

# On-line Master/Slave Robot System Synchronization with Obstacle Avoidance

Rogelio de J. Portillo-Vélez, Carlos A. Cruz-Villar, Alejandro Rodríguez-Ángeles

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav-IPN),  
Av. Instituto Politécnico Nacional, #2508, San Pedro Zacatenco, 07360. México D.F.

rportillo@cinvestav.mx (Corresponding author), cacruz@cinvestav.mx, aangeles@cinvestav.mx

**Abstract:** In this work, it is proposed a controller for the synchronization of master/slave robotic systems. The aim of the proposed controller is to provide autonomy to the slave robot, via obstacle avoidance capability. The controller includes two terms. The first term is a PID controller, which is mapped through the task Jacobian from the task space to the robot joint space. The second term is the on-line solution of an optimal control problem (OCP), which considers the dynamic model of the slave robot. The performance index of the OCP pursues three objectives. The first goal is synchronization of master/slave end-effector position. The second goal is to keep the joint positions of the slave robot within feasible limits. The third goal is the obstacle avoidance of the whole arm at the slave robot side. Experimental results show the effectiveness of our proposal when tested on a writing task.

**Keywords:** Synchronization, Optimization, Redundant Manipulators, Obstacle Avoidance.

## 1. Introduction

Controlled synchronization of robots is a well developed research area [1]. This kind of synchronization has potential applications to production processes, and master/slave systems. Typically, master-slave robotic systems perform synchronization tasks in hazardous and unstructured environments, which demand the slave robot to work in situations which might not be predicted a priori. These situations include information delays or changes in the environment as moving *obstacles*, which may hinder the task at the slave robot side.

The obstacle avoidance problem has been approached considering the path planning problem [2] and the control problem [3]. Depending on the knowledge of the obstacle trajectory, on-line or off-line approaches are considered. If a robot works in environments where obstacle trajectory is unknown, then off-line path planning schemes become useless. Therefore, schemes for on-line controlled synchronization of master/slave robotic systems must be adopted when the obstacle trajectory is not known in advance. Moreover, in a master/slave system, the operator guides the end-effector position of the slave robot through the master robot. However, the operator is not aware about the control of the rest of the slave robot structure, [4]. This fact imposes constraints on both, the human operator and the slave robot performance. Thus, increasing the autonomy of the slave robot greatly improves the performance of the master/slave robotic system.

Several approaches to achieve controlled synchronization of master/slave robotic systems have been adopted in the past. One approach is the direct control, where the motion of the slave robot is directly synchronized with the motion of the master robot [5]. Other approach is the supervisory control approach, where the operator monitors the task execution, and the slave robot has higher autonomy [6]. The shared control approach considers the direct control together with a local sensor based controller for the slave robot [7], [8]. Notice that, none of the cited approaches considers obstacle avoidance capabilities.

Some techniques have been explored in order to provide autonomy to the master/slave robotic systems. In [9], the robot is provided with sensor systems to acquire necessary information about the environment. On the other hand, it is well known that kinematic redundancy of a robot, i.e. the presence of additional degrees of freedom (DOF) to perform a task [10], allows rendering autonomy by performing additional goals, as energy consumption optimization or obstacle avoidance, see [11], [12], [13] and [14].

In this work, it is considered a master/slave robotic system commanded by a human operator. The master robot works in a structured environment, i.e. a well defined one. The slave robot synchronizes with the end-effector position of the master robot in an unstructured environment, where unexpected objects may appear and collide with the slave robot structure. Then, exploiting the slave robot

redundancy, its autonomy is increased while avoiding obstacles.

To achieve three objectives; the master/slave synchronization, the obstacle avoidance and robot joint limits avoidance at the slave side, an on-line optimal controller is proposed. The proposed controller simultaneously performs trajectory tracking and position time varying obstacle avoidance, while only instantaneous obstacle position is required. In this work, such a position is obtained via a CCD camera, to on-line locate a repulsive potential field around the obstacle. However, visual servoing approach is not followed as the robot closed loop feedback is performed by using the optical encoder attached to each joint actuator.

The design of the synchronization controller is considered as a dynamic optimization problem, which is on-line solved by means of the *gradient flow* approach [15]. Therefore, the state derivatives with respect to the optimizing controller input (sensitivities) are computed by on-line solving a set of adjoint differential equations.

Four features are distinguished in this work: i) consideration of the dynamic model of the slave robot in the controller design, ii) the inverse kinematic models of both robots are not required, iii) the fact that the operator is not aware of the obstacle presence, indeed, the controller does not know the obstacle trajectory but the instantaneous position, and iv) the controller considers joint limits avoidance at the slave robot side, in order to ensure feasible slave robot configurations.

The rest of the paper is organized as follows. The synchronization problem is stated in section 2. In section 3, the necessary mathematical models are provided. Section 4 describes the proposed synchronization controller. A case study is presented in section 5. In section 6 experimental results are discussed. Section 7 closes the paper with the conclusions.

