Estimation and Calibration of Robot Link Parameters with Intelligent Techniques

M. Barati*, A. R. Khoogar** and M. Nasirian***

Abstract: Using robot manipulators for high accuracy applications require precise value of the kinematics parameters. Since measurement of kinematics parameters are usually associated with errors and accurate measurement of them is an expensive task, automatic calibration of robot link parameters makes the task of kinematics parameters determination much easier. In this paper a simple and easy to use algorithm is introduced for correction and calibration of robot kinematics parameters. Actually at several end-effecter positions, the joint variables are measured simultaneously. This information is then used in five different algorithms; least square (LS), particle swarm optimization (PSO) , Genetic algorithms (GA), quadratic particle swarm optimization (QPSO) and simulated annealing particle swarm optimization (Sa_PSO) for automatic calibration and correction of the kinematics parameters. This process was also tested experimentally via a three degree of freedom manipulator which is actually used as a coordinate measuring machine (CMM). The experimental Results prove that the intelligent algorithms are useful for both parameter identification and calibration of link parameters.

Keywords: Calibration, Identification, Genetic Algorithms, Particle Swarm Optimization, Least Square, Robot Manipulator.

1 Introduction

Since the introduction of the first robot manipulators in 1960s, there has always been a demand for kinematic parameters identification and subsequent error correction operations in order to improve the ability of robot manipulators in reaching a specified position consistently and accurately [1]. It has been shown that as much as 95% of robot positioning inaccuracy arises from the inaccuracy in its kinematics model description [2]. Even if it is possible to dismantle a robot manipulator and determine the parameters in detached linkages kinematic frames using accurate measuring machines, the resulting model will still contain some inaccuracies arising from joint and link compliances changing with the manipulator configurations, thermal effects, wear, joint transducer errors, steady state errors in joint positions, inaccurate knowledge of the kinematics parameters, and payload carried by the manipulator [1].

It follows that automatic calibration of robot links parameters that can improve the manipulator accuracy will reduce the kinematics errors [3]. There is a wealth of literature on the kinematics identification and calibration of robotic systems: [2], [4], [5], [6], [7], [8].

A wide account of robot calibration consisting of (i) modeling, (ii) measurement, (iii) identification, and (iv) correction steps are available in [9] and [4].

To calibrate robotic manipulators, Everett et al. [10] presented a new kinematic model for achieving better kinematic representation. Chen and Chao [11], improve the manipulators positioning error by including the non-geometric error in kinematics model. For identification of manipulator link parameters, Stone et al. [12] have introduced the S model. Jang et al. [13] have presented a calibration methodology based on dividing the manipulator workspace into several local regions, and subsequently building a calibration equation using a three dimensional position measurement system consisting of a camera and infrared LED. Newman et al. [14] have reported on the calibration of a Motoman P-8 robot using circle point analysis technique, which requires external hardware to determine the manipulator end point positions in Cartesian space. Driels et al. [6] reported on the kinematic calibration of a PUMA 560 manipulator using a coordinate measuring machine that

Iranian Journal of Electrical & Electronic Engineering, 2011.
Paper first received 24 Aug. 2010 and in revised form 25 Jun. 2011.
* The Author is with the Department of Electrical Engineering, Ferdowsi University, Mashhad, Iran.
E-mail: baraty.m@gmail.com
** The Author is with the Science and Research Branch of the Islamic Azad University, Tehran, Iran.
E-mail: Khoogar@gmail.com
*** The Author is with the Department of Electrical Engineering, Khaje Nasiredin Toosi University, Tehran, Iran.
E-mail: nasir@ee.kntu.ac.ir

provided position and orientation data for randomly selected manipulator configuration. Also for manipulator calibration, Junhong et al. [15] have used the measurement technique by coordinate measuring machine. Renders et al. [5] presented a robot kinematic parameter identification technique based on a maximum likelihood algorithm in a recursive form. Drouet et al. [16] have presented a method to compensate for the geometric and elastic errors of a six-degree of freedom medical robot. An interferometer [17] and laser tracking [14] was, also, used for manipulator endeffector position measurement. For identification of robots kinematics parameters, Horning [18] has introduced and compared four different methods. A closed-loop method has been proposed that obviates the need for pose measurement by forming a manipulator into a mobile closed-loop kinematic chain. Actually kinematics parameters are determined from the joint angle readings alone [9]. Ruibo et al. [19] presents a generic error model, which is based on the product of exponentials (POEs) formula, for serial-robot calibration.

A calibration method is presented for kinematic parameters of space manipulator by Hui Li Zhihong Jiang [20]. This method utilizes the position and orientation information of a fixed target on the space station and adopts rank-one quasi-Newton method to calculate the errors of the kinematic parameters, the position and orientation of the fixed target can be measured by the camera mounted on the manipulator's endeffector. This method can calibrate the manipulator parameters online and has demand in working environment [20].

Calibration of a 6-PRRS parallel manipulator is studied by Yonggang Yang. et al. [21] a compensation method based on kinematic model is proposed. This method uses the D-H modeling method sets up for a 6-PRRS parallel manipulator kinematic model, it then identifies and compensates the error in model using vector chain [21].

As the above literature shows, it is virtually impossible to consider all the sources that contribute to the endeffector positioning errors in a single kinematic identification model of a robot manipulator. However, most of the positioning errors are related to the geometric parameters of linkages [2]. In this study the classical and intelligent identification techniques are used for compensation of the manipulator positioning errors which are produced by the inaccuracies of the geometric parameters. In recent years, stochastic optimization methods have gained increasing attention in parameter optimization of various systems. The most popular techniques are evolutionary computation and the simulating annealing algorithms. Since these methods do not require any gradient information, they are well suited for non-smooth or discontinuous optimization tasks occurring in nonlinear systems [22].

