Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Design of a Control Strategy for Teleoperation of a Platform with Significant Dynamics

Mattias Bratt, Christian Smith, and Henrik I. Christensen
Centre for Autonomous Systems
KTH
Stockholm, Sweden
Email: {bratt,ccs,hic}@kth.se

***Abstract* - A teleoperation system for controlling a robot with fast dynamics over the Internet has been constructed. It employs a predictive control structure with an accurate dynamic model of the robot to overcome problems caused by varying delays. The operator interface uses a stereo virtual reality display of the robot cell, and a haptic device for force feed-back including virtual obstacle avoidance forces.**

***Index Terms* – Internet, prediction, remote bilateral tele-operation, time delay, haptics.**

I. Introduction

Teleoperation is widely used in control of systems in many different types of applications, such as robot assisted surgery, space missions, handling of dangerous materials, etc. Most of these applications are characterized by limited dynamics. An obvious question is thus "what are the requirements for design of systems in the presence of significant dynamics?" A typical example of a task that is inherently dynamic is the catching of a flying object, e.g. a ball.

As part of the Neurobotics project one of the topical studies is related to teleoperation for such scenarios. Autonomous catching of such objects ([1] and [2]) or hitting them [3] has been reported in the literature. For teleoperation the control loop is closed through the operator. Here one of the key questions is the use of models. Humans have a long experience of catching flying objects and the performance depends upon the information presented to the operator and how this information matches to our models. Models for ball catching by humans have been studied extensively as for example reported by McIntyre [4].

To enable studies of use of such models for control, a high performance manipulation system (Fig. 1) has been designed for teleoperated ball catching. It was built from off-the-shelf components, and is further described in section III.

In this paper we describe the design of the teleoperation control strategy for the control of the system. To demonstrate the generality of the presented approach, a related application of mobile platform teleoperation is also presented. The differences between the two designs are described as well.

The research presented in this paper is funded in full by the 6th EU Framework Program, FP6-IST-001917, project name Neurobotics.

Fig. 1 The manipulator used for teleoperation studies

Initially the control strategy is discussed in Section II. The implementation of the chosen strategy is then presented in Section III - detailing both the catching and mobility systems. Early experimental results are presented in Section IV. A number of issues and an associated strategy for future studies are presented in Section V. Finally the study is summarized in Section VI.

II. Design of a Control Strategy

As mentioned earlier, teleoperation is widely used in a large number of applications today. Direct teleoperation has a number of different challenges depending on the problem at hand. Some of the challenges include:

- Kinematic transfer
- Handling of time delays
- Feedback generation

Kinematic transfer refers to the fact that in some cases there is a difference in kinematic structure between the actuation system (the manipulator) and the control system available to the operator. An example is the transfer from a force joystick to a manipulator. Another is the traditional command console used for operating cranes on trucks.

Handling of time-delay is another challenge to ensure stability and maximum performance. It can be divided into handling of deterministic time-delays and handling of

stochastic time-delays. In control theory the handling of deterministic delays is often modeled using a Smith predictor structure, whereas there are a number of different strategies for handling of stochastic delays such as [5].

Finally the issue of feedback generation is important as multiple modalities often can complement each other and provide efficient embedding of the operator in the scenario.

For the ball catching system and mobile platform we have chosen to use stereoscopic modeling and a force-reflectance joystick to provide a maximum of flexibility in the feedback generation and best performance in the type of experiments to be performed. The study is performed in the context of control of a system over the internet, so the system is assumed to have stochastic time characteristics.