## 2. Problem Statement

In the following, sub-index  $M$  refers to the master robot while sub-index  $S$  refers to the slave robot. Consider a master/slave system, composed of a master robot with  $n_M$  joint DOF, and a redundant slave robot with  $n_S$  joint DOF. Thus, dissimilar master/slave robot systems, i.e. without the same kinematic structure, can be

considered. The problem is stated as follows: to design a controller to synchronize the end-effectors position of a master/slave robot system, avoiding any possible collision between the obstacle and the slave robot, in spite of the lack of knowledge of the master robot operator about the obstacle trajectory.

### Remarks:

- The above stated problem considers only end-effector position communication from the master robot to the slave robot.
- Joint limits avoidance must be considered to guarantee feasible kinematic configurations of the slave robot.

## 3. Models of the Master/Slave Robots

Due to the proposed unilateral synchronization scheme, only the direct kinematic model is required on the master robot side. On the slave robot side, the direct kinematic model and the dynamic model are required.

### 3.1 Master robot model

Consider a fully actuated and rigid master robot with  $n_M$  joints. In general terms, the direct kinematics of the master robot is a nonlinear function  $h_M : \mathbb{R}^{n_M} \rightarrow \mathbb{R}^{m_M}$ , given in equation (1), which maps the vector of joint positions  $q_M \in \mathbb{R}^{n_M}$ , to the vector of cartesian positions  $X_M \in \mathbb{R}^{m_M}$ , where  $m_M$  is the Cartesian task dimension of the master robot.

$$X_M = h_M(q_M) \quad (1)$$

### 3.2 Slave Robot Model

Consider a  $n_S$  - joint, fully actuated and rigid slave robot. Its joint positions are denoted by  $q_S \in \mathbb{R}^{n_S}$ . The slave robot Cartesian space is  $m_S$  - dimensional, in such way that in general  $n_S \geq m_S$  in order to guarantee task feasibility, and for redundant robot manipulators  $n_S > m_S$ . The same holds for the master robot.

The direct kinematic model of the slave robot is a nonlinear function  $h_S : \mathbb{R}^{n_S} \rightarrow \mathbb{R}^{m_S}$ , given in equation (2), which maps the vector of joint

variables  $q_s$ , to the vector of Cartesian variables  $X_s \in \mathbb{R}^{m_s}$ .

$$X_s = h_s(q_s) \quad (2)$$

It is important to highlight that for trajectory design, motion planning, and for control implementations, the inverse kinematics model, which gives the inverse relationship in equations (1) and (2), is required. In general, the inverse kinematic model implies multiple solutions or even singular solutions, depending on the robot architecture. Our proposal is free of inverse kinematic model since the optimization formulation implicitly solves the joint trajectory generation problem.

Applying the Euler-Lagrange formalism, the joint space dynamic model of the slave robot is given in equation (3), where  $\dot{q}_s \in \mathbb{R}^{n_s}$  and  $\ddot{q}_s \in \mathbb{R}^{n_s}$  are the joint velocities and accelerations, respectively;  $M(q_s) \in \mathbb{R}^{n_s \times n_s}$  is the symmetric, positive-definite inertia matrix,  $G(q_s) \in \mathbb{R}^{n_s \times 1}$ , denotes the vector of gravity forces,  $B \in \mathbb{R}^{n_s \times n_s}$ , is a diagonal matrix containing the viscous friction coefficients of the slave robot joints,  $C(q_s, \dot{q}_s) \in \mathbb{R}^{n_s \times n_s}$ , represents the Coriolis and centrifugal forces matrix, and the vector of input torques for the slave robot is given by  $\tau_s \in \mathbb{R}^{n_s \times 1}$ .

$$M(q_s)\ddot{q}_s + C(q_s, \dot{q}_s)\dot{q}_s + B\dot{q}_s + G(q_s) = \tau_s \quad (3)$$

To fully relate the slave robot joint and Cartesian spaces, it is required to establish a relation among the joint torques  $\tau_s$ , and the Cartesian forces  $F_s \in \mathbb{R}^{m_s}$ , for that, the robot Jacobian  $J_s(q_s) = \frac{\partial X_s}{\partial q_s} \in \mathbb{R}^{m_s \times n_s}$  is considered. Notice that for redundant manipulators the Jacobian is not a square matrix. Thus as far as  $\text{rank}\{J_s(q_s)\} = m_s$ , and assuming there exists an admissible vector  $q_s$  such that  $X_s = h_s(q_s)$ , see [16], it is guaranteed that the Cartesian task  $X_s$  can be accomplished. Meanwhile, in the redundant case, an extra task such as obstacle avoidance, load distribution or joint limit avoidance can be performed.

## 4. Joint/Cartesian Synchronization Controller

The synchronization controller is conformed of two additive terms. The first term of the controller considers a PID Cartesian controller. This part of the controller is mapped through the Jacobian of the slave robot  $J_s(q_s) = \frac{\partial X_s}{\partial q_s} \in \mathbb{R}^{m_s \times n_s}$ , to the joint torques at the slave robot joint space. The second part considers an optimal controller, which is an on-line solution of an OCP. Thus, the input torque  $\tau_s$ , in equation (3), is proposed as in equation (4), where  $F_{PID_s} \in \mathbb{R}^{m_s \times 1}$  is the slave robot PID cartesian controller, and  $\tau_{O_s} \in \mathbb{R}^{n_s \times 1}$ , is the solution to a dynamic optimization problem.

$$\tau_s = J_s^T(q_s)F_{PID_s} + \tau_{O_s} \quad (4)$$

The proposed controller is developed as follows.

