Bilateral Control of a 2-DOF Teleoperated Manipulator Using UDP Scheme

Magdi Mohsen¹, Khalil Ibrahim²

¹,²Department of Mechanical Engineering, Faculty of Engineering, Assiut University

eng.magdimohsen@gmail.com, and Khalil.ibrahim@aun.edu.eg

Abstract — In this paper, a workspace-control system was developed to deal with the problem of position-tracking of bilateral teleoperated two degrees of freedom manipulators, and aid the human operator in driving and controlling the slave robot that performs critical and difficult tasks such as underwater works. The developed system consists of, “JUPITER XL SCARA” robot as a slave manipulator with only the first two links activated, a wireless User Datagram Protocol (UDP) network, and PC as a master controller. The system model parameters identified using the Least-Squares method. A PID and FLC controllers with different input signals were applied to the system. It was proved experimentally and by simulation on Matlab that the FLC gave a better performance than the PID controller. The two parallel loop subroutine technique proved a great enhancement to the system performance as it decreased the total delay time in the system to almost 25 milliseconds.

KEY WORDS: teleoperation, bilateral, position tracking, model identification, Least-Squares, FLC, PID, network time delay, UDP, MATLAB, SIMULINK.

I. INTRODUCTION

A Teleoperated manipulator is a manipulator remotely controlled by a human operator. A teleoperation structure usually consisting of a slave robot, a master center, and a communication network passing the data between the two subsystems. The master center is located at the operator site while the slave subsystem or robot usually located in the task environment, which always be a hazardous area, or unreachable for a human operator. Teleoperation used to exist in a wide variety of applications where the human operator cannot be there such as space exploration, underwater tasks, handling of radioactive sources in reactors, or telesurgery [1-3]. In addition, many advanced technologies entered the robotics industry, more and more generations of communication protocols, all this make the controlling of modern telerobotics is a seriously interesting issue.

The data transmitted between the master center and slave robot face the problem of network delays this serious problem affects the tracking performance of the teleoperated system and may destabilize the whole system. The problem of time delays and the tracking performance of the teleoperated systems took the interest of many researchers so they proposed different approaches to solve these problems. These solutions had been reviewed in [5-6]. The Proportional Derivative (PD) controller and the PD without measuring velocity signals [7-8] were an example of these solutions. in order to deal with, the uncertainty in teleoperation systems more advanced approaches was proposed, such as adaptive control, robust control, and other control schemes [9-15].

This paper studies the effect of using two different control schemes (a PID and fuzzy logic controllers) on the performance of a two degree of freedom teleoperated manipulator, the teleoperated system used in this study replaces the conventional master robot with a computer-based master control system called (workspace control system). This system has a big contribution in the enhancement of performing the difficult teleoperated tasks under delayed networks. Also, the workspace-control station has a virtual simulated graph of the slave manipulator which provides the operator with a kind of virtual sense to the task environment. A PD controller was used to test a program scheme that enhances the delay time in the network. A UDP wireless communication network was chosen in the implementation of the teleoperation system. A Selective Compliance Assembly Robot Arm (SCARA), robot with only two links activated was used as a slave robot, and robot dynamic parameter values were identified by the Least-squares method. An experimental and simulation studies were held for the system response under the PID and fuzzy logic controllers.

The paper is organized as follows: Section II explains in detail the description of the teleoperated system used in this study. Section III shows the identification of the model parameters. Section IV describes the control algorithm used and the enhanced UDP network. Section V shows the experimental results of using the proposed controllers. Section VI presents the simulation model used in SIMULINK and the simulated results of using the same controllers. Section VII concludes the paper.

II. SYSTEM DESCRIPTION

A. The slave robot:

The slave robot used to achieve the system is a SCARA robot. It has four degrees of freedom, but for simplicity, only the first two revolute joints were activated Fig.(1).
As mentioned before the teleoperated system used in this paper called ( workspace control system) Fig. 2. In this type, slave angle commands are generated from the master center depending on the operator commands and the inverse kinematics of the manipulator. The master center has to calculate the forward kinematics of the manipulator from the measured angles and generates a virtual graph for the slave manipulator posture. Also, the master center calculates motor torques that the slave robot needs to track the operator inputs. All these data transferred through the wireless User Datagram Protocol (UDP) communication channel.

