

An Online Admittance Control for Asymmetric Teleoperated Arm Robot Interacting with Unknown Environment

Adel Mohamed Outayeb*, Farid Ferguene*, Rabah Mellah**, Redouane Toumi*

* *University of Science and Technology Houari Boumediene, Bab Ezzouar, Algiers, Algeria (e-mail: aoutayeb@usthb.dz, fferguene@usthb.dz, rtoumi@usthb.dz)*

** *Faculty of Electrical and Computer Engineering, University of Tizi-Ouzou, Algeria. (e-mail: mellah.rab@gmail.com)*

Abstract: In many telerobotic tasks, robot manipulators interact with unknown environments. Moreover, the manipulators are high-coupled Asymmetric nonlinear multi-degrees of freedom (DOF) systems that present uncertainties and errors in modeling, which may affect safety and performance responses. In this paper, a novel adaptive control scheme is designed to achieve a trajectory and force tracking performance based on a three-channel teleoperation control framework. The adopted approach is based on an admittance control law combined with an inverse dynamics control strategy that avoids the use of force control loop and permits to deal with nonlinear terms. To cope with robots' uncertainties, neural network compensators (NNC) are implemented on both sides. Whereas the integration of weighted recursive least squares (WRLS) estimation method permits the identification of dynamic impedance of an unknown environment. Human in the loop experiment using a real Omni phantom, remote virtual PUMA560 and an environment show the effectiveness of the proposed approach.

Keywords: Admittance control, Environment, Estimation method, Neural network, Teleoperation, Workspace mapping.

1. INTRODUCTION

The considerable development of computer tools and technological evolutions have allowed the evolution of locally controlled systems to remotely controlled systems. Indeed, bilateral teleoperation allows robotic systems to benefit from advanced human problem-solving skills, which are necessary in many scenarios such as performing difficult tasks in inaccessible environments. Today, applications of teleoperation systems can be found in many areas including space technologies, underwater exploration, nuclear/toxic waste handling, surgery, and more recently training and virtual reality (Manocha et al., 2001; Guthart and Salisbury, 2000; Karan et al., 2018).

However, in many teleoperation tasks, robot manipulators interact with unknown environments. Due to the lack of dynamic information about the remote environment, the safety and performance response of such systems is potentially affected. Moreover, the presence of internal disturbances such as the uncertainties in dynamic high coupled robots, human operator, and communication networks of telerobotic systems may degrade the tracking performance and stability of the whole system either in free or constrained space.

Teleoperation control systems are designed for stable and transparent performance. The four-channel control architecture that was firstly proposed by Lawrence (Lawrence, 1993) offers perfect transparency under ideal conditions and compromising goals with robust stability under given communication delays.

In (Hashtrudi and Salcudean, 2002), the effect of local force feedback is evaluated, and ideal transparency conditions are revised. A global transparency analysis of Extended Lawrence

Architecture (ELA) is provided in (Naerum and Hannaford, 2009). In (Sakai et al., 2017), two channel F-P architecture with compliance control is proposed to provide stability in different modes with occurred small position error in the hard environment. However, in a teleoperation system, perfect knowledge of the leader and the follower dynamics may not be available due to the models uncertainties. Moreover, due to the existing time delays in the communication channel and under disturbances due mainly to unknown characteristics of the environment, stability and transparency are significantly compromised (Hashtrudi and Salcudean, 2001).

In the recent years, another alternative for a better trade-off between stability, robustness, and performance is the use of adaptive controllers. This type of control design is deployed to deal with uncertainties in the leader/follower model and operator dynamics. It is also well suited for situations with unstructured or time varying environment dynamics.

Considering linear models for both the leader and follower robots, in (Hashtrudi and Salcudean, 1996) have proposed an indirect adaptive controller that uses position, velocity, and acceleration of master and slave robots, instead of the slave robot force sensing. This architecture provides a good parameter convergence for applications with slow time-varying environments. Considering nonlinear models for the leader and follower, (Ryu and Kwon, 2001) assumed to have uncertainties in both robots. Moreover, by ignoring uncertainties of the operator and environment in the adaptation laws, position and force tracking performance were achieved.

Other adaptive control strategies are developed in the literature, (Love et al., 2004) offered an impedance adaptive controller using the RLS identification method to decrease

operator energy for stability, based on the estimated impedance. In (Liu and Tavakoli, 2011), the authors proposed a four-channel adaptive control scheme for force and position tracking based on the inverse dynamics approach, under the assumption of uncertain dynamics in both robots, operator and environment, and also without considering any communication delay. In (Yang et al., 2019), an adaptive admittance control approach combined with the radial basis function neural-networks (RBFNN) is developed to deal with uncertain robot manipulator interacting with human arms in the absence of a force sensor. According to the Lyapunov theory, a switching adaptive controller is designed to cope with the input saturations and nonhomogeneous models of the human operator and environment (Zhai and Xia, 2016). In (Wang et al., 2017), authors developed an adaptive neural control using radial basis function neural network approximation capabilities to deal with the backlash uncertainties. Under the four-channel architecture framework, guaranteed trajectory and force tracking performance is ensured. The control laws developed is based on adaptive parallel force/position control and inverse dynamic control strategies, under the assumption of uncertainties in follower robot, unknown environment, and noisy force sensor measurement (Adel et al., 2016). Later, in (Mohamed et al., 2017), the authors achieved the same performance by applying an adaptive external force control approach. Park and Lee applied an adaptive fuzzy control and new workspace mapping to improve the tracking performance on the discontinuous trajectory resulting from the mode transformation of the workspace (Park et al., 2020). In (Mellah et al., 2017), the authors proposed adaptive Neuro-fuzzy techniques to enhance teleoperation performance under the assumption of the dynamics uncertainties of leader and follower system. In (Kebria et al., 2020), an adaptive type-2 fuzzy neural-networks controller is developed and verified their effectiveness to learn the nonlinear model and uncertainties. In (Huang et al., 2019), a novel adaptive sliding mode control laws based on RBFNN is proposed to deal with nonlinear models when considering transmission delays and uncertainties.

In this paper, a novel adaptive bilateral controller approach is proposed for position tracking and force applied performance with linear human operator model, uncertain nonlinear leader and follower robots, and unknown linear dynamic of the environment. The considered approach is based on an admittance control law combined with inverse dynamics control strategy incorporated into the three-channel bilateral teleoperation control framework. The main advantages of this contribution are:

- The proposed approach uses only position control laws which is based on an admittance control strategy. It permits to generate a modified position in the direction of the constraint to overcome the absence of the force control law. The parameters of the admittance model (Damper and Stiffness) are obtained by using an online environment identification method based on the WRLS algorithm.
- Integration of the NNC on both the leader and follower controllers to cope with uncertainties due mainly to errors in modeling of robots and effects of end-effector payload.

- The use of Butterworth filter and limit function of boundaries permits to deal with the transient phase of the trajectories generated by the WRLS algorithm.
- We consider asymmetric robots that involve the problem of different scale and workspace mapping.

The reminder of this paper is organized as follows. In section 2, the mathematical formulation of the nonlinear multi-DOF teleoperation system is presented. Section 3 describes the methodology of the workspace mapping for the Omni PHANTOM device with respect to the PUMA560 robot. In Section 4, control laws for both the leader and follower sides using admittance controller and inverse dynamic approach are developed. Additionally, NNC and WRLS identification method are demonstrated in this Section. Human in the loop experiment results using a real Omni interface and a virtual PUMA560 are carried out in Section 5 to show the effectiveness of the proposed control method. Finally, a conclusion is drawn in Section 6.

2. TELEOPERATED SYSTEM DESCRIPTION AND MODELING

In this section, a presentation of a leader/fellow teleoperated system is given in both the sense of intuitive and a mathematical level. The system is composed of real haptic PHANTOM Omni handled by an operator and a virtual PUMA560 robot interacting with a virtual environment, as illustrated in Fig. 1.

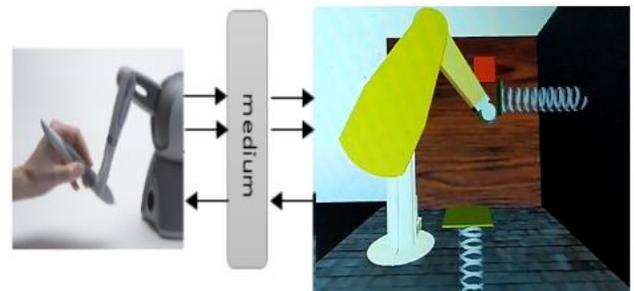


Fig. 1. The Leader/Follower system.

