(Article begins on next page)

# Bridging the gap between passivity and transparency

Michel Franken[1], Stefano Stramigioli[2], Rob Reilink[2], Cristian Secchi[3] and Alessandro Macchelli[4]

[1] Institute of Technical Medicine, University of Twente, Enschede, The Netherlands
m.c.j.franken@utwente.nl
[2] IMPACT Institute, University of Twente, Enschede, The Netherlands
[3] DISMI, University of Modena and Reggio Emilia, Reggio Emilia, Italy
[4] DEIS, University of Bologna, Bologna, Italy

*Abstract*— In this paper a structure will be given which in a remarkably simple way offers a solution to the implementation of different telemanipulation schemes for discrete time varying delays by preserving passivity and allowing the highest transparency possible. This is achieved by splitting the communication channel in two separate ones, one for the energy balance which will ensure passivity and one for the haptic information between master and slave and which will address transparency. The authors believe that this structure is the most general up to date which preserves passivity under discrete time varying delays allowing different control schemes to address transparency.

## I. INTRODUCTION

A telemanipulation chain is composed of a user, a master system, a communication channel, a slave system and an environment to act upon. Such a chain can be used to interact with materials in environments which are remote and/or dangerous or to augment the user's capabilities to facilitate the manipulation. In a bilateral telemanipulation chain the control algorithm usually tries to synchronize the motions of the slave device with the motions of the master device enforced by the user and to reflect the interaction forces between the slave device and the environment to the user by means of the master device. A proper haptic feedback is likely to increase the performance of the user with respect to effectiveness, accuracy and safety in many practical applications, e.g. robotic surgery [1].

Because the master and slave systems can be located at different sites, it is likely that time delays will be present in the communication channel. It is well known that due to these time delays it is not simply possible to exchange the power variables (forces and velocities) between the master and slave system. A direct exchange of these variables would create "fictitious" energy in the presence of time delays turning the communication channel into an active element. This energy could let the system become unstable.

An extensive overview of various approaches to the telemanipulation problem is given in [2]. Two criteria the "solution" to this problem should satisfy are transparency and stability, where transparency relates to how well the complete system is able to convey to the user the perception of directly interacting with the environment. A solution to the stability problem with respect to time delays can be found in the expression of the communication channel in scattering variables, which makes the system passive. It is shown that with these variables it is possible to "code" the power variables in such a way

that no energy is virtually generated in the communication channel [3] by effectively resembling the physics of an analog transmission line. This approach however has often been criticized with respect to the attainable transparency [4] and much effort has and is being put in developing adaptations of the original formulations to improve the transparency [2]. In any case, guaranteeing passivity of the telemanipulation chain is an effective and elegant solution to the stability problem.

Clearly, as a time delay is present, there will be limits to the achievable transparency, but in current schemes this limit is the result of the fact that both the energy exchanged between master and slave and the information which should transmit the haptic behavior for transparency are transmitted/coded in the same variables. This results in algorithms based on scattering variables to be passive but also limited in the achievable transparency, while other proposed methods without passivity property achieve better transparency but have discussible stability margins [4].

In this article we will introduce a new layered algorithm which shows that it is possible, in a remarkably simple way, to combine the benefits of passivity with the desire to achieve transparency. A two-layer framework is defined, where the top layer tries to satisfy the goals of the telemanipulation chain and the bottom layer has to make sure that the commands originating from the top layer do not violate the passivity condition.

In the rest of this paper we assume an impedance causality for both the master and slave device (velocities as input and forces as output to the robotic devices). For these devices the energy exchanged with the outside world can precisely be determined by the measured change of position, which is at the heart of Section IV [5]. The paper is organized as follows: Section II discusses the basic concepts of passivity and transparency and related work. Section III introduces the framework of the proposed algorithm. Sections IV and V contain the theory and proposed structures of the various parts of the framework and in Section VI an example is discussed. The paper ends with conclusions and a discussion of future work in Sections VII and VIII.

## II. BASIC CONCEPTS

Passivity and transparency are basic concepts in telemanipulation and for completeness will be explained shortly in this section. Afterwards, an overview of related work will be given

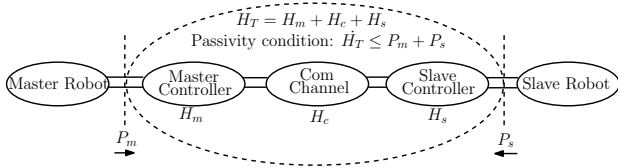to mark the differences with the proposed approach in Section III.



Fig. 1. Energy balance of the telemanipulation chain

### A. Passivity

A system is said to be passive if the energy which can be extracted from it is bounded from below by the injected and initial stored energy. Passivity of a system is a sufficient condition for stability and any proper combination of passive systems will again be passive. Physical energy exchange during operation is taking place between the user and the master system and between the slave system and the environment. Independent of anything else, including the goal of the telemanipulation chain, the only requirement therefore necessary to ensure passivity of the control system is:

$$\dot{H}_T(t) \leq P_m(t) + P_s(t) \qquad (1)$$

where $H_T(t)$ represents the total energy present in the control system at instant $t$ which is composed of all the energy present on the master controller $H_m(t)$, all the energy present on the slave controller $H_s(t)$ and the energy present in the communication channel $H_c(t)$, Fig. 1. $P_m(t)$ and $P_s(t)$ are respectively the power flowing from the master and slave robot into the master and slave controller.

