negligible*.

To obtain joint angular velocities and joint angular accelerations the model given by Eqns. (1) and (2) is utilized.

$$\boldsymbol{x_{i,k}} = \begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x_{i,k-1}} + \begin{pmatrix} 0 \\ 0 \\ p_i \end{pmatrix} \tag{1}$$

$$\bar{q}_{i,k} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \boldsymbol{x_{i,k}} + m_i \tag{2}$$

The state vector

$$\boldsymbol{x_{i,k}} = (q_{i,k}, \dot{q}_{i,k}, \ddot{q}_{i,k})^T \tag{3}$$

consists of the joint angle $q_{i,k}$, the joint angular velocity $\dot{q}_{i,k}$, and the joint angular acceleration $\ddot{q}_{i,k}$ of joint $i$ at time-step $k$. $m_i$ and $p_i$ denote the measurement noise and the process noise respectively; $\Delta t$ designates the sampling interval. Based on these Kalman filtered signals, the angular velocity ${}^{\boldsymbol{E}}\boldsymbol{\omega}$, linear velocity ${}^{\boldsymbol{E}}\boldsymbol{v}$, angular acceleration ${}^{\boldsymbol{E}}\boldsymbol{\alpha}$, and linear acceleration vector ${}^{\boldsymbol{E}}\boldsymbol{a}$ of the end-effector frame $\boldsymbol{E}$ of both the master and the slave are calculated by propagating the contributions of each revolute joint through the kinematic chain from frame $j-1$ to frame $j$ according to Eqns. (4) - (7) until reaching the end-effector frame [18]. $\boldsymbol{z}$ stands for $\boldsymbol{z} = [0,0,1]^T$ and ${}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}$ designates the rotation matrix relating the orientation of frame $j-1$ to that of frame $j$.

$${}^{\boldsymbol{j}}\boldsymbol{\omega} = {}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}\ {}^{\boldsymbol{j-1}}\boldsymbol{\omega} + \boldsymbol{z}\dot{q}_j \tag{4}$$

$${}^{\boldsymbol{j}}\boldsymbol{v} = {}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}\ {}^{\boldsymbol{j-1}}\boldsymbol{v} + {}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}\ \left({}^{\boldsymbol{j-1}}\boldsymbol{\omega} \times {}^{\boldsymbol{j-1}}\boldsymbol{l_j}\right) \tag{5}$$

$${}^{\boldsymbol{j}}\boldsymbol{\alpha} = {}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}\ {}^{\boldsymbol{j-1}}\boldsymbol{\alpha} + \boldsymbol{z}\ddot{q}_j + {}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}\ {}^{\boldsymbol{j-1}}\boldsymbol{\omega} \times \dot{q}_j\boldsymbol{z} \tag{6}$$

$$\begin{aligned} {}^{\boldsymbol{j}}\boldsymbol{a} \quad = \quad & {}^{\boldsymbol{j}}\boldsymbol{R_{j-1}}\left({}^{\boldsymbol{j-1}}\boldsymbol{a} + {}^{\boldsymbol{j-1}}\boldsymbol{\alpha} \times {}^{\boldsymbol{j-1}}\boldsymbol{l_j}\right. \\ & \left. + {}^{\boldsymbol{j-1}}\boldsymbol{\omega} \times \left({}^{\boldsymbol{j-1}}\boldsymbol{\omega} \times {}^{\boldsymbol{j-1}}\boldsymbol{l_j}\right)\right) \end{aligned} \tag{7}$$

where

$${}^{\boldsymbol{j-1}}\boldsymbol{l_j} = \begin{pmatrix} a_j \cos(q_j) \\ a_j \sin(q_j) \\ d_j \end{pmatrix} \tag{8}$$

with Denavit-Hartenberg parameters $a_j$ and $d_j$ [18]. Our experimental results in Section V highlight the advantages of this filtering approach.

## IV. System Description

Our bilateral teleoperation system consists of a Sensable Phantom Premium 1.5 HighForce/6DoF haptic device (6 *active* DoFs), a 6DoF Stäubli industrial manipulator (RX 60 or RX90) with a modified CS7B controller, and a 6D JR3 force-torque sensor (100M40A3-I63 400N40) or 6D force-torque and 6D acceleration sensor (85M35A3-I40-D 200N12) respectively. While the joint position controllers of the slave run at a rate of $10kHz$ and force-torque values may be sampled at rates of up to $8kHz$, the original driver of the haptic device merely provides update rates of $500Hz$, $1kHz$, and $2kHz$. However, when employing our driver, a software timer can be used to generate other rates. Based on these considerations, our system may be classified as a multi-rate sampled-data control system, i.e., the digital controllers in the system use different sampling rates. Fig. 2 shows our experimental setup.