A. Design of an overall control structure

A common technique for closing a teleoperation control loop over a communication medium with significant delays is the use of wave variables [6]. Whereas this can guarantee passivity (no net output of energy) of the communication link, and thereby help prove stability of the complete system, it does impose severe performance penalties in a scenario where fast dynamics are required despite the presence of substantial communication delays. For this reason a different approach, requiring an accurate dynamic model of the robot and its environment, was selected. In the employed scheme, all forces displayed to the operator through the haptic device are generated interacting exclusively with a dynamic simulation running on the user interface computer. Since this interaction does not involve a time delay, the risk of instabilities is much reduced. The modeling of the robot is facilitated by the fact that the model includes the controller local to the robot. Knowing the input command and the dynamic state of the robot, the future state can be predicted well after a time of the same magnitude as the communication delays.

A1. Basic structure of the teleoperation control system

To close the control loop over the robot via the communication link, we use a non-linear, multvariate Smith predictor control structure (depicted for the robot arm teleoperation case in Fig. 2). Because the system simulated (the robot and its controller at the robot site) in the predictor is highly non-linear, it is not sufficient to correct only the predictor output by the measured state y (the joint angles and speeds) arriving from the robot, as in a standard Smith predictor [7]. The simulation itself also needs access to the corrected state estimate to be able to use the correct dynamics for that particular point in state space. The simulation output however, is not corrected, and can still be compared to incoming measurements for generating new corrections.

The command and measurement communication delays, τ_c and τ_m respectively, are handled by

i) Adding an artificial delay τ_a to the stochastic command delay τ_c when a command packet arrives at the robot so that their sum, the virtual delay τ_v, is constant [8].

ii) Delaying the simulation result $y_{sim}(t+\tau_v)$ by $\tau_v+\tau_m$ before it is compared to the measured state $y(t-\tau_m)$ to form the simulation correction δ. This means that when a measurement packet arrives from the robot, the current value of τ_m is calculated from timestamps and an old simulation result $y_{sim}(t-\tau_m)$ retrieved from memory for comparison.

The net effect of this is that the simulation, the haptic controller, and the operator all perceive time τ_v ahead of real time. That allows the generated command signal u to travel to the robot before it is needed by the robot controller. τ_v values of up to 200 ms have been tested successfully.

Because the correction loop closed over the communication link runs at a much lower frequency (~20 Hz)

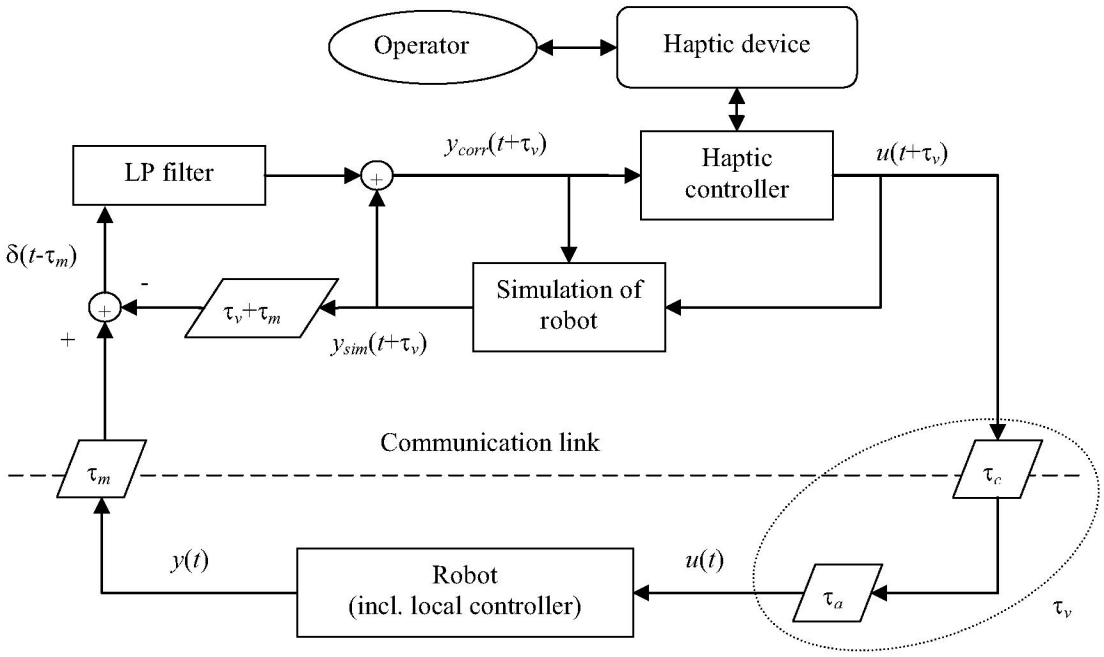

Fig. 2 Robot arm control structure (skewed boxes represent delays)

