Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Stochastic Switching Approach of Bilateral Teleoperation Systems: Part II - Experiment

Kevin Walker‡, Ya-Jun Pan† and Jason Gu‡

†*Department of Mechanical Engineering, Dalhousie University, Halifax, NS, B3J 2X4 Canada, Email: Yajun.Pan@Dal.Ca*
‡*Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS, B3J 2X4 Canada*

Abstract: In this paper, the theory of a new stochastic switching control strategy based on Linear Matrix Inequalities (LMIs) is applied to the bilateral teleoperation systems over networks with time delays and packet loss (Walker *et al.*, 2008). The control algorithm in the theory part is here extended to a case with asymmetric communication channels. In the experiment, two identical 1-DOF (degree of freedom) manipulators are used as master and slave systems. The characteristics of the network with random time delays and packet loss are thoroughly incorporated in the design. Correspondingly, a stochastic switching control is proposed to ensure that the tracking error is bounded by the rate of change of the external forces acting on the system. Finally, experimental setup and results are extensively discussed.

Keywords: Bilateral Teleoperation, Packet Loss, Network, Stochastic Control, Jump Linear Systems, Experiment, Manipulator

1. BRIEF INTRODUCTION

Increased attention has been paid to the area of bilateral teleoperation due to the presence of communication channels with random transmission delays into the overal system (Chopra *et al.*, 2003) (Pan *et al.*, 2006). When the communication channel has random delays, packet loss and bandwidth limitations, the system becomes unstable if the environmental force is fed back over networks without proper control designs.

Many teleoperation efforts in recent times have been based on the wave variable methods developed in (Niemeyer and Slotine, 1991) (Wang and Slotine, 2006), which don't guarantee the system performance. Some other methods require precise knowledge of the environment, (Bemporad, 1998), the operator (Prokopiou *et al.*, 1998), or both (Pan *et al.*, 2006) so that predictive methods can be used.

This paper is an extension of the stochastic switching approach for bilateral teleoperation system over networks with symmetric communication channels (Walker *et al.*, 2008). In the proposed paper, neither models for the operator nor for the environment are used when deriving the controllers to allow for general results that cover a wide range of scenarios and which would be robust to outside variations. The approach in this paper focuses on the decoupled asymmetric communication channels case, which is more general. It incorporates random time delays and packet losses and formulates the teleoperation system as a Markov Jump Linear System (MJLS) with stochastic properties. The extended algorithm is then applied to an experiment setup with two identical 1-DOF (degree of freedom) manipulators for master and slave systems. The experimental setup and recorded results are extensively discussed.

⋆ This research was supported by the Natural Sciences and Engineering Research Council (NSERC, Canada), the Canada Foundation for Innovation (CFI, Canada) and the Nova Scotia Health Research Foundation (NSHRF, Canada).

2. SYSTEM MODELLING

In this paper, a linear, single degree of freedom manipulator is modelled using state-space equations. The system modelling information are same as in (Walker *et al.*, 2008) while the network model is different as shown in the following part.

To determine the parameters of the network model for decoupled channels, an experiment was performed to characterize Internet communications. Internet Control Message Protocol (ICMP) packets were sent between Dalhousie University and Google.com (64.233.187.99). Various IP look-up services placed the location of this server in Mountain View, California, home of Google's headquarters, a distance of over 6,000km. Since ICMP packets record the round-trip time (RTT) the measured delays correspond to a packet that has travelled more than 12,000km, or roughly one third the circumference of the earth.

ICMP packets were sent approximately every second for a week from March 21st to March 28th, 2007 to determine the distribution of the delay at different times of day and different days of the week. Lost packets were logged as dropped after a time-out of 3 seconds. The `hrping` command line utility from cfos (http://www.cfos.de) was used to measure the delays because of its improved delay

measurement precision over other ping utilities. The packets had an IP header of 20 bytes, an ICMP header of 8 bytes and a payload of 32 bytes for a total size of 60 bytes. In practice the packets would be roughly this size since UDP headers are the same size as ICMP headers and 32 bytes of data allows for the transmission of 10 values with 24 bit precision.

