INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Impedance Controller Tuned by Particle Swarm Optimization for Robotic Arms

Regular Paper

Haifa Mehdi and Olfa Boubaker*

National Institute of Applied Sciences and Technology, INSAT, Tunisia
* Corresponding author E-mail: olfa.boubaker@insat.rnu.tn

Received 18 May 2011; Accepted 19 Aug 2011

Abstract This paper presents an efficient and fast method for fine tuning the controller parameters of robot manipulators in constrained motion. The stability of the robotic system is proved using a Lyapunov-based impedance approach whereas the optimal design of the controller parameters are tuned, in offline, by a Particle Swarm Optimization (PSO) algorithm. For designing the PSO method, different index performances are considered in both joint and Cartesian spaces. A 3DOF manipulator constrained to a circular trajectory is finally used to validate the performances of the proposed approach. The simulation results show the stability and the performances of the proposed approach.

Keywords Impedance control, Lyapunov stability, Particle Swarm Optimization, Trajectory tracking.

1. Introduction

In recent time, programming robot arms for maximizing their capabilities remains a challenging task [1]. Fine tuning of control parameters is particularly a time consuming and an expensive procedure [2, 3]. Indeed, control parameters must take account of coupling effects between multiple-joints. Furthermore, tuning procedure becomes a very complex problem when the robotic arm must operate in a constrained environment with high-accuracy performance and high speed. In fact, in such case, there is an increasing requirement for stable position/force control strategies [4, 5, 6]. Furthermore, not only position and speed parameters of the controllers must be tuned but also force controller parameters.

In many cases, using Artificial intelligence (AI) techniques is a good alternative to improve controller performances for robotic arms [7]. A wide range of AI techniques as neural network [8] and fuzzy logic [9] have been widely applied to proper tuning controller parameters of robot manipulators. Successful applications can be found in [10, 11, 12].

Alternatively, a wide range of new (AI) techniques, have submerged [13]. These novel techniques are biological-inspired optimization methods. They iteratively use random elements to transfer one candidate solution into a new better solution with regard to a given fitness function.

The main nature-inspired optimization approaches are Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms.

Some relevant applications of such strategies in robotic field can be found in [14, 15, 16, 17]. Among the variety of evolution algorithms, Particle Swarm Optimization (PSO) algorithm [18, 19] seems to be one of the most promising techniques [20]. Particularly, PSO algorithms have less computational complexities and gives better performances [21].

PSO algorithms were successfully applied in several robotic applications, "For example, see [22, 23, 24, 25]". Particularly, for tuning the controller parameters of robot arms in non constraint environment, several papers were recently proposed, "For example, see [26, 27]". However, for the constrained motion case study, very few papers were devoted to this topic [28, 29].

Based on the lack of results in this framework, this paper proposes an efficient and fast method for fine tuning the controller parameters of robot manipulators in constrained motion using PSO intelligence. The stability of the constrained robotic system is proved using a Lyapunov-based impedance approach whereas the optimal design of the controller parameters are tuned, in offline, by the PSO algorithm using different index performances in both joint and Cartesian spaces.

The remainder of this paper is organized as follows. In Section 2, the stability conditions using the Lyapunov based impedance controller for constrained robotic systems are presented. In section 3, PSO algorithm is exposed. The case study of fine tuning the controller parameters for a 3DOF robotic system constrained to a circular profile are developed in section 4 where the performances of the proposed strategy are shown by simulation results.

2. Impedance control

Let consider the dynamics of a serial-chain *n*-link constrained robotic arm with *n* degrees of freedom described, in the joint space, by:

$$M(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) = U - U_F \tag{1}$$

where θ is the $n\times 1$ vector of joint displacements, $\dot{\theta}$ is the $n\times 1$ vector of joint velocities, $\ddot{\theta}$ is the $n\times 1$ vector of joint accelerations, U is the $n\times 1$ vector of torque inputs and U_F is the $n\times 1$ vector of torques exerted by the contact generalized forces of the manipulator on the environment, $M(\theta)$ is the $n\times n$ symmetric positive definite inertia matrix, $H(\theta,\dot{\theta})$ is the $n\times 1$ vector of centripetal and Coriolis torques and $G(\theta)$ is the $n\times 1$ vector of gravitational torques.

The Cartesian position of the end-effector of the constrained robotic arm is characterized by a vector $X \in R^p$ where p is the task space dimension. Cartesian position is function of the joint displacements such that:

$$X = h(\theta) \tag{2}$$

where $h(\theta): R^n \rightarrow R^p$ denotes the direct kinematics. The time derivative of the direct kinematic model (2) yields the following differential kinematic model:

$$\dot{X} = J(\theta)\dot{\theta} \tag{3}$$

Where $\dot{X}$ is the $p\times 1$ Cartesian velocity vector and $J(\theta) \in R^{pxn}$ is the so-called analytical Jacobian matrix.
The vector of torques due to the contact generalized forces exerted by the manipulator on the environment can be written as:

$$U_F = J(\theta)^T F \tag{4}$$

The time derivative of the differential kinematic model (3) can be written as:

$$\ddot{X} = \dot{J}(\theta)\dot{\theta} + J(\theta)\ddot{\theta} \tag{5}$$

where $\ddot{X}$ is the $p\times 1$ Cartesian acceleration vector and $\dot{J}(\theta)$ is defined by:

$$\dot{J}(\theta) = \frac{d}{dt}J(\theta) \tag{6}$$

Given a desired Cartesian position of the end-effector $X_d \in R^p$ and a desired contact force $F_d \in R^p$, the control problem aim to ensure:

$$\lim_{t\rightarrow t_f} X - X_d = 0 \tag{7}$$

and

$$\lim_{t\rightarrow t_f} F - F_d = 0 \tag{8}$$

The last problem will be solved under the following assumptions:
A1: The entire vectors of force F, Cartesian position X, Cartesian velocity $\dot{X}$ and Cartesian acceleration $\ddot{X}$ are measured.
A2: The Jacobian matrix $J(\theta)$ is assumed to be full ranked and bounded for all $\theta \in R^n$.
A3: A desired impedance model for the contact point between the end-effector of the robotic arm and the constrained environment is imposed such that [30]:

$$Z_d = \frac{F_d - F}{X_d - X} = K_d + B_d s + M_d s^2 \tag{9}$$

where $K_d, B_d, M_d \in R^{pxp}$ are stiffness, damping and inertia matrices, respectively, and s is the Laplace operator.

A4: All control feedback gain matrices, used to solve the control problem are diagonal.