### B. Transparency

A system is said to be transparent if the user gets the experience that he is manipulating the environment directly. An interaction can be defined by the associated power variables, so a perfectly transparent system without time delays ensures:

$$\begin{aligned} \dot{q}_s(t) &= \dot{q}_m(t) \\ \tau_m(t) &= \tau_s(t) \end{aligned} \qquad (2)$$

where $\dot{q}_m, \dot{q}_s$ are the velocities of the master and slave device and $\tau_m, \tau_s$ are the forces associated with the interaction between the master device and the user and between the environment and the slave device. The transparency of such port-based systems can be compared as described in [6]. When increased time delays are introduced in the communication channel between the master and slave device, it is not enough to simply reflect the measured interaction forces towards the user, not even considering the possible problems with stability, as in the ideal situation this would lead to:

$$\begin{aligned} \dot{q}_s(t+T) &= \dot{q}_m(t) \\ \tau_m(t+T) &= \tau_s(t) \end{aligned} \qquad (3)$$

and a mismatch in the display of the interaction occurs which becomes worse with increasing time delays. An example of

this is presented in [7], which states that a wave variable travelling from the master to the slave system can be interpreted as a general *Move or Push* command and the returning wave describes the response of the slave to the received command. If the haptic feedback is introduced in order to safely interact with soft and delicate tissue then clearly this situation is not desirable as the tissue could be damaged before the imposed forces on the tissue are reflected to the user. Therefore a truly transparent system in the presence of arbitrary time delays should ensure:

$$\begin{aligned} \dot{q}_s(t+T) &= \dot{q}_m(t) \\ \tau_m(t) &= \tau_s(t+T) \end{aligned} \qquad (4)$$

meaning that the behavior at both interaction ports is equal but delayed in time. As this requires an acausal system, true transparency cannot be achieved. The best achievable result requires a predictor at the side of the master device that predicts the future interaction forces between the slave device and the remote environment, which requires a local model of the remote environment.

### C. Related work

In this section, literature will be discussed that bears relevance to the approach proposed in Section III. In [8] a wave variable based algorithm is presented in which a Smith Predictor is used to reduce the mismatch in time from Eq. 3 by correcting the incoming wave variable with the predicted future value of that variable without loss of passivity. Several algorithms based on the haptic interaction with a local model of the remote environment in the time domain, generally referred to as *Impedance Reflection* algorithms, have already been proposed, e.g. in [9] and [10], but offer no passivity of the telemanipulation chain which is a desirable property as it guarantees stability. A first attempt to extend such a transparency strategy with a passivity property is described in [11], where a time domain Passivity Observer is used to adapt the locally computed feedback force based on the actual measured, but delayed, interaction force to make the system passive. Also interesting is the work in [12] where an online energy bounding algorithm is presented for a spring-damper controller at the slave side.

## III. PROPOSED FRAMEWORK

In the opinion of the authors the most abstract view of an algorithm which combines the discussed transparency and passivity properties would be a two-layer structure as shown in Fig. 2. The *Transparency Layer* tries to satisfy the goals of the telemanipulation chain: movement synchronization on the slave side and force reflection on the master side, by computing a desired torque, $\tau_d$ based on the measured interaction data, $F_m, x_m, \dot{x}_m, ...$ and $F_s, x_s, \dot{x}_s, ....$ The *Passivity Layer* on the other hand has to make sure that the commands originating from the *Transparency Layer* do not violate the passivity condition. The benefit of this strict seperation in layers is that the strategy used to ensure optimal transparency does not depend on the strategy used to ensure passivity and
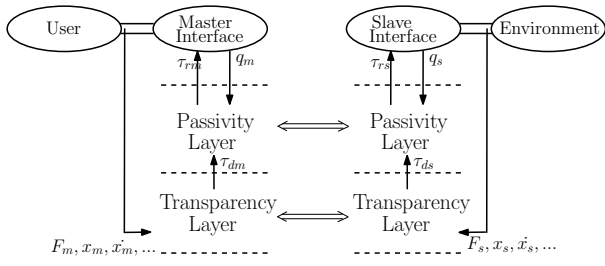
Fig. 2. Two layer algorithm for bilateral telemanipulation

vice versa. Most schemes only incorporate a single, possibly mixed layer and as a result a single two-way communication connection between the master and slave system. In this algorithm, however, the passivity and transparency are dealt with in seperate layers and therefore we can easily define two two-way communication channels between the master and slave system. One channel is used to communicate energy exchange related information between the *Passivity Layers* and the other to communicate information related to the behavior between the *Transparency Layers*.