In total, over 700,000 packets were transmitted and analyzed at every hour of the day and every day of the week. The average delay was 54.7ms and 7.46% of packets were lost. Variations were expected as the network load changed over time, but the results proved to be surprisingly stable over the different days of the week, shown in Fig.1, and different times of the day. After the data was collected, the delay distribution was calculated and the MC parameters were identified.

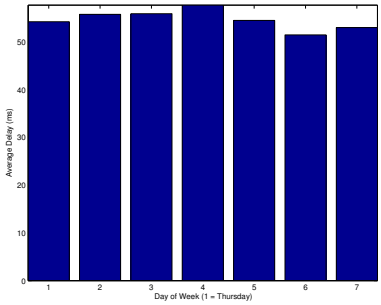

Fig. 1. Daily Histogram

A common assumption when designing NCSs is that the delay is bounded (Tipsuwan and Chow, 2004).These assumptions are necessary to apply a whole range of techniques that have been developed for NCSs over industrial networks. We can impose this condition on Internet communications by dropping packets that exceed the delay bounds. The lower bound τ_m is easy to establish as there is a minimum transmission delay that provides a sharp cutoff in the delay distribution. The delay distribution helps determine the upper bound τ_M so that the upper bound is as small as possible without inducing unduly large losses.

The probability density functions of the original experimental delays and the delays with imposed bounds are illustrated in Fig.2. Because packets with large delays were dropped for this simulation, the received packets have a smaller average delay of 51ms. Both distributions have a long tail, although not shown here to emphasize the nominal packet delays, which is typical in network communications.

The network data is pre-processed, marking all packets with a delay above τ_M and below τ_m as lost since they are dropped at the receiver. Choosing $\tau_m = 44$ms and $\tau_M = 64$ms, the new transition matrix is identified to get T_{bound} and π_{bound}:

$$T_{bound} = \begin{bmatrix} 0.8871 & 0.1129 \\ 0.7952 & 0.2048 \end{bmatrix}, \ \pi_{bound} = [0.8757 \ \ 0.1243]. \tag{1}$$

For asymmetric communication case and to derive a bidirectional communication model, two unidirectional links are combined into a meta-model. Two independent two-state MCs with identical transition matrices are combined

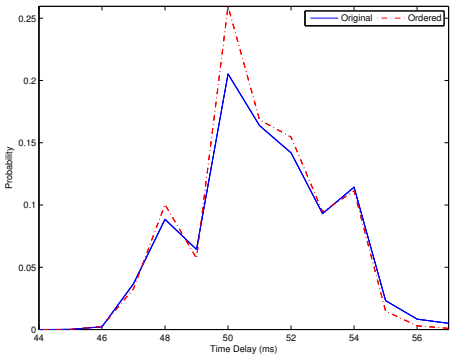

Fig. 2. Probability Density Function of Network Models

into a four-state MC describing all possible loss/reception scenarios. The transition matrices in Equation (2) are combined as described in Equation (3):

$$T_M = T_S = \begin{bmatrix} T_{R,R} & T_{R,L} \\ T_{L,R} & T_{L,L} \end{bmatrix}. \tag{2}$$

$$T_4 = \begin{bmatrix} T_{M_{R,R}}T_{S_{R,R}} & T_{M_{R,R}}T_{S_{R,L}} & T_{M_{R,L}}T_{S_{R,R}} & T_{M_{R,L}}T_{S_{R,L}} \\ T_{M_{R,R}}T_{S_{L,R}} & T_{M_{R,R}}T_{S_{L,L}} & T_{M_{R,L}}T_{S_{L,R}} & T_{M_{R,L}}T_{S_{L,L}} \\ T_{M_{L,R}}T_{S_{R,R}} & T_{M_{L,R}}T_{S_{R,L}} & T_{M_{L,L}}T_{S_{R,R}} & T_{M_{L,L}}T_{S_{R,L}} \\ T_{M_{L,R}}T_{S_{L,R}} & T_{M_{L,R}}T_{S_{L,L}} & T_{M_{L,L}}T_{S_{L,R}} & T_{M_{L,L}}T_{S_{L,L}} \end{bmatrix}. \tag{3}$$