Theorem:

For desired stiffness, damping and inertia matrices $K_d, B_d, M_d \in R^{pxp}$ and if there exist diagonal gain matrices $K_p, K_v, K_f \in R^{pxp}$ such that the following conditions:

$$\begin{aligned} &K_p + (I_{p\times p} + K_f)K_d > 0 \\ &K_v + (I_{p\times p} + K_f)B_d > 0 \\ &M_d = 0 \end{aligned} \tag{10}$$

or

$$\begin{aligned} &K_p > 0 \\ &K_v > 0 \\ &K_f = -I_{p\times p} \end{aligned} \tag{11}$$

are satisfied, then the robotic system described by the dynamical model (1) and the direct kinematic model (2) is asymptotically stable under the constrained force model:

$$F = F_d - K_d(X_d - X) - B_d(\dot{X}_d - \dot{X}) - M_d(\ddot{X}_d - \ddot{X}) \tag{12}$$

and the control law:

$$U = J(\theta)^T[K_p(X_d - X) + K_v(\dot{X}_d - \dot{X}) + K_f(F_d - F) + F_d] + G \tag{13}$$

where $K_p, K_v, K_f \in R^{pxp}$ are position, velocity and force gain matrices, respectively.

Proof:

Let $\Phi \in \Re^n$ and $Y(\Phi) \in \Re^p$ the error vectors defined in the joint and task space, respectively, by:

$$\Phi = \theta - \theta_d \tag{14}$$

$$Y(\Phi) = X(\theta) - X_d \tag{15}$$

Consider, now, the constrained robot system described by the dynamic model (1) for the force design (12) and the control law (13). Using the relations (14) and (15) we can write [31, 32]:

$$M(\Phi)\ddot{\Phi} + H(\Phi,\dot{\Phi}) + J^T(\Phi)K_1Y(\Phi) + J^T(\Phi)K_2\dot{Y}(\Phi) + J^T(\Phi)K_3\ddot{Y}(\Phi) = 0 \tag{16}$$

where

$$\begin{aligned} &K_1 = K_p + (I + K_f)K_d \\ &K_2 = K_v + (I + K_f)B_d \\ &K_3 = (I + K_f)M_d \end{aligned} \tag{17}$$

In order to evaluate the relationship between the dynamics of the constrained robotic system and its energy, we use Lagrange's equation described by [33]:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\Phi}}\right) - \frac{\partial T}{\partial \Phi} + \frac{\partial P}{\partial \Phi} + \frac{\partial D}{\partial \dot{\Phi}} = 0 \tag{18}$$

where $T(\Phi,\dot{\Phi})$ is the kinetic energy of the constrained robotic system defined by:

$$T(\Phi,\dot{\Phi}) = \frac{1}{2}\dot{\Phi}^T M(\Phi)\dot{\Phi} \tag{19}$$

and $P(\Phi)$, $D(\Phi,\dot{\Phi})$ are potential energy and dissipation function respectively. For the Lagrange equation (18) we can prove that [31]:

$$\frac{\partial P(\Phi)}{\partial \Phi} = J^T(\Phi)K_1Y(\Phi) \tag{20}$$

$$\frac{\partial D(\Phi,\dot{\Phi})}{\partial \dot{\Phi}} = J^T(\Phi)K_2\dot{Y}(\Phi) + J^T(\Phi)K_3\ddot{Y}(\Phi) \tag{21}$$

$$H(\Phi,\dot{\Phi}) = \sum_{i=1}^{n}\left(\frac{d\Phi_i}{dt}\frac{\partial M}{\partial \Phi_i}\right)\frac{\dot{\Phi}}{2} \tag{22}$$

Impose, now, to the system (16) to have a Lyapunov Hamiltonian function defined by [33]:

$$V(\Phi,\dot{\Phi}) = T(\Phi,\dot{\Phi}) + P(\Phi) - P(0) \tag{23}$$

The error system (16) is asymptotically stable if $V(\Phi,\dot{\Phi})$ satisfies the following conditions [34]:

$$V(0,0) = 0 \text{ if } \Phi = 0, \dot{\Phi} = 0 \tag{24}$$

$$V(\Phi,\dot{\Phi}) > 0 \text{ if } \Phi \neq 0, \dot{\Phi} \neq 0 \tag{25}$$

$$\dot{V}(\Phi,\dot{\Phi}) < 0 \text{ if } \Phi \neq 0, \dot{\Phi} \neq 0 \tag{26}$$

since $T(0,0) = 0$, then $V(0,0) = 0$ so the first Lyapunov condition (24) is verified. The second Lyapunov condition (25) will be verified if we prove that $V_p(\Phi)$ defined by:

$$V_p(\Phi) = P(\Phi) - P(0) \tag{27}$$

is positive definite [33] since the kinetic energy $T(\Phi,\dot{\Phi})$ is positive definite and $V_p(0) = 0$.

From (27) and (20) we can write:

$$\left[\frac{\partial V_p(\Phi)}{\partial \Phi}\right]=\left[\frac{\partial P(\Phi)}{\partial \Phi}\right]=J^T(\Phi)K_1Y(\Phi) \quad (28)$$

then :

$$\left[\frac{\partial V_p(\Phi)}{\partial \Phi}\right]_{\Phi=0}=J^T(0)K_1Y(0)=0 \quad (29)$$

Let consider the function W given by:

$$W=\frac{\partial}{\partial \Phi^T}\left(\frac{\partial V_p(\Phi)}{\partial \Phi}\right)=\frac{\partial}{\partial \Phi^T}\left(\frac{\partial P(\Phi)}{\partial \Phi}\right) \quad (30)$$

Substituting (20) in (30) we have:

$$W=\frac{\partial}{\partial \Phi^T}\left(J^T(\Phi)K_1Y(\Phi)\right)=w_{ij}+J^T(\Phi)K_1J(\Phi) \quad (31)$$

where:

$$w_{ij}=\left(\frac{\partial J_i}{\partial \Phi_i}\right)^T K_1Y(\Phi)$$

At the equilibrium point we have:

$$[W]_{\Phi=0}=J^T(0)K_1J(0) \quad (32)$$

Based on (32), the function W is positive definite at the equilibrium point if K_1 is positive definite. Hence the second Lyapunov condition (25) is satisfied if K_1 is positive definite.