than the inner loop containing the simulation and control of the haptic device (~500 Hz), the correction signal δ must be low pass filtered to reduce zero order hold noise that would otherwise be felt by the operator as vibrations. The low pass filter also reduces the high frequency gain of the correction loop which moderates the spike effect of quick state changes in the simulation and at the robot being slightly offset in time.

A2. Systems with multiple feedback channels

For the mobile Pioneer II platform, feedback of measurement data is slightly complicated by its being divided into two parts: relative odometry position measurements, and absolute, but less frequent, laser scanner localization data. The odometry data is calculated by the motion controller board of the Pioneer, and sent to the operator interface computer every 50 ms. Laser localization is done less frequently based on data from a SICK scanner on the robot. The modified control structure appears in Fig. 3. The differences from Fig. 2 are that there are now two measurement delays, τ_{mo} and τ_{ml}, for odometry and laser data respectively ($\tau_{ml}>\tau_{mo}$ because of the time required for computing the localization), and, correspondingly, two corrections, δ_o and δ_l, which are both added to the simulation result y_{sim} to form the corrected estimate y_{corr} of the robot state. δ_l is formed as a correction of the odometry data y_o, and is filtered through a slower low pass filter than δ_o because of its lower update frequency. Furthermore, though not visible in the diagram, the corrections can no longer be applied by plain vector addition because the state y now contains a rotation (apart from the two planar Cartesian coordinates, their first derivatives and the rotational velocity).

B. Feedback generation

The basic principle of the haptic controller of Fig. 2 and Fig. 3 is to map the haptic device handle to the chosen command space (velocity or, in the case of teleoperating the robot arm, alternatively position) and virtually attach it by a spring to the current corrected simulation value of the corresponding part of the robot state. I.e., when teleoperating the robot arm by velocity control, the current three-dimensional velocity of the end-effector is mapped to a point in the haptic device workspace to which the handle is attracted by a virtual spring. The command sent to the robot is just the position of the handle translated to a velocity by the inverse of the same mapping. In this way the force perceived by the operator is related to the resistance of the robot and its controller to adjust to the command.

In addition, virtual viscous damping is applied to the handle to help stabilize the system, and in the case of velocity control, a weak force toward zero velocity is added as well. To reflect the quality of the communication link, both of these forces can be strengthened in response to any sustained increase in the measurement communication delays.

B1. Obstacle avoidance forces

To avoid known obstacles, these are mapped into the used command space, and made to repel the handle with a force proportional to the inverse of the square or cube of the distance to the handle depending on the dimensionality of the space. For position control this mapping is trivial, but mapping obstacles to velocity space requires some thought. The chosen mapping is based on the principle that a point in velocity space is part of an obstacle if, starting at this velocity v, and at the current position of the robot, braking with a postulated constant negative acceleration a_{brake} to zero velocity results in a collision. The distance traveled during such a hypothetical braking maneuver from speed v is