2.1 Human modeling

In general, the dynamic characteristics of a human arm in interaction with a haptic interface can be modeled as a second-order system (Tzafestas et al., 2008). However, in this paper, the human arm impedance is approximately simplified as a spring-damper system, described by the following equation:

$$B_h \dot{X}_l + K_h (X_l - X_h) = F_h \quad (1)$$

where F_h is the force exerted by the human arm on the haptic device, X_h refers to the desired position issued from the central nervous system, whereas X_l and \dot{X}_l represent the position and velocity of the haptic interface, respectively.

2.2 Leader device

The leader device employed here is the PHANTOM Omni, which is a positional and sensing haptic device developed by SensAble Technologies. It is considered under actuated robot, composed of 6 DOF revolute joints that have a reachable workspace of 12cm×16cm. The initial condition corresponds

to the posture shown in Fig. 2 (Silva and al., 2009). The dynamic model of a leader robot in joint space establishes the relationship between the driving torque and joint motion, and it can be written in the following form

$$M_l(q_l)\ddot{q}_l + C_l(q_l, \dot{q}_l)\dot{q}_l + G_l(q_l) = \tau_l + J_l^T F_h \quad (2)$$

where the subscript "l" is used to indicate the leader. $M_l \in R^{6 \times 6}$ is the inertia matrix, $C_l \in R^{6 \times 6}$ is the Coriolis and centrifugal coupling matrix, and $G_l \in R^6$ represent the gravity acting on the joints, while q_l , \dot{q}_l and $\ddot{q}_l \in R^6$ are the joint position, velocity and acceleration, respectively. J_l^T is the transpose of Jacobian matrix. F_h is the force exerted by the human operator and τ_l is the generalized torque control signal acting on joints.

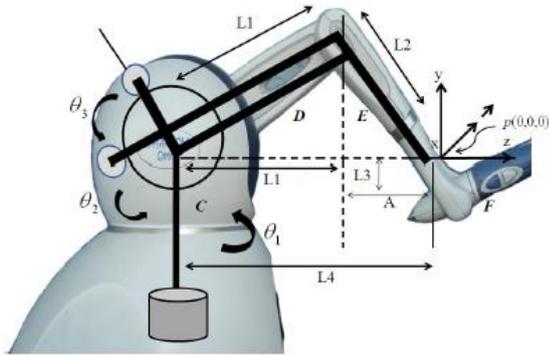


Fig. 2. The initial posture of Omni (Silva and al., 2009).

2.3 Follower device

The virtual teleoperated robot employed here is the PUMA560 robot. It has 6-DoF revolute joints. The first three joints are mounted in the arm which determine the position and the three last joints are mounted on the end effector (spherical wrist) which gives the orientation. The virtual scene is created by using V-realm editor under Matlab/Simulink environment. Table 1 gives the standard Denavit-Hartenberg (DH) parameters considered in Matlab Robotics Toolbox that represents the structure characteristics of the robot.

Table 1. Standard DH parameters of PUMA560.

Link _i (θ_i)	d_i (m)	a_i (m)	α_i (ra)	Range of joint (deg)
1 (θ_1)	0	0	$\pi/2$	-160 to +160
2 (θ_2)	0	0.4318	0	-225 to +45
3 (θ_3)	0.15005	0.0203	$-\pi/2$	-45 to +255
4 (θ_4)	0.4318	0	$\pi/2$	-110 to +170
5 (θ_5)	0	0	$-\pi/2$	-100 to +100
6 (θ_6)	0	0	0	-266 to +266

The dynamics model of a follower robot in joint space could be formulated as follows:

$$M_f(q_f)\ddot{q}_f + C_f(q_f, \dot{q}_f)\dot{q}_f + N_f(\dot{q}_f, q_f) = \tau_f - J_f^T F_e \quad (3)$$

where the subscript 'f' used to indicate the follower, $M_f \in R^{6 \times 6}$ is the inertia matrix, $C_f \in R^{6 \times 6}$ is the Coriolis and centrifugal coupling matrix, and $N_f \in R^6$ represent the gravity and other forces acting on the joints (Coulomb friction, load changes), while q_f , \dot{q}_f and $\ddot{q}_f \in R^6$ are the joint position, velocity and acceleration, respectively. J_f^T is the transpose of Jacobian matrix. F_e is the force exerted by the remote environment and τ_f is the generalized torque control signal acting on joints.

2.4 Environment modeling

Different models have been proposed in the literature, in order to provide a continuous representation of the contact level with a follower robot. In this paper, we use the simplest and the most common model (Fig. 3), which is the dynamic of a linear damper-spring system.

$$F_e = B_e \dot{x}_f + K_e (x_f - x_e) \quad (4)$$

where x_f and x_e are the cartesian position of the follower robot and the contact position with the environment, respectively. K_e and B_e are a (3×3) constant diagonal stiffness and damping matrices, respectively.

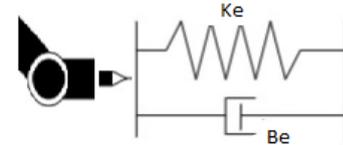


Fig. 3. Damper-spring environment.

2.5 Communication channel

In this work, a delay in the communication channel was neglected and considered very small, which is the case in many applications in which the distance between the follower manipulator and the operator is not too long. Also, in the case where the communication infrastructure is very effective such as fiber optics or 5G.

3. WORKSPACE MAPPING

The workspace of a robot manipulator is defined as the total locus of points at which the end-effector can be placed. Analyzing and mapping the workspace is extremely significant for improving follower robot working reliability and controllability, as it allows overlap between the leader and follower workspaces as much as possible. Before applying the mapping of workspace, it is necessary to adjust the frame axis direction of the PHANTOM Omni device that is different from the frame direction of PUMA560. Also, to correct the shift position of base coordinates of the Omni, the modified transform matrix could be calculated by using the relation below:

$$A' = R_z\left(\frac{\pi}{2}\right) * R_x\left(\frac{\pi}{2}\right) * (Tr_z(L4) * Tr_x(L3) * A) * R_y\left(\frac{\pi}{2}\right) * R_z\left(\frac{\pi}{2}\right) * \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (5)$$

where A is the homogeneous matrix of Omni device that gives the relationship between the position expressed on end-effector frame and base frame that could be formulated as follows:

$$A = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (6)$$

with

$$\begin{aligned} {}^{i-1}T_i &= D_{z_{i-1},d_i} R_{z_{i-1},\theta_i} D_{x_{i-1},a_i} R_{x_{i-1},\alpha_i} \\ &= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i c\alpha_i & a_i c\theta_i \\ c\theta_i & c\theta_i c\alpha_i & -c\theta_i c\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (7)$$

where 'c' indicates the function 'cos' and 's' is for 'sin'. ${}^{i-1}T_i$ is obtained from the product of four basic transformations based on DH parameters (Craig, 2009) using the parameters of link (i) and joint (i).

Moreover, the position related to modified cartesian axis coordinates is given by:

$$X_0' = A' * X_6 \quad (8)$$

where $X_0' = [x_l, y_l, z_l]^T$ represent the modified Cartesian coordinates of the end-effector. X_6 is a Cartesian coordinates of the end effector in the Omni frame.

3.1 Generating approximate workspace

The Monte Carlo method of random sampling is a widely used method for its simplicity since it involves no inverse Jacobian calculation. Especially well adapted for complex robots or even in presence of kinematic redundancy (Zhao and al., 2018). Consequently, in this work, the strategy considered is inspired from (Ju et al., 2014), in which we use homogeneous radial distribution to generate 7000 points to approximate separately the workspaces of the leader/follower robots. Then the cloud point that gives the volume ratio between the leader/follower workspaces is drawn in Fig. 4(a). The result indicates that the Omni workspace is of the order of one-tenth from the PUMA560 workspace.

