Advances in Engineering Software 78 (2014) 60–66

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

System identification and control of robot manipulator based on fuzzy adaptive differential evolution algorithm

Rawaa Dawoud Al-Dabbagh [a,b], Azeddien Kinsheel [c], Saad Mekhilef [d,*], Mohd Sapiyan Baba [e], Shahaboddin Shamshirband [f]

[a] Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[b] Department of Computer Science, Faculty of Science, University of Baghdad, Iraq
[c] Department of Mechanical and Industrial Engineering, Faculty of Engineering, University of Tripoli, Libya
[d] Power Electronics and Renewable Energy Research Laboratory (PEARL), Department of Electrical Engineering, University of Malaya, Malaysia
[e] Gulf University of Science and Technology, Kuwait
[f] Department of Computer Science, Chalous Branch, Islamic Azad University (IAU), 46615-397 Chalous, Mazandaran, Iran

ARTICLE INFO

Article history:
Received 17 July 2014
Received in revised form 12 August 2014
Accepted 24 August 2014

Keywords:
Robot manipulator
Adaptive differential evolution
Fuzzy logic
Parameter estimation
FADE algorithm
Parameter control

ABSTRACT

A requirement for new robotic manipulators is the ability to detect and manipulate objects in their environments. Robotic manipulators are highly nonlinear systems, and an accurate mathematical model is difficult to obtain using conventional techniques. Therefore, an efficient technique is required to deal with these types of complex and dynamic systems. Differential Evolution (DE) algorithm is a very powerful optimization technique and has become popular in many fields. Arguably, it is now one of the most predominant stochastic algorithms for real-parameter optimization. However, DE is very sensitive to its control parameters of the mutation operation (F) and crossover operation (CR) in such a way that their fine tuning greatly affect DE performance. Fuzzy Adaptive DE (FADE) algorithm is one of the well known adaptive DE variants that show superiority and reliability in solving different types of optimization problems. The objective of this article is to develop a new dynamic parameter identification framework to estimate the barycentric parameters of the CRS A456 robot manipulator based on FADE. The simulation results presented in this paper show the effectiveness of the FADE method over other conventional techniques, transcending the limits of the existing state-of-the-art algorithms in solving the problem of robot.

1. Introduction

There are many industrial applications where the robot manipulator is required to carry out precise task with high accuracy and repeatability. Recently, the application of robotic technology in clinical medicine has been a very active research area. For instance, in surgical operations the robot manipulator serves as an assistant to the doctor or as an extension of the doctor capabilities [1,2]. These kinds of advanced robot applications require an accurate model of the robotic system, which in turn, requires sufficiently accurate knowledge of the parameters of robot dynamics to be applied in advanced control system design, preoperative planning, process supervision, and simulation and training.

Dynamic models of robot arms used in model-based control schemes are designed in terms of various inertial and friction parameters that must be either measured directly or determined experimentally. However, direct measurements of such characteristics are rather impractical or even impossible in many cases. Inertial parameters of robot links cannot be measured without dismantling the robot arm, while highly nonlinear inherent phenomena at robot joints cannot be directly quantified. Therefore, models describing nonlinear effects such as friction should be addressed in conjunction with methods of determining parameters of the dynamic model of the arm based on experiments, in order to fully identify the dynamic model of the robot arm [3].

There are many traditional methods that have been used for dealing with dynamic robot parameter identification including Kalman filter [4] and least square method [5,6]. However, some model parameters such as link mass and link lengths cannot be easily measured using these methods especially with the effect of noise factor, or in other words their measurements relatively difficult [7]. Moreover, these traditional techniques are relatively effective for a class of specific issues. For example, the structural model is reliable but the data has limited accuracy. Furthermore, they depend on unrealistic assumptions that models must be

* Corresponding author.
E-mail address: saad@um.edu.my (S. Mekhilef).

http://dx.doi.org/10.1016/j.advengsoft.2014.08.009
0965-9978/© 2014 Elsevier Ltd. All rights reserved.

unimodal, continuous and derivable. These methods sometimes converge slowly, and sometimes at local optimum, or even not at all.