B. The master center:

The master center and slave robot always have a communication channel to transport their data on it. The type of communication channel can be any reliable know network. Two of the most known transport protocols for implementing communication channels are Transmission Control Protocol/Internet Protocol (TCP/IP), and UDP. In the present study, a wireless UDP communication network is used. Although TCP/IP provides more reliable communication and guarantees better data delivery, all of these advantages came on the cost of delay times due to long timeouts and several transmissions of lost data packets. The UDP protocol is the more preferred solution, as it uses shorter acknowledgments packets, data frames, and shorter timeouts. This makes UDP protocol better, and faster than TCP/IP.

III. EXPERIMENTAL IDENTIFICATION MODEL

The dynamics of the slave robot can be derived from the Newton–Euler method or the Lagrangian formulation. Using any method the dynamic model of the slave manipulator could be obtained and represented in joint space as seen in Eq.(1).

\[ \tau_i = M(\theta)\dot{\theta}_i + C(\theta, \dot{\theta}) + \tau_f(\dot{\theta}) \]  
(1)

Where \(t_i\) is the motor torque applied to joint (i), and \(\tau_f\) is the joint friction torque. \(M(\theta)\) is the mass matrix, \(C(\theta, \dot{\theta})\) the Coriolis/centrifugal torque, and \(\theta, \dot{\theta}, \ddot{\theta}\) are the joint position, velocity, and acceleration for each link. Considering the viscous and Coulomb friction forces, thus:

\[ \tau_f(\dot{\theta}) = F_v\dot{\theta} + F_s\text{sgn} (\dot{\theta}) \]  
(2)

And considering the drive characteristics as:

\[ \tau = K\psi \]  
(3)

Where \(K\) is the torque conversion constant.

The dynamic model then could be represented as:

\[
\begin{bmatrix}
    k_1\psi_1 \\
    k_2\psi_2
\end{bmatrix} = \begin{bmatrix}
    M_{11} & M_{12} \\
    M_{21} & M_{22}
\end{bmatrix} \begin{bmatrix}
    \dot{\theta}_1 \\
    \dot{\theta}_2
\end{bmatrix} + \begin{bmatrix}
    C_{11} & C_{12} \\
    C_{21} & C_{22}
\end{bmatrix} \begin{bmatrix}
    \dot{\theta}_1 \\
    \dot{\theta}_2
\end{bmatrix} + \begin{bmatrix}
    F_{v1} \\
    F_{v2}
\end{bmatrix} \text{sgn} (\dot{\theta}) + \begin{bmatrix}
    F_{c1} \\
    F_{c2}
\end{bmatrix} \text{sgn} (\ddot{\theta})
\]  
(4)

It is observed that the dynamic model has many parameters (i.e. inertial parameters, friction coefficients, geometrical parameters, etc.). Unfortunately, these parameters were not given in the robot datasheet, so a parameter identification process should be applied to obtain them.

Various methods have been proposed for estimating the model parameters of the robot manipulators. Some of the works concentrated on estimating the inertial parameters with known geometric parameters [17-18]. In the case of unknown geometric parameters, different identification algorithms were used [19-20]. These approaches are based on the analysis of the input/output behavior of the robot on some planned motion and on estimating the parameter values by minimizing the difference between a function of the real variables and its mathematical model.

In the way of experimental, due to the complexity and nonlinearity of the dynamics, it is difficult to estimate all robots’ individual parameters [21], since the link dynamic parameters are redundant to determine the manipulator dynamic model.
uniquely. The non-redundant and identifiable dynamic parameters whose values can determine the dynamic model uniquely is called base dynamic parameters or minimum dynamic. A method is proposed by Gautier and W. Khalil minimized the number of the model parameters to be identified through regrouping procedures, and formulated the identification model in an upper block triangular form. Many other methods were proposed for the identification of robot parameters such as the use of Adaline neural networks, and extended Kalman filter, which is based on the direct dynamic model. All these methods were reviewed in Refs.

Among these parameter estimation methods, the least-squares method takes a particular place for robot manipulator identification. It allows estimating the base inertial parameters from the measurement of the joint torques and the joint positions.

A. The Least squares method

The least-square method is a non-iterative method that finds the parameter estimates by optimizing the root-mean-square residual error of the model under the assumption that the measurement errors are negligible. To overcome the problem of Least square method sensitivity to measurement noise an excitation trajectory, and a filter to measured data were applied.

The least-squares method basically uses a set of linear differential equations driven from the inverse dynamic model, then uses the collected data pairs of the joints angles, and corresponding applied torques to solve these equations. With collecting data more than the unknown parameters, this yields overdetermined linear equations. The collected data usually are written in a matrix called “Regressor matrix” that can be easily solved by the pseudo inverse matrix. The identification process always followed by a type of verification process as illustrated in Fig. (3).

