

A Passivity-Based Bilateral Teleoperation Architecture using Distributed Nonlinear Model Predictive Control

Nicola Piccinelli, Riccardo Muradore

Abstract—Bilateral teleoperation systems allow the telepresence of an operator while working remotely. Such ability becomes crucial when dealing with critical environments like space, nuclear plants, rescue, and surgery. The main properties of a teleoperation system are the stability and the transparency which, in general, are in contrast and they cannot be fully achieved at the same time. In this paper, we will present a novel model predictive controller that implements a passivity-based bilateral teleoperation algorithm. Our solution mitigates the chattering issue arising when resorting to the energy tank (or reservoir) mechanism by forcing the passivity as a hard constraint on the system evolution.

I. INTRODUCTION

Even though several works in the last decades have investigated many aspects and proposed different control algorithms, teleoperation is still an active research field. Bilateral teleoperation systems allow humans to interact with remote environments by providing to the operator the reaction forces that occur during the task execution. This force feedback from the environment to the operator side improves human perception and understanding and consequently enhances performance. When the operator feels to be interacting with the remote environment, the teleoperation structure provides what is called telepresence or transparency as described in [1]. In bilateral teleoperation communication delay and/or package losses are critical challenges since they can induce instability in the system. The solutions in the literature can be subdivided into three main categories: no communication delay, constant delay, and time-varying communication delay [2].

In [1] the transparency of the teleoperation in case of no communication delays is achieved using a coupling between the transmitted impedance and the impedance of the environment. In case of constant delays, the operator and environment side are coupled using the so-called wave variables which allow dealing with the energy stored in the communication channel [3]. For time-varying delays, the main approach is based on the computation of the energy balance at run time. The work of [4] introduced two new components in the teleoperation architecture: the passivity observer and the passivity controller. These components keep track of the energy entering or leaving the operator and environment sides and in case of instability dissipate

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 779813 (SARAS project, www.saras-project.eu) and was partly supported by the project MIUR "Department of Excellence" 2018-2022.

Nicola Piccinelli and Riccardo Muradore are with the Department of Computer Science, University of Verona, Italy (email: {nicola.piccinelli, riccardo.muradore}@univr.it)

the exceeding energy adding artificial damping. Another approach proposed in [5] modifies the reference signals between the operator and the environment side to keep the system stable.

One of the most recent solution addressing time-varying delays is proposed in [6] and implemented successfully in [7] and [8] where the hierarchical control scheme consists of a transparency layer and a passivity layer. The verification of the passivity is done after the computation of the command and acts as a modulator before the command is sent to the robot. The hierarchical decoupling between control and passivity modulator has two characteristics. First, it allows the implementation of different transparency policies without changing the passivity layer. Second, this solution can prevent an optimal global control strategy. Since the passivity action is made after the computation of the control action the performance in terms of transparency can be arbitrarily degraded to keep the system passive. All the mentioned solutions sacrifice performance to guarantee the stability in a sort of "blind" way.

Model predictive controller (MPC) was also proposed as a solution in teleoperation. In [9] the MPC is applied to provide input/output constraints with unbounded communication delay. The proposed solution adds a recovery mode where the system in case of long delays switches in an open-loop mode; the MPC is only placed at the operator side. A different approach proposed in [10] incorporates the delay in the discrete state-space model of a Linear Quadratic Gaussian (LQG) controller to enhance transparency in the case of a known constant delay. The main drawback is that the state-space dimension is proportional to the delay. In [11] a model predictive controller is also used to achieve stable teleoperation in the presence of uncertain constant time delays and force feedback. The delay effect is managed to design a specific controller for free-motion and contact scenarios. In [12] the authors proposed a single MPC at the operator side allowing teleoperation under time-varying delay when the state at the environment robot is estimated using a Smith predictor.

In this paper, we propose a distributed passive nonlinear model predictive controller with the aims of obtaining the optimal control command maximising the transparency. Although passivity in MPC has been successfully integrated in [13] where it was imposed as a constraint of the system, there are no implementations in the particular bilateral teleoperation scenario. As in [6] and [4] we rely on the energy-storing principle. We modelled the operator and environment side robots with their corresponding tanks as a unique system

and we exploited the receding horizon principle to compute the optimal commands over the prediction horizon. The main contributions of this paper are the following:

- a distributed MPC for bilateral teleoperation,
- the exploitation of the passivity to avoid the modelling of the communication delay within the MPC,
- a formalisation of a less conservative energy-storing tank dynamics,
- a reduction of chattering on the computed commands when the energy within the tank reaches its lower bound.

The paper is organised as follows: in Section II we review the two-layer algorithm and propose a new formulation of the tank dynamics evolution. In Section III we define the MPC problem tackled in this paper. In Sections IV and V we show the simulation and experimental results. Finally in Section VI we draw a few conclusions and present the future work.