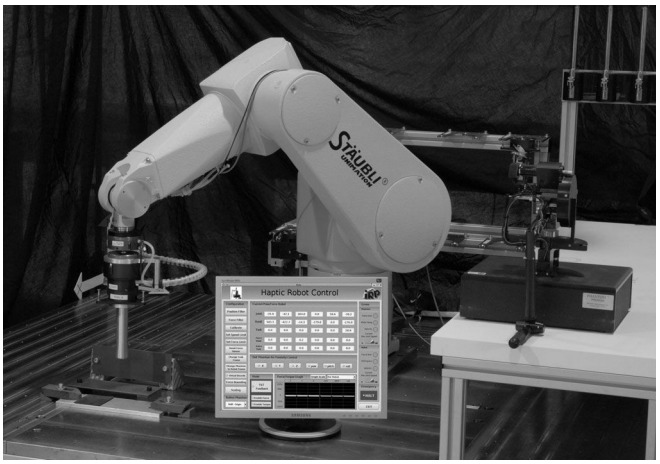


Fig. 2. Our bilateral teleoperation system consisting of a Stäubli RX90 manipulator, a Phantom 1.5 HighForce/6DoF haptic device, and a graphical user interface. Below the end-effector of the slave manipulator, objects with different stiffness values are located.

The following subsection addresses the development of a QNX Neutrino driver for Phantom Premium 1.5 haptic devices, which is the key to higher update rates. Subsection IV-B describes how standard manipulator hardware may be used to develop a high rate teleoperation system and also suggests design alternatives. The last subsection discusses remaining obstacles on the way to even higher update rates.

### A. High-Rate Phantom Driver

We use a Phantom Premium 1.5 HighForce/6DOF as master manipulator. This light-weight device, on the one hand, offers good backdriveability but, on the other hand, backlash and link elasticity deteriorate its performance. Since the Phantom family comprises very popular haptic devices, numerous works on parameter identification and performance characterization have been published. Çavuşoğlu et al. [15],

[19] studied several mechanical and electrical properties of the Phantom haptic device and could improve its performance significantly by replacing the motor electronics and adding measurement electronics. In [20] and [21] Tahmasebi et al. propose a method to estimate the parameters of a dynamic model of haptic devices and provide results for a Phantom Premium 1.5 device. Experiments revealing the position resolution and the structural response can be found in [22]. Apart from these hardware issues, the available device driver software has a significant drawback as well.

Unfortunately, the manufacturer does not offer drivers for commercial real-time operating systems. Our entire control system is based on the real-time operating system QNX Neutrino 6.3.2. Without a real-time OS running on the control computer, jitter-free update rates of more than $2kHz$ are definitely not achievable. Therefore, a QNX Phantom driver has been developed based on the communication of the Phantom with its host PC [23]. This driver also facilitates porting existing applications to QNX owing to its transparent interface. Furthermore, it overcomes several flaws of the original driver; especially the error handling has been improved.

The communication between the host PC and the Phantom device is based on callback functions. Both joint encoder data and joint torque setpoints are transmitted in subsequent data blocks after the control cycle has been triggered by an internal device timer. Our driver, however, enables operation at higher rates by replacing the device timer with a real-time OS timer service.

Essential components of such a driver are at least the forward kinematics and the manipulator Jacobian of the haptic device. The forward kinematics of the Phantom is addressed in [15], [19]. To calculate the motor torques based on Cartesian forces and torques, the transpose of the manipulator Jacobian [24] is employed. Since the encoders of our Phantom device merely offer resolutions of $1000$ increments per revolution (base motors) and $512$ increments per revolution (gimbal motors) and the gear ratios are relatively low, the joint velocity and hence the Cartesian end-effector velocity are estimated by a linear Kalman filter to reduce the effects of encoder noise – especially in the case of low velocities.