To go through the least squares procedure the following steps should be followed:

- Estimating link parameters using the least squares procedure.
- Executing a test trajectory.
- Calculating estimating torque using the estimated model.
- Compare measured torque with the estimated one.

Therefore, a robot model that is linear with respect to the inertial parameters needs to be generated to apply the least squares procedure. So, the dynamic model in equation (4) may be written as a set of unknown parameters linear equations as follows:

$$\tau = \varphi (\theta, \dot{\theta}, \ddot{\theta}) p$$

(5)

Where $\varphi$ is the regressor matrix, and $p$ is the base parameters matrix. Considering the dynamic equations of the manipulator, the model parameters were regrouped through a regrouping procedure mentioned in [20].

The dynamic model for estimating the lumped equivalent parameters can be formulated into a set of seven base parameters $[P_1, P_2, P_3, P_4, P_5, P_6, P_7]$ as follows:

$$Kv = \varphi (\theta, \dot{\theta}, \ddot{\theta}) p$$

(6)

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} k_1^{-1} & 0 \\ 0 & k_2^{-1} \end{bmatrix} \begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & \varphi_{14} & \varphi_{15} & \varphi_{16} & \varphi_{17} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \varphi_{24} & \varphi_{25} & \varphi_{26} & \varphi_{27} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \end{bmatrix}$$

(7)

Where:

$$\begin{align*}
\varphi_{11} &= \ddot{\theta}_1, & \varphi_{12} &= \ddot{\theta}_2, \\
\varphi_{13} &= (2\ddot{\theta}_1 + \ddot{\theta}_2) \cos \theta_2 - (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2) \sin \theta_2, \\
\varphi_{14} &= \dot{\theta}_1, & \varphi_{15} &= \text{sgn} (\dot{\theta}_1), \\
\varphi_{16} &= 0, & \varphi_{17} &= 0, & \varphi_{21} &= 0, \\
\varphi_{22} &= \ddot{\theta}_1 + \ddot{\theta}_2, & \varphi_{23} &= \ddot{\theta}_1 \cos \theta_2 + \ddot{\theta}_2 \sin \theta_2, \\
\varphi_{24} &= 0, & \varphi_{25} &= 0, & \varphi_{26} &= \dot{\theta}_2, \\
\varphi_{27} &= \text{sgn} (\dot{\theta}_2), \\
P_1 &= I_1 + I_2 + M_1 r_1^2 + M_2 r_2^2 + M_2 l_1^2, \\
P_2 &= I_2 + M_2 r_2^2
\end{align*}$$

Fig. (3) Model identification validation scheme.
\[
P_3 = M_2 L_2 r_2
\]
\[
P_4 = F_{v1}, \quad P_5 = F_{e1}
\]
\[
P_6 = F_{v2}, \quad P_7 = F_{e2}
\]

Where: \( M_i \): Mass of each link \((i=1,2) \).

\( L_i \): Length of each link \((i=1,2) \).

\( r_i \): Length from each link origin to its center of mass \((i=1,2) \).

\( l_i \): Moment of inertia of each link \((i=1,2) \).

When a system identification experiment is designed, it is necessary to consider a trajectory that excites sufficiently the manipulator parameters during the movement. Otherwise, some parameters will be impossible to identify or will become too sensitive to noise. Actually, no specific requirements are imposed on the test trajectories except that the actual trajectories of \( \theta, \dot{\theta}, \ddot{\theta} \) cannot be constant vectors for the existence of regressor inverse matrix. Hence, any trajectory that has continuous first and second derivatives may be a test trajectory. In the presented study a quintic polynomial equation described in Eq. (7) was set empirically within the robot workspace.

\[
\theta_1 = \theta_2 = 0.06 t^5 - 0.7 t^4 + 2.5 t^3 - 2.6 t^2
\]

For \( 0 < t < 4 s \) (8)

Each joint is commanded with a proportional controller to follow the test trajectory given by Eq. (8) while the joint positions and control voltages are recorded. The joint velocities and accelerations are calculated from the joint position data using cubic spline fitting to smooth the dataset, according to a balance parameter.

The torque conversion constants can be identified by performing a series of experiments with the robot constrained from motion. With known torque conversion constants values and joints position, velocity, and acceleration, the regressor matrix could be obtained. Each joint in the joints trajectory gave an equation, so the over determined equations generated. A least squares approximation is done on the data, using the left pseudoinverse.

\[
P = (\varphi^T \varphi)^{-1} \varphi^T K \nu
\]

But one of the disadvantages of the least square approximation is the feasibility of the solution (such as negative values). The method of least squares is a procedure to find the best-fit equation satisfying the collected data whoever of its physical meaning. To solve this problem, a condition or constraint relation should be added while the computer is solving the least-squares problem, and the problem then called a constraint least-squares problem. A special case from the constraint least squares is the non-negative least squares that was considered in this study.