## IV. PASSIVITY LAYER

This is the lowest-level control layer, whose only concern is to maintain passivity of the telemanipulation chain. Every movement the slave device makes will have an associated energetic cost and this energy will therefore have to be present at the slave side at the moment the movement is executed. In order to maintain passivity according to Eq. 1, the same energy will also have to be injected previously by the user at the master side. This clearly requires the transport of energy from the master to the slave system. To this end, the concept of a lossless energy tank is introduced in the *Passivity Layer* at both the master and the slave side, which can exchange energy. The level of these tanks can be interpreted as a tight energy budget from which actions can be fueled and which are continuously being replenished by the user at the master side.

If the energy level in the tanks is low, the movements the devices can make are restricted. An extreme situation occurs when the tank at the slave side is completely empty in which situation the slave device cannot make a controlled movement at all. In [12] also an energy tank is defined at the slave side to maintain passivity, but there the dissipated energy by the damping in the controller is accumulated, whereas in this framework the user directly makes energy available at the slave side regardless of the implemented controller for the desired behavior, this strict separation of energetic passivity and desired behavior is the key property of this two layered approach.

Suppose for the moment that no energy is exchanged between the user and the master system and no energy is exchanged between the slave system and the environment. In such a case an energy balance which would preserve passivity would be obtained if:

$$H_m(t) + H_c(t) + H_s(t) = 0 \qquad (5)$$

where in general, instead of 0 any positive constant could be chosen and $H_m(t)$ and $H_s(t)$ represent the energy levels of the tanks at the master and slave side and $H_c(t)$ is the total energy in the communication channel at time instant $t$.

As far as energy exchange is concerned, we can consider the possibility to send energy quanta from master to slave when there is energy available in the tank at that side and vice versa in the form of packets containing the amount of energy sent. When these packages arrive at the other side they are stored in a receiving queue. Both master and slave can implement completely asynchronously at any instant of time the following operations which, if we indicate it for simplicity for the master, would be:

$$H_{m+}(t) = \sum_{i \in Q_m(t)} \bar{H}_m(i) \qquad (6)$$

where $Q_m(t)$ represents the set of all energy packets present in the queue of the master at time $t$, $\bar{H}_m(i)$ represents the packet (energy quantum) $i$ present in that queue and $H_{m+}(t)$ represents the total amount of energy which is present in the receiving queue.

At time instant $t$ the receiving queue is then emptied, meaning that the energy present in the receiving queue is added to the energy tank, and if necessary a certain amount of energy is transmitted to the other side. The energy level of the tank at the master side after these operations, $H_m(t_+)$ will therefore be:

$$\begin{cases} H_m(t_+) = H_m(t_-) - H_{m-}(t) + H_{m+}(t) \\ Q_m(t_+) = \emptyset \end{cases} \qquad (7)$$

where $H_m(t_-)$ is the energy level of the tank before the operations at time $t$, $\emptyset$ represents an empty set and $H_{m-}(t)$ is the amount of energy which is sent via the channel to the other side in the form of a packet and which should clearly be any value smaller or equal to $H_m(t_-) + H_{m+}$ not to lose passivity of the tank itself. An illustration of this procedure is given in Fig. 3. It can be easily seen that by construction, this algorithm satisfies Eq. 5 and allows energy to be sent from the master to the slave and vice versa without losing passivity.
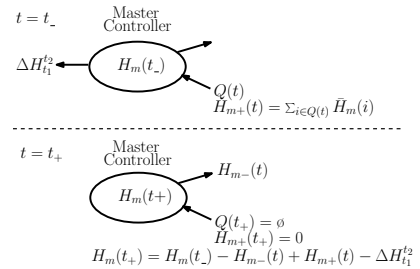


Fig. 3. Energy distribution at master side

### A. Interaction with the physical world

Unlike Eq. 7, Fig. 3 contains an energetic interaction port with the physical world. On the master and slave side, the controllers will have to control two robots which will interact

with the user and the environment. A Zero Order Hold (ZOH) is used to connect the discrete time controller to the actuators and will hold the control signal constant until a new value is set.

Let us indicate with $q(t)$ the position vector representing the angles/positions of the actuators at time $t$, with $\dot{q}(t)$ their velocities and with $\bar{\tau}(t)$ the torques/forces they are exerting on the mechanism, set by the controllers and held constant by ZOH during the sample interval. Suppose at a certain instant of time $t_1$ to have measured $q(t_1)$ and to have set a certain value $\bar{\tau}(t_1)$ for the actuators. At a subsequent time instant $t_2$ it is possible to exactly calculate the physical energy exchanged with the robot, by measuring $q(t_2)$:

$$\Delta H_{t_1}^{t_2} = \int_{t_1}^{t_2} \bar{\tau}(t_1)\dot{q}(t)dt = \bar{\tau}(t_1)(q(t_2) - q(t_1)) \quad (8)$$

where $\bar{\tau}(t_1)$ indicates the constant value held by the ZOH between $t_1$ and $t_2$. This is following the line of the method proposed originally in [5].