### 4.1 PID Cartesian controller

The PID controller for the slave robot is based on Cartesian task space variables. Therefore, by considering equation (2), it follows that this part of the controller can be expressed as in equation (5), where  $K_P, K_D, K_I \in \mathbb{R}^{m_s \times m_s}$  are proportional, derivative, and integral diagonal gain matrices, respectively.

$$F_{PID_s} = K_P e_{C_s} + K_D \dot{e}_{C_s} + K_I \int e_{C_s} dt \quad (5)$$

The Cartesian synchronization error is denoted by  $e_{C_s} \in \mathbb{R}^{m_s \times 1}$  while  $\dot{e}_{C_s} \in \mathbb{R}^{m_s \times 1}$  represents its time derivative, which are given in equations (6) and (7), where  $X_M$  and  $X_s$ , are given in equations (1) and (2), and  $\dot{X}_M$  and  $\dot{X}_s$  denote their time derivatives.

$$e_{C_s} = X_M - X_s = X_M - h_s(q_s) \quad (6)$$

$$\dot{e}_{C_s} = \dot{X}_M - \dot{X}_s = \dot{X}_M - J_s(q_s)\dot{q}_s \quad (7)$$

### 4.2 Optimization problem statement

The aim of the optimal controller is to improve the PID controller performance, by minimizing the Cartesian tracking error, as well as taking advantage of the redundancy to yield obstacle avoidance and considering robot joint position limits.

Let us consider the state vector  $\xi = [\xi_1 \ \xi_2 \ \xi_3]^T = [q_s \ \dot{q}_s \ q_I]^T \in \mathbb{R}^{(2n_s + m_s) \times 1}$ , where  $q_I = \int e_{C_s} dt \in \mathbb{R}^{m_s \times 1}$  corresponds to the vector state generated by the cartesian PID controller (5). The closed loop system given by the robot dynamic model (1) and (4) is given in equation (8), where the Coriolis, friction and gravitational effects are considered in the vector  $N(\xi_1, \xi_2) = C(\xi_1, \xi_2)\xi_2 + B\xi_2 + G(\xi_1)$ .

$$\dot{\xi} = \begin{bmatrix} \xi_2 \\ M^{-1}(\xi_1) \left[ -N(\xi_1, \xi_2) + J_s^T(\xi_1) F_{\text{pid}} + \tau_{O_s} \right] \\ X_M - F_{D_{M_s}}(\xi_1) \end{bmatrix} \quad (8)$$

The general OCP consists on minimizing the objective function  $I \in \mathbb{R}$ , subject to the closed loop dynamical model (8) and in general to  $p$  inequality constraints  $g \in \mathbb{R}^{p \times 1}$ . The objective function  $I$ , and the inequality constraints  $g$ , are function of the vector  $\xi$ , and the optimal controller  $\tau_{O_s}$ .

Thus, the constrained OCP is stated as: to minimize  $I(\xi, \tau_{O_s})$  under the optimization controller  $\tau_{O_s}$ , subject to the dynamical model (8), and the constraints  $g(\xi, \tau_{O_s})$ . This is stated as follows

$$\begin{aligned} \min_{\tau_{O_s}} \quad & I(\xi, \tau_{O_s}) \\ \text{subject to} \quad & \dot{\xi} = f(\xi, \tau_{O_s}) \\ & g_i(\xi, \tau_{O_s}) \leq 0 \quad i = 1, 2, \dots, p \end{aligned} \quad (9)$$

Solving constrained optimization problems as that one in equation (9) might be difficult, even for the linear case, as reported in [17].

In this work, inequality constraints  $g_i$  are used to consider robot obstacle avoidance and position joint limits. This is performed by considering the barrier functions approach [18]. Thus, the original OCP (9) can be rewritten as an OCP without inequality constraints (10), where the constraints are included in an augmented objective function  $I_a$ .

$$\begin{aligned} \min_{\tau_{O_s}} \quad & I_a(\xi, \tau_{O_s}) \\ \text{subject to} \quad & \dot{\xi} = f(\xi, \tau_{O_s}) \end{aligned} \quad (10)$$

### 4.3 On-line optimal synchronization controller

Most of the OCP are solved off-line, [19]. However, the proposed approach must consider an on-line solution, as the slave robot works at an unstructured environment. Therefore, in this proposal the optimization problem given in equation (10) is solved via the gradient flow approach [15]. The gradient flow approach states that the optimal controller  $\tau_{O_s}$ , which minimizes the objective function  $I_a$ , can be proposed as the solution to equation (11), where  $\Gamma \in \mathbb{R}^{n_s \times n_s}$ , is a diagonal gain matrix, which is related to convergence properties of the gradient flow approach.

$$\dot{\tau}_{O_s} = -\Gamma \left( \nabla_{\tau_{O_s}} I_a \right)^T \quad (11)$$

The gradient of the objective function, with respect to the independent optimization vector  $\nabla_{\tau_{O_s}} I_a \in \mathbb{R}^{1 \times n_s}$ , is computed by following the chain rule of differentiation, as indicated in equation (12).

$$\nabla_{\tau_{O_s}} I_a = \left( \frac{\partial I_a(\xi, \tau_{O_s})}{\partial \xi} \frac{\partial \xi}{\partial \tau_{O_s}} + \frac{\partial I_a(\xi, \tau_{O_s})}{\partial \tau_{O_s}} \right) \quad (12)$$

In equation (12), the term  $\frac{\partial \xi}{\partial \tau_{O_s}}$  denotes the states sensitivity function matrix with respect to the control action  $\tau_{O_s}$ , which is obtained from partial differentiation of the closed loop system (8), in a similar fashion as in [19], for the parametric sensitivity matrix.