A variety of methods have been applied to solve the non-negative least squares problem, in general, they can be roughly divided into active-set methods and iterative approaches. The MATLAB function (lsqnonneg) was based on the active methods described in [31-32].

A set of model parameters are identified with this approach, the experiment is repeated several times to obtain several sets of estimated parameters from the corresponding sets of recorded joint position and control voltage data. The average value of the estimates for each parameter is taken as the nominal value of that parameter. The standard deviation from the average is taken as its error bound.

### IV. NETWORK ENHANCEMENT AND CONTROL ALGORITHMS

#### A. Network enhancement

In the way of designing a good telemanipulation system, the global efficiency of the system is an important issue and it highly depends on the time delays existing in the system. The communication channel delays are the major term in the total telemanipulation system delays. So the implemented network should be tuned and revised to reduce the total delay time to a minimum. For this purpose a model-free PD control algorithm had been applied as a testing method, and predefined input signals were used in this phase to separate the effect of human operator errors and delays.

Figure (4) shows the block diagram of the bilateral teleoperated PD controller. The intermediate block represents the in/out delay times between the master and slave manipulators \((T_{in}, T_{out})\).

![Bilateral teleoperated PD controller block diagram.](image)

Two different software techniques were tested in implementing the control program, to find the best technique that minimizes the communication delay time between the master and slave controllers. The first flow chart of the controlling program used is shown in Fig. (5). In this algorithm, the controller calculations, and the communication processes were executed in one loop. But this algorithm gave a bad communication performance due to the time consumed in the controller calculations, the more data packets lost, and longer delay time.
To solve that problem another algorithm shown in Fig. (6) was implemented. In this algorithm, a loop for the communication process, and a loop for all controller calculations (inverse and forward kinematics, motor torques calculations, and visual displayingsubroutine), both loops work simultaneously in the program. This has good enhancement on the controller performance, system delay time, and stability. All of these have benefits were empirically proved by using step and sine trajectory as an input signal.

sometimes a problem of communication slowness or failure was happening. This problem happens due to miscommunication between the master and slave controller, and as a result of using the UDP protocol [33]. As there is no acknowledge mechanism between the master and the slave, the master may continue sending commands to the slave robot until filling its receiving buffer then the whole communication dropped out. To deal with that problem one item was added to the data being sent from slave to master station called “life signal”. The Life signal is the slave main program loop time index. the master program can get an indication of the communication status. In the case of bad communication, two important actions should be taken. Firstly neglect the rest of the received data items, and uses the last successfully received message. Also, pause the sending of new data to the slave controller to avoid filling the slave receiving buffer.

B. Control algorithms

1. PID controller

Proportional-Integral-Derivative (PID) controllers are the most common control algorithm in the industry [33-34]. They have been used for long times in industrial purposes due to their simplicity, low cost, easy designing, and stability. A PID control law consists of applying properly the sum of three types of control actions: a proportional action, an integral action, and a derivative action [36]. The PID controller represented by Eq. (10).

\[ U(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \dot{e}(t) \]  \hspace{1cm} (10)

Where \( K_p \), \( K_i \), and \( K_d \) are the proportional, integral, and the derivative gains of the controller respectively.

It is very important to tune the PID parameters to achieve an optimized value of the desired response. ZIEGLER-NICHOLLS method is one of the most used manual tuning methods of the PID controller. Ziegler-Nichols proposed simple rules for tuning PID controllers based on experimental step responses or based on the value of \( K_p \) that result in marginal stability when only proportional control action is used [37].
2. Fuzzy Logic Controller (FLC)

Fuzzy logic controller is a rule-based decision making good for process control and expert systems. A fuzzy controller acts or regulates by means of rules in a more or less natural language, based on the fuzzy logic rules that are invented by plant operators or design engineers [38-40]. The fuzzy controller block diagram proposed for this study is shown in Fig. (8). It is a PD-like FLC and has the traditional four main components: (the rule base, The inference mechanism, fuzzification, and defuzzification) [40].

The input linguistic variables were considered error e(t) and Change of error $\dot{e}(t)$, and the output is the joint torque value $U(t)$ as shown in Figs. (9-11) for joint (1,2) and Figs. (12) joint2 for output.

The rule base table for joint1 and joint2 fuzzy controllers is shown in table (1).