Using the values from Equation (1), the composite transition matrix is Equation (4). From the stationary probabilities π_4, it can be seen that a packet is received at both the master and the slave 76.7% of the time while either the master or the slave receive a packet 98.44% of the time:

$$T_4 = \begin{bmatrix} 0.7870 & 0.1001 & 0.1001 & 0.0127 \\ 0.7054 & 0.1817 & 0.0898 & 0.0231 \\ 0.7054 & 0.0898 & 0.1817 & 0.0231 \\ 0.6323 & 0.1629 & 0.1629 & 0.0420 \end{bmatrix},$$
$$\pi_4 = [0.7668 \ \ 0.1089 \ \ 0.1089 \ \ 0.0155]. \tag{4}$$

3. EXTENSIONS TO DECOUPLED CHANNEL

In the paper of (Walker *et al.*, 2008), the communication from Master to Slave (MtoS) and Slave to Master (StoM) were symmetric and coupled. If a packet was lost at the slave side, it was also lost at the master side. The delays were identical as well. In practise, this assumption will not hold. In this paper, we will model the MtoS and StoM communication separately. The statistical properties of the MtoS transmissions are assumed to be identical to those of the StoM transmissions. Since the packets will be travelling through the same network, they should experience the same network load and number of network hops and should therefore behave similarly. The MJLS formulation is adapted to consider the asymmetry in the communication channel.

3.1 Error Dynamics

The system switches between four states depending on what information was lost. The master and slave closed-loop equations (i.e. when packets are not lost) are in Equation (5) and Equation (6) and the open-loop equations (i.e. when packets are lost) in Equation (7) and Equation

(8). These equations combine to provide the 4 cases in Table 1 that encompass all the combinations of losses and receptions at the master and slave.

$$\begin{aligned}\mathbf{x}_m[k+1] =&(A+BK_1)\mathbf{x}_m[k] + Bf_h[k] \\ &- Bf_e[k-\tau] - BK_1\mathbf{x}_s[k-\tau] \quad (5)\\ \mathbf{x}_s[k+1] =&(A+BK_1)\mathbf{x}_s[k] - Bf_e[k] \\ &+ Bf_h[k-\tau] - BK_1\mathbf{x}_m[k-\tau] \quad (6)\\ \mathbf{x}_m[k+1] =&(A+BK_2)\mathbf{x}_m[k] + Bf_h[k] \\ &- B\hat{f}_e[k] - BK_2\hat{\mathbf{x}}_s[k] \quad (7)\\ \mathbf{x}_s[k+1] =&(A+BK_2)\mathbf{x}_s[k] - Bf_e[k] \\ &+ B\hat{f}_h[k] - BK_2\hat{\mathbf{x}}_m[k] \quad (8)\end{aligned}$$

Table 1. System States

State	Master to Slave	Slave to Master	Dynamics
1	Received	Received	(5) and (6)
2	Received	Lost	(5) and (8)
3	Lost	Received	(7) and (6)
4	Lost	Lost	(7) and (8)

Using the augmented error vector defined in Equation (10) and the disturbance vector in Equation (11), the various open-loop and closed-loop equations can be combined to derive useful error expressions.

$$\mathbf{s}_m[k] = \begin{bmatrix}\mathbf{x}_m[k]\\ \hat{\mathbf{x}}_m[k]\end{bmatrix}, \mathbf{s}_s[k] = \begin{bmatrix}\mathbf{x}_s[k]\\ \hat{\mathbf{x}}_s[k]\end{bmatrix}, \quad (9)$$

$$\mathbf{e}[k] = \mathbf{s}_m[k] - \mathbf{s}_s[k] = \begin{bmatrix}\mathbf{x}_m[k] - \mathbf{x}_s[k]\\ \hat{\mathbf{x}}_m[k] - \hat{\mathbf{x}}_s[k]\end{bmatrix}, \quad (10)$$

$$\mathbf{d}_k = \begin{bmatrix} f_h[k] - f_h[k-\tau_k] \\ f_e[k] - f_e[k-\tau_k] \\ f_h[k] - \hat{f}_h[k] \\ f_e[k] - \hat{f}_e[k] \end{bmatrix}. \quad (11)$$