Then, differentiating the closed loop system (8), with respect to  $\tau_{O_s}$  and inverting the order

of the linear operators  $\left[ \frac{\partial}{\partial \tau_{O_s}}, \frac{d}{dt} \right]$ , it results in a

time varying linear dynamical system, in terms of  $\left[ \frac{\partial \xi}{\partial \tau_{O_s}} \right]$ , as given in equation (13).

$$\frac{d}{dt} \frac{\partial \xi}{\partial \tau_{O_s}} = \left( \frac{\partial f(\xi, \tau_{O_s})}{\partial \xi} \frac{\partial \xi}{\partial \tau_{O_s}} + \frac{\partial f(\xi, \tau_{O_s})}{\partial \tau_{O_s}} \right) \quad (13)$$

Equations (11) and (13) are two dynamical systems that have to be on-line solved. They

depend on the dynamics of the slave robot (8) through the states sensitivity function matrix

$$\begin{bmatrix} \frac{\partial \xi}{\partial \tau_{o_s}} \end{bmatrix}.$$

#### 4.4 Non-incrementality of the objective function.

**Theorem 1.** If  $\tau_{o_s}^*(t)$  is the solution of the differential equation (11), with  $\Gamma \in \mathbb{R}^{n_s \times n_s}$  a diagonal definite positive matrix, then the objective function  $I_a(\xi^*, \tau_{o_s}^*)$ , is non-increasing along trajectories  $\xi^*(t)$  and  $\tau_{o_s}^*(t)$ , where  $\xi^*(t)$  is the solution to (8), with  $\tau_{o_s}(t) = \tau_{o_s}^*(t)$ .

**Proof:** The time evolution of  $I_a(\xi, \tau_{o_s})$  is given by

$$I_a(\xi(t), \tau_{o_s}(t)) = I_a(\xi(0), \tau_{o_s}(0)) + \int_0^t \frac{dI_a(\xi(t), \tau_{o_s}(t))}{d\sigma} d\sigma \quad (14)$$

where the time derivative of  $I_a(\xi, \tau_{o_s})$  is given by

$$\frac{dI_a(\xi, \tau_{o_s})}{dt} = \left( \frac{\partial I_a(\xi, \tau_{o_s})}{\partial \xi} \frac{\partial \xi}{\partial \tau_{o_s}} + \frac{\partial I_a(\xi, \tau_{o_s})}{\partial \tau_{o_s}} \right) \quad (15)$$

Finally, by considering equation (11), the time evolution of  $I_a(\xi, \tau_{o_s})$  is given by

$$I_a(\xi(t), \tau_{o_s}(t)) = I_a(\xi(0), \tau_{o_s}(0)) + \int_0^t \left[ \left( \frac{dI_a(\xi(t), \tau_{o_s}(t))}{d\tau_{o_s}} \right) \Gamma \left( \frac{dI_a(\xi(t), \tau_{o_s}(t))}{d\tau_{o_s}} \right)^T \right] d\sigma \quad (16)$$

#### 4.5 Remarks on the optimal controller

- The complexity of the optimal controller relies on the computation of the sensitivities (13), which represents an adjoint system to the robot model (3) in closed loop with controller (4). This adjoint system is derived in the next section, for a specific case study.
- The controller is valid for any locally convex and at least one time differentiable objective function.

## 5. Case Study

The synchronization of two *dissimilar* Master/Slave robot manipulators, performing a writing task while the slave robot avoids only one moving obstacle, is considered. The testbed is conformed of a three joint DOF serial robot as master robot, a three joint DOF planar robot as a slave robot and a CMOS camera as a sensor to detect the obstacle position. Both robots were constructed at Cinvestav-IPN and are built of aluminum (alloy 6063 T-5), with 9.525 [mm] thickness.

### 5.1 Master robot

The master robot (figure 1) is driven by 3 DC brushless servomotors of the brand Maxon©. Its direct kinematic model is given in equation (17), where,  $S_{i_m} = \sin(q_{i_m})$ , and  $C_{i_m} = \cos(q_{i_m})$ , for  $i=1,2,3$ . The lengths of the links corresponding to the master direct kinematic model are:  $l_{2_m} = 0.25 [m]$  and  $l_{3_m} = 0.26 [m]$ .

By locking the first joint of this robot, its workspace has the same dimension than the slave robot workspace, which is planar.