### *B. How To Achieve High Update Rates*

Basically, the maximum achievable update rate is determined by the weakest link in the chain, i.e., the slowest device in the control system. While current force-torque sensors can provide data at rates of up to $8kHz$ [25], common industrial manipulators are rather limited w.r.t. their maximum control rate. Current industrial manipulators of leading manufacturers (except for Stäubli and Comau) feature control interfaces with rates that are clearly less than $1kHz$ – if an interface that allows commanding desired Cartesian poses or joint angles is available at all. The TX generation of Stäubli manipulators features a so-called low-level interface [26] that enables control rates of up to approx. $1667Hz$. Apart from Stäubli, Comau offers a control interface [27] supporting low-level control at rates of $\geq 1kHz$. Nevertheless, higher control rates can hardly be reached using the control architectures provided by leading robot manufacturers. However, several current frequency inverters or motion controllers with EtherCAT or SERCOS III interfaces feature control rates of more than $1kHz$. Alternatively, a frequency inverter/motion controller with an analog interface may be employed. In general, update rates are limited by the control computer and the controller software, not by the performance of available frequency inverters/motion controllers or other components. Therefore, replacing the control computer (and – if necessary – unsuitable frequency inverters/motion controllers) is often a practicable approach to achieve higher control rates.

Most leading robot manufacturers offer control units that provide relatively poor computing power compared to standard PCs. Since higher control rates inevitably lead to higher computing demands, the original controller processing unit has to be replaced by a PC equipped with a motion control board that interfaces the encoders/resolvers and the frequency inverters. Our system uses a standard PC with a Vigilant motion control board instead of the CS7B control computer. Further details on our control architecture can be found in [28]–[30]. Fig. 3 presents a structural overview of our system hardware. As can be seen, a distributed computing architecture is employed.

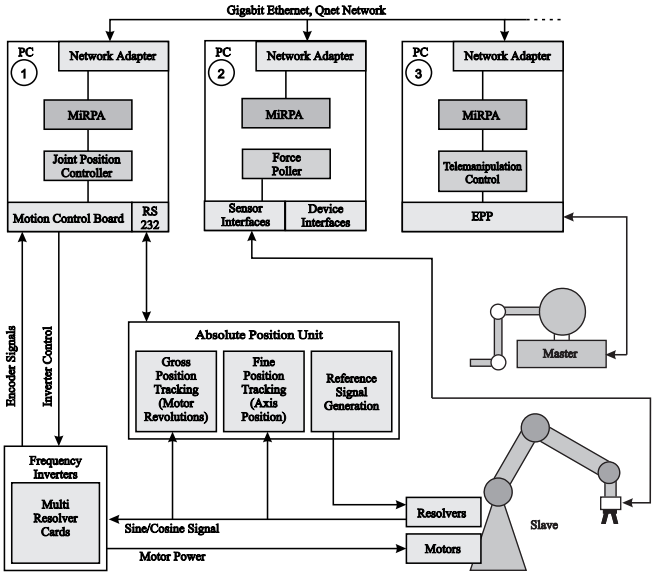


Fig. 3. Hardware overview of the bilateral teleoperation system. In the top part, the control computer system is depicted. It consists of three PCs but may be scaled up or down. The PCs are connected by a Gigabit Ethernet network using the real-time capable Qnet protocol. PC 1 performs low-level control of the slave manipulator. PC 2 solely acquires data from the JR3 force-torque sensor. PC 3 runs the teleoperation control application and interfaces the haptic device. In the bottom part of the figure, the frequency inverters, the absolute positioning unit responsible for referencing the slave, the master, and the slave are depicted.

As the separation of the slave control computer and the master control computer is generally desirable to increase computing resources, it also entails the problem of distributing the control data. To enable real-time capable communica-

tion between the control computers, a real-time middleware called MiRPA (Middleware for Robotic and Process Control Applications) [31] is employed that handles all network communication requests. As the name indicates, it has been especially designed for meeting robotics and process control software system demands. MiRPA allows the implementation of sophisticated distributed real-time software architectures. It handles message-based publisher/subscriber as well as client/server communication between local and distributed software modules. The average communication latency is less than $100\mu s$ for a network server request and less than $10\mu s$ for a local request (message payload: 100 bytes, Intel Desktop PRO/1000 family network adapter, Intel Core 2 Quad Q9300 "Yorkfield" processor). A detailed architecture description of our distributed real-time middleware including a thorough performance evaluation can be found in [31]. Although the communication latency of MiRPA for local requests seems to be negligible, it would sum up to a considerable time if the entire communication between the threads of the teleoperator control was performed using MiRPA. Therefore, local inter-thread communication is performed using efficient POSIX synchronization primitives and QNX message passing.

### C. Limitations

As already mentioned, an external real-time OS timer may be used to trigger an update cycle of the Phantom device at arbitrary rates – hence rates higher than $2kHz$ can be achieved. An ultimate limiting factor w.r.t. achievable update rate is due to the distributed architecture of our control system: Employing MiRPA entails an average latency of approx. $100\mu s$ per network request. Currently, the computing power of the teleoperation control computer (PC 3) prevents us from operating the system reliably at $4kHz$. Using the Intel C++ Compiler for QNX Neutrino instead of qcc might yield the required performance gain.