Remark: The following notation holds for both the robots at the operator and environment side, so we'll specify with the subscript i the corresponding teleoperation side (*op*, *env*) only when needed.

II. PROBLEM FORMULATION

Since our method is based on the two-layer algorithm, in this section we will briefly summarise its main concepts. The key component of the two-layer architecture is the energy-storing system, called tank. The purpose of the tank is to keep track of the energy exchanged between the operator and the environment side, and within each side (e.g. for the operator side: the operator, the controller, the haptic device). Due to the communication delays, there are two tanks: one at the operator side and one at the environment side. The controllers within the transparency layer provide the desired input commands to the operator and environment robots. These commands are eventually modulated by the passivity layer based on the energy available in the tanks.

The Cartesian dynamic model of a gravity compensated robot is

$$I(x)\ddot{x} + C(x, \dot{x})\dot{x} + B(x)\dot{x} = u + \tau \quad (1)$$

where x represents the task space position and orientation, u and τ are the command input and the external torque, $I(x)$ is the inertia matrix, $C(x, \dot{x})$ is the centrifugal and Coriolis terms and $B(x) \succcurlyeq 0$ is the damping matrix corresponding to the friction and the artificial damping terms as defined in [14].

Let $H(t)$ be the Hamiltonian function representing the stored energy of the system (1)

$$H(t) = -\frac{1}{2}\dot{x}^T I(x)\dot{x} + \dot{x}^T(u + \tau). \quad (2)$$

The passivity of the system follows from the power balance [15]

$$\dot{H}(t) \leq \dot{x}^T(u + \tau). \quad (3)$$

Since the dynamic model of the robot is passive ([16]) we can guarantee the passivity of the whole teleoperation system

by ensuring the passivity of the controllers and the communication channel. In this paper, we will implement a position-position teleoperation architecture where we will denote as \bar{u} the torque command computed by the transparency layer and u the torque effectively applied to the robot (i.e. after the passivity layer). As previously mentioned, to apply u the system should have in its tank E enough energy to implement it, otherwise the two-layer algorithm simply doesn't apply the command if no energy is available, or modulates the command to use only the energy available within the tank.

The tank is modelled as the dynamic system

$$\begin{cases} \dot{x}_t = \sigma \frac{D(x) + P^{in}(t)}{x_t} + \frac{-P^{out}(t) + P^u(t)}{x_t} \\ y_t = x_t \end{cases} \quad (4)$$

where x_t is the state of the tank and $E = \frac{1}{2}x_t^2$ is the energy stored in the tank. $D(x)$ is the power dissipated by the system (1), $P^{in}(t) \geq 0$ and $P^{out}(t) \geq 0$ are the incoming and outgoing powers exchanged between the tanks, and $P^u(t)$ is the power dissipated by the current desired command (the output of the transparency layer).

$$P^u = -\bar{u}^T \dot{x} \quad (5)$$

The model (4) describes the energy evolution of the tank: the power dissipated, $D(x)$, and the incoming power $P^{in}(t)$ are stored in the tank whereas $P^{out}(t)$ is the outgoing power. To avoid singularities in (4), an artificial minimum amount of energy in the tank is placed, i.e. $E(x_t(0)) > \epsilon > 0$, by preventing any energy extraction if $E(x_t) \leq \epsilon$. The tank must be also upper bounded to avoid excessive energy accumulation. In fact, in case of highly dissipative environment (i.e. in a teleoperation, human users often keep injecting energy) if the accumulated energy is released in a short period of time may cause safety issue to the operator [5]. The flags σ and $\hat{\sigma}$ are used to control the energy storage

$$\sigma = \begin{cases} 1, & \text{if } E(x_t) < T^{max} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$\hat{\sigma} = \begin{cases} 1, & \text{if } \sigma = 0 \text{ and } P^u(t) < 0 \\ 1, & \sigma = 1 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $T^{max} > 0$ is an application-dependant positive constant. Differently from [14] we introduced the new term $\hat{\sigma}$ in order to correctly upper bound the tank. In fact, in (4) we must take care about the direction of the power related to command input. Since the transparency layer is acting without any knowledge of passivity it can anytime insert or remove energy from the tank. The term $\hat{\sigma}$ defined in (7) differs from the classic upper bound term σ since it enables to store the incoming power either if the tank is full and the summation of the outgoing power $P^{out}(t)$ and input related power $P^u(t)$ is not positive, or if the tank is not full yet. This term allows also to have a faster response. In fact, the dissipated power of $P^u(t)$ at the operator side can be stored at the environment side if the operator robot doesn't need it (i.e. its tank has already reached the upper bound). Finally

the tanks can send or require energy from the tank on the other side of the network. These flows are controlled by terms T^{req} and β