Even if not strictly necessary, we can now synchronize the reading of $q(t_2)$ and the setting of a new value of $\bar{\tau}$ with the operations expressed in Eq. 7 extending them as in Fig. 3:

$$\begin{cases} H_m(t_{2+}) = H_m(t_{2-}) - H_{m-}(t_2) + H_{m+}(t_2) - \Delta H_{t_1}^{t_2} \\ Q(t_{2+}) = \varnothing \end{cases}$$
$$(9)$$

With this algorithm we are therefore able to ascertain the energy balance at each instant of time when sampling occurs.

*B. Energy Tanks*

We have shown that it is possible to exchange discrete energy quanta between the master and slave system in a passive manner with arbitrary time delays in the communication channel and also to quantify the energy exchange between the controllers and the external world (user/environment). Now a method is required to extract energy from the user to fill the energy tanks and to compute the energy quanta which needs to be transmitted to the other side.

In order to extract energy from the user a Tank Level Controller (TLC) is defined in the *Passivity Layer* of the master system. This TLC monitors the energy level of the local tank $H_m$ and tries to maintain this at a desired level $H_d$. Whenever $H_m$ is lower than $H_d$ the TLC applies a small opposing force $F_{TLC}$ to the user's movement to extract energy from the user into the energy tank. This force is superimposed on the force to obtain the desired behavior and could be implemented as a modulated viscous damper as:

$$\begin{aligned} F_{TLC}(k) &= -d(k)\dot{q}_m(k) \\ d(k) &= \begin{cases} \alpha(H_d - H_m(k)) & \text{if } H_m(k) < H_d \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (10)$$

where $\alpha$ is a parameter that can be used to tune the replenishment of the energy tank. It is important to note that although this strategy might appear similar at first glance to the Passivity Observer/Passivity Controller strategy [13] it is in fact very different. This damper is only activated to make

energy available at the slave side for the interaction with the remote environment and not to dissipate virtually generated energy. It should be noted that the presented strategy is but one way to extract energy from the user and that the framework can accommodate many alternatives.

Now that energy is being extracted from the user, this energy needs to be transported to the slave system. A simple way to accomplish this is for each tank to transmit a fixed fraction, $\beta$ of its energy level (when energy is available) to the other system. If both systems run at the same sample frequency and there is no energy exchange between the slave tank and the physical system, the energy levels in the two tanks will converge to the same value, which in this case will be the set desired level for the master tank. This is illustrated in Fig. 4.
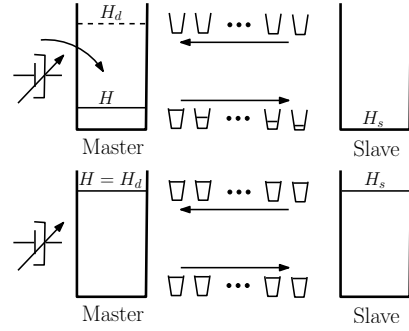


Fig. 4. Illustration of level synchronization between energy tanks

This means that energy quanta are continuously being exchanged between the master and slave system. Assuming equal sample frequencies $f_s$ at master and slave and constant time delays between master and slave, $\Delta T_{ms}$, and vice versa, $\Delta T_{sm}$, the total amount of energy present in the communication channel in steady state operation (no energy exchange with the physical systems) will be:

$$\begin{aligned} H_c &= H_{ms} + H_{sm} \\ H_{ms} &= \underline{\Delta T_{ms} f_s} \beta H_d \\ H_{sm} &= \underline{\Delta T_{sm} f_s} \beta H_d \end{aligned} \quad (11)$$

where $\underline{*}$ indicates a round down operation as there is a discrete number of packages present in the communication channel, $H_{ms}$ represents the energy traveling from the master to the slave and $H_{sm}$ the energy traveling in the other direction. The total energy in the communication channel $H_c$ can therefore become quite large for large time delays. Future research will look into this issue.

*C. Saturating the controlled effort*

The value which will be set as controlled torque $\tau_{d*}$ (the subscript $*$ replaced with $m$ refers to the master side and with $s$ to the slave side) for the interval to come, will be specified by the hierarchically higher level of control, the *Transparency Layer*, which will be treated in the next section. At both sides the *Passivity Layer* is used to limit the applied torque to the device with respect to what the *Transparency Layer* at that

side requests in order to maintain passivity. Three limits to which $\tau_{d*}$ can be subjected are given in Eq. 12, 13, 14.

When no energy is available in the *Passivity Layer*, the applied torque is set to zero:

$$\tau_{r*} = \begin{cases} \tau_{d*} & \text{if } H_* > 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Between two samples there will be no way to detect, act upon and therefore prevent a possible loss of passivity. If we know that the interval before a next sample will last $T$ seconds, the available energy is $H_*$ and the worst case motion velocity (highest in module) of the mechanism would be $v_*$, we can estimate an upper bound for the value of the applied torque $\tau_{r*}$ to which it should be constrained not to lose passivity. This tries to enforce:

$$T\bar{\tau}_{r*}v_* \leq H_* \quad (13)$$

An additional saturation method that can be useful is to define a mapping, $f_m$, from the current available energy in the tank to the maximum torque that can be applied. Meaning:

$$\tau_{r*(max)}(x) = f_m(H_*) \quad (14)$$

This mapping can be designed in such a way that a safe interaction in complex situations is guaranteed. Assume for instance that the slave is stationary but grasping an object in the environment and as a result no energy is exchanged between the slave and the environment. If at some point in time a loss of communication occurs, it could be desirable that the slave will smoothly release the object not to damage it by a continuous application of force. As each iteration the described algorithm for the energy tank sends an energy quantum to the other side, the energy level in the tank will drop even if no energy is exchanged with the environment. This means that the force exerted on the object in the environment will decrease over time and the storage function can be used to shape the manner in which it is released.