Apart from the update rate, hidden delays in the feedback loop and underlying control loops might contribute to instability. An important cause of delay are group delays of (implicit) IIR and FIR filters, e.g., the velocity averaging filter employed in the standard Phantom driver [15], or (cascaded) digital PID controllers used in the slave joint controllers. Our approach mitigates these effects by using a Kalman filter for velocity estimation and operating the joint controllers at a significantly higher rate than the feedback loop, viz., $10kHz$.

## V. Experimental Results

This section presents results of several experiments which have been conducted with our teleoperation system. In Subsection V-A the behavior of the system during impacts onto stiff surfaces is addressed. These results are especially interesting for shared control approaches [32]. Subsection V-B presents results on the relation between perceptual stability and update rate. Furthermore, a simple function describing this relation is derived.

### A. System Behavior during Impacts onto Stiff Surfaces

Often, a limitation of forces and torques in teleoperation is necessary to avoid damaging the environment or the manipulator, the slave, and/or the force-torque sensor. The probably simplest way is to stop the robot if a certain force-torque threshold is exceeded. With stiff environments or in high-velocity applications, however, this threshold will be clearly exceeded due to the kinetic energy of the manipulator and controller imperfections. If low update rates are used or delays are present in the feedback loop, the exerted forces and torques will increase further. Therefore, minimizing delays and increasing the update rate are suitable means – apart from predicting forces and considering manipulator dynamics – to reduce force overshooting in these situations.

Although elaborate methods incorporating manipulator dynamics exist, we use the following trivial approach in our experiments. If a force threshold $f_{th} = 50N$ is exceeded when impacting onto a surface, the slave pose setpoint is set to the currently measured pose. No force prediction based on an estimate of the environment stiffness and the motion state of the robot is employed. Hence, the desired force threshold is always exceeded since the slave robot has to decelerate, stop, and then reverse its direction of motion to reach the mentioned setpoint.

Fig. 4 shows an end-effector consisting of an aluminum rod with a ball tip as well as test surfaces, i.e., a sheet spring with a stiffness ranging from $18\frac{N}{mm}$ in its middle part to approx. $250\frac{N}{mm}$ near its supports and a separate aluminum surface. The depicted tool and the test surfaces have been used for the experiments presented in the following.

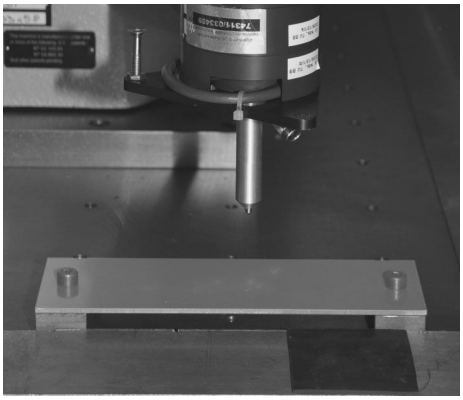

Fig. 4. Experimental test environment. An aluminum rod with a ball tip is mounted to the slave manipulator. This tool is used to impact onto the sheet spring below as well as the aluminum surface and its rubber-coated part in the foreground.

Fig. 5 presents an experimental result showing an impact onto a very stiff surface. The stiffness of the contact is mainly determined by the stiffness of the manipulator and the collision protection module since the end-effector and the environment consist of massive aluminum parts (cf. Fig. 4). This experiment demonstrates that

- the Kalman filter causes only a negligible lag of the velocity signal.

- the system reacts with a very low delay.