$$E^{req} = \begin{cases} 1, & \text{if } E(x_t) < T^{req} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$\beta = \begin{cases} 1, & \text{if } E(x_t) \geq T^{ava} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where T^{req} and T^{ava} are also application-dependant constants and satisfy the following inequalities $T^{max} \geq T^{ava} \geq T^{req} \geq \epsilon$. The flows $P_{op}^{out}(t)$ and $P_{env}^{out}(t)$ for the operator and environment tanks are defined as follow

$$P_{op}^{out}(t) = (1 - \sigma_{op})D_{op}(x) + E_{env}^{req}(t - d(t))\beta_{op}\bar{P} + (1 - \hat{\sigma}_{op})P_{op}^u(t) \quad (10)$$

$$P_{env}^{out}(t) = (1 - \sigma_{env})D_{env}(x) + E_{op}^{req}(t - d(t))\beta_{env}\bar{P} + (1 - \hat{\sigma}_{env})P_{env}^u(t) \quad (11)$$

$$P_{op}^{in}(t) = P_{env}^{out}(t - d(t))$$

$$P_{env}^{in}(t) = P_{op}^{out}(t - d(t))$$

where $d(t)$ is the communication channel delay (possibly time-variant) and \bar{P} is the rate of change of the energy flow and it is a design parameter. From (10) and (11) we can also see that if the tank is full then the dissipated power $D_{op}(x)$ and $D_{env}(x)$ are added to the outgoing power $P_{op}^{out}(t)$ and $P_{env}^{out}(t)$ respectively.

Proposition 1: The definition of $P_*^{out}(t)$ in (10) and (11) preserves the passivity with respect to the pair $(\tau_{op}, \dot{x}_{op})$ and $(\tau_{env}, \dot{x}_{env})$ even if the system is affected by communication delay.

Proof: By definition $D_*(x)$ and \bar{P} are positive terms and since $\sigma_*, E_*^{req}, \beta_* \in \{0, 1\}$ are positive as well, $P_*^{out}(t)$ is positive if and only if $(1 - \hat{\sigma}_*)P_*^u(t)$ is positive. By (7) $\hat{\sigma}_*$ is zero only if $P_*^u(t)$ is positive: this is the only case in which the input term contributes to $P_*^{out}(t)$. Then we

$$P_*^{out} \geq 0 \quad (12)$$

The total energy stored \mathcal{H} in the system is

$$\mathcal{H} = H_{op}(t) + H_{env}(t) + E_{op}(t) + E_{env}(t) + H^{ch}(t) \quad (13)$$

where $H_{op}(t)$ and $H_{env}(t)$ are the energy associated to (1), $E_{op}(t)$ and $E_{env}(t)$ are the energy stored in the tanks and $H^{ch}(t)$ is the energy flowing in the communication channel. Considering the derivative $\dot{\mathcal{H}}$ of the Hamiltonian \mathcal{H}

$$\dot{\mathcal{H}} = \dot{H}_{op}(t) + \dot{H}_{env}(t) + \dot{E}_{op}(t) + \dot{E}_{env}(t) + \dot{H}^{ch}(t) \quad (14)$$

and using (4) we end up with

$$\begin{aligned} \dot{\mathcal{H}} = & -D_{op}(t) - D_{env}(t) + u_{op}^T \dot{x}_{op} + u_{env}^T \dot{x}_{env} + \\ & + \sigma_{op}(D_{op}(t) + P_{op}^{in}(t)) - P_{op}^{out}(t) + P_{op}^u(t) + \\ & + \sigma_{env}(D_{env}(t) + P_{env}^{in}(t)) - P_{env}^{out}(t) + P_{env}^u(t) + \\ & + \tau_{op}^T \dot{x}_{op} + \tau_{env}^T \dot{x}_{env} + \dot{H}^{ch}(t) \end{aligned} \quad (15)$$

where $P_{op}^u(t) = -u_{op}^T \dot{x}_{op}$ and $P_{env}^u(t) = -u_{env}^T \dot{x}_{env}$ are the powers needed to execute the current robot commands. As

shown in [17] the power flowing through the communication channel is defined as

$$\begin{aligned} \dot{H}^{ch}(t) = & P_{op}^{out}(t) + P_{env}^{out}(t) + \\ & - P_{op}^{out}(t - d(t)) - P_{env}^{out}(t - d(t)) \end{aligned} \quad (16)$$

where $d(t) \geq 0$ is the communication delay. We can rewrite (15) using (11), (10) and (16) as

$$\begin{aligned} \dot{\mathcal{H}} = & -(1 - \sigma_{op})D_{op}(t) - (1 - \sigma_{env})D_{env}(t) + \\ & - (1 - \sigma_{env})P_{op}^{out}(t - d(t)) - (1 - \sigma_{op})P_{env}^{out}(t - d(t)) + \\ & + \tau_{op} \dot{x}_{op} + \tau_{env} \dot{x}_{env} \end{aligned} \quad (17)$$