As emphasized above, the aim is to let the workspace of each other overlap as much as possible, without being near to boundaries as it may cause singularity configurations. Toward this end, a simple method for mapping process of a haptic interface is employed, which includes scaling factors, rotational factors and translations terms that are given by:

$$\begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} = \begin{bmatrix} \cos\delta & -\sin\delta & 0 \\ \sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \left(\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \right) \quad (9)$$

where $[x_l, y_l, z_l]^T$ and $[x_f, y_f, z_f]^T$ represent the cartesian coordinates of the end effector of Omni and PUMA560 respectively. $[S_x, S_y, S_z]^T$ and $[T_x, T_y, T_z]^T$ represent the scaling factors, and translations terms about the X, Y and Z axes, whereas δ is the rotation angle of a Z axis for the base of a leader device. Thus, we get the matching parameters for the PUMA560 robot as: $[S_x, S_y, S_z]^T = [2.8, 2.9, 4.3]^T$, $[T_x, T_y, T_z]^T = [0, 0, 0]^T$ and $\delta = 0$. Hence, the results are illustrated in Fig. 4(b) and indicate the overlapped area after the mapping process obtained without being near to the boundaries that may cause singularity configuration.

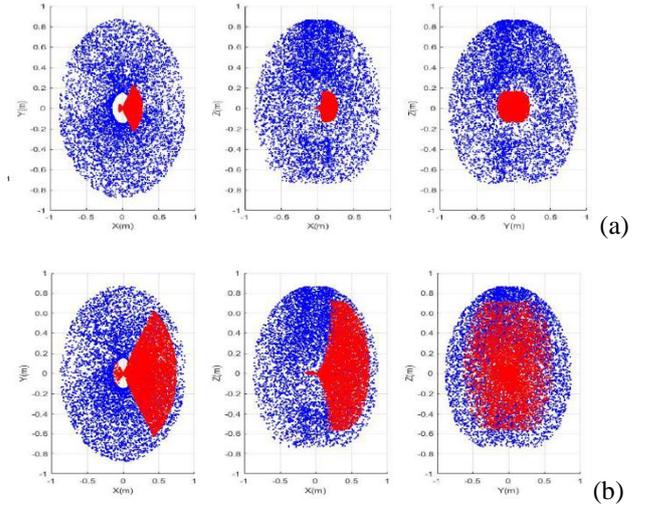


Fig. 4. (a) cloud point before workspace mapping, (b) cloud point after workspace mapping.

4. CONTROL STRATEGY

In this section, an adaptive online admittance control approach is presented for the follower device. The approach avoids the use of a force control loop and uses the WRLS method and boundaries function to generate modified trajectories that will be fed to proportional-integral-derivative PID position control law. Moreover, the position control laws in both leader and follower devices are combined with the nonlinear dynamic decoupling approach and integrated NNCs to cope with uncertainties raised by errors in modeling and effects of the end-effector payload. The scheme is based on three-channel architecture, where the force applied by the operator, the position of the leader robot after being converted by workspace mapping is transmitted to the follower side, and the position of the follower robot after being processed by inverse workspace mapping is reflected to the leader.

4.1 Leader controller

The human operator manipulates the haptic interface by applying a given force that is dependent on the voluntary motion desired. the leader output trajectory is converted into the desired trajectory through workspace mapping. Then all signals, force, and generated trajectory are measured and sent to the slave robot. On the other hand, the reflected trajectory of the follower robot after being processed by the inverse function of workspace mapping is used as inputs to subsequent torque controller. This signal is useful in producing a haptic sensation when interacting with an environment and also to correct the errors in positioning between the leader and follower robots. The adopted controller system is based on a classic PID (Oudjida et al., 2014) controller combined with the nonlinear dynamic decoupling approach and NNC based on feedback error learning control. Fig. 5(a) shows the block diagram of the leader controller.

Define the model of position error as

$$e_{p,l} = q_l - q_f \quad (10)$$

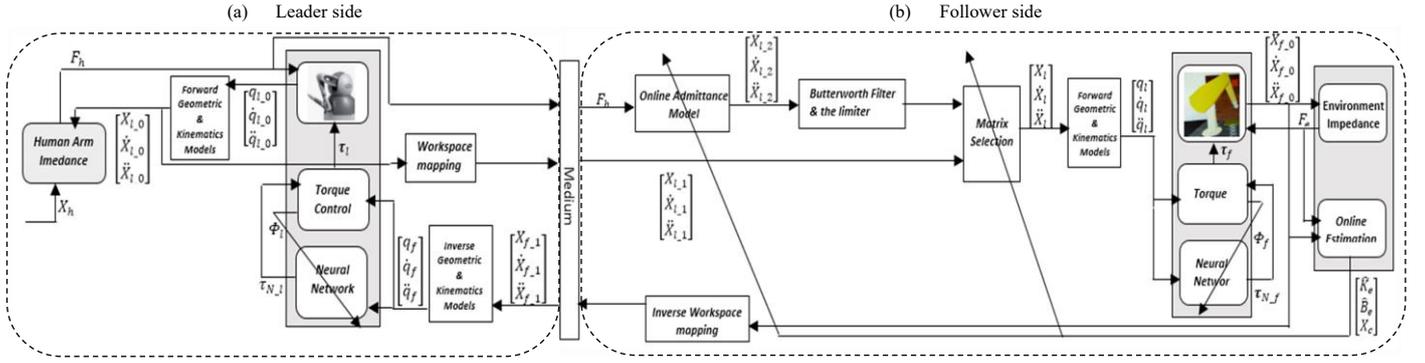


Fig. 5. Control diagram of whole teleoperated system.

Considering PID control law as

$$\tau_{PID,l} = \ddot{q}_f + K_{p,l} e_{p,l} + K_{i,l} \int e_{p,l} dt + K_{v,l} \dot{e}_{p,l} \quad (11)$$

where \ddot{q}_f is the desired acceleration joints of the follower robot, $K_{p,l}$, $K_{v,l}$ and $K_{i,l}$ are diagonal matrices with constants positives parameters chosen to approach zero asymptotically error $e_{p,l}$ between the desired and the actual endpoint positions. The controller law governing the system is expressed as

$$\begin{aligned} \tau_l = \widehat{M}_l(q_l) \tau_{PID,l} + \widehat{C}_l(q_l, \dot{q}_l) \dot{q}_l + \widehat{G}_l(q_l) + J_l^T F_h - \\ K_{NL,p} \tau_{N,l} - K_{NL,i} \int \tau_{N,l} dt \end{aligned} \quad (12)$$

where \widehat{M}_l , \widehat{C}_l , \widehat{G}_l denote the nominal and available terms of the dynamic model of Omni, F_h is the available force applied by the operator, J_l^T is the Jacobian transpose, $K_{NL,p}$ and $K_{NL,i}$ are diagonal matrices with positives parameters and $\tau_{N,l}$ is the output of the NNC. Consequently, the closed loop control law is given as

$$\begin{aligned} \ddot{e}_{p,l} + K_{v,l} \dot{e}_{p,l} + K_{p,l} e_{p,l} + K_{i,l} \int e_{p,l} dt \\ = \widehat{M}_l^{-1} (\Delta M_l \ddot{q}_l + \Delta C_l(q_l, \dot{q}_l) \dot{q}_l + \Delta G_l(q_l) + J_f^T \Delta F_h) \\ - K_{NL,p} \tau_{N,l} - K_{NL,i} \int \tau_{N,l} dt \end{aligned} \quad (13)$$

where $\Delta(\cdot)$ denotes the uncertainty terms.

The objective of the NNC compensator output $\tau_{N,l}$ is to minimize the uncertainties in (13). It is done by taking the desired trajectory as the input signal and the output of a stabilizing feedback controller Φ_l as a training signal for the neural network.

$$\Phi_l = \ddot{e}_{p,l} + K_{v,l} \dot{e}_{p,l} + K_{p,l} e_{p,l} + K_{i,l} \int e_{p,l} dt \quad (14)$$

In the ideal case, the right-hand side of (13) becomes zero and the output of the neural compensator is required to be

$$\begin{aligned} K_{NL,p} \tau_{N,l} + K_{NL,i} \int \tau_{N,l} dt = \widehat{M}_l^{-1} (\Delta M_l \ddot{q}_l \\ + \Delta C_l(q_l, \dot{q}_l) \dot{q}_l + \Delta G_l(q_l) + J_l^T \Delta F_h) \end{aligned} \quad (15)$$

as a result, the NNC realizes a nonlinear mapping between q_l , \dot{q}_l , \ddot{q}_l and F_h to $\tau_{N,l}$.

$$f(u_j) = \frac{1 - \exp^{-u_j}}{1 + \exp^{-u_j}} \quad (16)$$

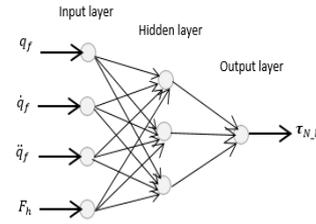


Fig. 6. Structure of NNC.