The experimental system employed here is a three degree of freedom manipulator which is actually used as a coordinate measuring machine (CMM). Aspects of this manipulator are depicted in Fig. 1.

To model the kinematics of this manipulator, the Denavit and Hartenberg (DH) standard is used. For measurement of the joint angles, ten bit absolute encoders are used. One of the encoders is shown in Fig. 2. In order to resolve the endeffector position, the tip of the endeffector was rested against a graduated plate which has been graduated with a CNC machine with an accuracy of $\pm 10\mu m$. For identification and calibration of the geometric parameters five different methods were used. Firstly the classical least square estimation technique was employed to determine the numerical values of the kinematic parameters. Then four intelligent techniques were used.

2 Kinematics Model

The schematic of the robot manipulator and the coordinate frames needed to generate a kinematics model are defined in Fig. 3. To model the kinematics of this manipulator, the Denavit and Hartenberg standard is used. In order to use this model, it is necessary to fix a coordinate frame to each linkage [23].

A set of possible body fixed coordinate frames is shown in Fig. 3. The DH parameters for the proposed manipulator are shown in Table 1.

These parameters are provided by the manufacture of the manipulator, in this work they are referred as the nominal parameters.

Fig. 1 The three degree of freedom manipulator.

Fig. 2 The ten bit absolute encoder.

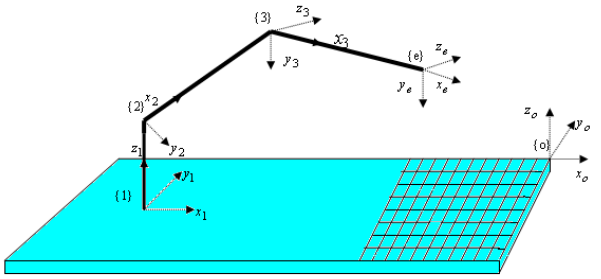


Fig. 3 Schematic representation of the robot manipulator with links coordinate frame.

Table 1 D-H parameters for the proposed manipulator.

Joint/Link	α_i (degree)	a_i (m)	d_i (m)	θ_i
1	0	-0.51	0.10	θ_1
2	-90	0	0.02	θ_2
3	0	0.31	-0.02	θ_3
e	0	0.15	0	θ_e

The homogeneous transformation matrix between two consecutive coordinate frames j and j+1, based on Devavit and Hartenberg convention, is ${}^{j}T_{j+1}$. With this convention, the overall transformation matrix between the base coordinate frame and the frame fixed to the endeffector is written as

$$ {}^{o}T_{e} = {}^{o}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\,{}^{3}T_{e} \tag{1} $$

3 Data Collection

For data collection it is necessary to measure the joint angles for each endeffector position. As mentioned earlier, for measurement of the joint angles ten bit absolute encoders are used. Encoder resolution is ± 0.351 degree. One of the encoders is shown in Figure 2. In order to have better accuracy, encoders with higher resolutions may be used. Now for data collection, we placed the endeffector in different position of the graduated plate and took note of the encoder values.

Note that at least 4 vector measurements are needed in order to estimate the 12 specified parameters [1]. Greater number of measurements would contribute to better convergence of the algorithm and to reduce the effect of measurement noise.

4 Kinematics Parameter Identification

To find the actual value of manipulator kinematic parameters that reduce the endeffector positioning error, we must first develop the relation between the endeffector position and the kinematic parameters, i.e. the forward kinematics equations

$$ P_n = f(\alpha, a, d, \theta) = f(\varphi) \tag{2} $$

where $P_n = [x \quad y \quad z]^T$ is the nominal manipulator end point position vector calculated with the nominal values of the parameters, $\alpha = [\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3]$, $a = [a_0 \quad a_1 \quad a_2 \quad a_3]$, $d = [d_1 \quad d_2 \quad d_3 \quad d_e]$ and the joint variables $\theta = [\theta_1 \quad \theta_2 \quad \theta_3]$. Next, the different identification algorithms used for this identification are described.

4.1 Nonlinear Least Square Method

Based on the nonlinear kinematic model $f(\varphi)$ of the manipulator expressed in (2), the kinematic parameters are estimated by minimizing the sum of the square of the 3×1 positioning error vector ΔP associated with m number of measurements in the objective function,

$$ E = \sum_{k=1}^{m} [\Delta P]^T [\Delta P] \tag{3} $$

where ΔP is expressed by

$$ \Delta P = [\delta x \quad \delta y \quad \delta z]^T = [P_r - P_n]_k \tag{4} $$

in which P_r is the measured (actual) position vector, and $\delta x, \delta y, \delta z$ are the computed position errors in the x, y, and z directions, respectively. This nonlinear least square optimization problem can be solved using either the interior-reflective Newton method or the Levenburg-Marquardt algorithm. These are two efficient optimization algorithms for large-scale nonlinear problems. It has been reported [24] that the former can solve complex nonlinear problems more efficiently than the latter. The interior-reflective Newton method employs the preconditioned conjugate gradients procedure to obtain the approximate solution of a large system of equations.

A good initial guess always helps the estimation algorithm to converge more quickly. Therefore, the nominal values of the parameters are taken as the initial guess for the parameters. The estimation techniques are realized iteratively until the position error is small enough to meet a termination condition. The position error in any particular Cartesian direction is described by the root mean square (RMS) of the position error, which is a 3×1 vector.

$$ \text{RMS} - \text{Position} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(P_r - P_n)_i^2} \tag{5} $$

For the 12 empirically obtained data, the RMS vector was evaluated in x, y, and z directions using both the nominal values of the parameters and estimated value of the parameters via the nonlinear least square technique, and the results are summarized in Table 2. The RMS (x, y, z) indicates the evaluated errors in the x, y, and z directions and $\sum \text{RMS}$ is the vector length of the positioning error.