Since $\sigma_{op}, \sigma_{env}, \hat{\sigma}_{op}, \hat{\sigma}_{env} \in \{0, 1\}$ we can upper bound $\dot{\mathcal{H}}$ by exploiting the inequalities $u_*^T \dot{x} \leq u_*^T \dot{x} + \sigma P_*^{out}(t)$ and $P_*^{out}(t) \geq 0$

$$\dot{\mathcal{H}} \leq \tau_{op}^T \dot{x}_{op} + \tau_{env}^T \dot{x}_{env} \quad (18)$$

which proves the theorem. ■

III. PASSIVE MODEL PREDICTIVE CONTROL

Figure 1 shows the block diagram of the control architecture. The plant model within the P-MPC (operator/environment) consists of the robot dynamics and the tank evolution (4). The passivity (i.e. $E_{op} \geq \epsilon$, $E_{env} \geq \epsilon$) of the system is implemented as a nonlinear constraint in the optimisation problem.

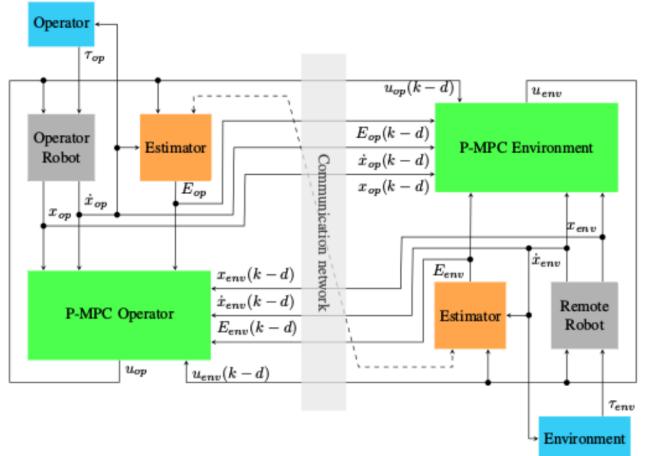


Fig. 1. The control architecture implemented. The green blocks are the controllers, the blue blocks are the sources of external torques, the grey blocks are the robots, the orange blocks are the tank estimators and the dashed line is the communication channel where P_*^{out} and E_*^{req} are exchanged.

A. Robot dynamic

The robot model used in the MPC is the exact linearisation via feedback of the robot dynamic defined in (1) [16]. Each degree of freedom is assumed to be independent of the other and modelled as a mass-damper equation. Let $h(x, u)$ be the update function of the equivalent SISO discrete-time second-order linear system

$$x(k+1) = h(x, u) = \bar{A}x(k) + \bar{B}(u(k)^T, \tau(k)) \quad (19)$$

where the vector x contains the joint position and velocity, u and $\tau \in \mathbb{R}$ are the torque applied by the motor and the external force. This implies that the state space matrices are the discretised version of the differential equation

$$j\ddot{x} + b\dot{x} = u + \tau \quad (20)$$

where $b \geq 0$ is the friction and $j > 0$ is the inertia. The external force τ is assumed to be an exogenous input. We have as many system (19) as degrees of freedom.

B. Tank dynamics

We rewrite (4) in order to express directly the energy stored in the tank to ease the modelling of the update function $g(E, \dot{x}, u)$ as a non linear discrete time system

$$\begin{aligned} E(k+1) = g(E, \dot{x}, u) = E(k) + \Delta t (\sigma(\frac{1}{2}b\dot{x}^2(k) + P^{in}(k)) \\ - P^{out}(k) + P^u(k)) \end{aligned} \quad (21)$$

where $E(k) \in \mathbb{R}^+$ represents the energy stored in the tank at time k , $P^{in}(k)$, $P^{out}(k)$ are the in/out power functions, and Δt is the sampling time of the discrete dynamics. In our formulation both the transparency and passivity layers will be implicitly taken into account within the P-MPC. $P^u(k)$ is defined as the power needed to apply the control input during the optimisation

$$P^u(k) = -u\dot{x} \quad (22)$$

where \dot{x} is the velocity of the robot. The damping term $D(x)$ in (1) in our case corresponds to the friction term and is equal to $\frac{1}{2}b\dot{x}^2$.

