

A Review of Bilateral Teleoperation Algorithms

Riccardo Muradore and Paolo Fiorini

Riccardo Muradore and Paolo Fiorini are with the Department of Computer Science, University of Verona, Strada Le Grazie 15, 37134 Verona, Italy.
riccardo.muradore@univr.it, paolo.fiorini@univr.it

Abstract: Bilateral Teleoperation is a key technology to allow humans to interact with remote environments by providing the operator with haptic feedback. Haptic feedback from the one hand improves human perception and therefore the quality of the human-robot interaction, on the other hand it can tamper with the stability of the system when the communication between the master side (where the operator is) and the slave side (where the remote robot interacts with the environment) is not instantaneous but affected by delay and packet drops. In the last 40 years many algorithms have been developed to guarantee the stability of haptic teleoperation in the presence of time delay, many of them based on passivity theory. In this paper we review and compare a few algorithms that are representative of the tools in the frequency or in the time domains that have been used to develop a safe and transparent physical human-robot interaction with unknown environments.

Keywords: Bilateral teleoperation, Passivity, Communication delay, Haptics, Force reflection.

1 Introduction

Teleoperation of mechanical arms marked the beginning of robotics development and it has been an important component of this field ever since. Teleoperation systems allow humans to interact with remote environments by providing the operator with sensory feedback similar to that s/he would experience as if they were at the remote site. To achieve full sensory feedback, teleoperation systems should communicate also use contact force/torque information, from the slave to the master side to improve human perception and understanding, improve task performance and achieve the telepresence. The specific teleoperation configuration, in which, the kinesthetic coupling between operator and environment is enhanced by dynamic coupling, is referred to as *bilateral* teleoperation, since it provides *force feedback*. This has been one of the fields pioneered by *Antal (Tony) Bejczy* and this review aims at showing the wide legacy of his initial research.

Current examples of teleoperation span from space applications [1, 2] to rehabilitation, to surgery [3] and to Unmanned Air/Ground Vehicles (UAV, UGV).

A large number of papers have been written on the various aspects of teleoperation as shown in many survey papers [4, 5, 6], thus indicating a significant interest in the field; however these survey papers present mostly the analytical aspects of the reviewed algorithms without comparison and discussion of experimental results.

To fill this gap, we have added to our research in teleoperation, a series of experiments to analyse in detail the most common teleoperation algorithms with the goal of developing a unified classification of the field and of formalising a curriculum in teleoperation studies that could be of benefit to the whole robotics community. We focus this review on *bilateral teleoperation systems* in which the master and the slave devices are connected through a packet-based communication channel which delivers the signals, such as commands from the masters and measurements from the slave. Although this architecture is not the most commonly used in real systems, because of the potential instability due to force feedback in the presence of communication time delay, it surely must be analyzed and discussed because it provides the operator the full perception of the remote side [7, 8, 9].

In this paper, we present the initial results of our classification and analysis of the hundreds of teleoperation algorithms available in the scientific literature. We focus on algorithms that guarantee stability of the bilateral teleoperation system and we group them according the time delay present in the communication channel. For each algorithm we show its block diagram and its basic equations.

In section 2 we discuss one algorithm for the communication without time delay. In section 3 we present two algorithms that compensate an unknown but constant communication time delay. Finally, in Section 4, we present three algorithms that support force feedback in the presence of variable communication time delay and of data packet losses. Section 5 describes the experimental set-up on which we compare the performance of the various algorithms, some of the hardware calibration details and the results of some experiments. It must be noted that the experimental data presented, were the results of the work of the students following the course in Advanced Robotics offered at our University. Finally, Section 6 summarizes the paper and presents our plans to continue the analysis of teleoperation algorithms.

2 Bilateral teleoperation with no communication delay

Even if most teleoperation algorithms face communication delays, there are many important applications where the teleoperated system uses dedicated communication channels (e.g. Da Vinci surgical robot, ESA system), without any time delay noticeable by the user.

2.1 Four channel teleoperation architecture

Among the many algorithms that consider the bilateral teleoperation without communication delay we analyse the one proposed by Lawrence in 1993, the *Four channel architecture* [10].

The teleoperation system is completely transparent if the operator feels that s/he is directly interacting with the remote environment (equal forces $f_m = f_s$ and velocities $v_m = v_s$). Transparency requires that the transmitted impedance Z_t is equal to the environment impedance $Z_e = \frac{f_s}{v_s}$. According to the block diagram in Figure 1 we have to design the blocks $C_s, C_m, C_1, \dots, C_4$ in such a way that the hybrid matrix H

$$\begin{bmatrix} f_m(s) \\ v_m(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} v_s(s) \\ f_s(s) \end{bmatrix} \quad (1)$$

is equal to the matrix $\begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$. This implies

$$Z_t = \frac{f_m}{v_m} = (H_{11} - H_{12}Z_e)(H_{21} - H_{22}Z_e)^{-1} = Z_e. \quad (2)$$

To achieve this goal it is necessary to have very accurate models of the master and slave robots, otherwise the transparency condition will not be satisfied. However it is worth highlighting, that good transparency is important mainly at low frequencies, i.e. where the operator is working and where, fortunately, the mathematical models are more accurate.