Percentage of error improvement is evaluated as

$$ \text{Error Improvement (\%)} = \frac{\text{RMS}_n - \text{RMS}_e}{\text{RMS}_n} \times 100 \tag{6} $$

where RMS_n is the RMS positioning error using the nominal parameters. And, the RMS_e is the RMS

positioning error computed using the estimated parameters and element symbols.

The corresponding positioning errors in the Cartesian coordinate are shown in the Figs. 4 and 5.

Table 2 The RMS position errors with nominal and identified parameters.

	RMS (x, y, z)	$\sum$RMS
with nominal Par.	$[2.14 \quad 10.6 \quad 7.7]^T \times 10^{-3}$	0.0204
with estimated Par.	$[1.7 \quad 0.884 \quad 1.6]^T \times 10^{-3}$	0.0041
Improvement (%)	$[20.56 \quad 91.66 \quad 79.22]^T$	79.90

4.2 The PSO Algorithm

This algorithm begins with generation of the initial swarm of particles in which each particle moves about the cost surface with an arbitrary velocity. The particles update their velocities and positions based on the local and global best solutions [25]. If $\vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $\vec{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ are the position and velocity of the i^{th} particle in an n dimension space, then the motion of this particle in the next step is computed as the vector sum of the present position and the velocity vectors as

$$\vec{x}_i^{k+1} = \vec{x}_i^{k} + \vec{v}_i^{k+1} \tag{7}$$

and, the velocity of the particle in the iteration k+1, $\vec{v}_i^{k+1}$, is obtained from the following equation

$$\vec{v}_i^{k+1} = w\vec{v}_i^{k} + c_1 \text{rand} \times (\vec{p}_i - \vec{x}_i^{k}) + c_2 \text{rand} \times (\vec{p}_g - \vec{x}_i^{k}) \tag{8}$$

where k is the iteration number, $\vec{p}_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ is the best particle position in the i^{th} iteration (pbest) and $\vec{p}_g = (p_{g1}, p_{g2}, \ldots, p_{gn})$ is the global best particle position in all iterations so far (gbest), and, c_1 and c_2 are the scaling factors that determine the relative pull of pbest and gbest.

As proposed in [26], the default values of c_1 and c_2 were selected as 2. In the standard PSO, the inertia weighting factor is quite important and it is usually chosen as a decreasing function. At the beginning it is set to $w_{initialize} = 1$ and finally it is reduced to $w_{final} = 0.5$. A linear relation is usually used as a function of iteration

$$w = (w_{initialize} - w_{final})\left(\frac{k_{max} - k}{k_{max}}\right) + w_{final} \tag{9}$$

In the above equation k is the running step number and k_{max} is the maximum number of steps [22].

Next steps summarized the algorithm:

1) Generate the initial values for the particles
2) Update the positions, the velocities and the inertia weighting factor w, in each step according to the cost function

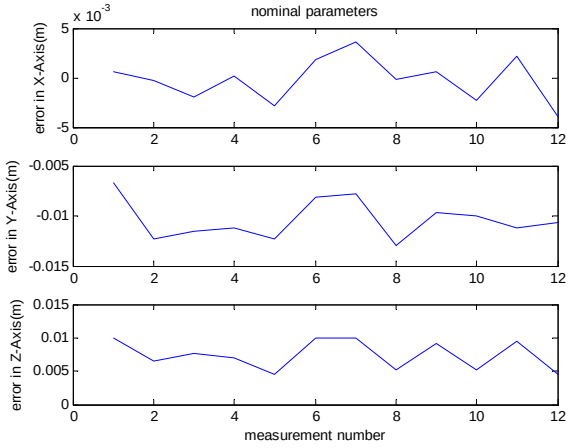


Fig. 4 Position error of endeffector using the nominal parameters.

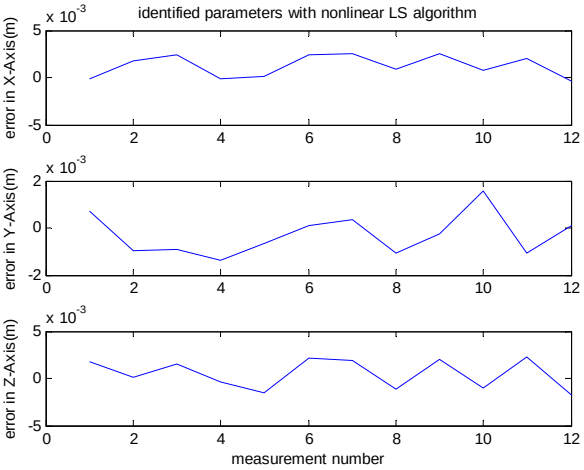


Fig. 5 Position error of endeffector using the identified parameters via nonlinear LS.

3) Repeat the loop until the desired solution is reached

We can consider a cost function for this algorithm similar to the cost function in the least square technique

$$E = \sum_{k=1}^{m} [\Delta P]^T [\Delta P] \tag{10}$$

In this algorithm, the nominal values of the parameters were also used as the initial value.

For the 12 empirically obtained data, the RMS vector was evaluated in x, y, and z directions using both the nominal values of the parameters and estimated value of the parameters via PSO algorithm. The results are summarized in Table 3.

The corresponding positioning errors in the Cartesian coordinate are shown in Figs. 6 and 7.

By comparing the results, we see that the PSO algorithm performs better. Also in this algorithm, the value of the fitness function for the best individual in each generation is shown in Fig. 8.

Table 3 The RMS position errors with nominal and identified parameters by PSO algorithm.