Recently, there have been intelligent proposed methods for estimation based on the use of universal approximations such as fuzzy logic and neural network methodologies. These methods seem to be very attractive because in the ideal case they allow the modeling of the dynamic effects even 'bad'-modeled, for example, friction. In recent years, Evolutionary algorithms such as Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) have been studied extensively. They have been used to improve the dexterity of robot manipulators in many fields such as control, parameter identification, robot design and planning [8–10]. They have been known to be better suited for noisy, discontinuous functions because they require no knowledge or gradient information about the response surface. This ability of Evolutionary algorithms has encouraged researchers to use these methods in order to moderate the difficulties of noise and nonlinearity that often arise in dynamic models. GAs is better suited for noisy, discontinuous functions because there is no requirement for a derivative in the fitness function. Moreover, GAs accumulate information about the system during the search process, which makes them more desirable than the traditional numerical methods [11] through the use of real-coded GA to estimate friction and torque sensor model parameters. The simulation approach demonstrates the effectiveness of the GA. By identifying the parameters, the position tracking error and the velocity tracking of the joint is enhanced. The performance of GA has been also analyzed and evaluated in optimizing the precision of kinematic parameters of the robot manipulator by developing a forward calibration algorithm which is based on GAs. The main problem for this approach is to find a good mathematical correction function and in [12] a suggestion has been made to enhance the accuracy of the robot manipulator by using some new techniques such as ANN and Fuzzy Logic technique.

Differential Evolution (DE) Algorithm is a new evolutionary approach proposed by Storn and Price in 1996 [13,14] to minimize nonlinear and non-differentiable continuous space functions. Price and Storn presented this algorithm to optimize a variety of problems. Similar to GA, it has been applied to various fields successfully. So far, there has been no attempt to optimize the design parameters of manipulator by which performance variations will be minimal. In [15] a modification in differential evolution optimization technique is proposed to incorporate the effect of noise in the optimization process and obtain the optimal design of manipulator, which is insensitive to noise. In this optimization process, the kinematic and dynamic models of the manipulators are used. The results indicate that the DE converges quickly with fewer generations and function evaluations than GA. Hence, fast performance of DE indicates that this approach can be a viable optimization technique. However, the performance of DE is still sensitive to its control parameters such as mutation factor (*F*) and crossover rate (*CR*).

Recently, the development of adaptive DE has shown more reliable performance than DE with manual settings [16–18]. Fuzzy Adaptive Differential Evolution algorithm (FADE) is one of the well-known adaptive DE variants. It is implemented by applying a mechanism in which the crossover and the mutation control parameters (*F* and *CR*) are both adaptive using fuzzy logic-based controllers; the input signal for the fuzzy system has been calculated from the population mean square diversity. In this algorithm, Fuzzy system plays a key role in updating the control parameters of DE as well as increasing the convergence rate.

In this paper, the application of FADE algorithm is proposed to estimate the barycentric parameters of the CRS A456 robot manipulator. This algorithm is used to off-line estimate the optimal parameters of the inverse dynamic model of the CRS A465 robot arm, which are expected to be insensitive to noise.

This paper is organized as follow. The detailed description of the CRS A465 robot arm and its barycentric parameters are presented in Section 2. The complete steps and structure of FADE algorithm is described in Section 3. Results and discussion of applying FADE algorithm as an estimator of the CRS A465 robot arm barycentric parameters is presented in Section 4. Section 5 concludes the paper.

2. Dynamic model of the CRS A456 robot manipulator

The CRS A465 arm considered in this work is used as a slave robot in a research cell for orthopedic robot-assisted surgery (see Fig. 1).

In this application, the end effector of the arm carries the surgical tool – the "drilling/machining tool". Due to the symmetry of the drilling tool, only five degrees of freedom is required. Therefore, only the first five joints of the arm are considered to be the subject for the modeling task in this work.