The error dynamics by state are

State 1, Equation (5) & Equation (6):

$$\mathbf{e}_{k+1} = A_{Rx}\mathbf{e}_k + A_{Rxd}\mathbf{e}_{k-\tau_k} + D_1\mathbf{d}_k.$$

State 2, Equation (5) & Equation (8):

$$\begin{aligned}\mathbf{e}_{k+1} =& A_{Rx}\mathbf{s}_m[k] - A_{Est}\mathbf{s}_s[k] \\ &+A_{Rxd}\mathbf{s}_m[k-\tau_k] + D_2\mathbf{d}_k.\end{aligned}$$

State 3, Equation (7) & Equation (6):

$$\mathbf{e}_{k+1} = A_{Est}\mathbf{s}_m[k] - A_{Rx}\mathbf{s}_s[k] + A_{Rxd}\mathbf{s}_s[k-\tau_k] + D_3\mathbf{d}_k.$$

State 4, Equation (7) & Equation (8):

$$\mathbf{e}_{k+1} = A_{Est}\mathbf{e}_k + A_{Estd} + D_4\mathbf{d}_k.$$

The augmented state matrices are defined as

$$A_{Rx} = \begin{bmatrix} A+BK_1 & 0 \\ 0 & 0 \end{bmatrix}, A_{Rxd} = \begin{bmatrix} BK_1 & 0 \\ I & 0 \end{bmatrix},$$

$$A_{Est} = \begin{bmatrix} A+BK_2 & BK_2 \\ 0 & A_{loss} \end{bmatrix}, A_{Estd} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} B & B & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, D_2 = \begin{bmatrix} 0 & B & B & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_3 = \begin{bmatrix} B & 0 & 0 & B \\ 0 & 0 & 0 & 0 \end{bmatrix}, D_4 = \begin{bmatrix} 0 & 0 & B & B \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The error dynamics of states 2 and 3 cannot be expressed solely in terms of the error vector and the disturbances, but it is shown that they can be combined when taking the expected value of the Lyapunov functional.

3.2 Stability Proof and Control Design

Note that the stability proof and control design (K_1 and K_2) are extensions of the theory as in (Walker *et al.*, 2008). It is omitted here due to the limited space.

4. EXPERIMENT SETUP AND RESULTS

4.1 Experiment Setup

To evaluate the performance of the controller design procedure, an experimental test-bed was developed. A linear system inspired by that of (Chopra *et al.*, 2003) was chosen that can evaluate a controller's performance under various conditions including motion in free space and contact with a hard surface. The overall block diagram is shown in Fig.3.

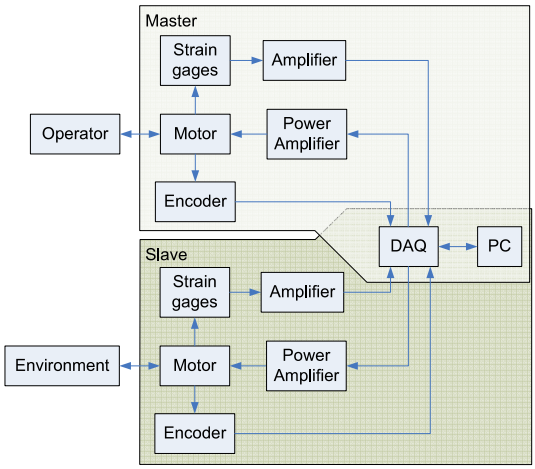

Fig. 3. Experimental Block Diagram

The master and slave systems are located in the same room and even share the same PC for their control, yet using the network data collected in the experiments described in Section 2, real world teleoperation conditions are recreated. Strain gages attached to the handles are used to measure the interaction forces which are digitized using a Data Acquisition Card (DAQ). The motor positions are measured using their built in position encoders and their velocity is derived. The motors are operated in a current-control mode to provide force control.