## V. Transparency Layer

In the previous section we have described an algorithm to maintain passivity in the presence of arbitrary time delays and to make energy available at the slave system. On top of this layer we can implement *any* algorithm to ensure that the desired behavior is displayed and transparency is obtained.

A first proposal for the structure of the *Transparency Layer* is given in Fig. 5. This structure arises from the discussion in Section II-B where it was derived that a local model of the environment is required at the master side to obtain the best transparency in the presence of time delays. In the following sections the functionality of each block of the structure in Fig. 5 will be discussed.

### A. Virtual Environment

This block contains a model which tries to mimic the remote environment and with which the user is interacting haptically. Based on the full state and interaction variables $F_m, x_m, \dot{x}_m, ...$ of the physical interaction the appropriate
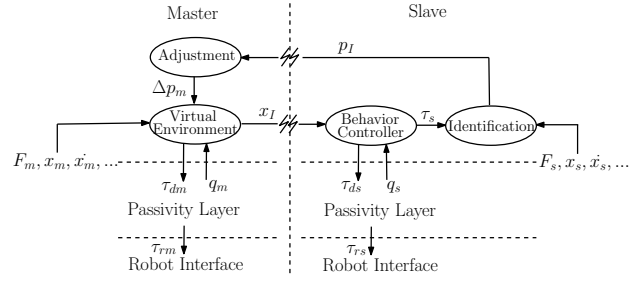


Fig. 5. Structure of the Transparency Layer

torque to apply to the master device is determined. The model is adaptable so that time varying environments can properly be displayed.

Many possibilities can be chosen for the structure of a model of the environment. A structure rich enough for many applications may be the one displayed in Fig. 6, where the right-end bar represents the interaction point between the slave robot and the environment.
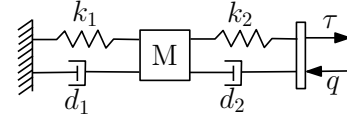


Fig. 6. A virtual model structure for the remote environment

$k_1$, $d_1$ and $M$ determine the dynamic behavior of the manipulated object in the remote environment and $k_2$ and $d_2$ the surface contact dynamics of the object itself. Simpler models can be implemented, e.g. based on a priori knowledge of the remote environment, to facilitate the parameter identification.

The model is running on a discrete medium with most likely a fixed sample frequency. This means that in *Impedance Reflection* algorithms using Euler discretized models without passivity properties, there exists an upper bound on the stiffness that can be reflected stably to the user [14]. Here the *Passivity Layer* will enforce passivity of the system and thus keep the system stable. However to minimize the interference of the *Passivity Layer* an elegant solution would be to implement a passive version of the model according to [5].

### B. Adjustment

At the master side the parameters of the local model of the remote environment have to be adjusted to match the identified parameters from the slave side. The delay in the communication channel in combination with the movement by the user can result in the generation of impulsive forces when the received parameters are directly introduced into the local model. These impulsive forces can lead to undesired contact behaviors at the master side and therefore a gradual convergence of the parameters of the model towards the received identified parameters is desired. To accomplish this, a smoothing function is implemented:

$$\Delta p_m = f(p_m, p_I) \quad (15)$$

where $p_m$ is the vector containing the current parameters of the model and $p_I$ contains the identified parameters received from the slave.

It is important to note that the change of parameters like mass, stiffness and length of a spring has energetical consequences in the model. When a passive implementation of the model is used this change of internal energy has to be accounted for. In order to have information on this energetical consequences, we can proceed as follows. Suppose that the state of the model would be $x \in \Re^n$. The energy function representing the total energy would then be a scalar function of $x$: $H(x) \in \Re$. We can express the energy then as a function of $p_m$ as well to get $H(x, p_m)$. By requiring a change in parameters $\Delta p_m$, the change in internal energy would be $\Delta H = H(x, p_m + \Delta p_m) - H(x, p_m)$. This energy can be drawn from the energy tank in the *Passivity Layer*.

### C. Behavior Controller

This controller has to make sure that the slave exhibits the desired behavior by the user. This can either be following an imposed trajectory, moving at a certain speed, exerting a certain force on the remote environment, etc. This controller receives the full state information about the interaction between the user and the master device. Based on the difference between this received information and the measured state information of the slave an appropriate control action is computed so that the slave mimics the behavior dictated by the user.