The inputs u_i in each layer are connected through weights, and the output of each layer is obtained by the following equation

$$y = f\left(\sum_{i=1}^n u_i w_i + w_0\right) \quad (17)$$

where w_i are the weights of connections from the preceding layer to this output, w_0 is the bias parameter. The function $f(\cdot)$ is called the activation function where the type is depending

on the problem. Therefore, the input-output relationship of the network is expressed as

$$\tau_{N,l} = \sum_{j=1}^{N_c} W_{jk}^2 \left(\frac{1 - \exp^{-\left(\sum_{i=1}^{N_e} u_i W_{ij}^1 + b_j^1\right)}}{1 + \exp^{-\left(\sum_{i=1}^{N_e} u_i W_{ij}^1 + b_j^1\right)}} \right) + b_k^2 \quad (18)$$

where N_e , N_c present the number of neurons in input and hidden layers respectively. W_{ij}^1 , W_{jk}^2 denote the weights of

connections from the input layer to the hidden layer and connections from the hidden layer to the output layer. b_j^1 is the bias of the j -th neuron in the hidden layer and b_k^2 is the bias of the k -th neuron in the output layer. Training proceeds with the back-propagation algorithm of the gradient (Mohamed et al., 2017), which is based on the minimization of candidate function of quadratic form calculated as

$$J = \frac{1}{2} E^T E \quad (19)$$

with $E = \Phi_l - \tau_{N,l}$

This algorithm consists of adjusting the output of Neural Network $\tau_{N,l}$ in the opposite direction to the derivative function of the error, which yields a gradient of J as :

$$\frac{\delta J}{\delta W} = \frac{\delta E^T}{\delta W} E = -\frac{\delta \tau_{N_l}^T}{\delta W} E \quad (20)$$

Therefore, the weight adaptation law is updated by the following equation at every sampling period.

$$\Delta W(t) = \eta \frac{\delta \tau_{N_l}^T}{\delta W} E + \mu \Delta W(t-1) \quad (21)$$

where η is called the learning rate and μ is the momentum coefficient. Finally, the resulting weight adaptation coefficients can be derived by making use of (20).

$$\Delta W_{ij}^1(t) = \eta 0.5(1 - Y_j^1)^2 u_i [\sum_{k=1}^n E_k W_{jk}^2] + \mu \Delta W_{ij}^1(t-1) \quad (22)$$

$$\Delta W_{jk}^2(t) = \eta E_k Y_j^1 + \mu \Delta W_{jk}^2(t-1) \quad (23)$$

$$\Delta b_j^1(t) = \eta 0.5(1 - Y_j^1)^2 [\sum_{k=1}^n E_k W_{jk}^2] + \mu \Delta b_j^1(t-1) \quad (24)$$

$$\Delta b_k^2(t) = \eta E_k + \mu \Delta b_k^2(t-1) \quad (25)$$

4.2 Follower Controller

Fig. 5(b) shows the block diagram of the follower side, the control system consists of a PID controller that is mounted on the output of Selection Matrix, which gives the decision about the type of desired trajectories input signals to be tracked, according to each of a direction X, Y and Z. The input to Selection Matrix is composed of two vectors: one indicates the desired trajectories position, velocity and acceleration issued from the online admittance model. The other vector is the desired trajectories of position, velocity and acceleration issued from the interaction of human operator on the Omni interface after having been converted by the mapping process. In the following, a proposed adaptive admittance model and online parameter estimation method of an unknown environment are presented.

4.2.1 Adaptive admittance model

Admittance control is developed to let a robot acts like specified admittance model. In our case, the admittance model is selected in adaptive fashion under the following form:

$$F_h = \widehat{B}_e \dot{x}_r + \widehat{K}_e x_r \quad (26)$$

where \widehat{B}_e , \widehat{K}_e are the estimated damping and stiffness matrices, respectively issued from an online environment estimation. The function of the admittance model is to generate desired robot trajectories in constrained space, which are passed through a Butterworth filter and limiter function, then it passes through inverse geometric and kinematics models to obtain equivalent trajectories in joint space as needed to subsequent control torque unit.

4.2.2 Butterworth filter and boundaries function

To suppress the jitter signals at the output of the admittance model, third-order low-pass Butterworth filters is used to adjust generating desired trajectories when passing the transient phase of an online estimation of the environment parameters. Hence, the transfer function can be expressed as follows

$$H(s) = \frac{1}{\left(\frac{s}{w_c}\right)^3 + 2\left(\frac{s}{w_c}\right)^2 + 2\left(\frac{s}{w_c}\right) + 1} \quad (27)$$

where $w_c = 8Hz$ is the cutoff frequency.

This choice is made as part of general information states that teleoperated system is mainly distributed in low frequency bandwidth, while noises are in high frequencies, also as part of work in (Geng et al., 2020), in which is concluded that a good position tracking is localized between 2Hz and 8Hz frequencies. Hence, the information above this limit should be suppressed.

The limiter function aims to limit the upper and lower boundaries of transient trajectories generated by the adaptive admittance model in the constrained space. High and low limits are chosen to avoid excessive contact when interacting with a rigid environment. These limits are given as logic conditions in the form:

- If transient position among the X direction exceeds a given constant limit, the penetration beyond this value is not considered and fixed within the upper limit. Furthermore, if the transient position among the X direction is negative, the position lower limit is fixed to zero. In similar manner, we consider the limit boundaries of Y and Z axis of both position and velocity transient signals.

To design a control torque that makes the follower (3) dynamics behave like the proposed adaptive admittance model (26). The nonlinear dynamic decoupling approach can be adopted, based on a conventional PID controller that is widely used to achieve complex position tracking.

Define the model of position error as

$$e_{p,f} = q_l - q_f \quad (28)$$

Consider PID control law as

$$\tau_{PID,f} = \ddot{q}_l + K_{p,f} e_{p,f} + K_{i,f} \int e_{p,f} dt + K_{v,f} \dot{e}_{p,f} \quad (29)$$

where \ddot{q}_l is the desired acceleration joint, $K_{p,f}$, $K_{v,f}$ and $K_{i,f}$ are diagonal matrices with constants parameters for the follower, the controller law governed the system is expressed as

$$\begin{aligned} \tau_f = & \widehat{M}_f(q_f) \tau_{PID,f} + \widehat{C}_f(q_f, \dot{q}_f) \dot{q}_f + \widehat{N}_f(\dot{q}_f, q_f) \\ & + J_f^T \widehat{F}_e - K_{Nf,p} \tau_{N,f} - K_{Nf,i} \int \tau_{N,f} dt \end{aligned} \quad (30)$$

where \widehat{M}_f , \widehat{C}_f , \widehat{N}_f denote the nominal and available terms of the dynamic model of PUMA560, \widehat{F}_e is the force sensor measurement, J_f^T is the Jacobian transpose, and $\tau_{N,f}$ is the output of NNC. Consequently, the closed loop control law is given by

$$\begin{aligned} \ddot{e}_{p,f} + K_{v,f} \dot{e}_{p,f} + K_{p,f} e_{p,f} + K_{i,f} \int e_{p,f} dt = & \widehat{M}_f^{-1} * \\ (\Delta M_f \ddot{q}_f + \Delta C_f(q_f, \dot{q}_f) \dot{q}_f + \Delta N_f(\dot{q}_f, q_f) + J_f^T \Delta F_e) - & \\ K_{Nf,p} \tau_{N,f} - K_{Nf,i} \int \tau_{N,f} dt \end{aligned} \quad (31)$$

By applying the same strategy as before, the NNC here is minimizing the uncertainties in (31). It is done by taking the desired trajectory as the input signal and the output of a

stabilizing feedback controller Φ_f as a training signal for the neural network

$$\Phi_f = \ddot{e}_{p_f} + K_{v_f} \dot{e}_{p_f} + K_{p_f} e_{p_f} + K_{i_f} \int e_{p_f} dt \quad (32)$$

In the ideal case, the right-hand side of (31) becomes zero, and the output of the NNC is required to be

$$K_{Nf_p} \tau_{Nf} + K_{Nf_i} \int \tau_{Nf} dt = \bar{M}_f^{-1} (\Delta M_f \ddot{q}_f + \Delta C_f(q_f, \dot{q}_f) \dot{q}_f + \Delta N_f(\dot{q}_f, q_f) + J_f^T \Delta F_e) \quad (33)$$

By applying the same procedure in the follower controller, the NNC realizes a nonlinear mapping between q_p, \dot{q}_p, q_l and F_e to τ_{Nf} .