A. Mechanical Design

A CMC T0852 direct drive motor was chosen that can generate significant torques at stall conditions. The motor is capable of generating 3 Nm of torque without gearing under 24V excitation. The lack of gearing improves the back-driveability of the device, meaning that the effects of friction are greatly reduced and the manipulator can be moved more easily so that the operator can perceive the environment more readily, as shown in Fig.4.

B. Electrical Design

The analog inputs (AI) to the DAQ come from the strain gages. The strain gages are fixed to the aluminium plate

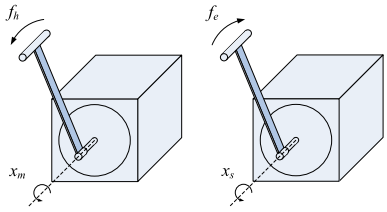

Fig. 4. System Conventions

in the handle near the motor shaft as shown in Fig.5. Four gages are used per handle, connected in a full Wheatstone bridge. Using a full bridge increases the sensitivity of the bridge to the strain and compensates for temperature variations.

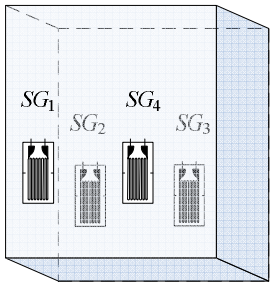

Fig. 5. Strain Gage Mounting Pattern

The gages are connected according to Fig.6 and their output is amplified. A large variable resistor (∼100 kΩ) is placed in parallel with the strain gage with the largest resistance to balance the bridge under no-load conditions. A three op-amp instrumentation amplifier structure was chosen to amplify the signal due to its high input impedance and was constructed using the low noise Texas Instruments TL074ACN op-amp. The strain gages are rated for up to 10V, so the excitation voltage for the bridge was chosen at 18V, which is also the maximum V_{CC+} for the op-amp.

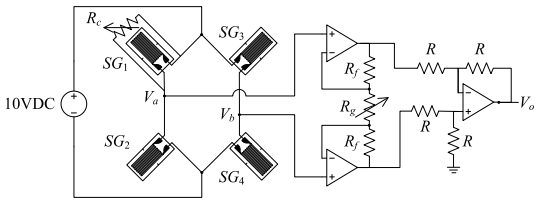

Fig. 6. Strain Gage Signal Conditioning

Connecting V_o to the AI of the DAQ, the measured signal is $V_o = (\frac{2R_f}{R_g} + 1)(V_b - V_a)$. Using a variable resistor for R_g, the system's sensitivity can be tuned. Fine-tuned calibration of the strain gage signal takes place in the custom control software. Small offsets are removed by clicking the "Calibrate" button before operation begins.

A conversion factor was determined experimentally to translate the strain gage signal from milli-volts to Newton-meters. Masses of known weight were suspended from the handles and the deviation in signal was measured. Given the length of the handle, the sensitivity of the bridge to given torques could be calculated. The master and slave systems were calibrated independently to accommodate variances in the circuitry and strain gage performance. The master and slave bridges were found to have a sensitivity of 8.54 V/Nm and 7.54 V/Nm, respectively.

The analog outputs from the DAQ are connected directly to the power amplifier. A Galil Motion Control Inc. AMP-19520 power amplifier was used to drive the motors which can provide up to 500W for each motor.

When the amplifier was turned on, it injected a great deal of noise into the system on the order of ±500 mV. Hardware filters were implemented to prevent damage to the digital components and reduce the vibration of the motors.

C. Counters

The motors come equipped with quadrature encoders with a resolution of 1000 ticks per revolution. The encoders are connected to the DAQ's counters with a D flip flop providing the direction of counter increment, as in Fig.7. The relative position of the handles is measured in this way, so it is important that the handles start in a "home" position when the system is powered up so that the system is synchronised.