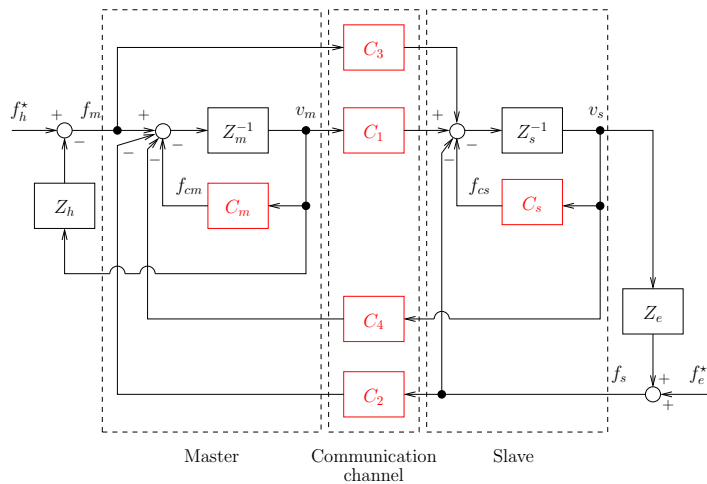


Figure 1

Four channel force-velocity architecture. Legend: Z_h operator arm impedance, Z_m^{-1} master robot admittance, Z_e environment impedance, Z_s^{-1} slave robot admittance; C_m master local controller, C_s slave local controller, C_1, \dots, C_4 communication link transfer functions; f_h^* operator intentional force, f_m, v_m force and velocity at the master side, f_s, v_s force and velocity at the slave side, f_e^* exogenous force at the remote site, f_{cm}, f_{cs} force controls at the master and slave side, respectively.

2.1.1 Discussion

The main tasks when using this algorithm are:

- Analyse the impact of the discretisation on transparency

- Identify the real values of the elements in H by using time series of velocities and forces, and compare them with the ideal values
- Consider the effects of model uncertainty and static friction

3 Bilateral teleoperation with constant and unknown communication delay

In several teleoperation systems the communication delay cannot be neglected but it can be assumed constant but unknown. The following approaches exploit the constant delay hypothesis to design control architectures that guarantee the stability of the system independently of the delay and the interaction with passive environments.

3.1 Wave variables and Scattering transformation

The architecture proposed in [11] transmits wave variables $\{u_m, w_m\}, \{u_s, w_s\}$ instead of power variables $\{f_m, v_m\}, \{f_s, v_s\}$ to guarantee the passivity of the system. The wave transformation relating wave and power variables is given by

$$u_m = \frac{1}{\sqrt{2b}}(f_m + bv_m), \quad u_s = \frac{1}{\sqrt{2b}}(f_s - bv_s), \quad (3)$$

$$w_m = \frac{1}{\sqrt{2b}}(f_m - bv_m), \quad w_s = \frac{1}{\sqrt{2b}}(f_s + bv_s) \quad (4)$$

where b is the characteristic impedance. The value of b is a design parameter. Figure 2 shows the architecture where we exploit the fact that the wave transformation is invertible and so the power variables can be retrieved from the wave variables

$$f_m = \sqrt{\frac{b}{2}}(u_m + w_m), \quad f_s = \sqrt{\frac{b}{2}}(u_s + w_s), \quad (5)$$

$$v_m = \frac{1}{\sqrt{2b}}(u_m - w_m), \quad v_s = -\frac{1}{\sqrt{2b}}(u_s - w_s). \quad (6)$$

Due to the communication delay τ , we have

$$w_s(t) = u_m(t - \tau) \quad (7)$$

$$w_m(t) = u_s(t - \tau). \quad (8)$$

It is worth remarking that when $\tau = 0$, transmitting wave variables is the same to transmit power variables.

The control architecture consists also of a velocity controller at the slave side whereas the force measured f_s is sent to the haptic devices for force rendering.

The solution of the time delay problem proposed in this algorithm shows why time delay may generate energy (and so destabilise the system) and how the wave transformation solves this problem [11].

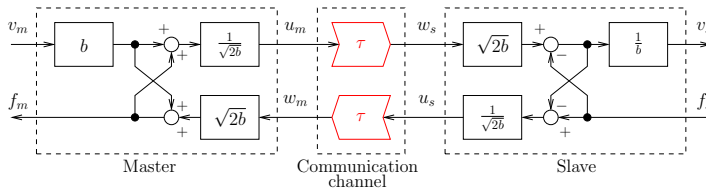


Figure 2

Wave variables. Legend: $\{u_m, w_m\}, \{u_s, w_s\}$ wave variables at master and slave side; $\{f_m, v_m\}, \{f_s, v_s\}$ power variables at master and slave side; τ constant communication delay; b characteristic impedance.

3.1.1 Discussion

The main aspects in the implementation of this algorithm are:

- The system at the master side behaves in a sluggish way also when the slave is in free motion. This behavior is function of the communication delay τ e the value of b .
- The parameter b should be tuned according to the delay τ .
- The transparency of the scheme in Figure 2 should be compared to the one having termination elements to match the impedances, which also introduce scaling in force (master side) and velocity (slave side).

3.2 PD and passivity terms

The solution proposed by Lee and Spong guarantees the stability of a position-position bilateral teleoperation by adding a dissipative term to the PD controller at each side of the architecture [12].