4.2.3 Online environment estimation

In this paper, we use an online estimation technique based on the WRLS method to identify unknown damping and stiffness parameters of the environment model, which was described in equation (4). The technique considered tracks and identifies the location and dynamic characteristics of any constraints in a robot workspace.

Let us define a regression vector $\varphi(t) = [X(t), \dot{X}(t)]^T$, and a parameters vector $\theta = [K, B]^T$. The linear equation with real parameters (4) is written as:

$$f(t) = \varphi(t)^T \theta + v(t) \quad (34)$$

with

$$\varphi(t) = \begin{bmatrix} X_f(t), X_f(t-1), \dots, X_f(t-k) \\ \dot{X}_f(t), \dot{X}_f(t-1), \dots, \dot{X}_f(t-k) \end{bmatrix}^T$$

$$\theta(t) = \begin{bmatrix} K_e(0), K_e(1), \dots, K_e(k) \\ B_e(0), B_e(1), \dots, B_e(k) \end{bmatrix}$$

where index k denotes the number of measurements, $v(t)$ is measurement noise at time t assumed to be sequence independent with zero mean and constant variance $v \sim N(0, V_f)$, and the equation of error in parameters estimation can be expressed by

$$E(t) = f(t) - \varphi(t)^T \hat{\theta}(t) \quad (35)$$

Considering the cost function is given in quadratic form

$$V(\hat{\theta}, t) = \frac{1}{2} \sum_{i=1}^t [f(i) - \varphi(i)^T \hat{\theta}(t)]^2 = \frac{1}{2} \sum_{i=1}^t E^T(t) E(t) \quad (36)$$

The objective here is to obtain the optimal solution for the parameter minimizing the cost function so that it can be calculated by differentiating it as

$$\frac{\delta V}{\delta \hat{\theta}} = 0 \quad (37)$$

Therefore, the minimum is given by

$$\hat{\theta} = [\varphi(t)^T \varphi(t)]^{-1} \varphi(t)^T f(t) \quad (38)$$

Then, the following covariance matrix is defined as

$$P(t) = [\varphi(t)^T \varphi(t)]^{-1} \quad (39)$$

Using a Matrix Inversion Lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (40)$$

where

$$A = P^{-1}(t-1), B = \varphi(t), C = I, D = \varphi(t)^T \quad (41)$$

By applying the details derivative method [Ljung, 1999], we get the recursive equations rather than the batch mode equations calculated in (38).

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t) (f(t) - \varphi^T(t) \hat{\theta}(t-1)) \quad (42)$$

with

$$L(t) = P(t) \varphi(t) = \frac{p(t-1) \varphi(t)}{1 + \varphi^T(t) p(t-1) \varphi(t)} \quad (43)$$

$$p(k) = p(t-1) (I - L(t) \varphi^T(t)) \quad (44)$$

Furthermore, in order to discard older data in favor of recent data that permits to track the variation in parameters, the concept of forgetting factor is established, and the equations (36) and (42) can be rewritten as:

$$V(\hat{\theta}, t) = \frac{1}{2} \sum_{i=1}^t \lambda^{t-i} [f(i) - \varphi^T(i) \hat{\theta}(t)]^2 \quad (45)$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t) (f(t) - \varphi^T(t) \hat{\theta}(t-1)) \quad (46)$$

where λ is called the forgetting factor, with :

$$L(t) = P(t) \varphi(t) = \frac{p(t-1) \varphi(t)}{\lambda + \varphi^T(t) p(t-1) \varphi(t)} \quad (47)$$

$$p(k) = \lambda^{-1} p(t-1) (I - L(t) \varphi^T(t)) \quad (48)$$

5. HUMAN IN THE LOOP EXPERIMENT

To evaluate the efficiency of the proposed method, two dissimilar robots are considered in the sense that they are geometrically asymmetric with different scales. The leader is a real robot (Omni) whereas the follower is a virtual PUMA560 robot whose parameters are taken from a real PUMA560 arm [Armstrong et al., 1986], which is interacting virtually with an unknown environment. Both virtual models are illustrated in Fig.7 which are built using V-realm editor under Simulink/ Matlab environment. In this paper, we assume that only the first three links of Omni and PUMA560 are considered. The environment is composed of three objects suspended separately over three directions X, Y, and Z. Accordingly, the environment torque is exerted at the end effector only in one direction for each object. On the assumption of an unknown dynamic of environment objects that can be defined as

$$\text{Object (1): } F_e = 10 \dot{x}_f + 500(x_f - 0.60) \quad (49)$$

$$\text{Object (2): } F_e = 900(y_f - 0.23) \quad (50)$$

$$\text{Object (3): } F_e = 20 \dot{z}_f + 2000(z_f - (-0.284)) \quad (51)$$

In order to calculate an online estimation of their parameters

(Stiffness and damping), we apply a WRLS method with $P(0) = 10000$, $\hat{\theta}(0) = 0$, and $\lambda = 0.90$. The matrices $\hat{M}_l(q_l)$, $\hat{C}_l(q_l, \dot{q}_l)$ and $\hat{G}_l(q_l)$ for the leader robot and $\hat{M}_f(q_f)$, $\hat{C}_f(q_f, \dot{q}_f)$, $\hat{N}_f(\dot{q}_f, q_f)$ for the follower, are the known dynamic parameters forming the nominal Models of the robots.

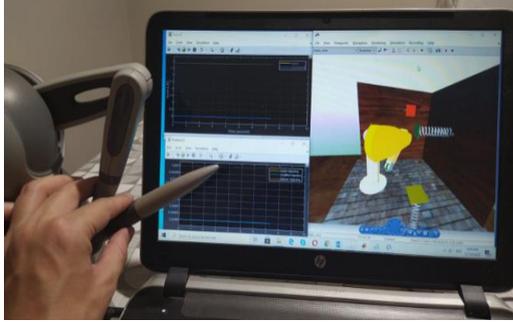


Fig. 7. Virtual scene of follower side.

Modeling uncertainties include 5 Kg Mechanical tool attached to the end-effector of follower robot, Coulomb friction, and viscous friction torques $\tau_f(\dot{q})$ added to each joint where $\tau_f = 0.8\dot{q} + 0.5\text{sign}(\dot{q})$. The NNC used on the leader side controller is composed of 12 neurons in the input layer and 9 neurons in the hidden layer. The backpropagation algorithm parameters for the leader side are: $\eta = 0.008$, $\mu = 0$, $K_{Nl,p} = 5 \times I$, $K_{Nl,i} = 0.07 \times I$. Whereas the NNC in the follower side kept the same number of neurons in each layer and the backpropagation algorithm parameters are: $\eta = 0.008$, $\mu = 0$, $K_{Nf,p} = 1.5 \times I$, $K_{Nf,i} = 0.01 \times I$. Initially, weights are randomly selected and then are adjusted every sampling time. The gains of human impedance and controller laws on each side are fixed experimentally and respectively as

$$\begin{aligned} K_h &= 160 \times I, & B_h &= 5 \times I \\ K_{p,l} &= 400 \times I, & K_{v,l} &= 0.1 \times I, & K_{i,l} &= 0 \times I \\ K_{p,f} &= 800 \times I, & K_{v,f} &= 40 \times I, & K_{i,l} &= 0 \times I \end{aligned}$$

In the beginning, the robots are situated at different positions, depending on the last configuration setting. Then, the operator starts moving the virtual PUMA560 in free space by applying a variable force F_h on the haptic interface, until it reaches the object to interact with. At that time, the operator keeps the same level of force intensity or decides to increase or decrease the applied force and that by pushing the integrated buttons on the Omni link according to the task to be performed and the direction of the interaction. In our case, we considered an arbitrary force profile for each repeated experience. And lastly, the operator exits the object under interaction and returns to the free space. Moreover, we repeated the experiment under different conditions by taking advantage of NNC integrated into the controllers with the presence of perturbations on the follower robot and without perturbations (ideal case).