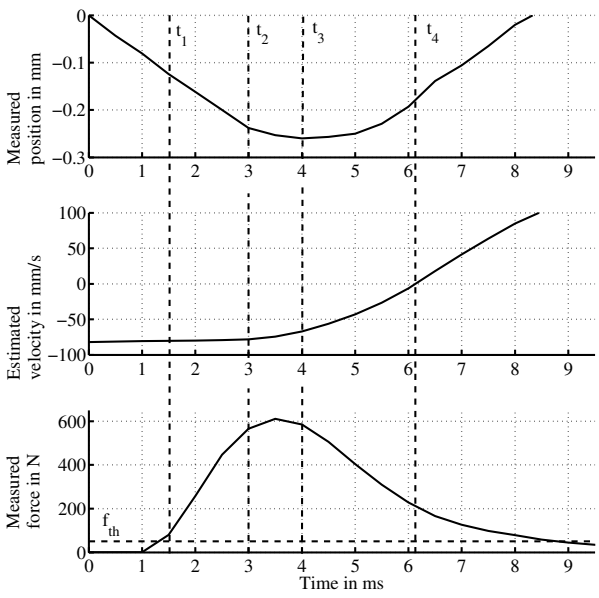


Fig. 5. End-effector position, velocity, and resulting force during an impact onto a very stiff ($> 3000\frac{N}{mm}$) surface with a velocity of approx. $8\frac{cm}{s}$ and an update rate of $2kHz$. At $t_1$, the force limiter recognizes that the force threshold $f_{th} = 50N$ is exceeded and hence sets the next pose setpoint to the currently measured pose. Only $1.5ms$ later at $t_2$, the velocity estimate of the Kalman filter shows a decrease of the end-effector velocity. At $t_3$, the direction of motion of the end-effector reverses, and with a delay of approx. $2.1ms$ (at $t_4$), the velocity estimate of the Kalman filter crosses the zero line as well. Depending on the permissible noise level, this delay may be reduced further.

In the second experiment, the relation between maximum forces during impacts and the update rate has been examined. Again, the robot is stopped as described above when a force threshold $f_{th}$ is exceeded. The surface has been approached with an approximately constant velocity of $10\frac{cm}{s}$. Fig. 6 shows the results of this experiment. In this figure and the following ones, the mean is marked by a point and the standard deviation is indicated by a vertical line. As can be observed, the maximum exerted force and its standard deviation clearly reduce with increasing update rate.

### *B. Relation between Perceptual Stability and Update Rate*

To underline that control stability is not a suitable criterion here, Fig. 7 shows an impact onto a stiff surface that caused perceptual instability, i.e., buzzing. Realistic haptic perception of stiff surfaces necessarily requires perceptual stability [8]. To obtain a relation between update rate and perceptual stability, a stiffness threshold $k_{th}$ has been determined for the following update rates: $125Hz$, $250Hz$, $500Hz$, $1000Hz$, and $2000Hz$. This stiffness threshold $k_{th}$ denotes the highest surface stiffness, which was evaluated as *perceptually stable* according to [8]. Note that the force transmitted to the haptic device has been scaled down by $\sigma_f = \frac{f_S}{f_M} = 25.0$ to keep $f_M$ within the maximum ratings of the haptic device.

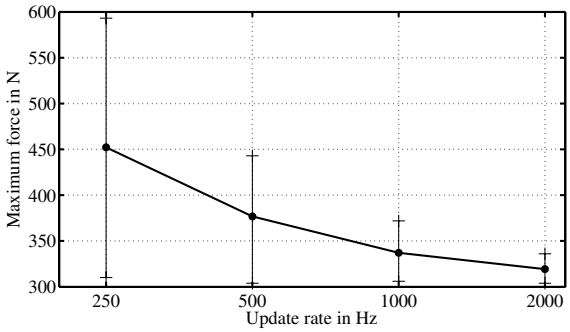


Fig. 6. Mean and standard deviation of the maximum exerted force (cf. Fig. 5) when impacting onto a stiff (approx. $700\frac{N}{mm}$) surface with a velocity of $10\frac{cm}{s}$. Note the significant decrease of the standard deviation of the maximum exerted force with increasing update rate. No experiments for an update rate of $125Hz$ have been performed since the force-torque sensor would have been overloaded.

Two right-handed male subjects (familiar with the Phantom device and with no known sensory or motor abnormalities of the upper extremities) performed the following experiment: The subjects were asked to impact onto the sheet spring (cf. Fig. 4) with a constant velocity (approx. $10\frac{cm}{s}$) and to determine the location where the contact became perceptually instable. This process was performed repetitively ($N = 10$ trials), advancing from the supports to the middle of the spring, and vice versa during each trial. This experiment was repeated for each of the update rates mentioned above with and without passivity control according to [33] on the master side.

The following two figures present the stiffness thresholds obtained by one subject as the differences between the subjects are rather slight. Fig. 8 shows the stiffness thresholds for the update rates mentioned above without passivity control. As can be observed, the threshold increases *linearly on the logarithmic scale*. In Fig. 9 the corresponding results with passivity control are presented.

Again, the threshold increases roughly linearly, but both the slope $k_s$ and the initial stiffness threshold $k_0$ at an update rate of $f_0 = 125Hz$ are clearly higher than without passivity control. For the sake of completeness, Fig.10 shows the difference of the mean of $k_{th}$ between the two subjects.

As can be observed, the means deviate slightly but no clear trend w.r.t. the update rate or the control approach can be observed; the same applies to the standard deviation. To generalize and validate the above mentioned results, the described experiment has to be repeated with a higher number of subjects.