As shown in Figure 3 the master and slave controllers are given by

$$u_m(t) = \underbrace{-K_p \tilde{x}_m(t) - K_v \tilde{v}_m(t)}_{f_{cm}} - \underbrace{(K_{dis} + P_\epsilon)v_m(t)}_{f_{pm, \text{dissipation}}} \quad (9)$$

$$u_s(t) = \underbrace{-K_p \tilde{x}_s(t) - K_v \tilde{v}_s(t)}_{f_{cs}} - \underbrace{(K_{dis} + P_\epsilon)v_s(t)}_{f_{ps, \text{dissipation}}} \quad (10)$$

where $\tilde{x}_m(t) := x_s(t - \tau_{s2m}) - x_m(t)$, $\tilde{v}_m(t) := v_s(t - \tau_{s2m}) - v_m(t)$ and $\tilde{x}_s(t) := x_m(t - \tau_{m2s}) - x_s(t)$, $\tilde{v}_s(t) := v_m(t - \tau_{m2s}) - v_s(t)$ are the position/velocity tracking errors at the master and slave side, respectively. The gain K_{dis} is needed to ensure passivity of the P-control action ($K_{diss} = \frac{\bar{\tau}_{RTT}}{2} K_p$, where $\bar{\tau}_{RTT}$ is an upper bound of the round trip time $\bar{\tau}_{RTT} \geq \tau_{RTT} := \tau_{m2s} + \tau_{s2m}$) and P_ϵ adds additional damping to guarantee the master-slave position coordination.

3.2.1 Discussion

The main issues to be considered when using this algorithm are:

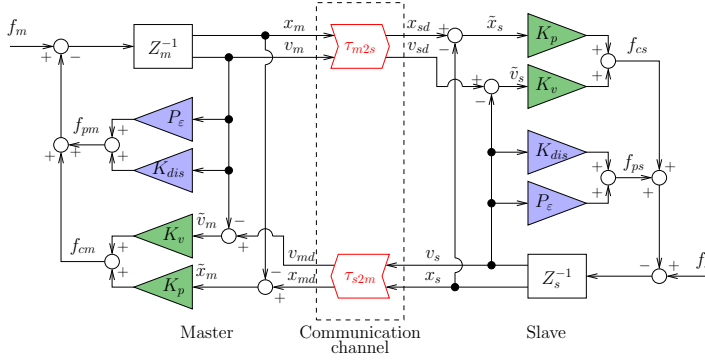


Figure 3

PD and passivity terms. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K_p, K_v position and velocity gains of the PD controller; K_{dis}, P_ϵ dissipative terms; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; f_{cm}, f_{cs} force controls at the master and slave side, f_{pm}, f_{ps} passivity-related forces.

- The effect of the discretization and of the velocity estimation.
- The speed of the system response as a function of K_{dis} (and so of K_p and τ_{RTT}).
- The effect of the values of K_p and K_v at the master and slave sides.
- The effect of P_ϵ on the level of transparency.
- The on-line adaptation of K_{dis} according to the measured delay.

3.3 Adaptive algorithm

The algorithm presented in this section for the constant communication delay scenario makes use of adaptive control to guarantee the passivity of the system [13]. In this algorithm, the PD controllers of the previous approach are replaced by an equivalent proportional controller on the new variables:

$$r_m(t) := v_m(t) + \lambda x_m(t) \quad (11)$$

$$r_s(t) := v_s(t) + \lambda x_s(t) \quad (12)$$

where $\lambda > 0$ is a tuning parameter. The overall architecture is shown in Figure 4 where the *Adaptive Estimation* blocks estimate the inertia and damping parameters of the DC motor at the slave and master side using an adaptive algorithm.

$$\begin{bmatrix} \dot{\hat{J}}_m(t) \\ \dot{\hat{B}}_m(t) \end{bmatrix} = \Gamma_m Y_m^T(x_m(t), r_m(t)) r_m(t) \quad (13)$$

$$\begin{bmatrix} \dot{\hat{J}}_s(t) \\ \dot{\hat{B}}_s(t) \end{bmatrix} = \Gamma_s Y_s^T(x_s(t), r_s(t)) r_s(t) \quad (14)$$

where Γ_m, Γ_s are constant positive definite matrices and Y_m, Y_s are the regression matrices.

The overall master and slave controls are computed as:

$$u_m(t) = f_{mc}(t) - \hat{J}_m(t)\lambda v_m(t) - \hat{B}_m(t)\lambda x_m(t) \quad (15)$$

$$u_s(t) = f_{sc}(t) - \hat{J}_s(t)\lambda v_s(t) - \hat{B}_s(t)\lambda x_s(t) \quad (16)$$

where $f_{mc}(t)$ and $f_{sc}(t)$ are the coordinating torques

$$f_{mc}(t) = K(r_m(t) - r_s(t - \tau_{s2m})) \quad (17)$$

$$f_{sc}(t) = K(r_s(t) - r_m(t - \tau_{m2s})). \quad (18)$$

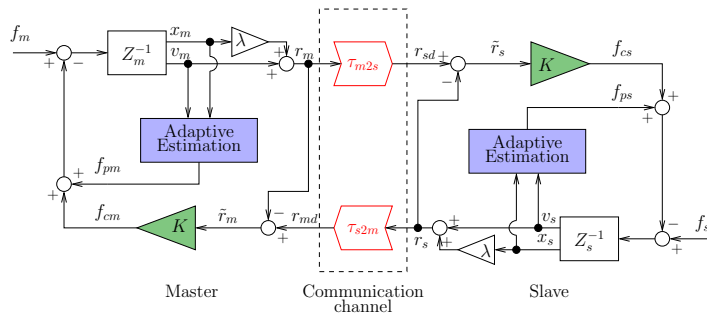


Figure 4

Adaptive-based algorithm. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K gain of the controller; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; λ tuning parameter; f_{cm}, f_{cs} force controls at the master and slave side, f_{pm}, f_{ps} passivity-related forces.