The obtained results consider the interaction with Object (1) and they are classified into two sets. The first set illustrated in Fig. 8 considers the ideal case. It presents the tracking trajectory among three axes (a), the performance of applied force during the direction of interaction (b), the trajectory

errors that occurred during the interaction and among X,Y,Z directions (c), the force errors during interaction (d), and finally, the quality of estimated impedance of the unknown environment (e). Whereas, the second set illustrated in Fig. 9 takes into consideration the presence of perturbations, which are organized in three Subsets (A), (B), and (C), and that to show clearly the effects of integrated NNC separately and jointly in leader and follower controllers.

The results show that the proposed adaptive controller under ideal conditions has less errors than the others either for trajectory tracking or compared force applied Fig. 8 (c), (d). Thus, the responses converge to a small neighborhood of zero, even though a presence of applied force delay. These errors occur when a human develops higher or lower force intensity Fig. 8 (b) and, consequently, they are reflected as errors in tracking trajectory that does not exceed 0.01m.

In terms of interpreting results of tracking trajectory, Fig. 8(a) shows that the Omni and modified trajectories are completely identical in free space, which is not the case for the constrained space. This difference is due to the role of the admittance model and matrix selector in giving priority to the force applied to the Omni than to the tracking of its trajectory. Therefore, it determines the depth of penetration that should be taken into the object. However, we can point out here that they are very close and the errors are still within the limits as illustrated in Fig. 8(c). Moreover, in Fig. 8(e), results indicate the quality of estimated impedance against real impedance in both free and constrained space, under the assumption of unknown characteristics of the object.

Considering the case where the perturbations are included, Fig. 9(A) shows the feasibility of integrating NNC on both sides to improve the performance of the PID controller and therefore obtain results much closer to the ideal case.

To highlight the potential usefulness of this concept and to conclude the effect of their absence, we have separately integrated into the controllers of the leader and follower sides and drawn results as given in Fig. 9(B) and (C). In subset (B), the responses show the efficiency of integrated NNC in compensating the tracking errors between the leader and follower trajectories when interacting with an object (a). On the other hand, the negative effect of its absence in the follower side causes failure in replicating the human force on the object and permanent error on it (b). Whereas in the subset (C), which represents the opposite case, it shows the efficiency of integrated NNC in follower controller to gain accuracy in replicating human force on the object as illustrated in (b). On the other hand, the negative effect of its absence in the leader controller generates a significant shift between trajectories of both sides, especially when interacting with the object (a), (c).

6. CONCLUSIONS

In this paper, a novel approach of adaptive admittance control laws combined with inverse dynamic strategy is designed for asymmetric nonlinear teleoperated arm robots, subject to dynamic uncertainties and interacting with an unknown environment. The proposed scheme is incorporated into the three-channel teleoperation control framework and uses only position control laws.