$$s = \frac{v^2}{2a_{brake}}. \tag{1}$$

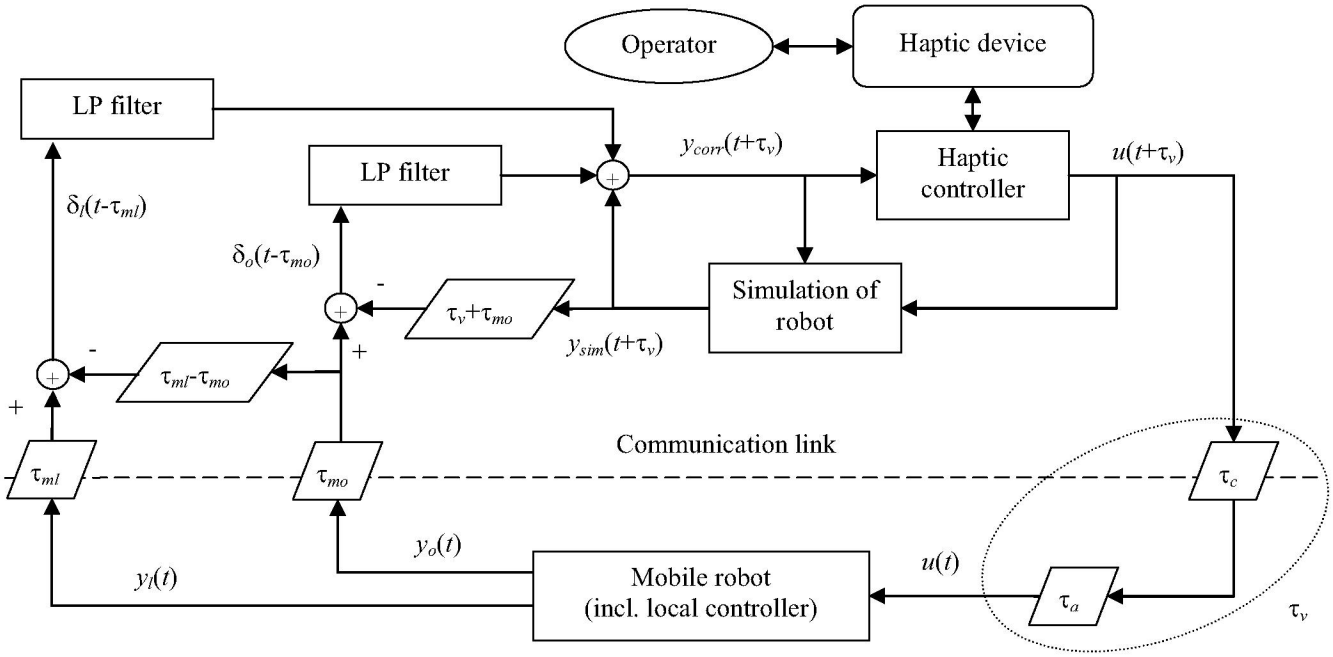

Fig. 3 Mobile robot control structure (skewed boxes represent delays)

III. Implementation of the Control Model

As mentioned earlier, teleoperation is studied in the context of the manipulation system shown in Fig. 1. It uses PowerCube actuator modules from Amtec, and is controlled at a frequency of 700 Hz by a standard PC using a CAN bus. Its kinematic configuration is of Puma560 type, allowing the use of analytical solutions for inverse kinematics and dynamics. Maximum end effector velocity and acceleration are 7 m/s and 140 m/s^2, respectively. The local controller PC communicates with the user interface computer by UDP/IP.

In addition, mobile platform experiments are performed with an ActivMedia Pioneer II robot with sonar and laser sensors. It has an onboard single board computer, which handles UDP/IP wireless communication with the user interface computer, and uses ActivMedia's Aria API for interfacing the motor controller board and sensors of the robot.

For interaction with the operator a stereoscopic display and a force reflectance joystick are used. The latter is an Omega unit from Force Dimension, which provides a large workspace, as well as high stiffness and force output.

For the stereoscopic display two different alternatives have been evaluated. The traditional stereoscopic display with shutter glasses has been tested using inexpensive glasses from an eDimensional 3D Vision System kit. However, Sharp has recently started to ship displays that allow free fusion of stereoscopic information without glasses. We use a Sharp 3D laptop (model Actius RD3D). The basic visualization is performed using Open Inventor 5 from Mercury Computer Systems and the force reflectance is implemented using the low level API provided by Force Dimension.