### Table 1 joint1, and joint2 rule base.

| OUTPUT  | ERROR  | ERR. CHANGE |
|---------|--------|-------------|
| BN      | Z      | Z           |
| MN      | SN     | Z           |
| SN      | Z      | Z           |
| Z       | MN     | SP          |
| MP      | BP     | BP          |
| BP      | SN     | SN          |
|        | Z      | Z           |

V. EXPERIMENTAL RESULTS

Experimental work includes testing of the two control schemes. Step and sine inputs were applied in each joint. Also, circle and lemniscate shape fed to the end effector as input trajectories. Initial conditions were kept the same to analyze the strength and weaknesses of each design. Experimental controllers were developed using NI-LABVIEW commercial software program.

1. Identification of model parameters

Firstly the torque conversion constants were identified by a force dial indicator in contact with the SCARA robot as shown in Fig. (13). Each joint is actuated by applying multi suitable control voltages and the resultant torques are measured to find the joint torque constant.
The average value of the estimated model parameters computed for each joint is taken as the nominal value, while the standard deviation from the average value is taken as the error bound. The identified joint torque conversion constants are listed in Table (2). The error bound is somewhat acceptable for joint 2, but for joint 1 it is relatively large. This may be due to the greater friction at joint 1 because the mass of joint 1 is large.

| Table 2 | The torque conversion constants values. |
|---------|----------------------------------------|
|         | $K_1$ | $K_2$ |
| Nominal Value [Nm V$^{-1}$] | 1.6 | 1.5 |
| Error Bound [Nm V$^{-1}$]   | 0.6 | 0.1 |

Secondly, the regressor matrix should be obtained for the parameter identification algorithm. For that purpose, a proportional controller applied for both joints, and the trajectories described in Eqs. (7) were fed to the end effector, then the joint response was logged. Figure (14) shows the position of joints; joint 1 (solid line), and joint 2 (dashed line) with the time. The corresponding input voltage of joints with time is shown in Fig. (15).

| Table 3 | Robot inertia parameters values. |
|---------|----------------------------------|
|         | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ |
| Nominal Value | 8.92 | 0.7 | 0.39 | 81.7 | 1.72 | 58.3 | 0 |
| Error Bound   | 0.32 | 0.12 | 0.07 | 1.4 | 0.2 | 0.29 | 0 |

Finally, the verification process carried by applying random test trajectories on the robot manipulator, then the joint angles, velocities, and accelerations are logged with its corresponding control voltages. Using the identified model and the measured data, the estimated control voltages are calculated. The verification is made by comparing the measured control voltage with the calculated estimated voltage for these test trajectories. Figures (16, 17) show the measured control voltage (solid line), and the calculated estimated voltage (dashed line) for the test trajectory of joint 1, and joint 2 respectively. The results show that the joint measured control voltage is so near to the voltage calculated from the estimated model.

The velocities and the accelerations of each joint were calculated by numerical differentiation and smoothed with cubic spline fitting. All these data analyzed with the LabView program. Also the LabView was used to generate the regressor matrix. The dynamic model parameters were obtained using MATLAB nonnegative least-squares solver function (lsqnonneg).
2. The network enhancement results

The experimental results of the two software architectures of the controlling program described in Figs. (5 - 6 ) are shown in this section. The schemes are checked with a sinusoidal wave input with a (1 rad) amplitude, and a frequency of (0.16 Hz). The experiments performed on the first arm as a testing medium. Fig. (18) shows the trajectory tracking of the PD controller while using the first scheme. In this figure, the red line describes the actual trajectory and the blue line describes the desired trajectory. The lag between input and output signals, or the sent wave and the reflected wave was nearly 500 milliseconds. Figure (19) shows the trajectory of the PD controller with the second scheme. The blue line describes the desired trajectory and the red line describes the actual trajectory. The total delay time was nearly 145 ms.

Fig. (18) The first scheme of trajectory tracking of the PD controller

Fig. (19) The second scheme of trajectory tracking of the PD controller

Based on the previous results, the second scheme (parallel loops execution) was considered in all controller's programs. Also, a lot of improvements in the slave software code were added, such as decreasing the number of bits in the sent and received messages to a minimum, also avoiding using predefined general functions included in LABVIEW libraries to decrease the program execution time. After these improvements and while using the FLC the delay time was measured again and found dropped down to a range of 25 ms as seen in Fig. (20).

3. PID controller experimental results

The proportional, Integral, and the derivative gains of the PID controller (K_p, K_i, and K_d) of the controlled joint are estimated using Ziegler–Nichols tuning rules, and found as listed in table (4).