The stiffness thresholds in Figs. 8 and 9 suggest a logarithmic function to approximate the mapping between update rate $f$ and stiffness threshold $k_{th}$, i.e.,

$$k_{th} = k_s \ \log_2\left(\frac{f}{f_0}\right) + k_0 \tag{9}$$

where $k_s$ denotes the slope and $k_0$ the initial stiffness threshold at $f_0$ ($125Hz$ in our experiments). When using a least-squares fit, the parameter values presented in Table I

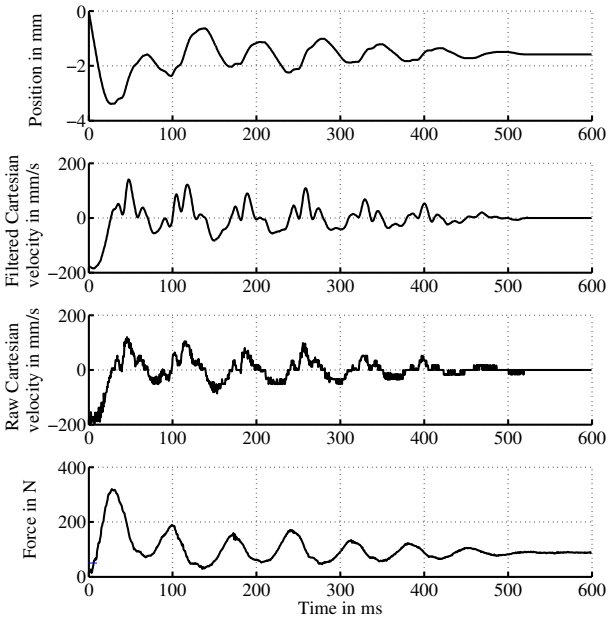


Fig. 7. Rather fast (approx. $19\frac{cm}{s}$) impact onto a stiff (approx. $100\frac{N}{mm}$) sheet spring using passivity control on the master side and an update rate of $1kHz$. After the impact, the measured forces and both the raw and the Kalman filtered Cartesian velocities show damped oscillations which cease after approx. $500ms$. Although the system is stable from a control engineering point of view, the contact is not *perceptually stable*, i.e., buzzing (high-frequency vibrations) is perceived. Hence, control stability is not an adequate criterion for performance evaluation here.

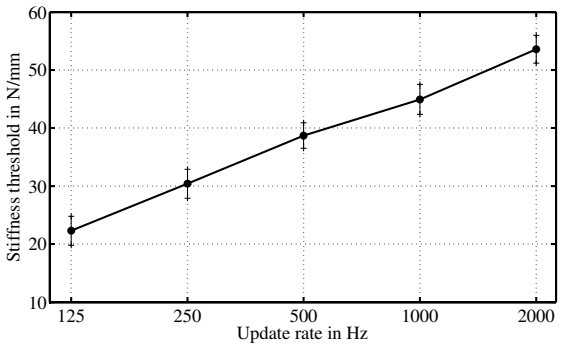


Fig. 8. Stiffness thresholds for different update rates *without* passivity control. As can be observed, the stiffness threshold increases linearly w.r.t. to the logarithmic rate scale.

are obtained for the subjects. As already indicated by Fig.10, the differences between the subjects are rather slight.

The function represented by Eq. (9) also agrees with the experimental results obtained by Choi et al. [2], [8] in a haptic rendering context.

To estimate the update rate which is necessary to achieve perceptually stable teleoperation in an environment with a known maximum stiffness, first $k_s$ and $k_0$ may be determined using rather low update rates. Then Eq. (9) may be employed to obtain an estimate of the necessary update rate.

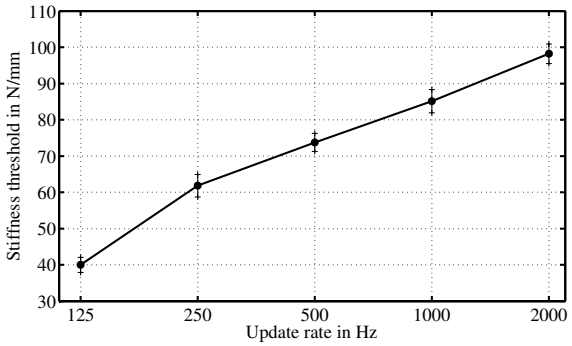


Fig. 9. Stiffness thresholds for different update rates *with* passivity control. Similar to Fig. 8, the stiffness threshold increases roughly linearly w.r.t. to the logarithmic rate scale.