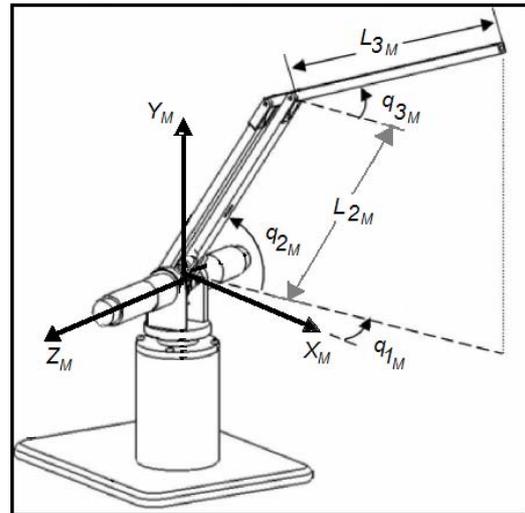


Figure 1. Master Robot.

$$\begin{aligned} x_m &= (l_{2_m} C_{2_m} + l_{3_m} C_{3_m}) C_{1_m} \\ y_m &= l_{2_m} S_{2_m} + l_{3_m} S_{3_m} \\ z_m &= (l_{2_m} C_{2_m} + l_{3_m} C_{3_m}) S_{1_m} \end{aligned} \quad (17)$$

### 5.2 Slave robot

The slave robot (Figure 2) is driven by DC brushless servomotors of the brand Micromo© Electronics Inc. Its direct kinematic model is

expressed in equation (18), were and

$$S_{ijk_s} = \sin(q_{i_s} + q_{j_s} + q_{k_s})$$

$$C_{ijk_s} = \cos(q_{i_s} + q_{j_s} + q_{k_s}).$$

$$x_s = -l_{1_s} S_{1_s} - l_{2_s} S_{12_s} - l_{3_s} S_{123_s}$$

$$y_s = l_{1_s} C_{1_s} + l_{2_s} C_{12_s} + l_{3_s} C_{123_s} \quad (18)$$

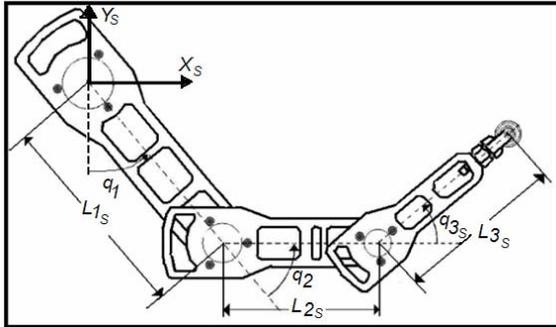
$$z_s = 0$$


Figure 2. Slave Robot.

Notice in Figure 1 and Figure 2 that the Cartesian coordinate systems have the same orientation. Thus, no rotations are required to obtain the relation between the Master/Slave Cartesian spaces. Nonetheless, a scaling factor  $S_F = 0.75$ , and an offset  $(x_r, y_r) = (0.05, -0.1)[m]$  are applied to the end-effector position of the master robot to fit into the slave robot workspace.

### 5.3 Obstacle position sensing and testbed

The Cartesian obstacle position  $X_o$ , which is constrained to the slave robot workspace, is acquired through a binarization of the images from a camera of the Photonfocus© brand, and processed in a personal computer. A frame grabber model X64 X-celera-CL PX4 from DALSA©, is used to preprocess the images from the CMOS camera. The camera is placed in front of the robot manipulator, such that the image plane of the camera is parallel to the work space of the slave robot. The slave robot and the background are in black color and, in order to increase the contrast for the image processing, the obstacle is in white color. The obstacle detection is sampled at 10 Hz.

The slave robot controller was programmed using another personal computer. The master/slave robot joint positions were obtained via a sensoray© 626 data acquisition board sampling at 500 Hz.

Both personal computers, the one which controls the slave robot and the one which computes the obstacle position, are equipped with Matlab©-Simulink© software. The above description of the testbed is shown in Figure 3.