C. P-MPC at the operator side

Let $z \in \mathbb{R}^6$ be the state vector of the MPC at the operator side

$$z(k) = \begin{bmatrix} x_{op} \\ \dot{x}_{op} \\ E_{op} \\ x_{env}(k-d(t)) \\ \dot{x}_{env}(k-d(t)) \\ E_{env}(k-d(t)) \end{bmatrix} \quad (23)$$

where $d(t)$ is the communication delay from environment to operator and $u_{op} \in \mathbb{R}$ is the operator robot control command. Let $f_{op}(z, u)$ be the update function of the robots state x_{op} , x_{env} and the tank values E_{op} , E_{env}

$$\begin{aligned} f_{op}(z, u_{op}) = \\ = \begin{bmatrix} h_{op}(z_1(k), u_{op}(k)) \\ g_{op}(z_3(k), z_2(k), u_{op}(k)) \\ h_{env}(z_4(k-d(t)), u_{env}(k-d(t))) \\ g_{env}(z_6(k-d(t)), z_5(k-d(t)), u_{env}(k-d(t))) \end{bmatrix} \end{aligned}$$

which yields to

$$z(k+1) = f_{op}(z, u_{op}) \quad (24)$$

where h is the evolution of the motor defined in (19), and g is the tank dynamics defined in (21). Due to the distributed approach the states about the opposite site and the last

applied command (i.e. the environment side for the operator side and vice-versa) are received after d steps of delay.

Assuming the controlled robot fully observable we can directly recover at each control cycle the state. The operator tank level E_{op} is computed using the observed states $x_{op}(k)$, $\dot{x}_{op}(k)$ and the control input $u_{op}(k)$. The environment tank level $E_{env}(k-d(t))$ is computed using the delayed environment robot states $x_{env}(k-d(t))$, $\dot{x}_{env}(k-d(t))$ and the delayed control input $u_{env}(k-d(t))$. The estimator computes at every control cycle the tank level applying (21). The cost function to be minimised would balance the tracking error between the operator and environment robot states (position and velocity) and moderates the control input \hat{u}_{op} and its rate of change $\Delta \hat{u}_{op}$. The resulting cost function at time k is

$$\begin{aligned} J_{op}(\hat{z}, \hat{u}_{op}) = \sum_{i=1}^{k_p} [q(\hat{x}_{op}(k+i) - \hat{x}_{env}(k-d(t)+i))^2 + \\ + w(\hat{\dot{x}}_{op}(k+i) - \hat{\dot{x}}_{env}(k-d(t)+i))^2] + \\ + \sum_{j=1}^{k_c} [c\hat{u}_{op}(k+j)^2 + d\Delta \hat{u}_{op}(k+j)^2] \end{aligned} \quad (25)$$

where q, w, c, d are application-dependant positive weights, and k_p and k_c are the prediction and control horizons, respectively.

The optimal control inputs $\hat{u}_{op}^*(k), \dots, \hat{u}_{op}^*(k+k_c)$ are obtained as a solution of the finite-horizon optimal control problem

$$\begin{aligned} \hat{u}_{op}^*(k+i)|_{i=0}^{k_c} = \arg \min_{\hat{z}, \hat{u}_{op}} J_{op}(\hat{z}, \hat{u}_{op}) \\ \text{s. t. } \hat{z}(k) = z(k) \\ \hat{z}(k+1) = f_{op}(\hat{z}, u_{op}) \\ \hat{E}_{op} \geq \epsilon_{op} \quad (26) \\ \hat{E}_{env} \geq \epsilon_{env} \quad (27) \end{aligned}$$

where we assume u_{env} constant over the optimisation horizon. As usual in MPC, the applied command to the robot is the first value

$$u_{op}(k) = \hat{u}_{op}^*(k).$$

The inequality constraints (26) and (27) guarantee that the tank levels would be always non negative. In this way, the overall teleoperation system is passive and so stable. Exploiting the prediction behaviour of the P-MPC the controller computes the command always taking into account the time evolution of the tank. In this way, the command is not forced to be zero when the tank is closer to the lower bound. Instead, we look for an optimal solution that will avoid the chattering behaviour of the classic two-layer implementation as shown in Section IV.

D. P-MPC at the environment side

For the environment side, the optimisation problem is the same as the operator side. The state vector $z \in \mathbb{R}^6$ is the

following

$$z(k) = \begin{bmatrix} x_{op}(k - d(t)) \\ \dot{x}_{op}(k - d(t)) \\ E_{op}(k - d(t)) \\ x_{env}(k) \\ \dot{x}_{env}(k) \\ E_{env}(k) \end{bmatrix} \quad (28)$$

where the communication delay is moved from the environment robot state variables to the operator ones. The equations for f_{env} , J_{env} , and the optimisation problem must be changed accordingly with the new definition of z and to the control input for the environment robot u_{env} .

IV. SIMULATION RESULTS

The proposed solution has been initially validated using *Simulink* and the *Optimisation Toolbox* provided by *Matlab R2019b* on 1 degree of freedom robots. As shown in Figure 1, the robot at the operator side is moved by a PD controller, acting as the operator, which takes as input the reference signal and the delayed position and velocity of the environment robot; the output is the torque to be applied at the operator side to track the reference. This feedback loop models the classic video streaming used in a real setup. The P-MPC controllers provide commands both to the operator and environment robots. At the operator side, the command acts as force feedback to the operator, while at the environment side it drives the robot. We implemented the two-layer controller proposed in [14] with position PD controllers in the transparency layer. The passivity layer shares the same tank dynamics.