Using the expression (23) we have:

$$\frac{dV(\Phi)}{dt}=\frac{dT(\Phi,\dot{\Phi})}{dt}+\frac{dP(\Phi)}{dt} \quad (33)$$

From equation (19) we can write:

$$\frac{dT(\Phi,\dot{\Phi})}{dt}=\dot{\Phi}^TM(\Phi)\ddot{\Phi}+\dot{\Phi}^T\frac{dM(\Phi)}{dt}\frac{\dot{\Phi}}{2}=\dot{\Phi}^TM(\Phi)\ddot{\Phi}+H(\Phi,\dot{\Phi}) \quad (34)$$

Furthermore,

$$\frac{dP(\Phi)}{dt}=\dot{\Phi}^T\frac{\partial P(\Phi)}{\partial \Phi} \quad (35)$$

Substituting (20) in (35) we have:

$$\frac{dP(\Phi)}{dt}=\dot{\Phi}^TJ^T(\Phi)K_1Y(\Phi) \quad (36)$$

Substituting (34) and (36) in (33) we obtain:

$$\frac{dV(\Phi,\dot{\Phi})}{dt}=\dot{\Phi}^TM(\Phi)\ddot{\Phi}+\dot{\Phi}^TH(\Phi,\dot{\Phi})+\dot{\Phi}^TJ^T(\Phi)K_1Y(\Phi) \quad (37)$$

From (16) we can write:

$$M(\Phi)\ddot{\Phi}+H(\Phi,\dot{\Phi})+J^T(\Phi)K_1Y(\Phi)=-(J^T(\Phi)K_2\dot{Y}(\Phi)+J^T(\Phi)K_3\ddot{Y}(\Phi)) \quad (38)$$

Substituting the second member of (38) in (37) gives:

$$\frac{dV(\Phi,\dot{\Phi})}{dt}=-\dot{\Phi}^T(J^T(\Phi)K_2\dot{Y}(\Phi)+J^T(\Phi)K_3\ddot{Y}(\Phi)) \quad (39)$$

Using relations (3), (14) and (15) gives:

$$\frac{dV(\Phi,\dot{\Phi})}{dt}=-\dot{Y}^T(\Phi)(K_2\dot{Y}(\Phi)+K_3\ddot{Y}(\Phi)) \quad (40)$$

The third Lyapunov condition (26) is then verified if K_2 is positive definite and K_3 is null. Note that K_3 is null if $I+K_f=0$ or $M_d=0$. The control objectives (7) and (8) are then reached if the conditions (10) or (11) are satisfied. However, to ensure some desired performances, tuning the controller gain matrices K_p and $K_v \in R^{pxp}$ remain a complex and time consuming procedure. This difficult problem will be solved in the next section.

3. PSO-Impedance Controller

3.1 Overview of PSO algorithm

The particle swarm optimization (PSO) method was firstly introduced by Kennedy and Eberhart [18]. It is a relatively new evolutionary algorithm that may be used to find optimal or near optimal solutions in big search space. The PSO method is developed from study on swarm such as fish schooling and bird flocking. It can be easily implemented, and has stable convergence characteristic with good computational efficiency.

During flight, each particle adjusts its position according to its own experience, and the experience of other particles, making use of the best position encounter by itself and its neighbors. The swarm direction of a particle is defined by the set of particles neighboring the particle and its history of experience. The best previous position of ith particle is recorded and represented as $pbest_i$. The best particle among all the particles in the group is represented as gbest [35].

The PSO concept consists of, at each time step, changing the velocity (or acceleration) of each particle i toward its pbest and the gbest position. For the ith particle, a new velocity and position are updated such that [19]:

$$V_i^{(k+1)} = w^{(k)} \cdot V_i^{(k)} + c_1 \cdot \text{rand}() \cdot (\text{pbest}_i - p_i^{(k)}) + c_2 \cdot \text{Rand}() \cdot (\text{gbest}^{(k)} - p_i^{(k)}) \quad (41)$$

$$p_i^{(k+1)} = p_i^{(k)} + V_i^{(k+1)} \quad (42)$$

for $k = 1,2,...,N$ and $i = 1,2,...,M$ and where:

M	Number of particles in a group
N	Maximum number of iterations
$V_i^{(k)}$	Velocity of ith particle at iteration k
$p_i^{(k)}$	Position of ith individual at iteration k
$\text{pbest}_i^{(k)}$	Best position of ith individual at iteration k
$\text{gbest}^{(k)}$	Best position of the group until iteration k
$w^{(k)}$	Inertia weight factor at iteration k
c_1, c_2	Acceleration factors
rand ()	Random numbers between 0 and 1
Rand ()	Random numbers between 0 and 1

In general, the inertia weight is calculated according to the following equation:

$$w^{(k)} = \frac{w_{max} - w_{min}}{N} \cdot k \quad (43)$$

The design steps for implementing the PSO algorithm are given as follow:

1. Initialize a population array of particles with random positions and velocities on d- dimensional space.
2. For each particle, evaluate the desired optimization fitness function.
3. Compare particle's fitness evaluation with its pbest_i. If current value is better than pbest_i, then set pbest_i equal to the current value.
4. Identify the particle in the neighborhood with the best success so far, and assign it to gbest .
5. Change the velocity and position of the particle according to equations (41) and (42) respectively.
6. If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), go to 7 else go to 2.
7. The particle that generates the latest gbest is an optimal controller parameter.

3.2. PSO-Controller tuning

To design the impedance controller (13), only the stability condition (11) will be adopted in this paper to ensure that $M_d \neq 0_{p\times p}$. In this case, the force matrix gain is well defined as $K_f = -I_{p\times p}$. Taking on account of the assumption A_4, the member of each individual of the swarm will be composed of the diagonal elements of the gain matrices $K_p, K_v \in R^{pxp}$. Consequently, whatever the number of degrees of freedom for the constrained robotic arm, the dimension of the individual member will be always 4 for a planar robot and 6 for a three dimensional robot.

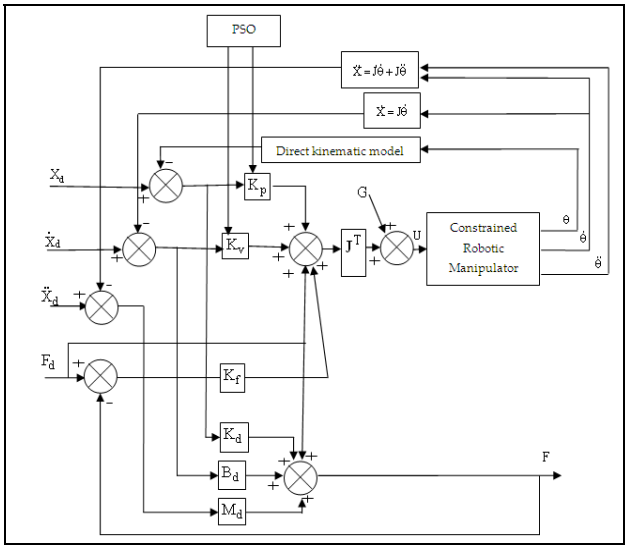

Figure 1. Impedance controller tuned with PSO

Fig.1 illustrates the block diagram of optimal design of the proposed Lyapunov-based impedance controller for constrained robotic arms using PSO algorithm.