A. Design of VR environment

The 3D graphics rendering is done using Open Inventor 5 from Mercury Computer Systems on top of OpenGL. It employs hierarchical scene graphs containing shape primitives, surface and lighting properties, and transformations.

For building the scene graph a system of C++ classes have been constructed, allowing high level specification of rooms and their subcomponents, including windows, furniture, etc. The same C++ objects define the free space used in obstacle avoidance for the mobile platform.

The visual appearance of objects can be imported by Open Inventor as VRML files and inserted into the scene graph. This is very useful for complex objects, and makes it possible to do the modeling of e.g. a chair in any 3D modeling package instead of having to define it in the C++ source code. A screen dump of the operator interface together with a photograph of the same scene is in Fig. 4.

OpenGL quad buffered stereo with asymmetric frustum perspective mapping is supported, and the dependence of teleoperation performance on any static or dynamic virtual camera position can be explored. A web cam feed on a separate connection acts as a reference.

B. Obstacle avoidance force generation

As mentioned in section II part B1 above, force generation is based on mapping obstacles to the command space used. To achieve this when commands are given as velocities, obstacles are mapped to velocity space using the rule that if the robot would collide with an obstacle even if it started braking immediately, then its current velocity is part of an obstacle in velocity space.

B1. Mobile platform 2D forces

Because the differential drive Pioneer II platform is non-holonomic, it cannot be controlled in Cartesian velocity space, but is instead controlled in 2D speed-curvature ($v\kappa$) space, with the haptic device locked to a plane. This is similar to driving a car, where the wheel controls the curvature, and the accelerator and brake dictate the speed. A drawback is that, though physically possible on a differential drive platform, it does not allow turning on the spot as this would require infinite curvature. Additional problems result from not being able to calculate the curvature from measured wheel speeds