The dynamics of the behavior displayed by the user can be time-varying and/or uncertain and the same can apply for the dynamics of the environment. To avoid oscillations and maintain a proper reference tracking the implemented control structure has to be able to cope with this variation/uncertainty. Suitable control structures to implement here will therefore be either robust or adaptive control structures, of which the adaptive controller might be more suitable as the magnitude of the variation/uncertainty can be difficult to determine a priori.

### D. Identification

In order to accurately reflect the interaction forces to the user, the parameters of the local model have to be identified. A lot of research is being done in the field of impedance estimation of for instance soft tissue environments [15]. Any suitable identification algorithm can be implemented here as long as it is able to estimate the parameters of the model. This means that the complexity of the implemented model of the environment will largely determine the required complexity of the identification algorithm.

Several important characteristics of a suitable identification algorithm can be identified. The remote environment might contain delicate structures, e.g. soft tissues, so it is desirable that the identification algorithm does not require a strong excitation of the environment. The slave is probably used to

manipulate objects in the remote environment so that hard and soft contacts and free space movements will be interchanging. During this process the stability of the estimation will have to be guaranteed.

## VI. EXAMPLE

In order to illustrate the algorithm described above a simple simulation example will be presented here. The values of all the parameters are listed in Table I. The master and slave devices are modelled as two identical one-dimensional arms having a mass $m_*$ and a small viscous friction $c_*$, where like before the subscript $*$ replaced with $m$ refers to the master side and with $s$ to the slave side.

$$
\begin{aligned}
\tau_h + \tau_m &= m_m \ddot{q}_m + c_m \dot{q} \\
\tau_e + \tau_s &= m_s \ddot{q}_s + c_s \dot{q}
\end{aligned}
\tag{16}
$$

The user imposes a force $\tau_h$ on the master device to make it follow a sinusoidal motion. This is modelled as a PD-controller driven by a sinusoidal reference, $q_d$:

$$
\begin{aligned}
\tau_h &= k_h e_h + b_h \dot{e}_h \\
e_h &= q_d - q_m \\
q_d &= -Q_d \sin(\omega t)
\end{aligned}
\tag{17}
$$

The environment contains a wall at position $q_w$ with visco-elastic contact properties located on the trajectory the user tries to follow, $q_p$ is the penetration of the slave device into this wall:

$$
\begin{aligned}
\tau_e &= k_e q_p + b_e |q_p| \dot{q}_p \\
q_p &= \begin{cases} 0 & \text{if } q_s > q_w \\ q_w - q_s & \text{otherwise} \end{cases}
\end{aligned}
\tag{18}
$$

The proposed algorithm is implemented and runs at a sample frequency of 1000 Hz. The communication channels between the master and slave devices are lossless and have a fixed time delay of $1s$. All simulations have been carried out in the software program 20-sim [16].

### A. Passivity Layer

The *Passivity Layer* is implemented as discussed in Section IV. At the slave side a mapping in the form of a simple linear spring with stiffness $k_{ss}$ is associated with the energy level to saturate the control effort:

$$
|\tau_{max}| = \sqrt{2 H_s k_{ss}}
\tag{19}
$$

### B. Transparency Layer

For this simple initial example, some a priori knowledge about the environment is used. With respect to Fig. 6 the stiffness $k_1$ is known to be infinitely stiff and $k_2$ a linear spring and $d_2$ a non-linear damper where the damping coefficient depends linearly on the compression of spring $k_2$. The following implementations were chosen:

### 1) Virtual Environment:

The sample frequency of the control algorithm is more than sufficient to stably reflect the chosen stiffness of the environment. Therefore a simple discretized model of the environment with adjustable parameters $x_0, k_m, d_m$ is implemented:

$$
\begin{aligned}
p_w(k) &= q_m(k) - x_0 \\
\dot{p}_w(k) &= \frac{p_w(k) - p_w(k-1)}{\Delta T} \\
\tau_{dm}(k) &= \begin{cases} -k_m p_w(k) - b_m\,|p_w|\,\dot{p}_w(k) & \text{if } p_w(k) < 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{20}
$$

### 2) Adjustment:

The model is not passively implemented. Therefore there are no energetic consequences connected with parameter changes which need to be handled. The smoothing function limits the change in parameters to a percentage of the difference between the currently used and identified parameters:

$$
\Delta p_m = \gamma(p_I - p_m) \tag{21}
$$

### 3) Behavior Controller:

The behavior we want the slave device to mimic is trajectory tracking. The environment and user dynamics in this example are time invariant and therefore the controller parameters are fixed and tuned specifically for this example. Only the position of the master device $q_m$ is transmitted and $q_{\hat{m}}(k)$ is the position of the master system which is received at time instant $k$ by the slave system:

$$
\begin{aligned}
e_s(k) &= q_{\hat{m}}(k) - q_s(k) \\
\dot{e}_s(k) &= \frac{e_s(k) - e_s(k-1)}{\Delta T} \\
\tau_{ds}(k) &= -k_s e_s(k) - b_s \dot{e}_s(k)
\end{aligned} \tag{22}
$$