3.3. Performance indexes

A fundamental step in applying PSO algorithm remains in choosing the cost function or performance index which is used to evaluate fitness of each particle. In this paper the following cost functions:

1. Mean of Root Squared Error (MRSE)
2. Mean of Absolute Magnitude Error (MAE)
3. Mean of Time-weighted Magnitude Error (MTE)

Based on (7) and (8) we consider furthermore a multiobjectif optimization problem to be solved where the two objectives to be reached are really in conflict. Furthermore the multiobjectif problem will be solved under the stability constrained conditions (11).
We first try to solve the problem in the joint space by considering the following multiobjectif cost functions:

$$MRSE_\theta = \frac{1}{N}\left(\sum_{k=1}^{N}\sqrt{e_k^T \cdot e_k} + \sum_{k=1}^{N}\sqrt{F_k^T \cdot F_k}\right) \quad (44)$$

$$MAE_\theta = \frac{1}{N}\left(\sum_{k=1}^{N}\sum_{i=1}^{n}\left|e_{i,k}\right| + \sum_{k=1}^{N}\sum_{j=1}^{p}\left|F_{j,k}\right|\right) \quad (45)$$

$$MTE_\theta = \frac{1}{N}\left(\sum_{k=1}^{N} t \cdot e_k^T \cdot e_k + \sum_{k=1}^{N} t \cdot F_k^T \cdot F_k\right) \quad (46)$$

where:

$$F_k = \left[F_d(k) - F(k)\right]^T \in \Re^p$$

$$e_k = \left[\theta_{1,d}(k) - \theta_1(k) \quad \cdots \quad \theta_{n,d}(k) - \theta_n(k)\right]^T \in \Re^n$$

$$e_{i,k} = \left|\theta_{i,d}(k) - \theta_i(k)\right| \in \Re, \quad i = 1, \cdots, n$$

$$F_{j,k} = \left|F_{j,d}(k) - F_j(k)\right| \in \Re, \quad j = 1, \cdots, p$$

$\theta_{i,d}(k)$ design the ith element of desired angular displacement vector. $F_{j,d}(k)$ and $F_j(k)$ design the jth element of the desired force vector and the jth element of the force vector, respectively.

The optimization problem will be also solved in the Cartesian space by considering the following multiobjectif cost functions:

$$MRSE_x = \frac{1}{N}\left(\sum_{k=1}^{N}\sqrt{E_k^T \cdot E_k} + \sum_{k=1}^{N}\sqrt{F_k^T \cdot F_k}\right) \quad (47)$$

$$MAE_x = \frac{1}{N}\left(\sum_{k=1}^{N}\sum_{j=1}^{p}\left|E_{j,k}\right| + \sum_{k=1}^{N}\sum_{j=1}^{p}\left|F_{j,k}\right|\right) \quad (48)$$

$$MTE_x = \frac{1}{N}\left(\sum_{k=1}^{N} t \cdot E_k^T \cdot E_k + \sum_{k=1}^{N} t \cdot F_k^T \cdot F_k\right) \quad (49)$$

where:

$$E_k = \left[X_d(k) - X(k)\right]^T \in \Re^p$$
$$E_{j,k} = \left|X_{j,d}(k) - X_j(k)\right| \in \Re, \quad j = 1, \cdots, p$$

$X_j(k)$ and $X_{j,d}(k)$ design the jth element of Cartesian displacement vector and the desired Cartesian displacement vector, respectively.

4. Application to a 3DOF constrained robotic arm

To verify the stability and performances of the proposed controller tuned by PSO intelligence, a 3DOF robotic arm constrained to circular trajectory is considered (see Fig. 2).

4.1. The 3DOF constrained robotic arm

The dynamical model (1), kinematic model (2) and differential kinematic models (3) of the 3DOF constrained robotic arm are defined, respectively, by:

$$M(\theta) = \begin{bmatrix} I_1 + m_1k_1^2 + m_2L_1^2 + m_3L_1^2 & a \cdot c_{12} & b \cdot c_{13} \\ a \cdot c_{12} & I_2 + m_2k_2^2 + m_3L_2^2 & c \cdot c_{23} \\ b \cdot c_{13} & c \cdot c_{23} & I_3 + m_3k_3^2 \end{bmatrix}$$

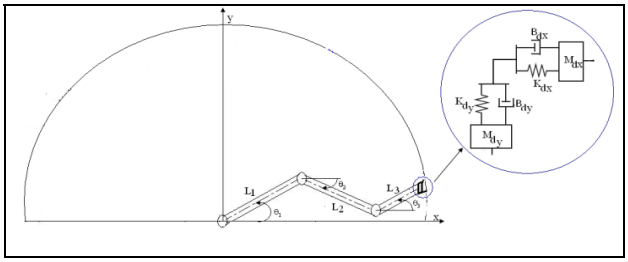

Figure 2. 3DOF manipulator at constrained circular motion

$$H(\theta, \dot{\theta}) = \begin{bmatrix} 0 & a \cdot s_{12} & d \cdot s_{12} \\ -a \cdot s_{12} & 0 & c \cdot s_{23} \\ -d \cdot s_{12} & -c \cdot \sin s_{23} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix}$$

$$G(\theta) = g\begin{bmatrix} (m_1k_1 + m_2L_1 + m_3L_1) \cdot c_1 \\ (m_2k_2 + m_3L_2) \cdot c_2 \\ m_2k_3 \cdot c_3 \end{bmatrix}$$

$$h(\theta) = \begin{bmatrix} L_1 \cdot c_1 + L_2 \cdot c_2 + L_3 \cdot c_3 \\ L_1 \cdot s_1 + L_2 \cdot s_2 + L_3 \cdot s_3 \end{bmatrix}$$

$$J(\theta) = \begin{bmatrix} -L_1 \cdot s_1 & -L_2 \cdot s_2 & -L_3 \cdot s_3 \\ L_1 \cdot c_1 & L_2 \cdot c_2 & L_3 \cdot c_3 \end{bmatrix}$$

where:

$$c_{12} = \cos(\theta_1 - \theta_2),\ c_{13} = \cos(\theta_1 - \theta_3),\ c_{23} = \cos(\theta_2 - \theta_3)$$
$$s_{12} = \sin(\theta_1 - \theta_2),\ s_{23} = \sin(\theta_2 - \theta_3)$$
$$c_1 = \cos\theta_1,\ c_2 = \cos\theta_2,\ c_3 = \cos\theta_3$$
$$s_1 = \sin\theta_1,\ s_2 = \sin\theta_2,\ s_3 = \sin\theta_3$$
$$a = m_2L_1k_2 + m_3L_1L_2$$
$$b = m_3L_1k_2$$
$$c = m_3L_2k_3$$
$$d = m_3L_1k_3$$