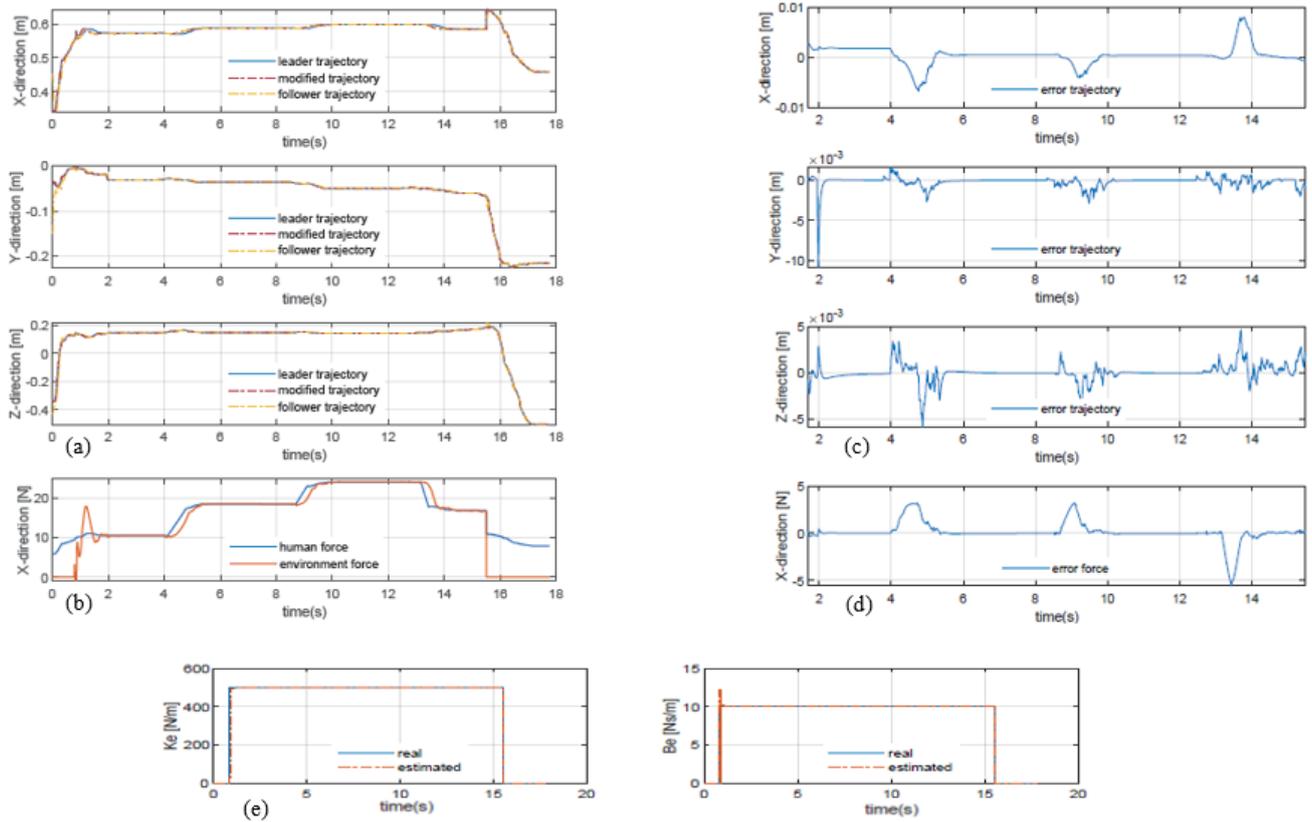
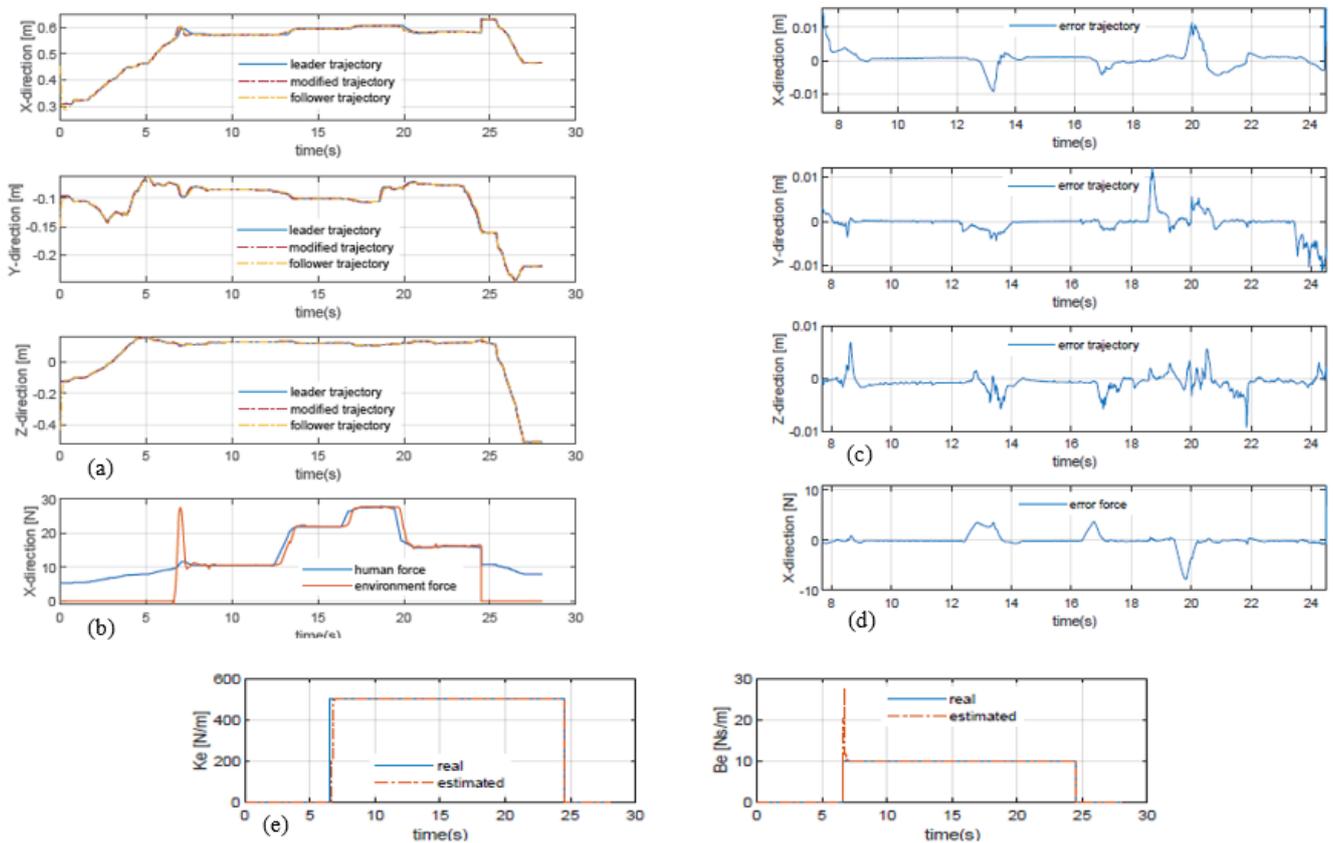
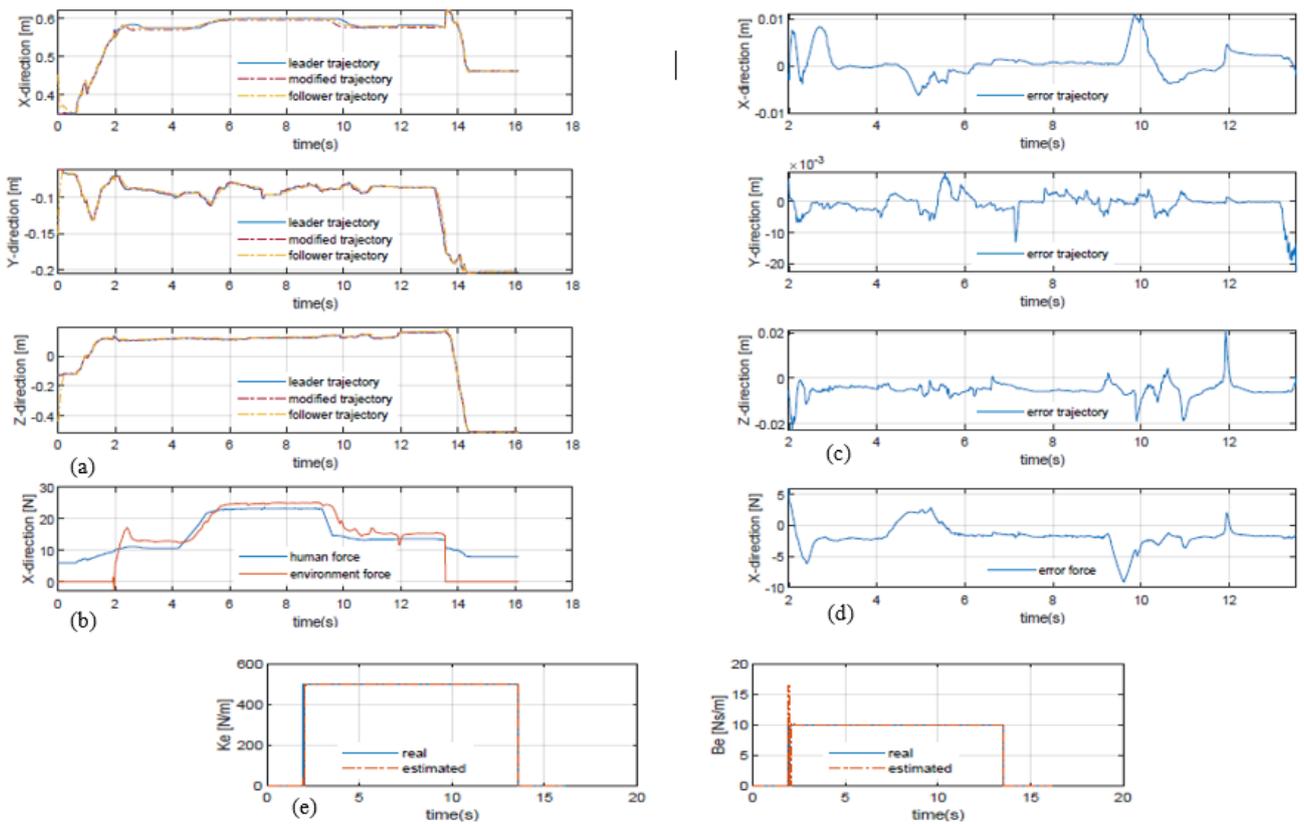


Fig. 8. Results without perturbations. (a) : tracking trajectory among X,Y,Z directions (b) : performance of applied force among X direction (c) : trajectory errors during the phase of the interaction and among X,Y,Z directions (d) : force errors during interaction (e) : reel against estimated impedance of unknown environment.

Subset (A)



Subset (B)



Subset (C)

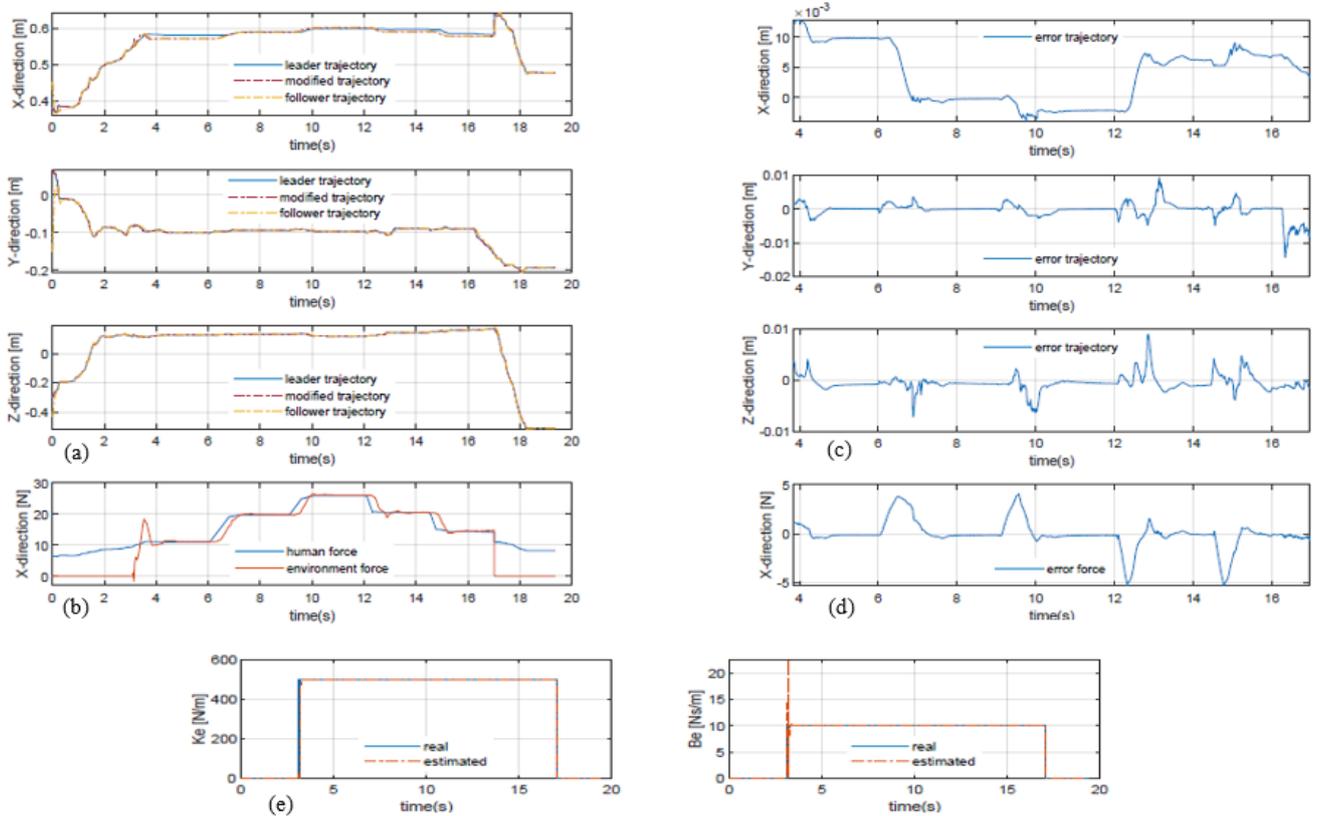


Fig. 9. Results with perturbations: **subset (A)**: implemented NNC in both leader and follower controllers **subset (B)**: implemented NNC in leader controller **subset (C)**: implemented NNC in follower controller