Table 4 The values of proportional, integral, and derivative tuned gains.

|      | K_p | K_i | K_d |
|------|-----|-----|-----|
| Joint1 | 7.8 | 50  | 0.3 |
| Joint2 | 1.8 | 9.3 | 0.086 |

Using a PID controller with several input for each joint like step input and sinusoidal input to validate our system [33]. The PID controller also has been tested using circle and lemniscate trajectories. The desired angle and actual angle for joint1 and joint2 corresponding to the circle trajectory are shown in Figs. (21, and 22).

Fig. (20) The system total delay time while applying a FLC.

Fig. (21) Joint 1 measured angle due to circle input while using PID controller.
The desired angle and actual measured angle for joint1 and joint2 with the PID controller and the lemniscate shape input are shown in Figs. (23, and 24).

Fig. (22) Joint 2 measured angle due to circle input while using PID controller.

Fig. (23) Joint 1 measured angle due to lemniscate input while using PID controller.

Fig. (24) Joint 2 measured angle due to lemniscate input while using PID controller.

Fuzzy logic controller response with the circular trajectory for joint1 and joint2 are shown in Figs. (25 , and 26), respectively.

Fig. (25) Joint 1 measured angle due to circle input while using FLC controller.

Fig. (26) Joint 2 measured angle due to circle input while using FLC controller.

Response of fuzzy logic controller with the lemniscate shape for joint1 and joint2 are shown in Figs. (27, and 28), respectively.

Fig. (27) Joint 1 measured angle due to lemniscate shape input while using FLC controller.

4. Fuzzy logic controller results

In this section, the system response results relevant to the different inputs are presented, after implementation of the fuzzy controller described in section IV. The different Inputs used with the PID with the same characteristics, also tested with the fuzzy controller.
VI. SIMULATION RESULTS

The simulation of the system done by the MATLAB software program, and the "Simulink" tool. The dynamic model of the bilateral system is consisting of two main parts; the slave robot model, and the communication channel model. The slave dynamic equations used in the simulation were based on Eq. (4), with the values of its parameters, namely: The torque conversion constants, given in Table (2), and robot inertia parameters, given in Table (3). As all experiments start while the robot was in the rest position, then all initial conditions of the joints angles and joints derivatives were set to zero. Using Laplace transformation for the slave robot dynamic equations, the open loop feed forward transfer functions of the robot were obtained as follows.

\[
\frac{\theta_1(s)}{V_1(s)} = \frac{0.15}{s^2 + 8.3 s}
\]

\[
\frac{\theta_2(s)}{V_2(s)} = \frac{2.17}{s^2 + 84.3 s}
\]

For the SCARA robot used in this study, the output is saturated when the motor driver reaches its maximum output voltage which is 48 volts (As per motor driver datasheet). The driving motors also a dead zones experimentally obtained and found ± 6 Volt for joint 1, and ± 1.5 Volt for joint 2. It is assumed that the communication channel have a fixed delay time that was previously measured empirically in Sec.VI, and it was nearly 25 ms. The Simulink model for the complete system is shown in Figs. (29, and 30) for the PID controller, and the fuzzy logic controller, respectivlley. The PID controller also was tested by a circular shape trajectory. The response of the simulated model of joint 1 is shown in Fig. (31), and the response of joint 2 is shown in Fig. (32).

The PID controller also was tested by lemniscate shape trajectory. The response of the simulated model of joint 1 is shown in Fig. (33), and joint 2 is shown in Fig. (34).
The fuzzy logic controller that was used in the real system, also applied in the simulation program. The different inputs used with the PID with the same characteristics, also tested with the fuzzy controller.

While using fuzzy logic controller, simulated response to the circular trajectory for joint1 and joint2 are shown in Figs (35) and (36), respectively. And the corresponding response to the lemniscate shape trajectory for joint1 and joint2 are shown in Figs. (37, and 38).

To analyze the different experiments data, different comparison criteria were used for each input function. For step response it can easily identify The time response parameters, including percent Maximum overshoot (Mp), settling time (ts) and steady state error (Ess) visually from figures. For other inputs the scalar value Root Mean Square (RMS) error methods of the tracking error is applied to compare the performance results of the two controllers, which is defined as follows:

$$e_{RMS} = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (e_j)^2}$$  \hspace{1cm} (13)

Where N is the number of sample points.