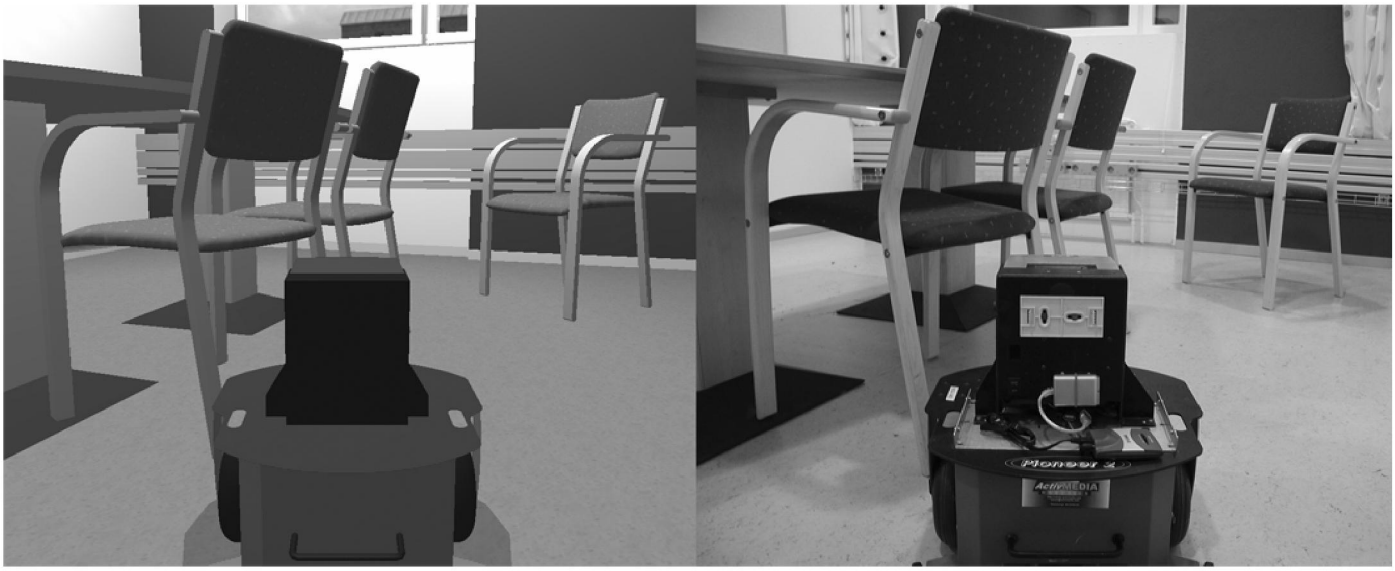

Fig. 4 Visualization (left) of the robot in a room with some furniture (right).

near zero velocity, but these can be handled by using an exponentially weighted (by traveled distance) moving average filter, in combination with a weak influence from the current commanded curvature.

Mapping of obstacles to $v\kappa$ space has been implemented using sampling of $v\kappa$ space at 40x40 points evenly distributed around the current (v,κ) of the handle. In Fig. 5 the not yet sampled points are represented by dots, and the (v,κ) of the handle by a square. For each curvature, speeds successively more different from the current speed are sampled in the positive and negative direction until a collision is found (a cross in the figure) or all points have been sampled (circles).

To build the obstacle avoidance force, each found collision point contributes a force proportional to the inverse of its squared distance to the handle and in the direction away from the (v,κ) of the collision point.

Points in $v\kappa$ space are sampled by calculating the position in 2D Cartesian space where the robot would stop if braking from speed v while following an arc with curvature κ. Then all obstacles are checked to see if this position is occupied, which would cause a collision. To correct for the non-zero size of the robot, obstacles are expanded by the radius of the robot platform (approximating it as cylindrical). A visualization of the positions corresponding to the sampling points is in Fig. 6. Dark points are on a constant curvature arc ending at a collision, light points are not.

The current implementation ignores the fact that the length of the obstacle curve in $v\kappa$ space approximated by the sampled collision points is not proportional to the number of points. Ideally, the force influence of line segments connecting neighboring collision points should be used instead.

The complete process of finding collision points and calculating the avoidance force (involving sampling of 1600 points if no collisions are found) is repeated at frequency equal to that of the haptic controller (~500 Hz).

B2. Robot arm 3D forces

In the three dimensions of the Cartesian velocity space of the robot arm, a similar sampling approach (involving 40^3 sampling points) becomes infeasible. However, in this case a point on an obstacle is easily mapped into velocity space since the distance to the point where the end-effector would stop after constant deceleration is proportional to the square of the initial velocity (Eq. 1), and the direction of course the same as that of the velocity. This leads to the relationship

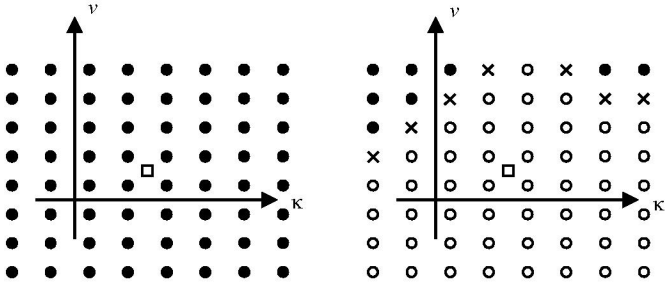

Fig. 5 Sampling of speed-curvature space, start and end of process.