### 4) Identification:

The identification algorithm implemented here is a linear regression algorithm based on [15]. The estimator tries to minimize the cost function:

$$
\begin{aligned}
V_N(k_e, d_e) &= \frac{1}{N} \sum_{k=1}^{N} \epsilon^2(k) \\
\epsilon(k) &= \tau_e(k) - (\hat{k}_e \hat{x}_s(k) + \hat{b}_e\,|\hat{x}_s(k)|\,\dot{\hat{x}}_s(k))
\end{aligned} \tag{23}
$$

and is implemented by computing the following recursive equations each iteration:

$$
\begin{aligned}
\hat{\theta}(k) &= \hat{\theta}(k-1) + Q(k)[\phi(k) - \mu^T(k)\hat{\theta}(k-1)] \\
Q(k) &= R(k-1)\mu(k)[\beta_e + \mu^T(k)R(k-1)\mu(k)]^{-1} \\
R(k) &= \frac{1}{\beta_e}[\mathbb{I} - Q(k)\mu^T(k)]R(k-1)
\end{aligned} \tag{24}
$$

where $\theta = \begin{vmatrix} \hat{k}_e \\ \hat{b}_e \end{vmatrix}$, $\mu = \begin{vmatrix} x_s \\ |\hat{x}_s|\,\dot{\hat{x}}_s \end{vmatrix}$, $\beta_e$ is a forgetting factor to limit the estimation to more recent measurements, $\mathbb{I}$ is a 2×2 identity matrix and Q and R are 2×1 and 2×2 matrices. $R$ is initialized as an identity matrix and no prior information about the parameters of the environment is assumed: $\hat{x}_0(0) = \hat{k}_e = \hat{d}_e = 0$, $\beta_e$ is chosen as 0.99.

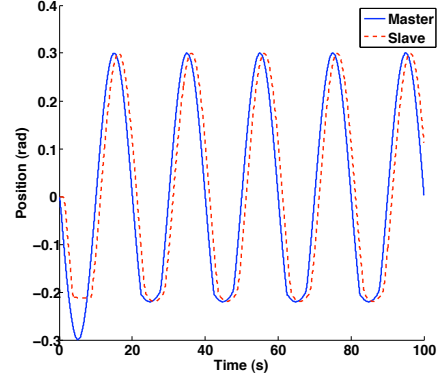| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $m_{m/s}$ | 0.1 $kg$ | $c_{m/s}$ | 0.1 $Ns/m$ |
| $k_h$ | 5 $N/rad$ | $b_h$ | 1 $Ns/rad$ |
| $Q_d$ | 0.3 $rad$ | $\omega$ | 0.31 $rad/s$ |
| $k_e$ | 25 $N/rad$ | $b_e$ | 10 $Ns/rad$ |
| $q_w$ | -0.2 $rad$ | $H_d$ | 0.01 $J$ |
| $\alpha$ | 100 | $\beta$ | 0.01 |
| $k_{ss}$ | 10 $N^2/J$ | $\gamma$ | 0.01 |
| $k_s$ | 100 | $b_s$ | 10 |

TABLE I

PARAMETERS OF THE SIMULATION



Fig. 7.   Position tracking

## C. Simulation Results

Fig. 7 shows the position tracking performance of the system. It is clearly visible that the user can initially complete the sinusoidal movement as the parameters of the local model are not yet set. The force exerted on the environment, Fig. 8, however, is limited due to the saturation strategy of Eq. 19. After this start-up phase, a very good correspondence between the master and slave trajectory is obtained. Fig. 8 shows the
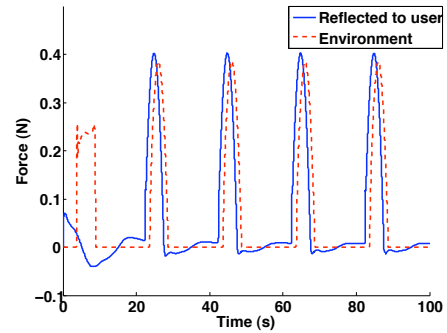


Fig. 8.   Forces during position tracking

interaction forces. At the beginning the user is moving in free space, as the model parameters are initially not set, and the force applied is used to extract energy from the user to fill the tanks. After the initialization period there is also a good correspondence between the reflected force to the master and the remote interaction force. The reflected force is indeed leading in time with respect to the interaction force. The peak value of the reflected force is slightly higher and this is caused due to a deeper penetration in the virtual wall by the user than really occurs in the remote environment. Some spurious forces are present in the reflected force which accounts for the energy

dissipated by the friction present in the slave system.