3.3.1 Discussion

The main steps for the practical implementation of this algorithm are:

- The effect of the discretization on the adaptive estimation.
- System transparency as a function of $K, \lambda, \Gamma_m, \Gamma_s$.
- The computation of the value of K , assuming to exactly know the dynamic parameters of the motors, and of λ to ensure the equivalence of this algorithm to the algorithm presented in the previous section.

4 Bilateral teleoperation with time-varying communication delay

In this section we address the case of bilateral teleoperation when the communication delay is time-varying and unknown. We present three solutions that share the

same idea: the passivity is guaranteed in the time domain by computing the energy balance at run time.

4.1 Time Domain Passivity Approach

The Time Domain Passivity Approach (TDPA) proposed in [14] uses two tools to evaluate the energy and to cope with energy shortage:

- The passivity observers PO compute the monotonically increasing energy leaving or entering the master and the slave side

$$E_{in}^*(k) = \begin{cases} E_{in}^*(k-1) + T_s P^*(k), & \text{if } P^*(k) > 0 \\ E_{in}^*(k-1), & \text{if } P^*(k) \leq 0 \end{cases} \quad (19)$$

$$E_{out}^*(k) = \begin{cases} E_{out}^*(k-1) - T_s P^*(k), & \text{if } P^*(k) < 0 \\ E_{out}^*(k-1), & \text{if } P^*(k) \geq 0 \end{cases} \quad (20)$$

where $*$ = m, s , $P^*(k) = f_*(k)v_*(k)$ is the actual discrete-time power computed at time $t = kT_s$, with T_s the sample time.

- The passivity controllers PC intervene any time an unstable behavior is going to be applied by activating dissipative elements

$$\beta_k = \begin{cases} \frac{\Delta E_s(k)}{T_s f_{sd}^2} & \text{if } \Delta E_s(k) > 0 \text{ and } f_{sd} \neq 0 \\ 0 & \text{if } \Delta E_s(k) \leq 0 \end{cases} \quad (21)$$

$$\alpha_k = \begin{cases} \frac{\Delta E_m(k)}{T_s v_{md}^2} & \text{if } \Delta E_m(k) > 0 \text{ and } v_{md} \neq 0 \\ 0 & \text{if } \Delta E_m(k) \leq 0 \end{cases} \quad (22)$$

where

$$\Delta E_s(k) := E_{out}^s(k) - E_{in}^m(k - \tau_{m2s}(k)) \quad (23)$$

$$\Delta E_m(k) := E_{out}^m(k) - E_{in}^s(k - \tau_{s2m}(k)). \quad (24)$$

Figure 5 shows this architecture where also low pass filters F are implemented in a bi-directional fashion (to preserve passivity) to reduce bumps and oscillations at the operator side.

4.1.1 Discussion

The following are important considerations that a designer using this algorithm should consider:

- How conservative is this “distributed” energy control.
- The effect of the communication delay on the performance of the TDPA algorithm.
- If and how to implement a position-position teleoperation instead of the proposed position-force scheme.

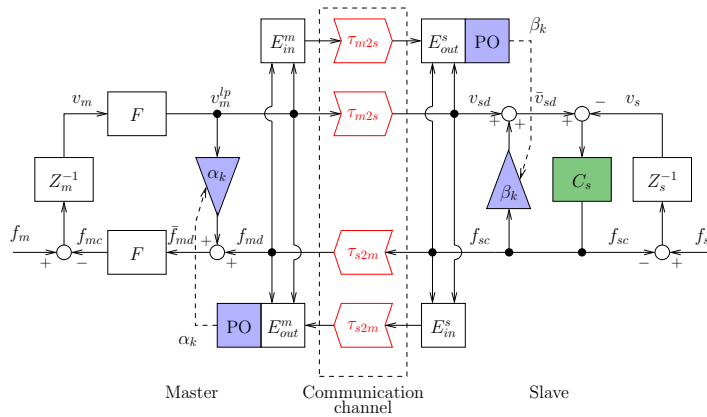


Figure 5

TDPA algorithm. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; C_s velocity controller; f_m, v_m force and velocity at the master side, f_s, v_s force and velocity at the slave side; $v_{sd}(t) = v_m^{lp}(t - \tau_{m2s})$ desired velocity at the slave side with $v_m^{lp} = F(s)v_m$, \bar{v}_{sd} modified velocity reference, $f_{md}(t) = f_{sc}(t - \tau_{s2c})$ desired force, \bar{f}_{md} modified force reference with $f_{md} = F(s)\bar{f}_{md}$.

4.2 Passive Set-Position Modulation

The Passive Set-Position Modulation (PSPM) algorithm is another approach to modify the value of the reference signal received from the master or the slave to comply with the passivity constraint.