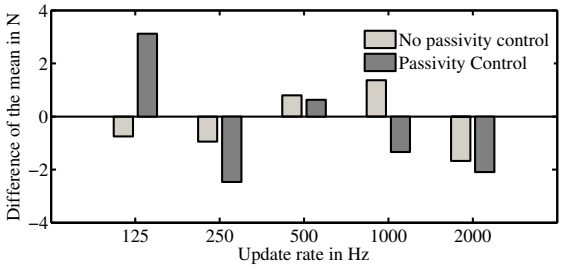


Fig. 10. Differences of the mean of the stiffness thresholds $k_{th}$ between the two subjects. The differences are slight and no clear trend w.r.t. the update rate or the control approach can be observed.

## VI. Conclusion and Outlook

This paper presents a bilateral teleoperation system architecture comprised of commercial hardware, that is, a 6 DoF Stäubli RX60 or RX90 manipulator, a 6DoF Phantom Premium 1.5 haptic device, and a JR3 force-torque sensor. In contrast to previously proposed solutions, our system features update rates of more than $2kHz$ and very low communication latencies of less than $100\mu s$. Thus, it provides a high performance 6DoF bilateral teleoperation solution. Experimental results presented in the previous section demonstrate that

- update rates significantly higher than $1kHz$ may be necessary if stiff environments and high approach velocities are targeted.
- the update rate and the maximum allowable environment stiffness are related by a logarithmic function whose parameters may be derived from the system

TABLE I

Values of $k_s$ and $k_0$ determined by a least-squares fit.

| Parameters | $k_s$ in $N/mm$ | $k_0$ in $N/mm$ | $f_0$ in $Hz$ |
|---|---|---|---|
| Subject I | | | |
| No passivity control | 7.71 | 22.57 | 125 |
| Passivity control | 13.98 | 43.82 | 125 |
| Subject II | | | |
| No passivity control | 7.76 | 22.23 | 125 |
| Passivity control | 13.09 | 45.25 | 125 |

behavior (at lower update rates).

Thus, a rough estimate of the update rate necessary to guarantee perceptual stability w.r.t. a given environment may be derived from the system behavior at lower rates; hence, the necessary computing power and resulting upgrade costs may be estimated. Both control and perceptual stability can clearly be improved by increasing the update rate and avoiding delays in the feedback loop. While humans are capable of perceiving vibrotactile stimuli of up to $1kHz$ thus necessitating a minimum sampling rate of $2kHz$, teleoperation systems may require considerably higher update rates (depending on the control approach, the targeted environment, etc.) to guarantee perceptual stability.

Further research will show whether significant performance gains with regard to perceptual and control stability can be achieved with update rates that are significantly higher than $2kHz$. Additionally, more advanced control algorithms will be implemented and evaluated extensively w.r.t. the relation between perceptual stability and update rate. This process will include experiments with a higher number of subjects to underpin our findings.

## ACKNOWLEDGMENTS

We would like to thank *QNX Software Systems* for providing free software licenses. Moreover, we are indebted to Torsten Kröger, who designed our robot control architecture, without which this system would not exist.

## REFERENCES

[1] P. F. Hokayem and M. W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, December 2006.

[2] S. Choi and H. Z. Tan. Discrimination of virtual haptic textures rendered with different update rates. In *WHC '05: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 114–119, 2005.

[3] M. O'Malley and M. Goldfarb. The effect of force saturation on the haptic perception of detail. *IEEE/ASME Transactions on Mechatronics*, 7(2):280–288, 2002.

[4] M. O'Malley and M. Goldfarb. The implications of surface stiffness for size identification and perceived surface hardness in haptic interfaces. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1255–1260, 2002.

[5] G. Upperman, A. Suzuki, and M. O'Malley. Comparison of human haptic size discrimination performance in simulated environments with varying levels of force and stiffness. In *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS04)*, pages 169–175, 2004.

[6] H. Z. Tan, M. A. Srinivasan, B. Eberman, and B. Cheng. Human factors for the design of force-reflecting haptic interfaces. In *Proc. of the 3rd Ann. Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 353–359, 1994.

[7] S. Booth, F. De Angelis, and T. Schmidt-Tjarksen. The influence of changing haptic refresh-rate on subjective user experiences - lessons for effective touch-based applications. In *Proceedings of the EuroHaptics 2003 Conference*, pages 374–383, 2003.

[8] S. Choi and H. Z. Tan. Effect of update rate on perceived instability of virtual haptic texture. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3577–3582, 2004.

[9] A. Křenek. Haptic rendering of complex force fields. In *EGVE '03: Proceedings of the workshop on Virtual environments 2003*, pages 231–239, 2003.