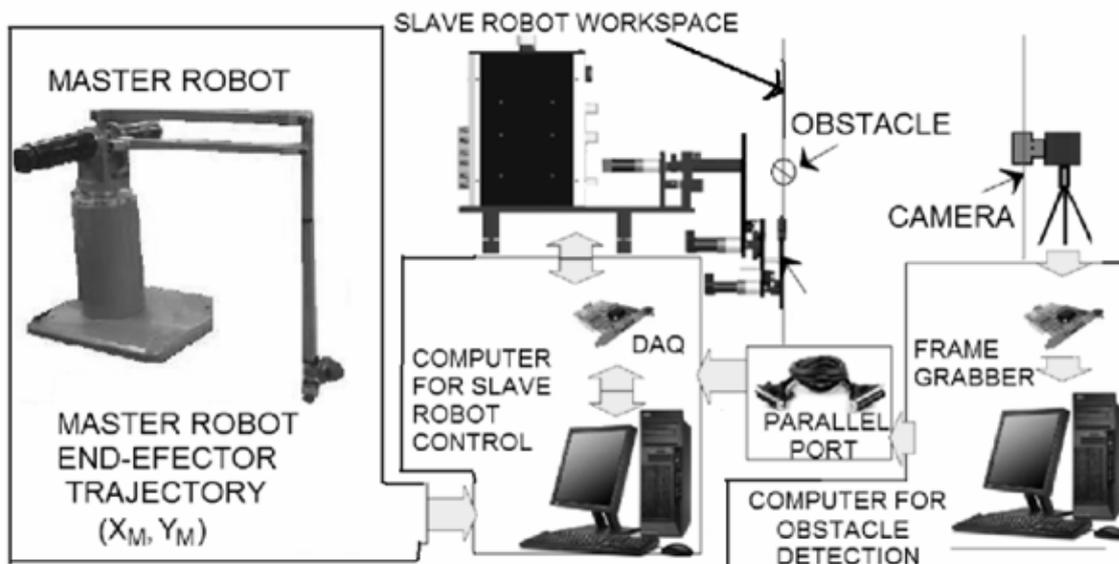


Figure 3. Testbed

## 5.4 Proposed Objective Function

The proposed objective function is designed to achieve the end-effector synchronization of the master/slave system. This is achieved by considering a convex function of the Cartesian synchronization error, which will be defined in the optimization problem. The obstacle and joint limit avoidance tasks are considered as inequality constraints at the optimization problem.

Obstacle avoidance is considered by using artificial potentials, which have similar form as it is addressed in [3]. It is considered a distance between the slave robot links and the obstacle, denoted by  $e_{O_{Si}}$ , given in equation (19). This distance is related to the Cartesian obstacle position  $X_o$ , and to the Cartesian position of the extreme of the  $i^{th}$  - link of the slave robot  $X_{Si}$ , for  $i = 1, 2, 3$ .

$$e_{O_{Si}} = (X_o - X_{Si})^T (X_o - X_{Si}) \quad (19)$$

Therefore, the OCP is formulated as a minimization problem under the independent vector  $\tau_{O_s} \in \mathbb{R}^{3 \times 1}$ , with the constraints given by the obstacle avoidance through  $e_{O_{Si}}$ , and the robot joint limits  $q_{i_{\min}}$  and  $q_{i_{\max}}$ , for  $i=1,2,3$ . This is given in the problem (20).

$$\begin{aligned} \min_{\tau_{O_s}} \quad & I = e_{C_s}^T \cdot e_{C_s} \\ \text{subject to} \quad & \\ & e_{O_{Si}} - \rho_i \leq 0 \quad i=1,2,3 \\ & q_{i_{\min}} - q_i \leq 0 \\ & q_i - q_{i_{\max}} \leq 0 \\ & \dot{\xi} = f(\xi, \tau_{O_s}) \end{aligned} \quad (20)$$

To avoid overcoming the slave robot joint limit, the barrier function approach is used, see [18]. For this, joints limits are considered to be the same for the three links and symmetrical, i.e.  $q_{i_{\min}} = -q_{i_{\max}} = q^*$ .

Then, the optimization problem (20), can be written as an unconstrained optimization problem as follows

$$\begin{aligned} \min_{\tau_{O_s}} \quad & I_a = e_s + \sum_{i=1}^3 \frac{\alpha_i}{e_{O_{Si}} + \rho_i} - \sum_{i=1}^3 \frac{2\mu_i q^*}{(q_i)^2 - (q^*)^2} \\ \text{subject to} \quad & \\ & \dot{\xi} = f(\xi, \tau_{O_s}) \end{aligned} \quad (21)$$

where  $q^* > 0$ , is the parameter related to the motion constraints at the slave robot joint limits,  $q_i$  is the  $i^{th}$  - slave joint link position,  $\alpha_i$  and  $\rho_i$  are optimization parameters related to the influence area of the artificial potentials, and  $\mu_i$  is a parameter related to the slave robot joints limits.

## 6. Results and Discussion

At the initial condition, i.e.  $t = 0$ , the slave robot is configured with  $q_{1_s} = q_{2_s} = q_{3_s} = 0$ , thus the initial end effector position is located at the Cartesian coordinates  $x_s = 0[m]$  and  $y_s = -0.452[m]$ . The master robot joints were selected to be  $q_{1_M} = 0$  for all  $t$ ;  $q_{2_M} = \pi/2$  and  $q_{3_M} = 0$  only for  $t = 0$ , thus the master end-effector coordinates are  $x_M = 0.26[m]$ ,  $y_M = -0.25[m]$  and  $z_M = 0[m]$ . The slave robot joint limits are considered, with  $q^* = 2.618[rad]$ , which are part of the mechanical design of the slave robot structure due to safety reasons.

The Cartesian PID controller gains and optimal controller parameters were selected via simulation tests. The main diagonal elements of the PID gain matrices for each one of the cartesian-DOF are listed in Table 2.

**Table 2.** Cartesian PID Gains

	$K_{P_s}$	$K_{D_s}$	$K_{I_s}$
$x_s$	1000	70	50
$y_s$	1000	70	50

The optimal part of the controller uses the following parameters:  $\Gamma_i = 0.1$ ,  $\alpha_i = 0.002$ ,  $\rho_i = 0.001$  and  $\mu_i = 0.01$ , for  $i = 1, 2, 3$ .