The scheme proposed in [15] is a position-position teleoperation architecture where the original set-position signal (x_{md} or x_{sd}) is modulated in such a way that the new value (\bar{x}_{md} or \bar{x}_{sd}) satisfies the passivity constraint when applied to a spring $K_p(x_*(t) - \bar{x}_{*d}(k))$ with damping injection $K_v v_*$, $* = m, s$.

The novelty of this approach is to explicitly consider the possibility of passivity-breaking due to spring energy jumps at the switching instances. The scheme is shown in Figure 6 where E_m is the virtual energy reservoirs at the master, and the PSPM block solves the following minimization problem any time a new reference value is received:

$$\begin{aligned} \min_{\bar{x}_{md}(k)} \quad & \|x_{md}(k) - \bar{x}_{md}(k)\| \\ \text{s.t.} \quad & E_m(k-1) + \Delta E_s(k) + D_m(k-1) - \Delta \bar{P}_m(k) \geq 0 \end{aligned} \quad (25)$$

where $E_m(k-1)$ is the available energy, $\Delta E_s(k)$ is the energy received by the slave (*energy-shuffling*), $D_m(k-1)$ is the causal approximation of the damping dissipation (*energy reharvesting*, $\frac{1}{2}K_v x_m^2(k)$) and $\Delta \bar{P}_m(k)$ is the spring energy jump

$$\Delta \bar{P}(k) = \frac{1}{2} \|x_m(k) - \bar{x}_{md}(k)\|_{K_p}^2 - \frac{1}{2} \|x_m(k) - \bar{x}_{md}(k-1)\|_{K_p}^2 \quad (26)$$

In this algorithm, the local controllers are assumed continuous: in real implementation we can only assume that the local controllers have a sample time T_s smaller than the sample time at which the slave and master send their commands/measurements

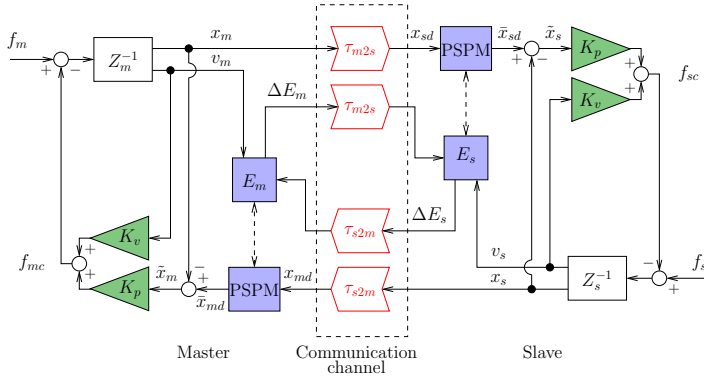


Figure 6

PSPM algorithm. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K_p, K_v position and velocity gains; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; x_{md}, \tilde{x}_{md} desired and modulated reference position at the master side; x_{sd}, \tilde{x}_{sd} desired and modulated reference position at the slave side; f_{mc}, f_{sc} force controls at the master and slave side.

to the other side of the bilateral architecture. The PSPM at the master side can also decide to send energy to the slave side (ΔE_m) by checking the current energy (*energy ceiling*).

Exactly the same approach works at the slave side proving that this architecture is symmetric as shown in Figure 6. The idea is to compute x_{md} (x_{sd}) in such a way that its value is as close to \tilde{x}_{md} (\tilde{x}_{sd}) as possible (transparency) but without violating the passivity constraint (stability).

4.2.1 Discussion

Since the original formulation assumes that the local controllers are continuous and only the communication channel is discrete-time, in a real implementation one should consider that also the controllers are discrete-time, and re-evaluate the performance of the algorithm as a function of the ratio of the sample time of the network and of the controllers. Other important aspects to be considered are:

- The effect of the size of ΔE_m and of ΔE_s on the transparency of the algorithm.
- The use of different techniques to solve in an efficient way the minimization problem.

4.3 A two-layer approach

The last algorithm discussed implements a hierarchical two-layer approach: the top layer (transparency layer) is used to implement the control strategy that better fits the performance requirement, and the lower layer (passivity layer) ensures that no “virtual” energy is generated [16].

The scheme is reported in Figure 7 for a position-force configuration. It is interesting to see that the approach of this algorithm is similar to the one of the two previous solutions but the passivity verification is done after the computation of the commands and not before.

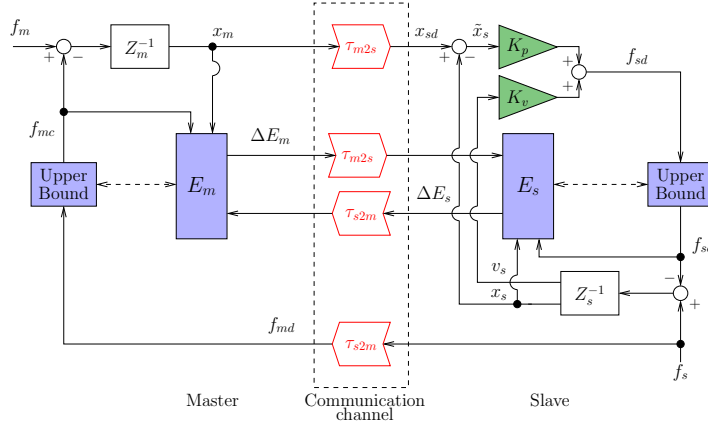


Figure 7

Hierarchical two-layer approach. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K_p, K_v position and velocity gains; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; f_{md}, f_{mc} desired and modulated forces at the master side; f_{sd}, f_{sc} desired and modulated force at the slave side.