A. Simulation setup

The test is conducted with a constant delay of 0.1 s between the operator and the environment side. The sampling time of the controller was set to $\Delta t = 0.02$ s, the prediction horizon of the P-MPC equal to 5 steps and the control horizon is 3 steps. The energy thresholds of the two tanks defined in Section II are set equal to small values in order to induce the operator and environment tank towards the upper and the lower bound respectively ($T^{max} = 1.5$, $T^{ava} = 0.6$, $T^{req} = 0.4$, $\epsilon = 0.001$, $\bar{P} = 0.01$) and in this way to validate the proposed architecture. The tanks are initialised to $T^{max}/2$ at the operator side and to ϵ at the environment side. The robot model parameters (i.e. inertia and friction) used during the simulation are identified on the experimental setup and reported in Section V.

B. Soft contact

The experiment is conducted using as a reference signal for the operator a low-passed step signal. The two controllers are tuned to behave similarly when the energy stored in the system is enough to perform the action required. We simulate a soft contact placing an obstacle along the path of the environment robot (at 0.5 rad) so it can't reach the desired set point. Figure 2 shows the P-MPC and the original two-layer algorithm driving the environment robot to track the position and the velocity of the operator robot until the remote robot

interacts with the environment (around 2 s). Figure 2g shows a high frequency chattering in the torque commanded to the robot at the environment side: this is because the environment tank has reached its lower bound, as shown in Figure 2h and Figure 3. Since the passivity layer simply sends zero torque in case of lack of energy the resulting controller *jitters* around the lower bound until the required energy to apply the torque is transmitted by the operator side tank. On the other hand Figure 2c shows a smoother command than Figure 2g when the environment tank reaches the lower bound. In fact, as shown in Figure 2d and Figure 3, the tanks evolution is also smoother since the P-MPC avoids a high rate of change in the command. It's also clear how the MPC avoids the chattering: the position and velocity tracking is less precise when the environment tank reaches the lower bound. In fact, the cost function has to trade off the tracking error and the rate of change of command. However, in this way, we guarantee a smoother control action (i.e. no vibrations on the motors and reflected to the operator).

In Figure 2h and Figure 2d the increasing of the environment tank at the end of the experiment, starting from 4 s, is due to the incoming power \bar{P} from the operator side. In fact, even if the robots do not move, the environment tank is still requesting energy to the operator side since $E_{env} < T_{env}^{req}$ and $E_{op} \geq T_{op}^{ava}$.

V. EXPERIMENTAL RESULTS

The P-MPC has been implemented in C++ using NLOpt to solve the optimisation problem, the solver adopted is the Sequential Least Squares Programming (SLSQP). The overall teleoperation architecture has been implemented using ROS. The experimental setup is composed of two brushless DC motors controlled via current feedback loops by two *Maxon Escon* boards (see Figure 4). The motor's axes are connected through gearboxes and a pair of high precision encoders (4096 readings per rotation) provide position measurements. The operator and environment sides exchange data through a simulated network connection that allows delaying and/or losing packets. The network component queues packets and assigns them a delay value. The communication delays can be constant or randomly time-varying, with the possibility of receiving packets out-of-order.

A. Experimental setup

We conducted two experiments, the first with a constant delay of 0.1 s and the second with a uniform random delay between 0.1 and 0.5 s. The sampling time of the controller was set to 0.02 ms, the prediction horizon of the P-MPC is equal to 5 steps and the control horizon is 3 steps. The threshold values used in the two tanks are: $T^{max} = 1.5$, $T^{ava} = 1.0$, $T^{req} = 0.5$, $\epsilon = 0.001$, $\bar{P} = 0.01$. The tanks are initialised to $T^{max}/2$ for the operator side and $T^{req}/2$ for the environment side.

The inertia and the friction of the motor have been identified using a least square approach and the resulting parameters for (20) are: $j = 0.0266 \text{ kg m}^2$ and $b = 0.0218 \text{ N s}$. The parameters are the same for both motors. Since the