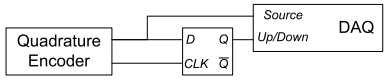

Fig. 7. Quadrature Encoder Connection

D. Software Design

A real-time platform was used to ensure successful control at frequencies of 1 kHz or more. The Real-Time Application Interface (RTAI) provides a hardware abstraction layer that allows Linux to run as a non-real-time process on top of a real-time operating system. In addition, the Linux Control and Measurement Device Interface (COMEDI) library provides real-time drivers for many DAQs, including the NI 6040E used here.

Two processes were written for the experiments. The first is a real-time process written as a kernel module in C that reads data from the DAQ, calculates and outputs the control signals to the DAQ and forwards all the data to the second process. The second process runs in user-space and has been written in Java. The two processes communicate via a real-time FIFO queue, with the user-space process blocking until more data is available.

4.2 Parameter Identification

The final experimental setup is shown in Fig.8. To design the control law for this experimental system, the moment of inertia and the damping must be calculated. The documentation of the motor states the rotor has an inertia of $J_r = 129.94 \times 10^{-6}$ $\mathtt{kgm}^2$.

The handles' inertia was determined by experiment. The handles were allowed to swing freely as a pendulum about the motor shaft's axis and their period of oscillation was recorded. Their weight is measured using a scale and their center of gravity (COG) is determined by balancing the handles on a thin edge. The handles' inertia is then given by $J_h = \frac{mgaT^2}{4\pi^2}$, where m is the mass of the handle in kilograms, g is the gravitational acceleration constant of 9.81 m/s^2, a is the distance from the axis of rotation to

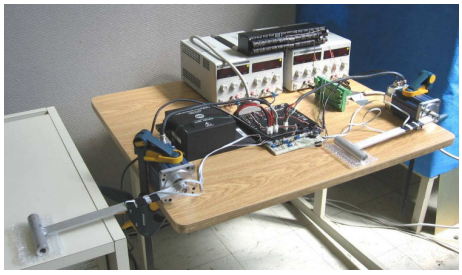

Fig. 8. Experimental Setup

the COG in meters and T is the period of oscillation in seconds. The measured parameters are listed in Table 2.

Table 2. Inertia Related Measurements & Calculations

Master Inertia		
Periods	Time (s)	Average Period (s)
20	20.464	1.0232
23	23.352	1.0153
20	20.417	1.0208
Average :	1.0198 s	
Weight :	0.277 kg	
Distance to COG :	116.6 $\times 10^{-3}$ m	
Inertia :	8.4482 $\times 10^{-3}$ kgm^2	
Slave Inertia		
Periods	Time (s)	Average Period (s)
25	25.796	1.03184
30	29.872	0.99573
28	28.094	1.0033
12	12.123	1.01025
Average :	1.0103 s	
Weight :	0.279 kg	
Distance to COG :	117.6 $\times 10^{-3}$	
Inertia :	8.2511 $\times 10^{-3}$ kgm^2	

The nominal inertia value is taken as 8.3496×10^{-3} kgm^2. The total inertia of the system is the sum of $J = J_r + J_h = 8.4796 \times 10^{-3}$ kgm^2. The handle is sufficiently rigid so that it does not introduce any dynamics into the system. The damping is then determined from the parameters of the motor. The motor's viscous friction is given as $b_m = 114.6 \times 10^{-6}$ Nms/rad according to the manufacturer.

The identified system parameters yields the system matrices after discretizing the continuous system.

$$A = \begin{bmatrix} 1.0000 & 9.9999 \times 10^{-4} \\ 0 & 1.0000 \end{bmatrix}, \; B = \begin{bmatrix} 5.8965 \times 10^{-5} \\ 0.11793 \end{bmatrix}.$$

The control gains were calculated as $K_1 = [-3.0212 \; -7.1933]$ and $K_2 = [0.0002 \; -0.0306]$.

4.3 Experimental Results

The designed controller was stable under real world Internet performance including losses and delays. The plots from two experiments are presented here. The first involves the manipulators moving through space under Internet conditions and the second case is that the manipulators interact with an environment under Internet conditions.