The level of the energy tank E_m (the same hold for E_s) consists of three terms

$$E_m(k) = E_m(k - 1) + E_{s2m}(k) - E_H(k) \tag{27}$$

where $E_m(k - 1)$ is the past value of the energy, $E_{s2m}(k) = \Delta E_s(k - \tau_{s2m})$ is the energy received by the slave side and $E_H(k)$ is the energy exchange between the discrete-time controller and physical world. It is computed as

$$E_H(k) = u_m(k)(x_m(k) - x_m(k - 1)) \tag{28}$$

where $u_m(k)$ is the torque applied to the motor over the interval $[k - 1, k]$.

This energy is used to compute an upper bound for the command to guarantee the stability of the teleoperated system, e.g. $u_m^{max} = \frac{E_m(k)}{\hat{v}_m(k)T_s}$ where T_s is the sample time of the local controller and $\hat{v}_m(k)$ is an estimation of the future velocity.

If $E_m(k)$ is large enough the master sends energy $E_{m2s}(k)$ to the slave side and updates $E_m(k)$ accordingly

$$E_m(k) \leftarrow E_m(k) - E_{m2s}(k). \tag{29}$$

The same architecture is implemented at the slave side as shown in Figure 7.

4.3.1 Discussion

Additional points to be considered are:

- The implementation of a position-position structure within the hierarchical two-layer approach.
- The use of different policies for the master-slave energy exchange.
- The amount of energy present “in the network” as a function of τ_{m2s} , τ_{s2m} and the energy transfer policy.

5 Experimental setup

The experimental setup is shown in Figure 8 and it supports the implementation of C++ different types of control architecture in a real time middleware. This teleoperation architecture has one degree of freedom: this simplifies the analysis because we can avoid the complication of nonlinear control of multi-input multi-output manipulators, and is also more suitable to educational purpose and to compare the algorithms’ characteristics.



Figure 8
Bilateral teleoperation test bench.

The hardware consists of two DC motors, a gear at the master side with ratio equal to 4, two single axis force sensors mounted on rods, and a control board. The board has an interface to a Beckhoff™ EtherCAT® controller. The motor controllers are logically separated at software level sharing only the physical communication bus.

The software is developed in OROCOS [17], an open source component-based frame-work for robotic applications developed by K.U. Leuven, Belgium, in collaboration with LAAS Toulouse, France, and KTH, Sweden. It allows program flexibility (in C++), real time performance and code reusability.

The software runs onto a low latency Linux kernel patched for real time-preemption whereas the communication with the board is handled by RTnet [18]. This results in a stable computational platform capable of executing any teleoperation algorithm.

The main software component is called *E-Board* and interacts with the physical board by using the SOEM library [19]. Each side of the teleoperation architecture is represented by a logical abstraction of the control and of the DC motor and are indicated with *Master Side* and *Slave Side* in Figure 9. Both components receive the current angular positions from the board and send voltage commands to the motors' amplifiers. The master and slave sides exchange control/measurement data through two instances of a component that simulates a network connection by delaying and/or losing packets. The network component queues packets and assigns them a delay value. The delay can be constant or random; we implemented the following probability distributions: uniform (between an upper and lower bounds), Gaussian or exponential. When the current packet is received, the queue content is sorted according to the corresponding delay and the last valid packet is written to the output port. It is worth highlighting that in case of multiple packets with the same exit time (receiving time + communication delay), the queue is emptied of any packet with equal exit time and the one with the largest packet identifier (pktid) is sent out.

The state of master and slave is contained within a *struct* type called *DataStructure* (indicated as "D" in Figure 9):

```
struct DataStructure
{
    unsigned int pktid;
    char sender;
    double joint;
    double voltage;
    double speed;
    double absorbedCurrent;
    double timestamp;
    double genericField1;
    double genericField2;
    double genericField3;
    double genericField4;
};
```

The *Generic Fields* within the *DataStructure* are needed to exchange further information between the master and slave sides according to the particular solution under study (e.g. wave variables, energy quanta).

6 Results

In this section we show some plots of data collected during the test and comparison of the algorithms.

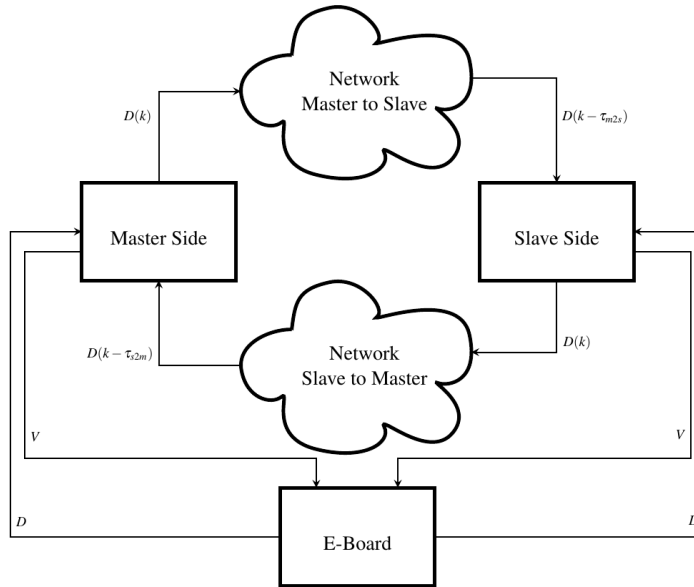


Figure 9
Components organisation.