	RMS(x, y, z)	$\sum$ RMS
With nominal par.	$[2.14 \quad 10.6 \quad 7.7]^T \times 10^{-3}$	0.0204
With estimated par.	$[1.1 \quad 0.673 \quad 1.1]^T \times 10^{-3}$	0.0028
Improvement (%)	$[48.6 \quad 93.65 \quad 85.71]^T$	86.27

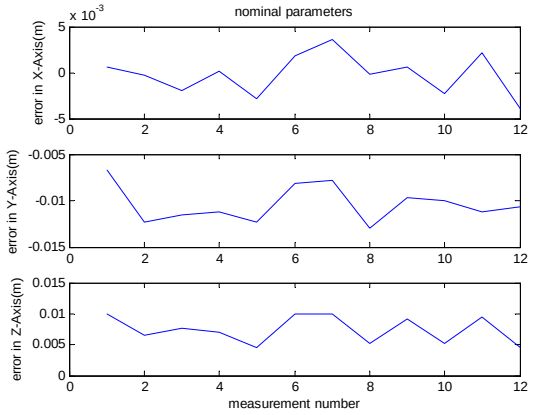


Fig. 6 Position error of endeffector using the nominal parameters.

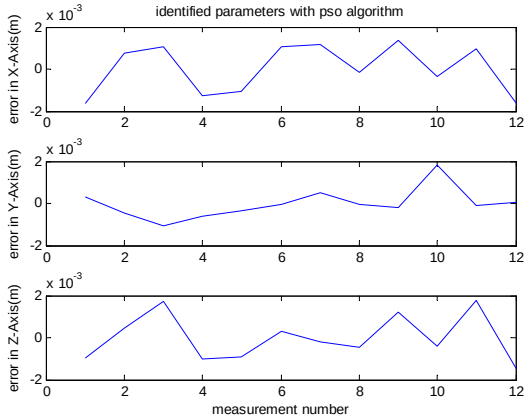


Fig. 7 position error of endeffector using the identified parameters via PSO algorithm.

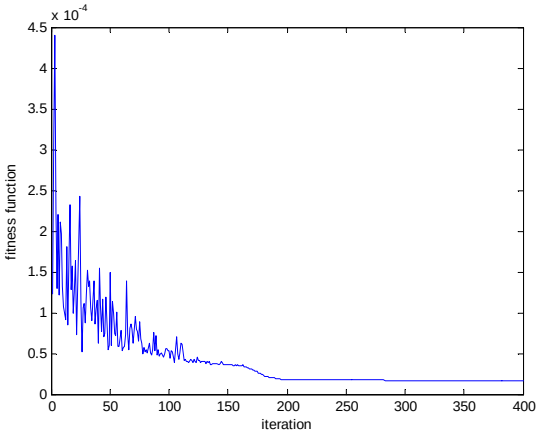


Fig. 8 The value of fitness function per iteration.

4.3 Genetic Algorithm

Genetic algorithms operate based on the theory of evolution in the nature, i.e., the algorithms search for a best solution from the population of potential solutions. In every generation, the better individuals are selected. Successive populations are generated through reproduction, crossover and mutation. In this process, better individuals reproduce in the next generation with a greater probability [27].

When the algorithm begins, the initial population is generated randomly and the fitness of each individual is evaluated from the cost function. If the termination condition is not reached, choose the parents of the next generation based on their fitness functions. The next generation is produced through crossover between parents of the previous generation. A mutation operator is also included. This process is continued iteratively until the desired termination condition is reached [29].

Consider a cost function for this algorithm similar to the cost function used in the least square technique. In this algorithm, the nominal values of the parameters were also used as the initial value.

For the 12 empirically obtained data, the RMS vector was evaluated in x, y, and z directions using both the nominal values of the parameters and estimated value of the parameters via Genetic algorithm, and the results are summarized in Table 4.

The corresponding positioning errors in the Cartesian coordinate are shown in Figs. 9 and 10**.**

The value of the fitness function for the best individual in each generation is shown in Fig. 11.

Table 4 The RMS position errors with nominal and identified parameters via Genetic algorithm.

	RMS(x, y, z)	$\sum$ RMS
With nominal Par.	$[2.14 \quad 10.6 \quad 7.7]^T \times 10^{-3}$	0.0204
With estimated Par.	$[1.2 \quad 0.725 \quad 1.3]^T \times 10^{-3}$	0.0033
Improvement (%)	$[43.92 \quad 93.16 \quad 83.11]^T$	83.82

4.4 The QPSO algorithm

The QPSO algorithm is a simple and modified integrated version of basic PSO (BPSO) and EA. The quadratic crossover operator suggested in [31] is a nonlinear multi parent crossover operator which makes use of three particles (parents) of the swarm to produce a particle (offspring) which lies at the point of minima of the quadratic curve passing through the three selected particles. The nonlinear nature of the quadratic crossover operator used in this work helps in finding a better solution in the search space.

$$\tilde{x}^i = \frac{1}{2} \times \frac{(b^{i^2} - c^{i^2}) \times f(a) + (c^{i^2} - a^{i^2}) \times f(b) + (a^{i^2} - b^{i^2}) \times f(c)}{(b^i - c^i) \times f(a) + (c^i - a^i) \times f(b) + (a^i - b^i) \times f(c)} \tag{11}$$

The computational steps of the QPSO algorithm are given below:

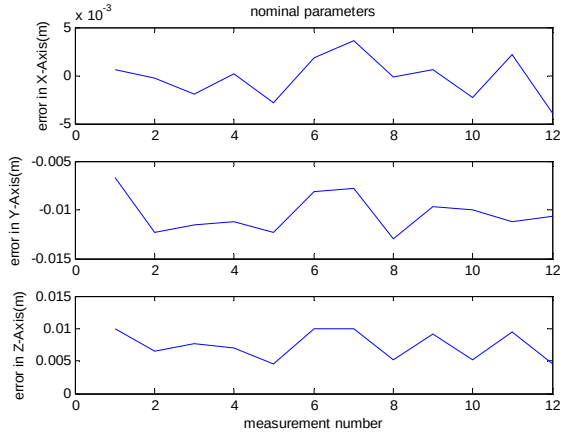


Fig. 9 Position error of endeffector using the nominal parameters.