The movement through free-space under Internet conditions is shown below in Figures 9, 10, 11 and 12. The average position error was 0.1298 rad and the average force error was 0.0979 Nm. The master handle was pushed half way around and back without having the slave handle contact anything. The position tracking is fairly good, as seen in Fig.9, although periodic ripples can be observed in the slave's position. These ripples are due to the rotor's transition from one set of magnetic poles to the next and could be felt on the master side as indicated by the ripples in f_h in Fig.10. The control signal is quite noisy due to numerical differentiation and makes it difficult to compare f_h to u_s and f_e to u_m as was done throughout the simulations. The control signals involve many peaks, as seen in Fig.11. These spikes are due to the differentiation of the position signal to calculate the velocity term and from the packet losses. Filtering the signal introduces additional time delay, so the signal was left unfiltered. The disturbances, as the derivative of the forces, look like noise as in Fig.12.

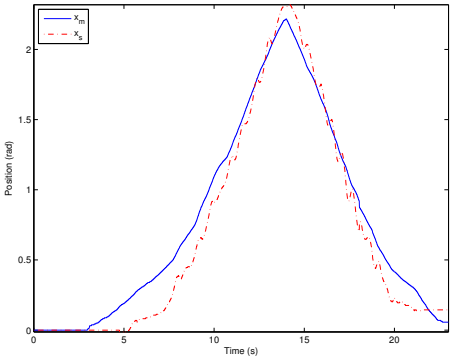

Fig. 9. Teleoperation - Manipulator Positions

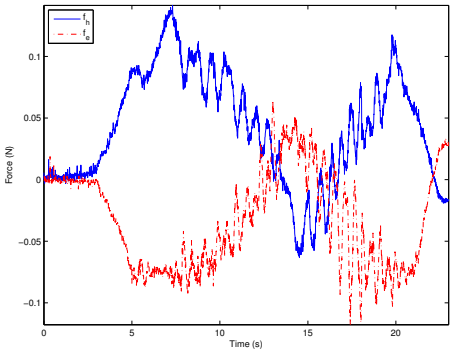

Fig. 10. Teleoperation - External Forces

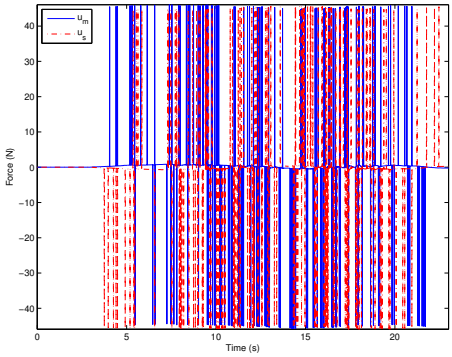

Fig. 11. Teleoperation - Motor Command Signals

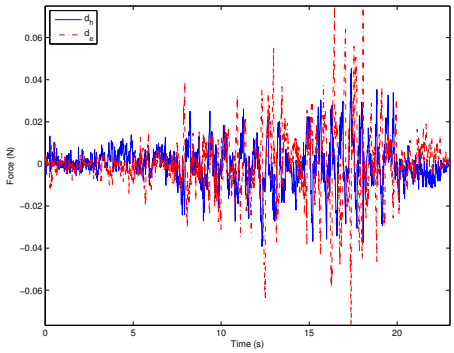

Fig. 12. Teleoperation - Disturbance Signals

There was still a fair level of noise on the signals from the power amplifier which caused a barely audible buzzing as the handles vibrated very slightly. This vibration caused problems when measuring the manipulator positions. Over time, an error accumulated and at the end of the experiment the positions of the master and slave are out of synch on the plot although both handles were returned to their "home" positions.

The second experiment suffered from the same position drift problems, but to a greater degree. In this case, a cardboard box was placed in the path of the slave manipulator in the hope that the master side would experience this obstacle. The positions are recorded in Fig.13 and the forces are in Fig.14.