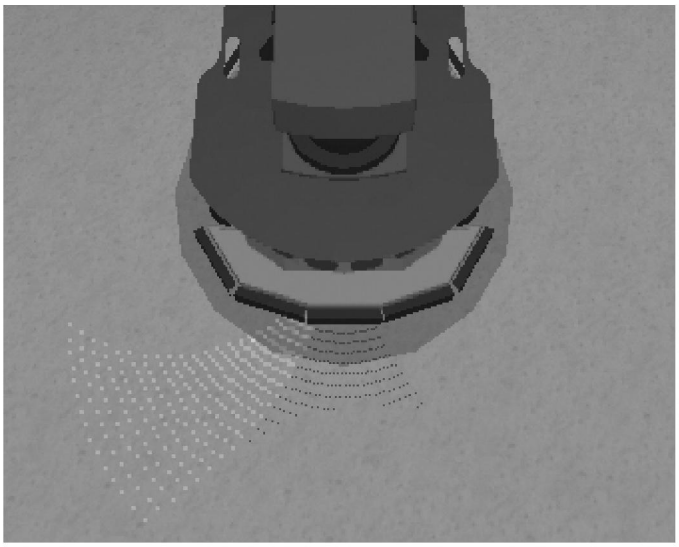

Fig. 6 Points sampled for obstacles (obstacle not shown).

$$\mathbf{v} = C\frac{\mathbf{r}}{(\mathbf{r}\cdot\mathbf{r})^{1/4}} \tag{2}$$

where $\mathbf{r}$ is the position of a point on an obstacle relative to the robot, $\mathbf{v}$ is the same obstacle point mapped into velocity space, and C is a constant that depends on a_{brake}. Using this, an obstacle surface can be mapped into velocity space, and the resulting obstacle avoidance force vector on the handle of the haptic device becomes

$$\mathbf{F}_o = \iint_{\Omega}\frac{\mathbf{v_h}-\mathbf{v}}{\left((\mathbf{v_h}-\mathbf{v})\cdot(\mathbf{v_h}-\mathbf{v})\right)^2}d\Omega\,. \tag{3}$$

Here, $\mathbf{v_h}$ is the velocity pointed to by the handle and Ω is the obstacle surface in velocity space.

Sampling an obstacle surface in Cartesian space, converting the samples to velocity space, and estimating the $d\Omega$ surface element (whose analytical formulation is a bit complicated even for a plane in Cartesian space) using the distance between converted samples allows estimation of the integral. The spacing of the sampling points must probably be adjusted according to the variation of the force contribution (the integrand in (3)) to keep the required number of points down. Obstacle avoidance forces in three dimensions have yet to be implemented.

C. Internet communication delays

The communication medium between the operator interface and the robot is the Internet or another IP network. Whereas this gives several advantages, including world-wide availability and abundant hardware and software support, it also poses some problems. There are no bandwidth guarantees, and, most importantly, there is a stochastic communication delay.

The magnitude and distribution of the delay [9] depends on several factors, such as physical distance, time of day etc. Typical round trip times are about 50 ms inside a city, 100 ms US coast to coast, and 150-200 ms for a transatlantic

connection. Different kinds of fluctuations in the delay include random noise, changes due to variable network load, delays from transmitting too much data, bunching of packets not originally transmitted together, and lost packets [9]. Improved real time communication properties can be achieved by using IPv6 where available [10].

On top of the IP protocol, UDP (user datagram protocol) is used for the teleoperation communication. For transmission of real time data UDP is preferable to TCP (transport control protocol), because it avoids the overhead of e.g. detecting and retransmitting lost packets, at the expense of not being able to guarantee transmission of all data or constant ordering [5]. If the packets are self contained, i.e. they each contain samples of all the transmitted signals, only the newest packet available at the receiving side is relevant, making retransmission of old data and enforcing constant ordering pointless.

To avoid the problems caused by variable delays, the teleoperation system uses a constant virtual delay equal to a postulated maximum communication delay as detailed in section II part A1 above. The virtual delay is realized by embedding a timestamp in each packet and adding an artificial delay at the receiving side so that the sum of the communication delay and the artificial delay is equal to the virtual delay. It is chosen to make occasions when the communication delay is larger than the chosen value relatively rare. They can then be treated as exceptions by the control system causing the robot to stop gracefully.