The follower robot controller is based on an admittance control strategy that permits the generation of a modified position in the constraint direction, and therefore, it overcomes the absence of the force control law. In order to interact with an unknown environment, the desired admittance model is obtained by online estimation of environment impedance parameters based on the WRLS method. Additionally, the NNCs are developed and added as auxiliary controllers to the leader and follower dynamic controllers that act as compensators of uncertainties effects on position and force tracking, raised by errors in modeling and effects of the end-effector payload. To verify the effectiveness of our proposed approach, a comparison of human-in-the-loop experiments interacting with a real Omni haptic interface and virtual remote robot and environment have been performed. The results illustrate that the whole system is stable. Moreover, position tracking errors are guaranteed to be ideally small in both free motion and situations when force contact is established, so satisfactory performance of position tracking and forces applied between the leader and follower robots.

For future works, we propose to test and validate our proposed approach by using a real remote robot, as well as developing other advanced control approaches to compare performances.

REFERENCES

- Adel, O., Farid, F. and Toumi, R. (2016). "Bilateral control of nonlinear teleoperation system using parallel force/position control approach and online environment estimation," in *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 1110-1115.
- Armstrong, B., Khatib, O. and Burdick, J. (1986). "The explicit dynamic model and inertial parameters of the PUMA 560 arm," in *Proceedings of IEEE international conference on robotics and automation* pp. 510-518.
- Craig, J.J. 2009. "Introduction to robotics: mechanics and control," 3/E: Pearson Education India.
- Geng, C., Xie, Q., Chen, L., Li, A. and Qin, B. (2020). "Study and Analysis of a Remote Robot-assisted Ultrasound Imaging System," in *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 389-393.
- Guthart, G. and Salisbury, J. (2000). "The intuitive telesurgery system: overview and application" In *Proceedings of IEEE International Conference on Robots and Automation*. San Francisco.
- Hashtrudi-Zaad, K. and Salcudean, S. E. (1996). "Adaptive transparent impedance reflecting teleoperation," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1369-1374.
- Hashtrudi-Zaad, K. and Salcudean, S. E. (2001). "Analysis of control architectures for teleoperation systems with impedance /admittance master and slave manipulators," *The International Journal of Robotics Research*, vol. 20, pp. 419-445.
- Hashtrudi-Zaad, K. and Salcudean, S. E. (2002). "Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 108-114.
- Huang, F., Zhang, W., Chen, Z., Tang, J., Song, W. and Zhu, S. (2019). "RBFNN based adaptive sliding mode control design for nonlinear bilateral teleoperation system under time-varying delays," *IEEE Access*, vol. 7, pp. 11905-11912.
- Ju, Z., Yang, C., Li, Z., Cheng, L. and Ma, H. (2014). "Teleoperation of humanoid baxter robot using haptic feedback," *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, pp. 1-6.
- Karan, P. R., Siddharth, B., Steven, A. K. and Matthew, H. R. (2018). "Potential for Virtual Reality and Haptic Feedback to Enhance Learning Outcomes Among Construction Workers," In *Proceedings of the 18th International Conference on Construction Applications of Virtual Reality*, pp 246–255.
- Kebria, P. M., Khosravi, A., Nahavandi, S., Wu, D. and Bello, F. (2020). "Adaptive type-2 fuzzy neural-network control for teleoperation systems with delay and uncertainties," *IEEE Trans. Fuzzy Syst*, vol. 28, no. 10, pp. 2543–2554.
- Lawrence, D. A. (1993). "Stability and transparency in bilateral teleoperation," *IEEE transactions on robotics and automation*, vol. 9, pp. 624-637.
- Liu, X. and Tavakoli, M. (2011). "Adaptive inverse dynamics four-channel control of uncertain nonlinear teleoperation systems," *Advanced Robotics*, vol. 25, pp. 1729-1750.
- Ljung, L. (1999). *System identification (2nd ed.): theory for the user*: Prentice Hall PTR.
- Love, L. J. and Book, W. J. (2004). "Force reflecting teleoperation with adaptive impedance control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, pp. 159-165.
- Manocha, K. A., Pernalet, N. and Dubey, R. V. (2001). "Variable position mapping based assistance in teleoperation for nuclear cleanup," in *Proceedings ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, 2001, pp. 374-379.
- Mellah, R., Guermah, S. and Toumi, R. (2017). "Adaptive control of bilateral teleoperation system with compensatory neural - fuzzy controllers," *International Journal of Control, Automation and system*, vol. 15, pp. 1-11.
- Mohamed, O. A., Farid, F. and Redouane, T. (2017). "Adaptive Neural Control for Bilateral Teleoperation System using External Force Approach," in *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics - Volume 2*, pages 309-315.
- Naerum, E. and Hannaford, B. (2009). "Global transparency analysis of the lawrence teleoperator architecture," in *IEEE International Conference on Robotics and Automation*, pp. 4344-4349.
- Oudjida, A.K., Chaillet, N., Liacha, A., Berrandjia, M.L. and Hamerlain, M. (2014). "Design of High-Speed and Low-Power Finite-Word-Length PID Controllers," *Journal of Control Theory and Applications (JCTA)*, vol. 12, N° 1, pp. 68-83.
- Park, T. M., Lee, S. R. and Yi, H. (2020). "Workspace mapping with adaptive fuzzy control for robotic manipulator in teleoperation," *Journal of Mechanical Science and Technology*, vol. 34, pp. 2171-2178.
- Ryu, J. H. and Kwon, D. S. (2001). "A novel adaptive bilateral control scheme using similar closed-loop dynamic characteristics of master/slave manipulators," *Journal of Robotic Systems*, vol. 18, pp. 533-543.
- Sakai, H., Tomizuka, D. and Ohnishi, K. (2017). "Compliance control for stabilization of bilateral teleoperation system in the presence of time delay," in *IEEE International Conference on Mechatronics (ICM)*, pp. 62-67.
- Silva, A. J., Ramirez, O. A. D., Vega, V. P. and Oliver, J. P. O. (2009). "Phantom omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, pp. 193-198.
- Tzafestas, C., Velanas, S. and Fakiridis, G. (2008). "Adaptive impedance control in haptic teleoperation to improve transparency under time-delay," in *IEEE International Conference on Robotics and Automation*, pp. 212-219.
- Wang, H., Liu, P. X. and Liu, S. (2017). "Adaptive neural synchronization control for bilateral teleoperation systems with

- time delay and backlash-like hysteresis," *IEEE transactions on cybernetics*, vol. 47, pp. 3018-3026.
- Yang, C., Peng, G., Cheng, L., Na, J. and Li, Z. (2019). "Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks", *IEEE Trans. Syst. Man Cybern. Syst.*
- Zhai, D.-H. and Xia, Y. (2016). "Adaptive control for teleoperation system with varying time delays and input saturation constraints," *IEEE Transactions on industrial electronics*, vol. 63, pp. 6921-6929.
- Zhao, Z., He, S., Zhao, Y., Xu, C., Wu, Q. and Xu, Z. (2018). "Workspace Analysis for a 9-DOF Hyper-redundant Manipulator Based on An Improved Monte Carlo Method and Voxel Algorithm," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2018, pp. 637-642.