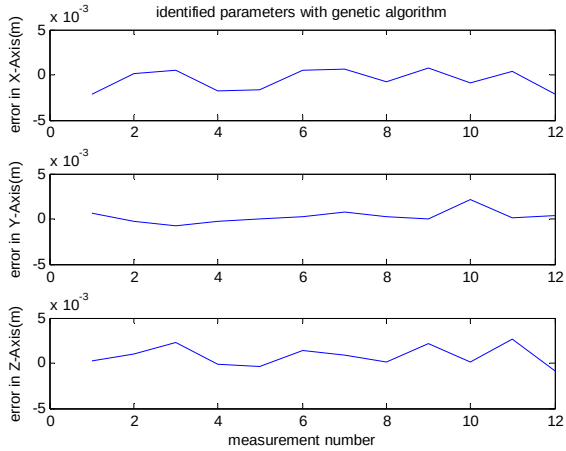


Fig. 10 Position error of endeffector using the identified parameters via Genetic algorithm.

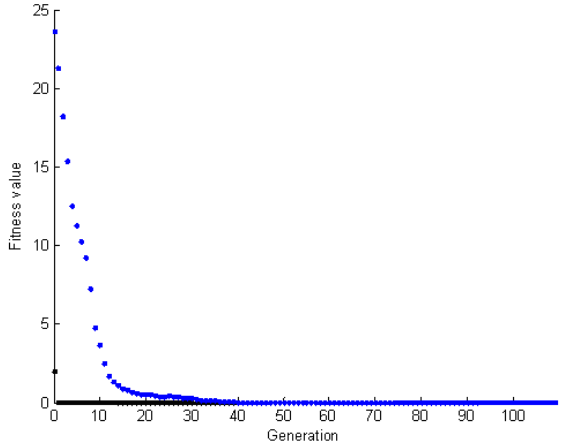


Fig. 11 The value of fitness function per generation.

1-Initialize the swarm
2-For each particle
Update velocity
Update position
Update personal best
Update global best
3-Find a new particle using equation (11)
4-Replace the worst Particle by the new Particle While (Stopping condition is not reached), [31].

We can consider a cost function for this algorithm similar to the cost function in the least square technique. In this algorithm, the nominal values of the parameters were also used as the initial value.

For the 12 empirically obtained data, the RMS vector was evaluated in x, y, and z directions using both the nominal values of the parameters and estimated value of the parameters via QPSO algorithm, and the results are summarized in Table 5.

The corresponding positioning errors in the Cartesian coordinate are shown in Figs. 12 and 13.

Table 5 The RMS position errors with nominal and identified parameters via QPSO algorithm.

	RMS(x, y, z)	$\sum$RMS
With nominal Par.	$[2.14 \quad 10.6 \quad 7.7]^T \times 10^{-3}$	0.0204
With estimated Par.	$[1.1 \quad 0.69 \quad 1.1]^T \times 10^{-3}$	0.0029
Improvement (%)	$[47.06 \quad 93.43 \quad 85.76]^T$	85.67

4.5 The Sa-PSO Algorithm

The idea of simulated annealing algorithm is presented by Metropolis in 1953, and was used in compounding optimization by Kirkpatrick in 1983. It accepts the current optimal solution at a probability after searching, which called Metropolis law. And Sa-PSO algorithm become a global optimal algorithm by using this new acceptance rule, the theory has been proved [32]. The basic idea of simulated-annealing particle swarm optimize algorithm (Sa-PSO) is shown below.

At the beginning, the individual best point and the global best point were accepted by the Metropolis rule, the hypo-best point was accepted at probability, the aim function is allowed to become worse at a certain extent, the acceptance rule was decided by the coefficient T, where T is the anneal temperature. With the T descending, the searching region would be around the best point, the accepted probability of the hypo-best point become small also, when the T descend to the lower limit, the accepted probability of the hypo-best point is zero, the algorithm only accept the best solution as the basic PSO algorithm. The relation between the annealing temperature and the inertial weight was built, the inertial weight changes with the temperature, and then the searching precision was changed following the inertial weight, so the searching speed was increased [31].

$f(x_i)$ is the i^{th} particle solution, p_i is the historical best solution, T is the annealing temperature. The steps of Sa-PSO are shown below:

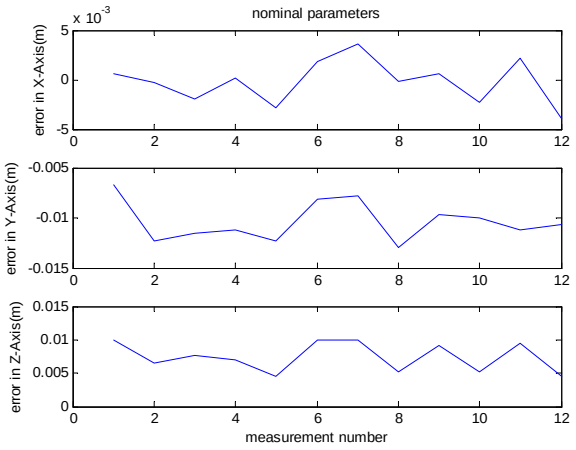


Fig. 12 Position error of endeffector using the nominal parameters.