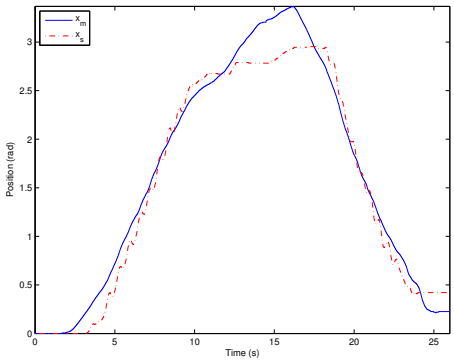

Fig. 13. Teleoperation Contact - Manipulator Positions

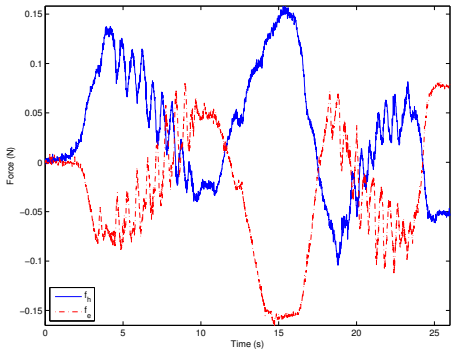

Fig. 14. Teleoperation Contact - External Forces

The presence of the box was transmitted to the master side, as evidenced by peak in the f_h force at around 11 seconds, but the recorded slave position slipped due to vibrations. The vibrations caused the position encoder to oscillate faster than the DAQ could read. Contact was made at 10 seconds and the slave manipulator was not moving again until 18 seconds, but the measured position continued to drift. The average position error was 0.1637 rad and the average force error was 0.1136 Nm, although these errors provide little insight, given the position measurement drift.

The designed controller was stable and the position tracking error was low enough to ensure intuitive operation. The operator force f_h is the mirror image of the environmental f_e, which shows how the forces were successfully transmitted over the communication channel. The proposed design scheme is thus validated.

5. CONCLUSION

The main contribution of this paper is the incorporation of network models into a teleoperation design scheme to guarantee performance without imposing restrictive conditions on the operator or the environment. The random packet losses and delays of the Internet were modelled and that knowledge was used in the control design to better represent the teleoperation system. Separate MtoS and StoM communications were considered and bounded delays were imposed. The final design was verified through experiments that were designed and built from the ground up. It was shown that Internet based teleoperation is possible in the face of random delays and packet losses.

REFERENCES

Bemporad, A. (1998). Predictive control of teleoperated constrained systems with unbounded communication delays. In: *Proceedings of the 37th IEEE Conference on Decision & Control*. Tampa, Florida. pp. 305–310.

Chopra, N., M. Spong, S. Hirche and M. Buss (2003). Bilateral teleoperation over internet: the time varying delay problem. In: *Proceedings of the American Control Conference*. Denver, Colorado. pp. 168–173.

Niemeyer, G. and J. J. Slotine (1991). Stable adaptive teleoperation. *IEEE Journal of Oceanographic Engineering* **16**(1), 152–162.

Pan, Y. J., C. Canudas de Wit and O. Sename (2006). A new predictive approach for bilateral teleoperation with applications to drive-by-wire systems. *IEEE Transactions on Robotics* **22**(6), 1146–1162.

Prokopiou, P. A., W. S. Harwin and S. G. Tzafestas (1998). Fast, intuitive and time-delays-robust telemanipulator designs using a human arm model. In: *Proceedings 6th Symposium on Intelligent Robotic Systems*. Edinburgh, UK. pp. 7–16.

Tipsuwan, Y. and M. Y. Chow (2004). Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation - part i: Networked control. *IEEE Transactions on Industrial Electronics* **51**(6), 1218–1227.

Walker, K., Y. J. Pan and J. Gu (2008). Stochastic switching approach of bilateral teleoperation systems: Part I - Theory. In: *Proceedings of the 17th IFAC World Congress*. Seoul, South Korea.

Wang, W. and J. J. E. Slotine (2006). Contraction analysis of time-delayed communications and group cooperation. *IEEE Transactions on Automatic Control*.