The parameters m_i, L_i, k_i and I_i , $(i = 1,2,3)$, design mass, length, position of gravity center and moment of inertia of the links of the 3DOF constrained arm.

4.2 Trajectory generation

The desired Cartesian displacement, speed and acceleration of the constrained robotic system are used as inputs of the control laws (13). It is essential then to generate desired trajectories. Assume that joint motion must begin and end regularly such that:

$$\theta_i(t_0) = \theta_{i0};\ \dot{\theta}_i(t_0) = 0$$
$$\theta_i(t_f) = \theta_{id};\ \dot{\theta}_i(t_f) = 0$$

where t_0 and t_f design initial and terminal times. Reference joint trajectories are chosen as:

$$\theta_{i,d}(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3$$

$$\dot{\theta}_{i,d}(t) = a_{i1} + 2a_{i2}t + 3a_{i3}t^2$$

where:

$$\begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}^{-1} \begin{bmatrix} \theta_{i0} \\ \theta_{id} \\ 0 \\ 0 \end{bmatrix}$$

In the Cartesian space, we impose a circular motion. To realize this profile, we will introduce a sinusoidal signal on each axis. So, the axis references correspond to a harmony excitation in quadratic phase. Therefore, for a circle radius R, the desired end-effector motion of the constrained robotic system will be defined by:

$$X_d = \begin{bmatrix} R\cos(a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3) \\ R\sin(a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3) \end{bmatrix}$$

4.3. Simulation Results

Simulation results are conducted solving the dynamical robotic model (1) using relations (2), (3), (4), (5) and (6) for the control law (13) and the force design (12). The parameter data of the robotic system are given by Table 1.

Link i	m_i(Kg)	L_i(m)	k_i(m)	I_i(Kg m^{-2})
1	1.960	0.321	0.140	0.016
2	1.120	0.253	0.109	0.006
3	0.420	0.187	0.095	0.001

Table 1. Physical parameters of the robotic arm

Desired circular motion is obtained by imposing $\theta_{i0} = [0 \quad 0 \quad 0]^T$ and $\theta_{id} = [\pi \quad \pi \quad \pi]^T$ as initial and final joint positions, $t_0 = 0$ and $t_f = 1s$ as initial and final times and $R = 0.76m$ for the circle Radius. The desired circular trajectory is then given by:

$$X_d = \begin{bmatrix} 0.76\cos(3\pi t^2 - 2\pi t^3) \\ 0.76\sin(3\pi t^2 - 2\pi t^3) \end{bmatrix}$$

PSO intelligence is applied using the variables given by Table 2. To carry out simulation results, two case studies for desired contact force, stiffness, damping and inertia parameters were assessed (see table 3). Fig. 3 and Fig. 4 show the convergence characteristics of cost functions for the two cases with respect to index performances (44)-(49).

Designation	Variable	Value
Number of particles in a group	M	30
Maximum number of iterations	N	50
Minimum inertia weight factor	w_{min}	0.4
Maximum inertia weight factor	w_{max}	0.9
Search interval for k_p	$[k_{p_{min}} \quad k_{p_{max}}]$	$[100 \quad 900]$
Search interval for k_v	$[k_{v_{min}} \quad k_{v_{max}}]$	$[30 \quad 500]$
Search interval velocity of k_p	$[0 \quad k_{p_{max}}/2]$	$[0 \quad 450]$
Search interval velocity of k_v	$[0 \quad k_{v_{max}}/2]$	$[0 \quad 250]$
acceleration constants	c_1 c_2	2

Table 2. Variables of the PSO algorithm

	Case study 1	Case study 2
F_d	$[6 \quad 0]^T$	$[6 \quad 6]^T$
M_d	$diag[0.02 \quad 0.02]$	$diag[0.05 \quad 0.05]$
B_d	$diag[5 \quad 5]$	$diag[10 \quad 10]$
K_d	$diag[20 \quad 20]$	$diag[10 \quad 10]$

Table 3. Case study parameters

	$MRSE_\theta$	MAE_θ	MTE_θ
k_{p_1}	799.3	640.9	757.8
k_{p_2}	839.6	727.2	322.4
k_{v_1}	358	307.2	500
k_{v_2}	297.2	319.5	290.6
Processing time (s)	1961	1965	2093
Best fitness value	0.0369	0.0408	0.0374

Table 4. PSO-best controller tuning in the joint space: case1

	$MRSE_\theta$	MAE_θ	MTE_θ
k_{p_1}	691	646.8	746.6
k_{p_2}	860.2	699.6	380.9
k_{v_1}	450.1	464.8	451.4
k_{v_2}	272.6	458.8	438.3
Processing time (s)	2973	2958	3210
Best fitness value	0.0457	0.0458	0.0582

Table 5. PSO-best controller tuning in the joint space: case2

	$MRSE_X$	MAE_X	MTE_X
k_{p_1}	255	566.1	551
k_{p_2}	801.5	572.4	391.4
k_{v_1}	500	269.3	200.5
k_{v_2}	367.4	222.2	324.5
Processing time (s)	1988	1960	2052
Best fitness value	0.0286	0.0292	0.0355

Table 6. PSO-best controller tuning in the Cartesian space: case1

	$MRSE_X$	MAE_X	MTE_X
k_{p_1}	821.3	493.3	900
k_{p_2}	792.1	487.2	620.5
k_{v_1}	434.2	331.5	485.2
k_{v_2}	364.5	356.2	436.3
Processing time (s)	2969	2966	3305
Best fitness value	0.0342	0.0341	0.0543

Table 7. PSO-best controller tuning in the Cartesian space: case2

Best cost functions and their corresponding controller gain parameters were reported in Tables 4 and 5 using the performance indexes (44)-(46) and in Tables 6 and 7 using the performance indexes (47)-(49). It is clear that optimization procedure with respect to the Cartesian space errors gives better results.