IV. Early Experimental Evaluation

Teleoperation using the proposed control structure has been demonstrated for the mobile platform, and for the robot arm. In each case stable operation has been achieved for varying delays up to 200 ms.

The manipulator system has only recently become available and early experiments with obstacle avoidance forces were thus performed on the mobile platform. It has been teleoperated in the presence of significant clutter to challenge the operator and to require both visual and force feedback. The test environment used for early experiments is that shown in Fig. 4 above.

With obstacles mapped to velocity space, their velocity distance from the robot (and therefore their influence on the obstacle avoidance force) depends not only on the Cartesian distance, but also on the speed of the robot. When driving fast through a narrow passage the generated force almost completely takes over control of the robot and guides it along a good path. Conversely, when there is a lot of free space in front of the robot or if driving very slowly, almost no force will be generated.

V. Issues for Future Research

The present system has been designed to provide a flexible basis for studying teleoperation of systems with significant dynamics, and for investigating several different questions. The obvious questions to be studied include

- Design of visual feedback strategies to assist in grasping.
- Types of force feedback to assist in grasping.
- Balance between autonomous control and operator interaction.
- Impact of better communication performance (a la IPv6) on teleoperation.
- Use of human motion models for design of a model based control strategy.
- Selection of visual frame of reference to maximize human performance.

These are merely a few of the questions to be studied in the design of efficient interfaces.

VI. Summary

The present paper has described the design of a system for teleoperation experiments in the presence of significant dynamics. The overall control model was developed and it was demonstrated how the model can be used with one or more sensory feedback modalities. In addition the basic feedback models were presented. Initial experiments with a mobile platform and with a manipulator system demonstrates that the system has a high fidelity of reality in the presence of stochastic delays of up to 200 ms. Finally a number of issues for future studies were outlined.

References

[1] Frese, U., et al., "Off-the-shelf vision for a robotic ball catcher," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Maui, Hawaii, USA, pp. 1623-1629, Oct. 2001.

[2] Borst, C., Fischer, M., Haidacher, S., Liu, H., and Hirzinger, G., "DLR Hand II: Experiments and experiences with an anthropomorphic hand," *Proc. IEEE Int. Conf. Robotics and Automation*, Taipei, Taiwan, pp. 702- 707, Sept. 2003.

[3] Senoo, T., Namiki, A., and Ishikawa, M. ,"High-speed batting using a multi-jointed manipulator," *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, pp. 1191-1196, April 2004

[4] McIntyre, J., Zago, M., Berthoz, A., and Lacquaniti, F., "Does the brain model Newton's laws?," *Nature Neuroscience* 4, pp. 693-694, 2001

[5] Munir, S., and Book, W.J., "Internet-based teleoperation using wave variables with prediction," *IEEE/ASME Trans. Mechatronics*, Vol. 7, no. 2, pp. 124-133, June 2002.

[6] Niemeyer, G. and Slotine, J.-J.E., "Using wave variables for system analysis and robot control," *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, pp.1619-1625, April1997.

[7] Smith, O. J. M., "A Controller to Overcome Dead Time," *ISA Journal* 6 (2): 28-33, 1959

[8] Kosuge, K., Murayama, H., and Takeo, K., "Bilateral feedback control of telemanipulators via computer network," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Osaka, Japan, pp. 1380-1385, Nov. 1996.

[9] Niemeyer, G., and Slotine, J.-J. E., "Toward bilateral Internet teleoperation," in Beyond webcams: an introduction to online robots, Goldberg, K. and Siegwart, R. Eds., ISBN: 0-262-07225-4, MIT Press, Cambridge, MA, 2001.

[10] Nuño, E., and Basañez, L., "Using force feedback for assisted robot teleoperation via IPv6 protocol in high-tech networks," *36th International Symposium on Robotics (ISR)*, Tokyo, Japan, Nov. 2005.