The performed experiment is described as follows. The operator guided the position of the master robot end-effector, during 60 seconds,

writing of the word 'meca' as commanded by the operator. On the other side, the slave robot end-effector performs a writing task on a board, synchronizing the master robot end-effector trajectory. An obstacle approached to the second link of the slave robot, while the writing task is performed.

Figure 4 shows the synchronized end-effector trajectories of the master and the slave robot.

It is notable that the controller performance keeps synchronization errors small (less than 0.01 [m]) as shown in Figure 5.

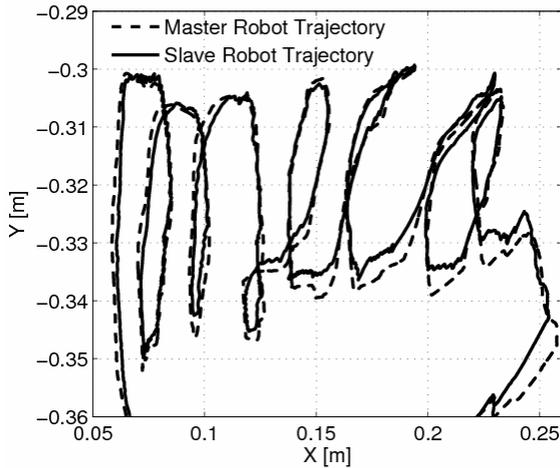


Figure 4. Master/Slave Cartesian Trajectories

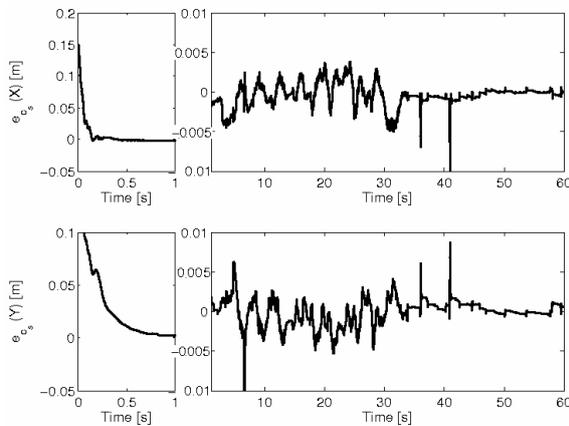


Figure 5. Cartesian errors of the slave robot

Figure 6 shows the different kinematic configurations achieved in order to avoid the obstacle. The dashed line represents the robot configuration with the presence of the obstacle, and when the optimal controller is turned off ( $\tau_{O_s} = 0$ ). In this case a collision occurs between the obstacle and the second slave robot link, thus, the writing task is hindered.

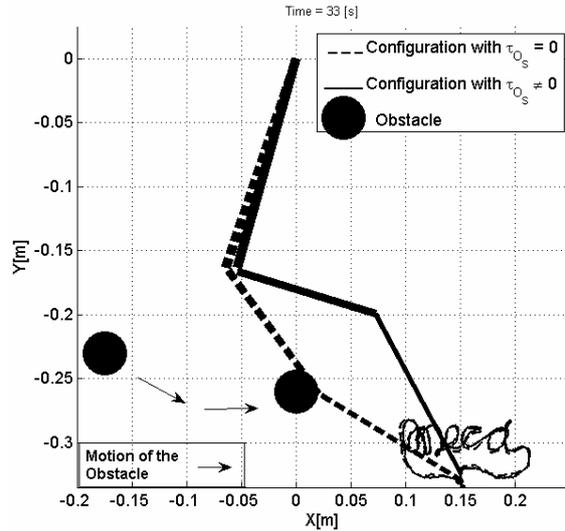


Figure 6. Kinematic configurations of the Slave Robot and Obstacle

On the other hand, the solid line represents the new configuration reached to avoid a collision with the obstacle, this time when the optimal controller is turned on ( $\tau_{O_s} \neq 0$ ). Due to the optimal controller part, the second link moves far away from the obstacle. This movement is caused by the obstacle approaching to the second link of the slave robot, which is sensed with the CMOS camera and feedback to the optimal controller. Thus, a different slave robot configuration is reached, this time avoiding the obstacle and fulfilling the writing task.

Figure 7 shows the slave robot joint trajectories during the entire experiment. Notice that the slave robot joint limits are never reached. If the joint limits are not considered in the optimization problem, a physical configuration that is not feasible may be reached, i.e. such that the slave robot joint position can overcome its joint limit.

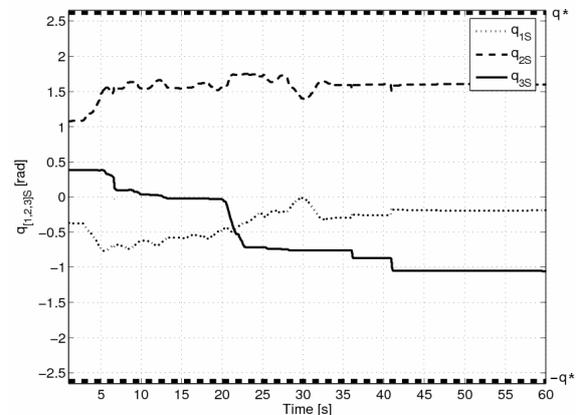


Figure 7. Joint trajectories of the Slave Robot

## 7. Conclusions

The proposed slave controller for synchronization of master/slave systems is joint/cartesian type; nonetheless inverse kinematic model is not required. It is necessary to highlight that the operator does not require knowledge about the obstacle trajectory. Thus, the autonomy of the entire master/slave system is enhanced.