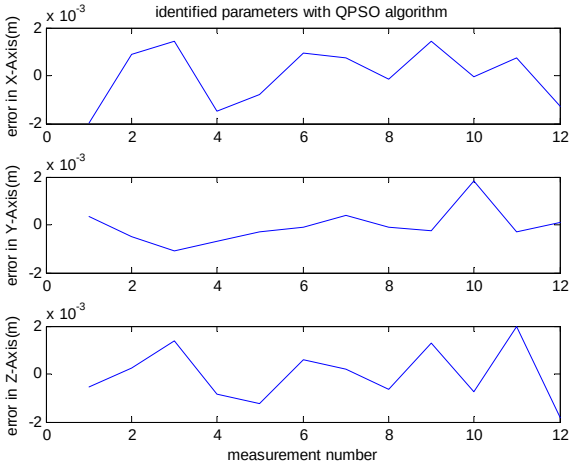


Fig. 13 Position error of endeffector using the identified parameters via QPSO algorithm.

Step 1: Initialize the coefficients, This includes the annealing temperature T and w, c_1, c_2. To initialize the particle swarm, it includes the particle random position and the first speed;

Step 2: Evaluate each particle's adaptive value $f(x_i)$;

Step 3: For each particle, the adaptive value $f(x_i)$ is compared with one of the historical best position p_i if the adaptive value is better than one of p_i. Then, x_i is consider as the best position p_i, otherwise, using the accept-probability law function (12) to decide if this point is accepted.

$$P = \exp(-\Delta f / T) \tag{12}$$

Setp 4: For each particle, the best point p_i itself was compared with the whole best point p_g, if p_i is better than p_g, then reset p_g, otherwise, the global point is acceted according to the probability function (12).

Step 5: The position and speed of each particle were changed following functions (7) and (8) (The functions (7) and (8) define the basic PSO algorithm)[33], several steps later, in order to adjust the temperature T and the inertial weight w, the functions presented in (13) and (14) can be used.

$$T(k) = T_0 \alpha^{k^{1/N}} \tag{13}$$

$$w = w_0 (1 - \frac{T_0 - T(k)}{\beta \times T_0}) \tag{14}$$

Step 6: if the desired condition is not satisfied, then go back to step 2, otherwise stop.

Consider a cost function for this algorithm similar to the cost function used in the least square technique. In this algorithm, the nominal values of the parameters are also used as the initial value.

For the 12 empirically obtained data, the RMS vector was evaluated in x, y, and z directions using both the nominal values of the parameters and estimated value of the parameters via Sa-PSO algorithm, the results are summarized in Table 6.

The corresponding positioning errors in the Cartesian coordinate are shown in Figs. 14 and 15.

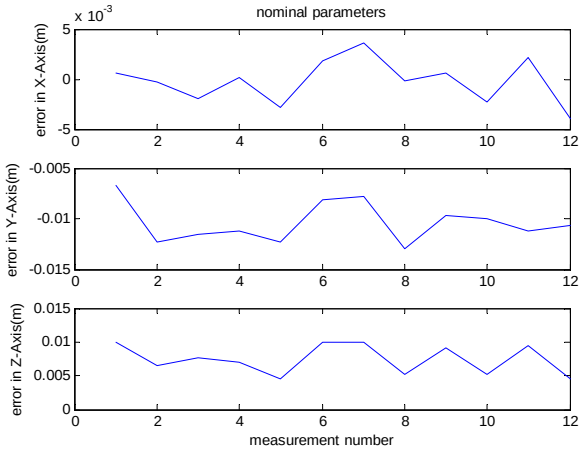


Fig. 14 Position error of endeffector using the nominal parameters.

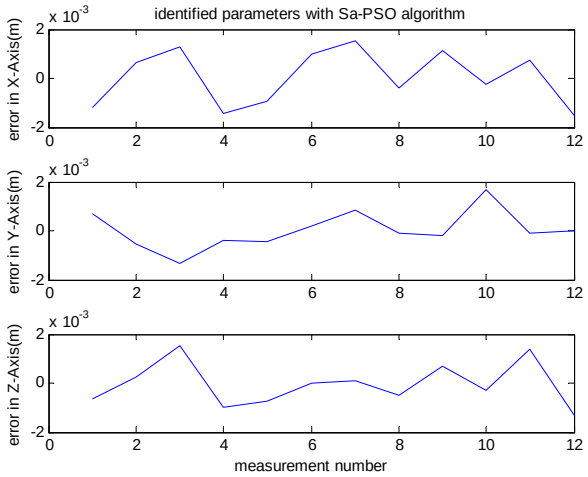


Fig. 15 Endeffector position error using the identified parameters via the Sa-PSO algorithm.

Table 6 The RMS value of position errors with nominal and identified parameters via Sa-PSO algorithm.

	RMS(x, y, z)	$\sum$ RMS
With nominal Par.	$[2.14 \quad 10.6 \quad 7.7]^T \times 10^{-3}$	0.0204
With estimated Par.	$[1.1 \quad 0.74 \quad 0.85]^T \times 10^{-3}$	0.0027
Improvement (%)	$[49.22 \quad 92.97 \quad 88.92]^T$	86.85

Table 7 The Nominal and Identified parameters using the LS method.

Identified parameters	Nominal	L S
a_0(cm)	-51	- 51.62
a_1(cm)	0	1.5
a_2(cm)	31	31.14
a_3(cm)	14	14.99
α_0(degree)	0	0.2521
α_1(degree)	-90	-88.9286
α_2(degree)	0	-3.9534
α_3(degree)	0	-9.683
d_1(cm)	10	10.1
d_2(cm)	2	1.91
d_3(cm)	-2	-1.89
d_e(cm)	0	1.12
Iteration	--	5
Final value of the fitness function	--	3.4×10^{-5}
Elapsed time of the algorithm	--	1.039 Sec

Table 8 The Identified parameters using the Genetic Algorithm and Particle Swarm Optimization.