The response for other desired input angles (sinusoidal, circle, and lemniscate) using PID and fuzzy controllers is analyzed in terms of \(e_{RMS}\) as shown in Fig. (39). The summary of results are given in Figs. (40, and 41). These results indicate that the error decreased when using fuzzy control in both joints.
especially in joint2, as the average percent of error decrease was (30% for $\Theta_1$, and 62% for $\Theta_2$).

A comparison between the experimental results and the simulation results was held to investigate how the simulated model get along with the real system.

The differences between the experimental, and the simulation results for other inputs (sinusoidal, circle, and lemniscate shape) in the case of using the PID, and FLC controllers are shown in Fig. (40, and 41). In order to clarify the differences between experimental and simulation results, the experimental results will be considered as a baseline. The comparison shows that most of the simulation results deviated with a percent of 50% on average from the experimental results except the highlighted cells are shown in Fig. (40, and 41). Sinusoidal response under both controllers for joint 1, and under FLC only for joint 2. These differences may be due to a kind of inaccuracy in the dynamic model estimated parameters or because of the assumption of a fixed delay time in the communication channel.

Experimental and simulated results show clearly that the fuzzy logic controller gave a better satisfying response than the traditional PID controller in term of minimum overshoot, rise time, settling time, and RMS of tracking error. That main conclusion is in agreement with [41]. Also he observed that the controller gave better performance on joint 2 than joint 1 (note: he uses same JUPITER XL SCARA robot).

VII. CONCLUSIONS

The main conclusions from this work can be represented in the following points

- The well-designed network is a main part of the teleoperated system, as it highly depends on the total delay time in the communication channel.
- The least squares method used to estimate the SCARA robot mathematical model parameters proved a satisfying estimated model for the system in direct straight forward steps. However, the differences observed in joint 1 between the experimental and simulation results.
- The two parallel loop subroutine techniques prove a great enhancement to the total delay time in the system. Also resetting of the receiving buffer in case of lost data packets increases the performance of the communication network. All of these improvements contribute to reduces the system's total delay time to a range of 25 ms.
- The root mean square error decreased when using fuzzy control in both joints especially in joint 2, as the average percent of error decrease was (30% for $\Theta_1$, and 62% for $\Theta_2$).
- Experiments were implemented on two joints only, but the proposed algorithm can be extended to any number of joints.
Acknowledgment
This study was supported in parts by Mechatronics Lab. Mechanical department Faculty of engineering Assiut University.

Magdi Mohsen and Khalil Ibrahim have equal contributions. Authors declare no conflict of interest on this work.

Availability of data and materials
Not applicable

Funding
Not applicable

REFERENCES
1. Sivčev S, Rossi M, Coleman J, et al. Fully automatic visual servoing control for work-class marine intervention ROVs. Control Eng Pract 2018; 74: 153–167.
2. Lee S-J, Lee S-C, Ahn H-S. Design and control of tele-matched surgery robot. Mechatronics 2014; 24: 395–406.
3. Ya-kun Z, Hai-yang L, Rui-xue H, et al. Shared control on lunar spacecraft teleoperation rendezvous operations with large time delay. Acta Astronaut 2017; 137: 312–319.
4. Sun D, Naghdy F, Du H. A novel approach for stability and transparency control of nonlinear bilateral teleoperation system with time delays. Control Eng Pract 2016; 47: 15–27.
5. Hokayem PF, Spong MW. Bilateral teleoperation: An historical survey. Automatica 2006; 42: 2035–2057.
6. Sheridan TB. Teleoperation, telerobotics and telepresence: A progress report. Control Eng Pract 1995; 3: 205–214.
7. Nuño E, Ortega R, Barabanov N, et al. A globally stable PD controller for bilateral teleoperators. IEEE Trans Robot 2008; 24: 753–758.
8. Nuño E, Basañez L, López-Franco C, et al. Stability of nonlinear teleoperators using PD controllers without velocity measurements. J Franklin Inst 2014; 351: 241–258.
9. Yang Y, Hua C, Guan X. Coordination control for bilateral teleoperation with kinematics and dynamics uncertainties. Robot Comput Integr Manuf 2014; 30: 180–188.
10. Martínez CAL, Polat I, van de Molengraft R, et al. Robust high performance bilateral teleoperation under bounded time-varying dynamics. IEEE Trans Control Syst Technol 2014; 23: 206–218.
11. Yang Y, Ge C, Wang H, et al. Adaptive neural network based prescribed performance control for teleoperation system under input saturation. J Franklin Inst 2015; 352: 1850–1866.
12. Yang Y, Hua C, Guan X. Adaptive fuzzy synchronization control for networked teleoperation system with input and multi-state constraints. J Franklin Inst 2016; 353: 2814–2834.
13. Hua C, Yang Y, Liu PX. Output-feedback adaptive control of networked teleoperation system with time-varying delay and bounded inputs. IEEE/ASME Trans Mechatronics 2014; 20: 2009–2020.
14. Hua C-C, Yang Y, Guan X. Neural network-based adaptive position tracking control for bilateral teleoperation under constant time delay. Neurocomputing 2013; 113: 204–212.
15. Shim H, Jun B-H, Lee P-M, et al. Workspace control system of underwater tele-operated manipulators on an ROV. Ocean Eng 2010; 37: 1036–1047.
16. Knapp ED, Langill JT. Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems. Syngress, 2014.
17. Atkeson CG, An CH, Hollerbach JM. Estimation of inertial parameters of manipulator loads and links. Int J Rob Res 1986; 5: 101–119.
18. Khosla PK. Estimation of robot dynamics parameters: Theory and application. Carnegie-Mellon University. The Robotics Institute, 1987.
19. Wakui S, Mita T. A parameter identification method of horizontal robot arms. Adv Robot 1989; 4: 337–352.
20. Urrea C, Pascal J. Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot. Comput Electr Eng 2018; 67: 791–806.
21. Radkhah K, Kulic D, Croft E. Dynamic parameter identification for the CRS A460 robot. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2007, pp. 3842–3847.
22. Mayeda H, Yoshida K, Osuka K. Base parameters of manipulator dynamic models. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation. IEEE, 1988, pp. 1367–1372.
23. Gautier M, Khalil W. Direct calculation of minimum set of inertial parameters of serial robots. IEEE Trans Robot
24. Urrea C, Pascal J. Parameter identification methods for real redundant manipulators. J Appl Res Technol 2017; 15: 320–331.