Best solutions are reported in Table 8 and will be used in the following to illustrate the performances of the PSO-impedance controllers. Fig. 5 and Fig. 6 show the convergence of the PSO algorithm to these solutions with respect to the number of iterations. Fig. 7 and Fig. 8 show the evolution of joint positions and velocities using best controller gains whereas Fig. 9 and Fig. 10 illustrate the tracking performances of the desired forces and the smooth profile of control laws.

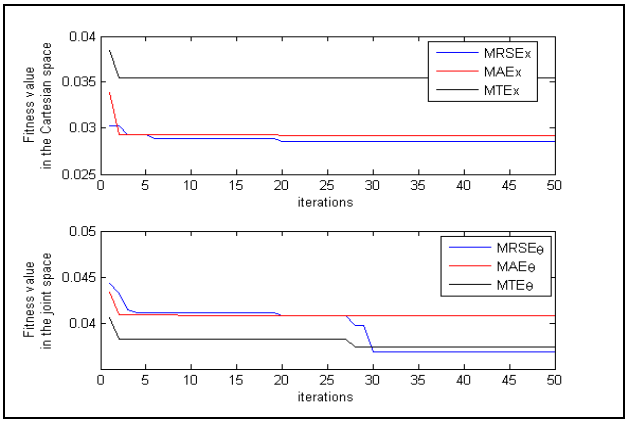

Figure 3. Evolution of fitness functions: case1

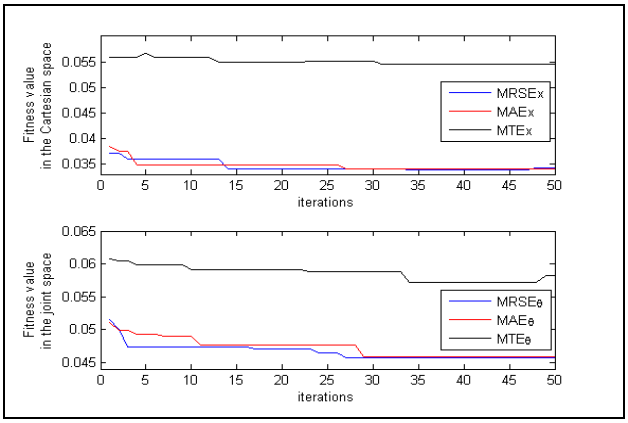

Figure 4. Evolution of fitness functions: case2

	k_{p_1}	k_{p_2}	k_{v_1}	k_{v_2}
Case study 1	255	801.5	500	367.4
Case study 2	493.3	487.2	331.5	356.2

Table 8. Case study controller parameters

Position and force errors are depicted in Fig. 11 and Fig. 12, respectively. Results prove the tracking performances. Fig.13 shows the evolution of the constrained robotic arm to a circular profile.