That the system is indeed passive can be seen in Fig. 9. The energy which flows into the remote environment is bounded by the energy injected by the user. The difference between the energy flows is stored in the energy tanks and the communication channel between the *Passivity Layers*. As mentioned, this energy is at the moment a quite large part of the energy injected by the user $(0.04J)$.
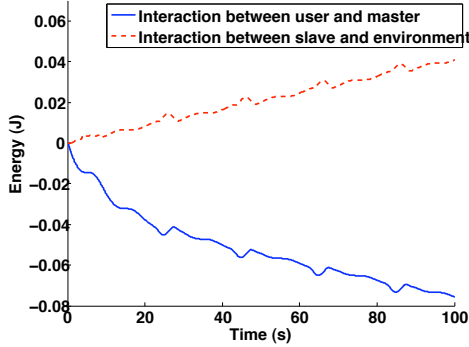


Fig. 9.    Energy exchange at interaction ports to the external world

A situation that demonstrates the benefit of the inclusion of the *Passivity Layers* is a loss of communication between the master and slave system. Fig. 10 shows the slave system response and at $t = 46$s all communication between master and slave system is eliminated. At this time instant the slave robot is still in contact with the remote environment but smoothly moves to a position where no force is applied. When the communication is restored at $t = 64$s the slave robot smoothly restarts following the master system.
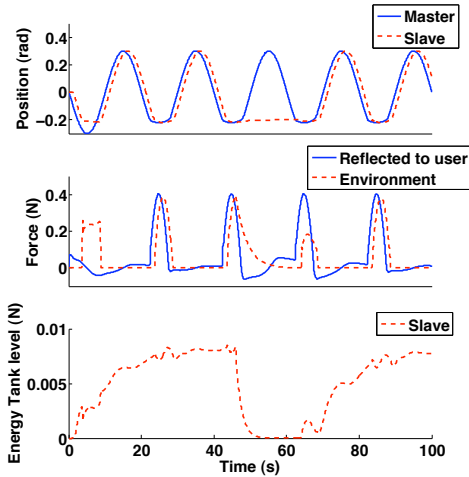


Fig. 10.    System response when communication loss occurs

## VII. Conclusions

A structure is presented that allows the combination of passivity and transparency in a very intuitive manner. The layered approach can contain different strategies and allows the separate design of the passivity and transparency properties of the telemanipulation chain. Simulation results of a first implementation show that this approach is very promising and

deals very well with large time delays. It should be noted that the achievable performance in the proposed implementation relies heavily on the ability to properly characterize the remote environment, which can be problematic in practical implementation. The proposed framework, however, can accommodate many variations of the proposed implementation.

## VIII. Future Work

The initial implementation as discussed in this paper will be implemented on a physical master-slave system to compare its performance with existing bilateral telemanipulation algorithms. As the structure can easily accomodate different implementations of the *Passivity* and *Transparency Layer*, further research will investigate different implementations based on this framework. One of the first additions to the *Passivity Layer* will be a friction compensation technique to extend this approach to manipulators with high internal friction.

## References

[1] B. Bethea, A. Okamura, M. Kitagawa, T. Fitton, S. Cattaneo, V. Gott, W. Baumgartner, and D. Yuh, "Application of haptic feedback to robotic surgery," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 14, no. 3, pp. 191–195, 2004.

[2] P. Hokayem and M. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, pp. 2035–2057, 2006.

[3] S. Stramigioli, A. van der Schaft, B. Maschke, and C. Melchiorri, "Geometric scattering in robotic telemanipulation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 588–596, August 2002.

[4] P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," *Robotics and Autonous Systems*, vol. 38, no. 1, pp. 49–64, 2002.

[5] S. Stramigioli, C. Secchi, A. van der Schaft, and C. Fantuzzi, "Sampled data systems passivity and discrete port-hamiltonian systems," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 574–587, August 2005.

[6] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Transparency in port-hamiltonian-based telemanipulation," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 903–910, August 2008.

[7] G. Niemeyer, "Using wave variables in time delayed force reflecting teleoperation," Ph.D. dissertation, Massachusetts Institute of Technology, September 1996.

[8] H. Ching and W. J. Book, "Internet-based bilateral teleoperation based on wave variable with adaptive predictor and direct drift control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 86–93, 2006.

[9] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 426–434, 1989.

[10] C. Tzafestas, S. Velanas, and G. Fakiridis, "Adaptive impedance control in haptic teleoperation to improve transparancy under time-delay," *Proceedings of ICRA 2008*, pp. 212–219, 2008.

[11] K. Kawashima, K. Tadano, G. Sankaranarayanan, and B. Hannaford, "Model-based passivity control for bilateral teleoperation of a surgical robot with time delay," *Proceedings of IROS 2008*, pp. 1427–1432, 2008.

[12] D. Lee and K. Huang, "Passive position feedback over packet-switching communication network with varying-delay and packet-loss," *Symposium on haptic interfaces for virtual environment and teleoperator systems, 2008*, pp. 335–342, March 2008.

[13] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on Robotics*, vol. 20, no. 2, pp. 365–373, 2004.

[14] J. Colgate, P. Grafting, M. Stanley, and G. Schenkel, "Implementation of stiff virtual walls in force reflecting interfaces," *Virtual Reality Annual International Symposium*, pp. 202–208, 1993.

[15] N. Diolaiti, C. Melchiorri, and S. Stramigioli, "Contact impedance estimation for robotic systems," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 925–935, 2005.

[16] Controllab Products B.V., "20-sim version 4.0," http://www.20sim.com/, 2008.