Identified par.	PSO	GA
a_0(cm)	-52.12	-52.10
a_1(cm)	0.38	7.03
a_2(cm)	30.91	30.53
a_3(cm)	15.37	14.34
α_0(degree)	0.8422	0.8422
α_1(degree)	-89.9885	-87.3578
α_2(degree)	-4.1138	-4.1769
α_3(degree)	0.1604	2.6643
d_1(cm)	10.20	10.25
d_2(cm)	2.37	3.11
d_3(cm)	1.3	-2.10
d_e(cm)	0.25	-0.21
Iteration	400	800
Final value of the fitness function	0.68×10^{-6}	5.11×10^{-6}
Elapsed time of the algorithm	3.65 Sec	3.21 Sec

Nominal parameters and Identified parameters using the Least Square technique are summarized in Table 7.

The Identified parameters using the Genetic Algorithm and Particle Swarm Optimization are summarized in Table 8.

Identified parameters using the QPSO Algorithm and Sa-PSO Algorithm are summarized in Table 9.

Table 9 Identified parameters using the QPSO Algorithm and Sa-PSO Algorithm.

Identified par.	QPSO	Sa-PSO
a_0(cm)	-52.16	-52.20
a_1(cm)	1.43	8.8
a_2(cm)	30	30.53
a_3(cm)	15.19	15.21
α_0(degree)	0.8537	0.7678
α_1(degree)	-91.2605	-89.6963
α_2(degree)	-4.068	-4.0795
α_3(degree)	-0.636	2.6643
d_1(cm)	10.18	10.14
d_2(cm)	1.94	2.37
d_3(cm)	-1.09	-1.62
d_e(cm)	0.4	0.61
Iteration	400	108
Final value of the fitness function	1.68×10^{-6}	8.11×10^{-6}
Elapsed time of the algorithm	3.87 Sec	5.43 Sec

5 Conclusions

In this study, the classical technique of the least square and four intelligent algorithms i.e. genetic algorithms, particle swarm optimization algorithm, QPSO and Sa-PSO was used for identification and calibration of a manipulator kinematics parameters. Numerical and experimental results demonstrate that these techniques are effective in reduction of positioning error.

Advantages of using intelligent methods in comparison with classical calculus based methods are,

- They do not require complex derivative evaluations
- Their applications are simple
- They don't get caught in local minima as easily as the classic method

The obvious advantage of the proposed algorithm is that it does not require very advance equipments for data collection. Inexpensive experimental devices can be used to obtain valuable kinematic parameters. The proposed algorithms were able to compensate as much as 87% of the positioning error.

Using other objective functions in the intelligent methods may improve the identification results.

Using better encoders with higher resolutions can improve the algorithms performance. Other sources of

error, thermal errors, joint transducer errors, steady state errors in the joint positions and …, can be incorporated in the calibration model in future works.

References

[1] Alici G. and Shirinzadeh B., "A Systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing", *Elsevier Journal of Mechanism and Machine Theory*, Vol. 40, No. 8, pp. 879-906, 2005.

[2] Zhong X. L. and Lewis J. M., "A New Method for Autonomous Robot Calibration", *IEEE Int. Conf. on Robotic and Automation*, pp. 1790-1795, 1995.

[3] Abderrahim M., Khamis A., Garrido S. and Moreno L., *Accuracy and Calibration Issues of Industrial Manipulators, Industrial Robotics: Programming, Simulation and Application*, Edited by: Low Kin Huat. Germany, 2006.

[4] Roth Z. S., Mooring B. W. and Ravani B., "An Overview of Robot Calibration", *IEEE Journal of Robotics and Automation*, Vol. 3, No. 5, pp. 377-384, 1987.

[5] Renders J. M., Rossignol E., Becquet M. and Hanus R., "Kinematic Calibration and Geometrical Parameter Identification for Robots", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 6, pp. 721-732, 1991.

[6] Driels M. R., Swayze W. and Potter U. S., "Full-pose calibration of a robot manipulator using a coordinate-measuring machine", *International Journal of Advanced Manufacturing Technology*, Vol. 8, No. 1, pp. 34-41, 1993.

[7] Alici G. and Shirinzadeh B., "Laser interferometry based robot position error modeling for kinematic calibration", *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, pp. 3588-3593, 2003.

[8] Hayati S. A., "Robot arm geometric link parameter estimation", *Proc. 22th IEEE Decision and Control Conf.*, pp. 1477-1483, 1983.

[9] Maric P. and Potkonjak V., "Geometrical Parameter Estimation for Industrial Manipulators Using Two-step Estimation Schemes", *Journal of intelligent and Robotic System*, Vol. 24, No. 1, pp. 89-97, 1999.

[10] Everett L., Driels M. and Mooring B., "Kinematic Modeling for Robot Calibration", *IEEE Int. Conf. on Robotic and Automation*, pp. 183-189, 1987.

[11] Chen J. and Chao L. M., "Positioning Error Analysis for Robot Manipulator with all Rotary Joints", *Pro. IEEE Int. Conf. on Robotics and Automation*, pp. 1011-1016, 1986.

[12] Stone H. W., Sanderson A. C. and Neumann C. P., "Arm Signature Identification", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 41-47, 1986.

[13] Jang J. H., Kim S. H. and Kwak Y. K., "Calibration of geometric and non-geometric errors of an industrial robot", *Robotica*, Vol. 19, pp. 311-321, 2001.

[14] Newman W. S., Birkhimer C. E., Horning R. J. and Wilkey A. T., "Calibration of a Motoman P8 robot based on laser tracking", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3597-3602, 2000.

[15] Junhong J., Lining S. and Lingato Y., "A New Pose Measuring and Kinematics Calibration Method for Manipulators", *IEEE Int. Conf. on Robotic and Automation*, pp. 4925-4930, 2007.