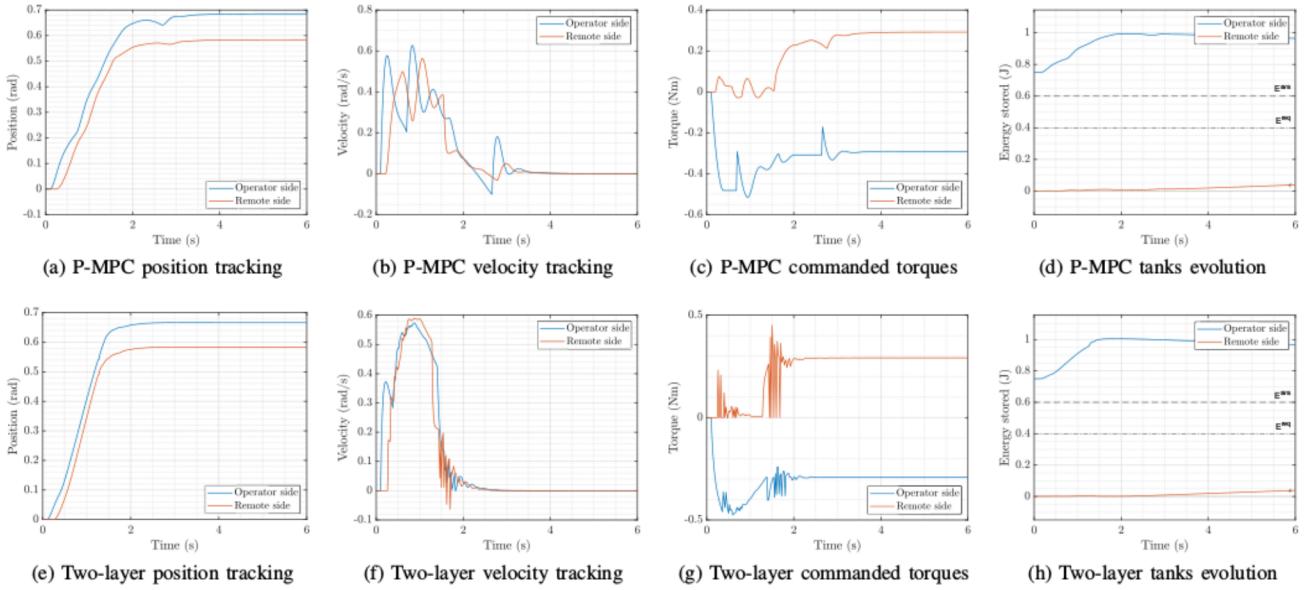


Fig. 2. Simulated soft contact comparison between two-layer and P-MPC approach. The blue and orange lines are the operator and environment robot trajectories respectively. a) P-MPC position tracking, b) P-MPC velocity tracking, c) P-MPC torque commanded, d) P-MPC stored energy, e) two-layer position tracking, f) two-layer velocity tracking, g) two-layer torque commanded, h) two-layer stored energy.

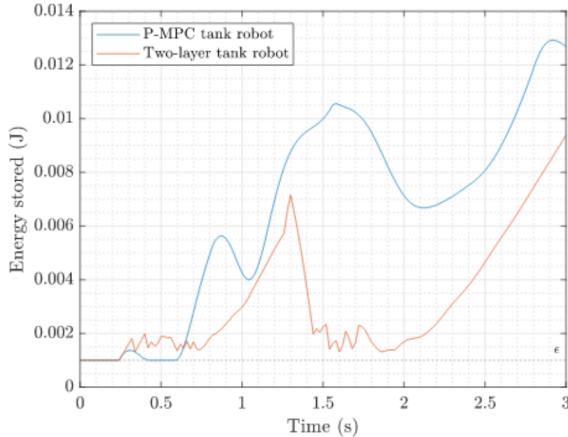


Fig. 3. A magnification of the environment side tank behaviour. The blue and orange lines are the environment tank evolution for the P-MPC and the two-layer approaches respectively. The time axis and the data are the same of Figure 2.

experiments focus only on hard contact, the environment stiffness is assumed to be infinite.

B. Hard contact with constant delay

In the first experiment, the operator moves the robot and the environment motor will touch an obstacle placed along the path. The delay is constant and as it can be seen in Figure 5a and Figure 5b the remote robot goes in contact several times with the environment. The contact position moves a bit during the experiments and this is due to a backlash in the handle attached to the motor.

Figure 5c shows the command computed by the P-MPCs, the operator and the environment torque commanded match

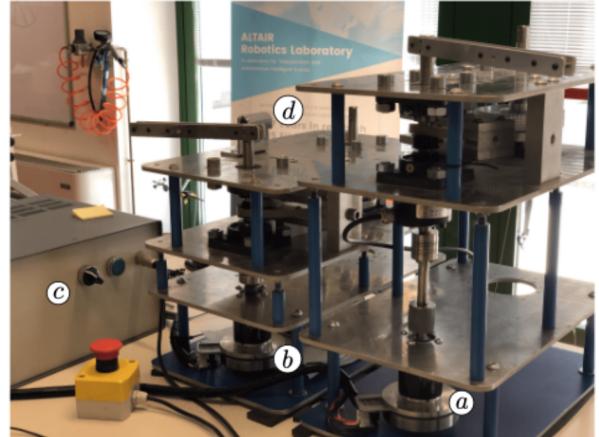


Fig. 4. Experimental teleoperation setup. (a), (b) DC motors with encoder and gearbox, (c) motors controller, (d) environment side obstacle.

during the contacts (of course with opposite signs). Figure 5d shows the evolution of the tanks, as explained in Section IV. The tanks are correctly upper-bounded and the action of the new P^{out} definition can be seen between 7 and 9 s. In that interval the operator tank is full and the power dissipated by the operator is not “wasted” but transferred to the environment side. This causes a faster increase of the available energy for the environment robot allowing a more reactive response.