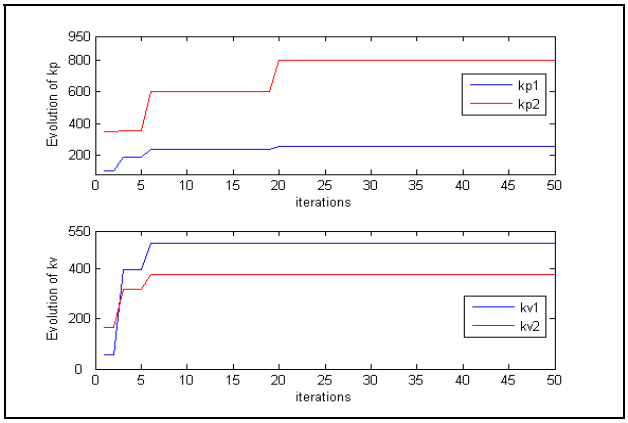

Figure 5. Convergence of controller parameters: case1

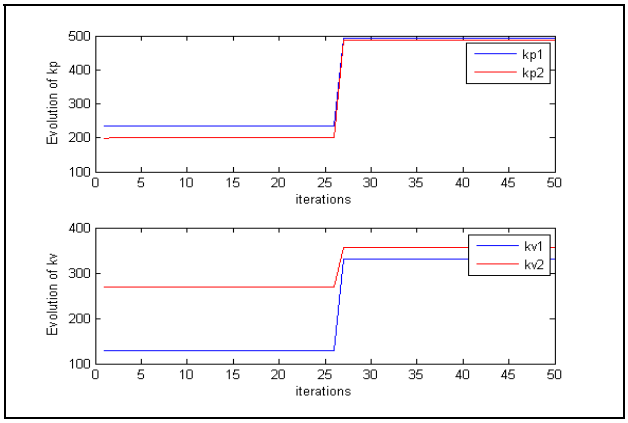

Figure 6. Convergence of controller parameters: case2

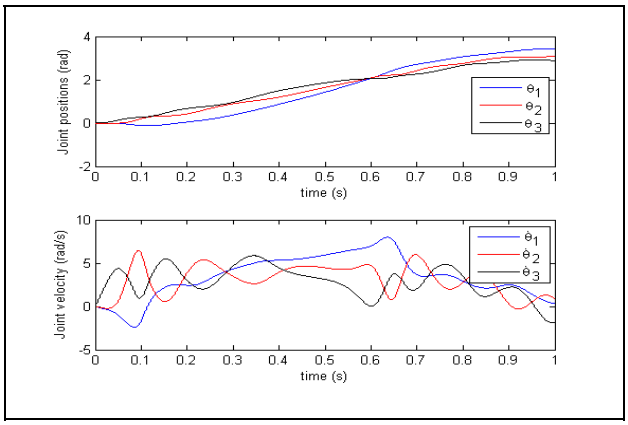

Figure 7. Evolution of joint position and velocity: case1

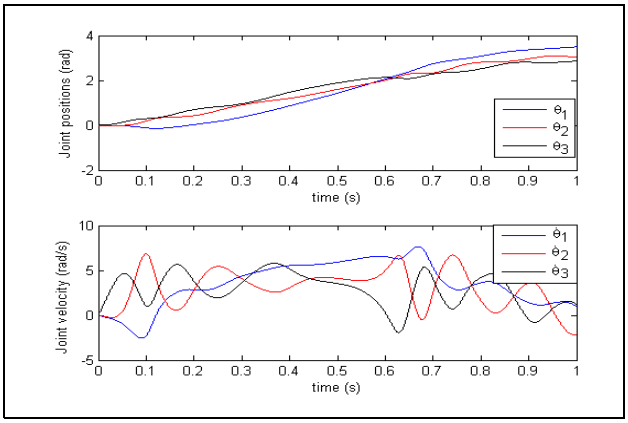

Figure 8. Evolution of joint position and velocity: case2

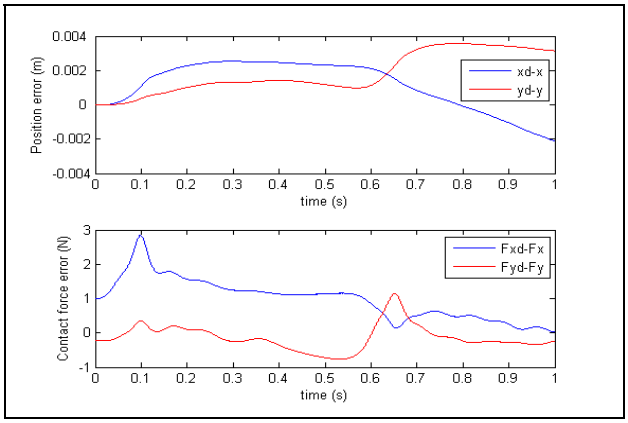

Figure 11. Position and force errors: case 1

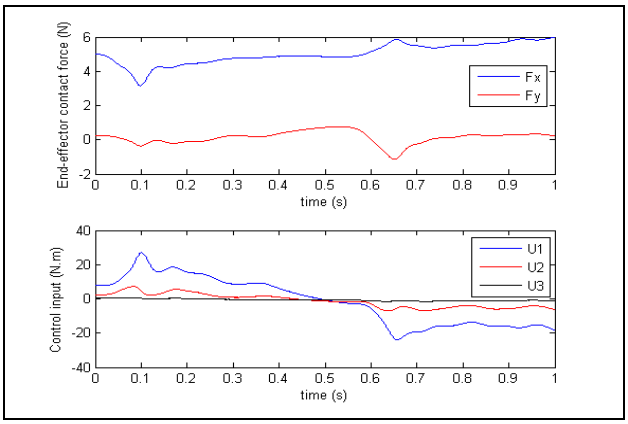

Figure 9. Contact force response and control laws: case1

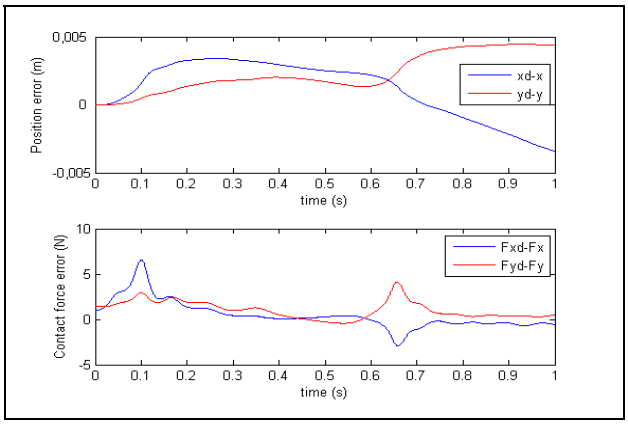

Figure 12. Position and force errors: case 2

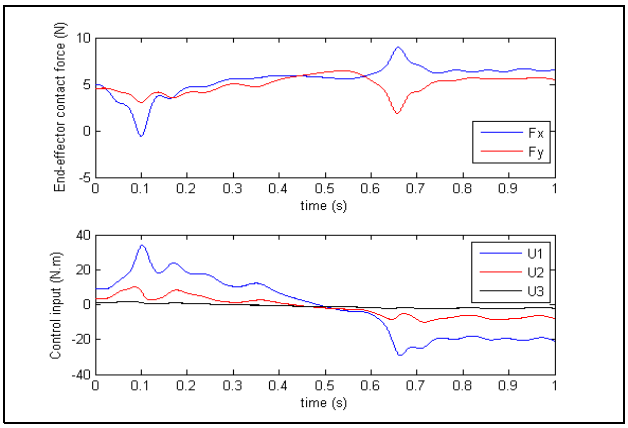

Figure 10. Contact force response and control laws: case2

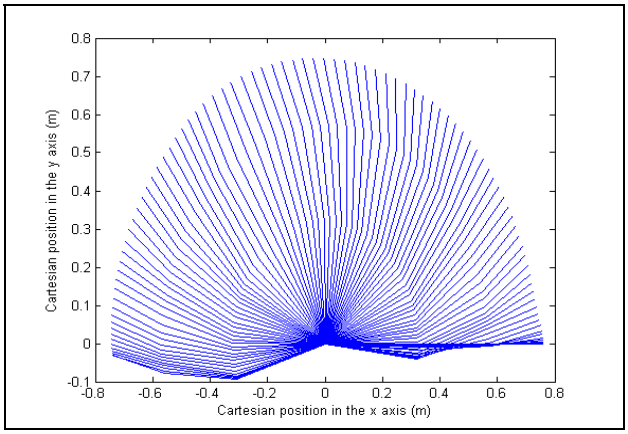

Figure 13. Robotic arm constrained to a circular profile

6. Conclusion

This paper proposes a Lypunov-based impedance controller tuned by a multiobjectif Particle Swarm Intelligence. In order to examine effects of the cost functions on the controller parameter optimization, different performance indexes were used in both join and Cartesian spaces. The PSO- impedance controller was tested on a 3DOF robot constrained to a circular trajectory. The simulation results prove the stability and the performances of the offline tuned controller.

7. References

[1] B. Siciliano, O. Khatib, *Springer Handbook of Robotics*, Berlin: Heidelberg, 2008.

[2] Y. Chen, "Formulation of Automatic Parameter Tuning for Robot Arms," *in Proceedings of the 26th IEEE Conference on Decision and Control*, Los Angeles, 1987, pp. 595-598.

[3] Y. Chen, "Parameter Fine-Tuning For Robots," *IEEE Control Systems Magazine,* vol. 9, pp. 35-40, 1989.

[4] G. Zeng, A. Hemami, "An overview of robot force control," *Robotica*, vol. 15, pp.473-482, 1997.

[5] S. Chiaverini, B. Siciliano, L.Villani, "A Survey of Robot Interaction Control Schemes with Experimental Comparison," *IEEE/ASME Transactions on Mechatronics*, Vol.4, pp. 273-285, 1999.