25. Sinha SK, Nagaraja T. Extended Kalman filter algorithm for continuous system parameter identification. Comput Electr Eng 1990; 16: 51–64.

26. Wu J, Wang J, You Z. An overview of dynamic parameter identification of robots. Robot Comput Integr Manuf 2010; 26: 414–419.

27. Khalil W, Gautier M, Lemoine P. Identification of the payload inertial parameters of industrial manipulators. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007, pp. 4943–4948.

28. Gautier M. Optimal motion planning for robot’s inertial parameters identification. In: [1992] Proceedings of the 31st IEEE Conference on Decision and Control. IEEE, 1992, pp. 70–73.

29. Chan SP. An efficient algorithm for identification of robot parameters including drive characteristics. J Intell Robot Syst 2001; 32: 291–305.

30. Schittkowski K. The numerical solution of constrained linear least-squares problems. IMA J Numer Anal 1983; 3: 11–36.

31. Coleman TF, Li Y. An interior trust region approach for nonlinear minimization subject to bounds. SIAM J Optim 1996; 6: 418–445.

32. Coleman TF, Li Y. On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. Math Program 1994; 67: 189–224.

33. Magdi M, Khalil I and Abdelfatah M Design and Control of a Two Degree of Freedom Teleoperated Manipulator. IOP Conference Series: Materials Science and Engineering Shanghai, China 2018; 435:1-5.

34. Mohamed A, Omar S, Bishoy M, K. Ibrahim Design and Analysis of Low Cost Upper Limb Exoskeleton. The 12th IEEE International Conference on Computer Engineering and Systems (ICCES 2017) 2017; 80-84.

35. Bennett S. A brief history of automatic control. IEEE Control Syst Mag 1996; 16: 17–25.

36. Cervantes I, Alvarez-Ramirez J. On the PID tracking control of robot manipulators. Syst Control Lett 2001; 42: 37–46.

37. Ogata K, Yang Y. Modern control engineering. Prentice hall Upper Saddle River, NJ, 2010.

38. Precup R-E, Hellendoorn H. A survey on industrial applications of fuzzy control. Comput Ind 2011; 62: 213–226.

39. Abdeen AA, Ibrahim K and Nasr AM. Active suspension system design using fuzzy logic control and linear quadratic regulator. In: Proceedings of the International conference on advanced intelligent systems and informatics, Cairo, Egypt, 29 August 2018, pp. 152–166.

40. Khalil I and Abdel Badie S. A hybrid PID control scheme for flexible joint manipulators and a comparison with sliding mode control. Ain Shams Eng J 2018; 9: 3451–3457.

41. Farrage A, Sharkawy AB. Experimental Investigation of an Adaptive Neuro-Fuzzy Control Scheme for Industrial Robots. J Eng Sci Fac Eng Univ 2014; 42: 703–721.