C. Hard contact with variable delay

In the second scenario, the communication delay changes randomly over time. The operator moves the robot and the motion at the environment side is blocked by an obstacle. Figure 6c shows the command torques: also in this case the

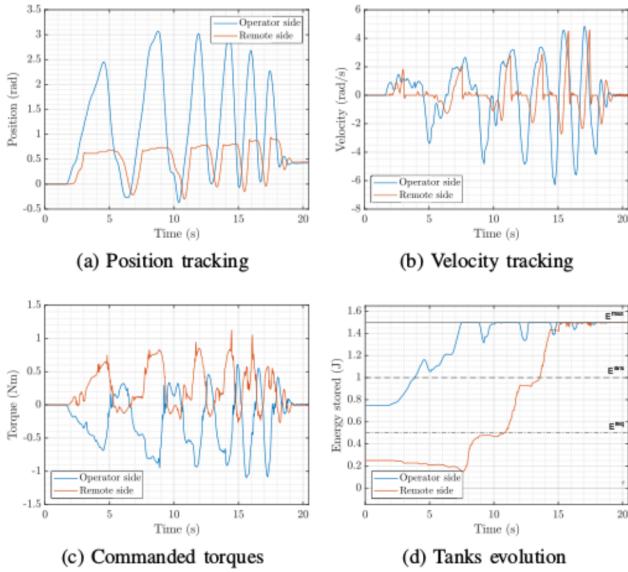


Fig. 5. Hard contact with constant delay P-MPC. The blue and orange lines are the operator and environment robot trajectories respectively. a) position tracking, b) velocity tracking, c) torque commanded, d) stored energy.

interaction with the environment is correctly reflected to the operator. The tanks are initialised with the same values as in the previous experiment and as before between 10 and 13 s the environment tank receives the dissipated power by the operator which helps to have a faster response. As before Figure 6a shows a small movement of the contact position due to the backlash in the robot handle.

VI. CONCLUSION

In this paper, we proposed a novel implementation of the two-layer teleoperation algorithm exploiting a nonlinear passive model predictive controller P-MPC. Such a distributed controller works in a smoother way than the original two-layer algorithm. It can better deal with chattering induced by the hard threshold that controls the time evolution of the tank. Moreover, we introduced a variation on the tank dynamic to cope with the upper bound of the tank allowing the sharing of the dissipated power between the operator and the environment side. We plan to extend our work on multiple degrees of freedom robot.

REFERENCES

- [1] D. A. Lawrence, "Stability and Transparency in Bilateral Teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, 1993.
- [2] R. Muradore and P. Fiorini, "A review of bilateral teleoperation algorithms," *Acta Polytechnica Hungarica*, vol. 13, no. 1, pp. 191–208, 2016.
- [3] G. Niemeyer and J. J. E. Slotine, "Stable Adaptive Teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 152–162, 1991.
- [4] 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.
- [5] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.
- [6] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.
- [7] M. Minelli, F. Ferraguti, N. Piccinelli, R. Muradore, and C. Secchi, "An energy-shared two-layer approach for multi-master-multi-slave bilateral teleoperation systems," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 423–429.
- [8] E. Sartori, C. Tadiello, C. Secchi, and R. Muradore, "Tele-echography using a two-layer teleoperation algorithm with energy scaling," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 1569–1575.
- [9] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 2, Dec 1998, pp. 2133–2138 vol.2.
- [10] S. Sirospour and A. Shahdi, "Model predictive control for transparent teleoperation under communication time delay," *IEEE Transactions on Robotics*, 2006.
- [11] J. Sheng and M. W. Spong, "Model predictive control for bilateral teleoperation systems with time delays," in *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, vol. 4, May 2004, pp. 1877–1880 Vol.4.
- [12] T. Slama, A. Trevisani, D. Aubry, R. Oboe, and F. Kratz, "Experimental analysis of an internet-based bilateral teleoperation system with motion and force scaling using a model predictive controller," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3290–3299, Sep. 2008.
- [13] P. Falugi, "Model predictive control: A passive scheme," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, no. 3, pp. 1017–1022, 2014.
- [14] F. Ferraguti, N. Preda, G. De Rossi, M. Bonfe, R. Muradore, P. Fiorini, and C. Secchi, "A two-layer approach for shared control in semi-autonomous robotic surgery," in *2015 European Control Conference, ECC 2015*, 2015.
- [15] B. Brogliato, R. Lozano, B. Maschke, and O. Egeland, "Dissipative systems analysis and control," *Theory and Applications*, vol. 2, 2007.
- [16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. Springer Publishing Company, Incorporated, 2008.
- [17] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4948–4953.