6.1 Force sensors tuning

The test bench is equipped with two single-axis force sensors at the master and slave side. Some of the bilateral teleoperation algorithms explicitly use force measurements to provide *direct* force feedback to the user, whereas others provide an *indirect* force feedback related to the displacement between master and slave positions. The second set of algorithms requires fewer sensors but generates force feedback to the user even if the slave is not in contact with the environment.

An important step is to relate the measurement force to the motor torque and to the voltage command (ignoring the dynamics of the electrical subsystem). Figure 10 shows the linear interpolation relating voltages and force measurements. It is interesting to see that the force sensors have an important negative bias. Even though the sensors are the same, the slopes are different because of the gear ratio at the master side ($N = 4$). Moreover there is a “dead zone” around zero. This is not due to the sensor but to the high static friction of the motors. Voltages below 0.5 V do not move the motors. These force sensors have also a rather high quantization error (around 0.15 N) that will be clearly visible in Section 6.2.

6.2 Experiments

Figure 11 shows the behavior of the algorithm described in Section 3.2 when the slave gets in contact with the environment (~ 3.1 s). The bilateral teleoperation system remains stable despite the communication delay and the hard contact. The

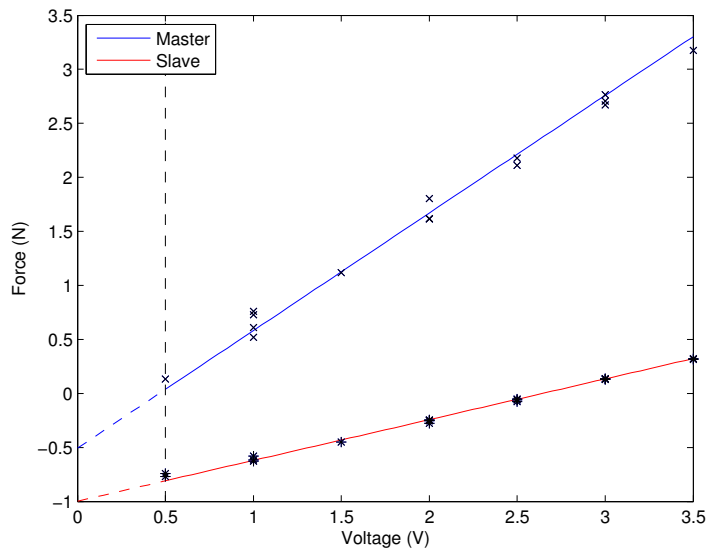


Figure 10
Calibration of the force sensors.

voltages at the master and slave side have very similar values (in absolute value but opposite signs) and are strongly correlated to the force measurements at the master side.

After the contact, the position of the slave does not change whereas the master force is proportional to the difference between the master and slave positions. The temporal misalignment that can be seen in the master and slave voltages is mainly due to the round trip time $\tau_{m2s} + \tau_{s2m}$ and partially on the haptic paddle dynamics and the control parameters (K_p , K_v and K_{dis} , P_{ε}). High values for τ_{m2s} , τ_{s2m} imply a high value for K_{dis} : the system behaves in a sluggish manner. This is the price to pay to have the system passive with high round trip time.

The plots in Figure 12 show how the PSPM algorithm of Section 4.2 modulates the desired reference position x_{sd} when the energy level E_s is close to zero. In this case there is no contact but the instability arises due to the discretization and the fast movement compared with the energy shuffling and reharvesting. Any time there is a lack of energy (positive energy in the bottom plot in the figure) the minimization problem (25) determines the proper modulated reference position \bar{x}_{sd} to satisfy the stability constraint.

Conclusions

In this paper we reviewed classic and more recent algorithms that guarantee the stability of bilateral teleoperation systems. The crucial concept of passivity is implemented in each algorithm in different forms to safely interact with unknown environment despite communication delay from the master and slave sides. We list the assumptions behind each algorithm, their advantages and disadvantages, and we