[10] Z. Kabeláč. Rendering stiff walls with phantom. In *Proc. of the 2nd PHANToM Users Research Symposium*, 2000.

[11] N. Diolaiti, G. Niemeyer, F. Barbagli, and J.K. Salisbury. Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects. *TransR*, 22(2):256–268, 2006.

[12] D. Reintsema, C. Preusche, T. Ortmaier, and G. Hirzinger. Toward high-fidelity telepresence in space and surgery robotics. *PRESENCE - Teleoperators and Virtual Environments*, 13(1):77 – 98, 2004.

[13] M. Kawai and T. Yoshikawa. Haptic display with an interface device capable of continuous-time impedance display within a sampling period. *IEEE/ASME Trans. on Mechatronics*, 9(1):58–64, 2004.

[14] Günter Niemeyer, Carsten Preusche, and Gerd Hirzinger. *Handbook of Robotics*, chapter 31, pages 741–755. Springer, 2008.

[15] M. Çavuşoğlu and D. Feygin. Kinematics and dynamics of phantom(TM) model 1.5 haptic interface. Technical Report UCB/ERL M01/15, University of California at Berkeley, Electronics Research Laboratory, 2001.

[16] P. R. Belanger. Estimation of angular velocity and acceleration from shaft encoder measurements. In *Proc. of IEEE International Conference on Robotics and Automation*, 1992.

[17] S. H. Lee, T. A. Lasky, and S. A. Velinsky. Improved velocity estimation for low-speed and transient regimes using low-resolution encoders. *IEEE/ASME Transaction on Mechatronics*, 9(3):553–560, September 2004.

[18] P. J. McKerrow. *Introduction to Robotics*. Addison-Wesley Publishers Ltd., 1991.

[19] M. Çavuşoğlu, D. Feygin, and F. Tendick. A critical study of the mechanical and electrical properties of the phantom(TM) haptic interface and improvements for high performance control. *Presence*, 11:555–568, 2002.

[20] A. M. Tahmasebi, B. Taati, F. Mobasser, and K. Hashtrudi-Zaad. Dynamic parameter identification and analysis of a phantom(TM) haptic device. In *IEEE Conference on Control Applications*, pages 1251–1256, 2005.

[21] B. Taati, A. M. Tahmasebi, and K. Hashtrudi-Zaad. Experimental identification and analysis of the dynamics of a phantom premium 1.5A haptic device. *Presence*, 17(4):327–343, August 2008.

[22] G. Campion and V. Hayward. Fundamental limits in the rendering of virtual haptic textures. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 263–270, 2005.

[23] I. Weidauer. Implementation of a QNX driver for phantom premium haptic devices. Project thesis. Technische Universität Braunschweig, Institut für Robotik und Prozessinformatik ⟨http://www.rob.cs.tu-bs.de⟩. Internet, 2008.

[24] R. P. Paul. *Robot Manipulators: Mathematics, Programming and Control*. The MIT Press, 1981.

[25] JR3, Inc., 22 Harter Ave., Woodland, CA 95776. *JR3 DSP-Based Force Sensor Receivers Software and Installation Manual*.

[26] F. Pertin and J. Bonnet des Tuves. Real time robot controller abstraction layer. In *Proceedings of the International Symposium on Robotics 2004*, 2004.

[27] COMAU S.p.A. Robotics, Via Rivalta, 30, 10095, Grugliasco (Turin), Italy. *C4G OPEN Instruction Handbook, System Software Rel. 3.1x*, 2008.

[28] B. Finkemeyer, T. Kröger, and F. M. Wahl. Executing assembly tasks specified by manipulation primitive nets. *Advanced Robotics*, 19(5):591–611, June 2005.

[29] J. Maass. Hard- und Software Schnittstellen einer Robotersteuerung. Project thesis. Technische Universität Braunschweig, Institut für Robotik und Prozessinformatik, 2004.

[30] J. Maass. Entwurf, Realisierung und Verifikation einer offenen, PC-basierten Robotersteuerung. Master's thesis, Technische Universität Braunschweig, Institut für Robotik und Prozessinformatik, 2004.

[31] B. Finkemeyer, T. Kröger, D. Kubus, M. Olschewski, and F. M. Wahl. Mirpa: Middleware for robotic and process control applications. In *International Conference on Intelligent Robots and Systems 2007, Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware*, 2007.

[32] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press, 1992.

[33] C. Preusche, G. Hirzinger, J. H. Ryu, and B. Hannaford. Time domain passivity control for 6 degrees of freedom haptic displays. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2944–2949, 2003.