This approach can be extended to more complex robotic systems, such as mobile manipulators, which are used in unpredictable environments. Experimental results show that the task synchronization is performed successfully, avoiding mobile obstacles, without overcoming slave robot joint limits at any time.

## Acknowledgements

The first author acknowledges the support of CONACyT Mexico by means of the scholarship 28753. All authors acknowledge support from CONACyT grants 133527 and 084060.

## REFERENCES

1. NIJMEIJER, H., A. RODRIGUEZ-ANGELES, **Synchronization of Mechanical Systems**. World Scientific Series on Nonlinear Science, Series A. Vol. 46, Ed. World Scientific, Singapore, 2003.
2. LOZANO-PEREZ, T., **A Simple Motion Planning Algorithm for General Robot Manipulators**. IEEE Journal of Robotics and Automation, Vol. 3. No. 3, 1987, pp. 224-238.
3. KHATIB, O., **Real-Time Obstacle Avoidance for Manipulators and Mobile Robots**. The International Journal of Robotics Research, Vol. 5, No. 1, 1986, pp. 90-98.
4. KRAGÍC, D., L. PETERSSON, H. I. CHRISTENSEN, **Visually Guided Manipulation Tasks**. Robotics and Autonomous Systems. Vol. 40, No. 2-3, 2002, pp. 193-203.
5. CHOPRA, N., M. W. SPONG, R. LOZANO, **Synchronization of Bilateral Teleoperators with Time Delay**. Automatica, Vol. 44, No. 8, 2008, pp. 2142-2148.
6. PARK, J. H., T. B. SHERIDAN, **Supervisory Teleoperation Control Using Computer Graphics**. IEEE International Conference on Robotics and Automation. Vol. 1, 1991, pp. 493-498.
7. BLAINE, G. W. **Shared Control for Dexterous Telemanipulation with Haptic Feedback**. Ph. D. Dissertation. Stanford University. Stanford, 2003.
8. NUDEHI, S. S., R. MUKHERJEE, M. GHODOUSSI, **A Shared-Control Approach to Haptic Interface Design for Minimally Invasive Telesurgical Training**. Transactions on Control Systems Technology. Vol. 13, No. 4, 2005, pp. 588-592.
9. NANAYAKKARA, T., K. KIGUCHI, T. MURAKAMI, K. WATANABE, K. IZUMI, **Enhancing the Autonomy of Teleoperated Redundant Manipulators Through Fusion of Intelligent Control Modules**. International Journal of Robotics and Mechatronics, Vol.14, No. 3, 2002, pp. 534-545.
10. CONKUR, E. S., R. BUCKINGHAM, **Clarifying the Definition of Redundancy as Used in Robotics**. Robotica, Vol. 15, No. 5, 1997, pp. 583-586.
11. LUMELSKY, L., E. CHEUNG, **Towards Safe Real-Time Robot Teleoperation Automatic Whole-Sensitive Arm Collision Avoidance Frees the Operator for Global Control**. Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, 1991, pp. 797-802.
12. FEDDEMA, J. T., J. L. NOVAK, **Whole-arm Obstacle Avoidance for Teleoperated Robots**. IEEE International Conference on Robotics and Automation, Vol. 4, 1994, pp. 3303-3309.
13. PERDEREAU, V., C. PASSI, M. DROUIN, **Real-time Control of Redundant Robotic Manipulators for Mobile Obstacle Avoidance**. Robotics and Autonomous Systems. Vol. 41, No. 1, 2002, pp. 41-59.
14. GARCIA-HERNANDEZ, N., V. PARRA-VEGA, **Haptic Teleoperated Robotic System for an Effective Obstacle**

- Avoidance.** Second International Conferences on Advances in Computer-Human Interactions, 2009, pp.255-260.
15. HELMKE, U., J. B. MOORE, **Optimization and Dynamical Systems.** (Communications and Control Engineering), Springer-Verlag, London, 1996.
  16. KELLY, R., A. COELLO, **Analysis and Experimentation of Transposed Jacobian-based Cartesian Regulators.** Robotica, Vol. 17, 1999, pp. 303-312.
  17. HAMAKER, C., **Optimization of a Constrained Quadratic Function.** Studies in Informatics and Control, Vol. 18, No. 1, 2009, pp. 21-32.
  18. F BAZARAA, M. S., H. D. SHERALI, C. M. SHETTY, **Nonlinear Programming, Theory and Algorithms.** John Wiley & Sons Inc. Second edition, 1993.
  19. BRYSON, E. A., Y-H. HO, **Applied Optimal Control: optimization, estimation and control.** John Wiley & Sons, 1<sup>st</sup> edition, 1975.
  20. KHALIL, H. K., **Nonlinear Systems.** Prentice Hall, 3<sup>rd</sup> edition, 2001.