[16] Drouet P., Dubowsky S., Zeghloul S. and Mavroidis C., "Compensation of geometric and elastic errors in large manipulators with an application to a high accuracy medical system", *Robotica*, Vol. 20, No. 3, pp. 341-352, 2002.

[17] Newman W. S. and Osborn D. W., "A New Method for Kinematic Parameter Calibration via Laser Line Tracking", *IEEE Int. Conf. on Robotic and Automation*, Vol. 2, pp. 160-165, May 1993.

[18] Hayati S. A., "Robot arm geometric link parameter estimation", *Proc. 22th IEEE Decision and Control Conf.*, pp. 1477-1483, 1983.

[19] Ruibo H., Yingjun Z., Shunian Y., Shuzi Y., "kinematic-Parameter Identification for Serial-Robot Calibration Based on POE Formula", *IEEE Transactions on Robotics*, Vol. 26, No. 3, pp. 411-423, 2010.

[20] Hui L., Zhihong J. and Qiang Huang Y. H., "Vision-based space manipulator online self-calibration ", *IEEE International conference on Robotics and Biomimetics (ROBIO)*, pp. 1768-1772, 2009.

[21] Yonggang Y., Yubin L., Yongsheng P. J. and Shi W. L., "An Calibration of a 6-PRRS parallel manipulator using D-H method combined with vector chain", *International Conference on Mechatronics and Automation (ICMA)*, 2009.

[22] Sadoghi H. and Effati S., "Eigenvalue Spread Criteria in the Particle Swarm Optimization algorithm for Solving of Constraint Parametric Problem", *Elsevier Journal of Mathematics and computation*, Vol. 192, No. 1, pp. 40-50, 2007.

[23] Graig J. J., *Introduction to Robotics*, Addition,-Weseley, 2nd edition, 1989.

[24] Coleman T. F. and Li L., "An interior, trust region approach for nonlinear minimization subject to bounds", *Journal of Optimization*, Vol. 6, No. 1, pp. 418-445, 1996.

[25] Kennedy J. and Eberhart R., "Particle Swarm Optimization", *Proc. IEEE Int. Conf. Neural Network*, pp. 1942-1948, 1995.

[26] Kennedy J., Eberhart R. and Shi Y., *Swarm Intelligence*, Morgan Kaufman Publishers, USA, 2001.

[27] Poorzaker S. A., *Artificial intelligence and artificial neural network & algorithm genetic*, Nedaye Sabze Shomal Publisher, 1th edition. 2006.

[28] Ali M. M. and Torn A. "Population Set Based Global Optimization Algorithms", *Computer & Operations Research journal.* Vol. 31, No. 10, pp. 1703-1725, Sept 2004.

[29] Zahiri S., Rajabi Mashhadi H. and Seyedin S. A., "Intelligent and Robust Genetic Algorithm Based Classifier", *Iranian Journal of Electrical & Electronic Engineering (IJEEE)*, Vol. 1 , No. 3, pp. 1-9, 2005.

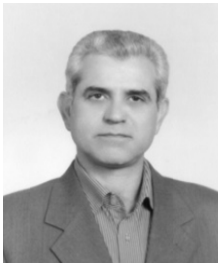
[30] Lucas C., Tootoonchian F. and Nasiri-Gheidari Z., "Multi-Objective Design Optimization of a Linear BrushlessPermanent Magnet Motor Using Particle Swarm Optimization (PSO)", *Iranian Journal of Electrical & Electronic Engineerin (IJEEE)*, Vol. 6, No. 3, pp. 183-189, 2010.

[31] Pant R. and Thangaraj A. A., "A new PSO Algorithm with Crossover Operator for Global Optimization Problems" *Springer Innovations in Hybrid Intelligent Systems (ASC)*, Vol. 44, No. 1 , pp. 215-222, 2007.

[32] Chaojun D. and Zulian Q., "Particle Swarm Optimization Algorithm Based on the Idea ofSimulated Annealing", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 6, No.10, pp. 152-157 ,Oct. 2006.

[33] Lucas C., Nasiri-Gheidari Z. and Tootoonchian F., "Using Modular Pole for Multi-objective Design Optimization of a Linear Permanent Magnet Synchronous Motor byParticle Swarm Optimization (PSO)", *Iranian Journal of Electrical & Electronic Engineering (IJEEE)*, Vol. 6, No. 4, pp. 214-223, 2010.

Maryam Barati was born in Sabzevar, Iran in 1984. She received her B.S degree from Ferdowsi University of Mashhad in 2006. She graduated in Control engineering in M.S in 2008. She is interested in intelligent technique and She is also interested in Robotic. She works in Sairan Spatial Industrial Group until now.



Ahmad R. Khoogar received three degrees from The University of Alabama including a Ph.D. degree in Robotics and Control in 1989. After that he worked in the Space System Branch of the MacDonnell Douglas Space System Company until 1991. He is currently a Assistant Professor in the Science and Research Branch of the Azad University in Tehran and he is also the director of an Industrial Automation Laboratory. He has supervised more than 40 MS and Ph.D. Program's Thesis and dissertations during this time. He has Published over 40 technical papers, Authored a book in "Control System Design Using Matlab" and Translated the book "Flight Dynamics" to the Persian language. His research interests span a broad interdisciplinary curriculum ranging from robotics, Intelligent System learning, planning and optimization in autonomous systems to Industrial Vibrations.

Mehrzad Nasirian received his M.S degree from Iran University of science and technology (IUST) in 1995. He graduated in Ph.D. degree in Control engineering from Khaje Nasiredin Toosi in 2007. He worked in the Search Site in surface effect from 1993 until 1996. After that he worked in the didactic and Research institution of defence ministry from 1999 until 2003 after that he works in Sairan Spatial Industrial Group until now. He has Published 17 technical papers and 3 journals until now. His research interests to satellite, ground station and robotics.