[6] B. Siciliano, L. Villani, *Robot Force Control*, Boston: Kluwer Academic Publishers, 1999.

[7] R. R. Murphy, *Introduction to AI Robotics*, Cambridge: MIT Press, 2000.

[8] F. L. Lewis, A. Yesildirak, J. Suresh, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Bristol: Taylor & Francis, 1998

[9] M.P. Kevin, Y. Stephen, *Fuzzy Control*, Menlo Park: Addison Wesley Longman, 1998.

[10] K. Kiguchi, K. Watanabe, T. Fukuda "Generation of Efficient Adjustment Strategies for a Fuzzy-Neuro Force Controller Using Genetic Algorithms – Application to Robot Force Control in an Unknown Environment," *Information Sciences*, vol. 145, pp. 113-126, 2002.

[11] M.S. Ju, C.CK. Lin, D.H. Lin, I.S. Hwang, S. M. Chen, "A Rehabilitation Robot with Force-Position Hybrid Fuzzy Controller: Hybrid Fuzzy Control of Rehabilitation Robot," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 13, pp. 349-358, 2005.

[12] Z. Sun, R. Xing, C. Zhao, W. Huang, "Fuzzy Auto-Tuning PID Control of Multiple Joint Robot Driven by Ultrasonic Motors," *Ultrasonics*, vol. 46, pp. 303-312, 2007.

[13] T. Weise. (2009). Global Optimization Algorithms - Theory and Application. Available: http://www.it-weise.de/projects/book.pdf.

[14] D.P. Kwok, F. Sheng, "Genetic Algorithm and Simulated Annealing for Optimal Robot Arm PID Control," in Proceeding of the *IEEE Conference on Evolutionary Computation*, Orlando, 1994, pp. 707-713.

[15] D. Daney, Y. Papegay, B. Madeline, "Choosing Measurement Poses for Robot Calibration with the Local Convergence Method and Tabu Search," *International Journal of Robotics Research, vol.* 24, pp. 501-518, 2005.

[16] M.A.P. García, O. Montiel, O. Castillo, R. Sepúlveda, P. Melin, "Path Planning for Autonomous Mobile Robot Navigation with Ant Colony Optimization and Fuzzy Cost Function Evaluation," *Applied Soft Computing Journal,* vol. *9*, pp. 1102-1110, 2009.

[17] T. Mohammad, B. Vakil, S. Mina, "The Design of PID Controllers for a Gryphon Robot Using Four Evolutionary Algorithms: A Comparative Study," *Artificial Intelligence Review*, vol. 34, pp. 121-132, 2010.

[18] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," in the Proceeding of the *IEEE International Conference of Neural Networks,* Perth, 1995, pp.1942–1948.

[19] Y.Shi, R.C. Eberhart, "A Modified Particle Swarm optimizer." in the Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, 1998, pp.69–73.

[20] P. Riccardo, "Analysis of the Publications on the Applications of Particle Swarm Optimization," *Journal of Artificial Evolution and Applications*, vol. 2008, 2008.

[21] R.C. Eberhart, Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *Lecture Notes Computer Science*, vol. 1447, pp. 611-618, 1998.

[22] A. Chatterjee, K. Pulasinghe, K. Watanabe, K. Izumi, "A Particle-Swarm-Optimized Fuzzy-Neural Network for Voice-Controlled Robot Systems," *IEEE Transactions on Industrial Electronics*, vol. 52, pp. 1478-1489, 2005.

[23] G. G. Rigatos, "Multi-robot motion planning using swarm intelligence." *International Journal of Advanced Robotic Systems*, vol. 5, pp. 139-144. 2008.

[24] Q. Xu, Y. Li, "Kinematic analysis and optimization of a new compliant parallel micromanipulator," *International Journal of Advanced Robotic Systems,* vol. 3, pp. 351-358, 2006.

[25] W. Xu, C. Li, B. Liang, Y. Liu, Y. Xu, "The Cartesian Path Planning of Free-floating Space Robot using Particle Swarm Optimization," *International Journal of Advanced Robotic Systems,* vol. 5, pp. 301-310, 2008.

[26] C. Wei-Der, S. Shun-Peng, "PID Controller Design of Nonlinear Systems Using an Improved Particle Swarm Optimization Approach," *Communications in Nonlinear Science and Numerical Simulation,* vol.15, pp.3632–3639, 2010.

[27] B. Zafer, K. Oguzhan, "A Fuzzy Logic Controller Tuned With PS0 For 2 DOF Robot Trajectory Control," *Expert Systems with Applications,* vol. 38, pp.1017-1031, 2011.

[28] M. Salehi, G.R. Vossoughi, M. Vajedi, M. Brooshaki, "Impedance Control and Gain Tuning of Flexible Base Moving Manipulators Using PSO Method," in Proceedings of the *2008 IEEE International Conference on Information and Automation,* Zhangjiajie, 2008, pp. 458-463.

[29] S. Rubrecht, E. Singla, V. Padois, P. Bidaud, M. De Broissia, "Evolutionary Design of a Robotic Manipulator for a Highly Constrained Environment," *Computational Intelligence*, vol. 341, pp. 109-121, 2011.

[30] N. Hogan, "Impedance Control: An Approach to Manipulators: Part 1, 2, 3," *ASME Journal of Dynamic Systems Measurement and Control*, vol.107, pp. 1-24, 1985.

[31] H. Mehdi, O. Boubaker, "Position/force control for constrained robotic systems: A Lyapunov approach," in Proceedings of the *2010 IEEE International Symposium on Robotics and Intelligent Sensors*, Nogoya, 2010, pp. 299-304.

[32] H. Mehdi, O. Boubaker, "Rehabilitation of a Human Arm Supported by a Robotic Manipulator: A Position/Force Cooperative Control," *Journal of Computer Science*, vol.6, pp. 912-919, 2010.

[33] T.Yabuta, A.J. Chona, G. Beni, "On The Asymptotic Stability of the Hybrid Position/Force Control Scheme for Robot Manipulators," in the Proceedings of the *IEEE International Conference on Robotics and Automation*, Philadelphia, 1998, pp. 338-343.

[34] J.J.E. Slotine, W. Li, *Applied Nonlinear Control*, New Jersey: Prentice Hall, 1991.

[35] I.M. Willjuice, S. Baskar, "Evolutionary Algorithms Based Design of Multivariable PID Controller," *Expert Systems with Applications*, vol.36, pp.9159-9167, 2009.