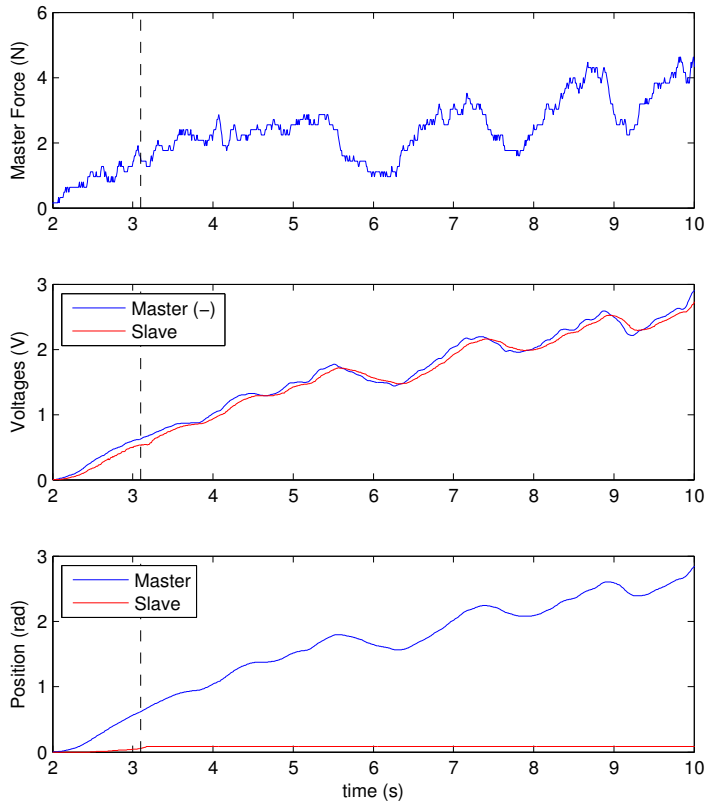


Figure 11

PD plus passivity terms. Hard Contact scenario, $\tau_{m2s} = \tau_{s2m} = 10ms$, $F_s = 100Hz$, $K_p = 1.0$, $K_v = 0.02$, $P_e = 0.001$. The dashed vertical line around 3.1s is when the hard contact occurs.

highlight some important points that a designer should take into account to choose the best algorithm for his/her applications. Experimental results obtained by students of the Advanced Robotics course, offered at the University of Verona, are also reported to demonstrate the behavior of the different algorithms.

References

- [1] ESA, <http://esa-telerobotics.net/>, (Access date: Dec 14, 2015).
- [2] NASA, <http://www.nasa.gov/>, (Access date: Dec 14, 2015).
- [3] Intuitive Surgical, <http://www.intuitivesurgical.com>, (Access date: Dec 14, 2015).
- [4] T. B. Sheridan, "Space teleoperation through time delay: review and prognosis," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 592–606, 1993.

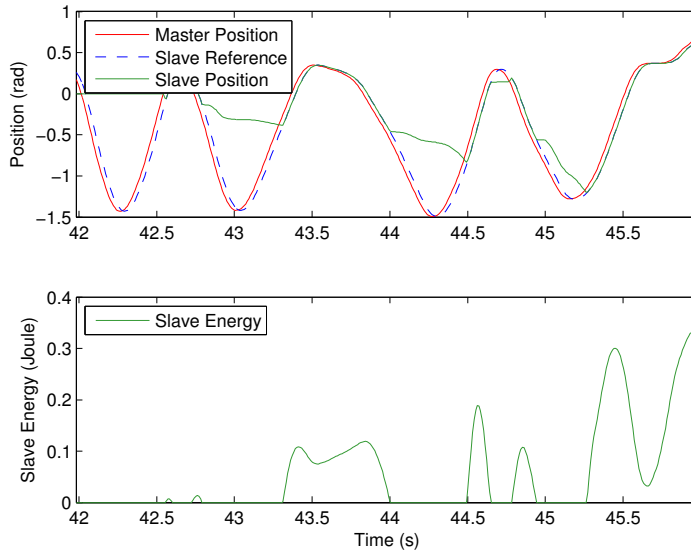


Figure 12

PSPM algorithm. $\tau_{M2S} = \tau_{S2M} = 10ms$, $F_s^{ctrl} = 500Hz$, $F_s^{network} = 100Hz$, $K_p = 1.0$, $K_v = 0.01$.

- [5] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.
- [6] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, no. 3, pp. 485–495, 2011.
- [7] Aliaga, I., A. Rubio, and E. Sanchez, "Experimental quantitative comparison of different control architectures for master-slave teleoperation," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 1, pp. 2–11, 2014.
- [8] De Gerssem, G., H. Van Brussel, and F. Tendick, "Reliable and enhanced stiffness perception in soft-tissue telemanipulation," *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 805–822, 2005.
- [9] Malysz, P. and S. Sirouspour, "Nonlinear and filtered force/position mappings in bilateral teleoperation with application to enhanced stiffness discrimination," *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1134–1149, 2009.
- [10] D. Lawrence, "Stability and transparency in bilateral teleoperation," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 624–637, 1993.
- [11] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," *Oceanic Engineering, IEEE Journal of*, vol. 16, no. 1, pp. 152–162, 1991.
- [12] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 269–281, 2006.

- [13] N. Chopra, M. W. Spong, and R. Lozano, “Synchronization of bilateral teleoperators with time delay,” *Automatica*, vol. 44, no. 8, pp. 2142–2148, 2008.
- [14] J.-H. Ryu, J. Artigas, and C. Preusche, “A passive bilateral control scheme for a teleoperator with time-varying communication delay,” *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010.
- [15] D. Lee and K. Huang, “Passive-set-position-modulation framework for interactive robotic systems,” *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 354–369, 2010.
- [16] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, “Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency,” *Robotics, IEEE Transactions on*, vol. 27, no. 4, pp. 741–756, 2011.
- [17] OROCOS, <http://www.orocos.org>, (Access date: Dec 14, 2015).
- [18] RTnet, <http://www.rtnet.org/>, (Access date: Dec 14, 2015).
- [19] SOEM, <http://openethercatsociety.github.io/>, (Access date: Dec 14, 2015).