The equation of motion for the robot is developed using the *L–E* formulation. The *L–E* is non-recursive method that allows the development of the robot model using a set of equations derived from the energy model [19]. Based on this formulation the torque acting on any joint axis is:

$$\tau_i = \sum_{j=1}^{N} D_{ij}(q,\chi)\ddot{q} + \sum_{j=1}^{N}\sum_{k=1}^{N} H_{ijk}(q,\chi)\dot{q}_j\dot{q}_k + G_i(q,\chi) + \tau_{fi} \tag{1}$$

where τ_i is the torque acting on joint *i*, *i* = 1,2...*N*, *N* is the number of degrees of freedom, $q, \dot{q}, \ddot{q}$ are the position, velocity and acceleration of robot joints, respectively, χ is the model parameters, D_{ij} is the effective and coupling inertia, H_{ijk} is the centripetal and Coriolis effect, G_i is the Gravity loading, and τ_{fi} is the joint friction.

The details of the coefficients D_{ij} and H_{ijk} is given in [19] through examination of Eq. (1) shows that the equation of motion is linear in the robot physical parameters, that is the mass, center of gravity locations moments and products of inertia of each link see Fig. 2.

The terms of the equation of motion given in Eq. (1) are linear in the model parameters χ that are the mass, center of gravity locations moments and products of inertia of each link. Therefore it can be written in:

$$\tau = \Phi(q,\dot{q},\ddot{q})\chi \tag{2}$$

where $\boldsymbol{\tau}$ is the torque vector, $\Phi(q,\dot{q},\ddot{q})$ represents an $(N \times R)$ observation matrix, and the R- length vector χ, contains the effective inertial parameters of the manipulator grouped in the barycentric or base parameters. The identification "observation" matrix $\Phi(q,\dot{q},\ddot{q})$ depends on the joint angles, velocities, and accelerations. The barycentric parameters of a link are combinations of its inertial parameters and its descendants in the kinematic chain [21]. The categorization and grouping of the barycentric parameters is done symbolically or by applying a set of rules. Normally, special computer programs are developed for automatic generation of the symbolic model and the associated barycentric parameters. For the CRS A465 the set of the barycentric parameters χ are given [20].

In this study, in order to make a clear comparison among the estimation methods, the problem is simplified to consider only a single joint arm of the CRS A465 to estimate its parameters. The CRS 465 single joint arm has four parameters a_i, *i* = 1,..., 4 to be identified; they are the inertia, the viscous friction coefficient, the positive side Coulomb friction, and the negative side Coulomb friction, respectively. The system equation becomes:

$$\tau = a\chi \tag{3}$$

where τ is the torque, and χ is the barycentric parameters that have been reduced to four parameters, they are the angular acceleration

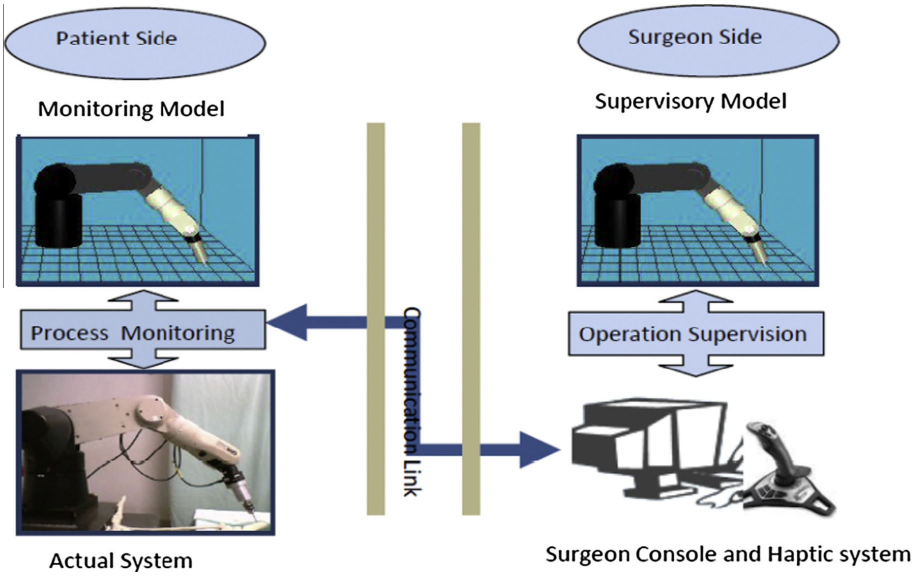


Fig. 1. Structure of a single robotic cell for robot assisted orthopaedic surgery [20].

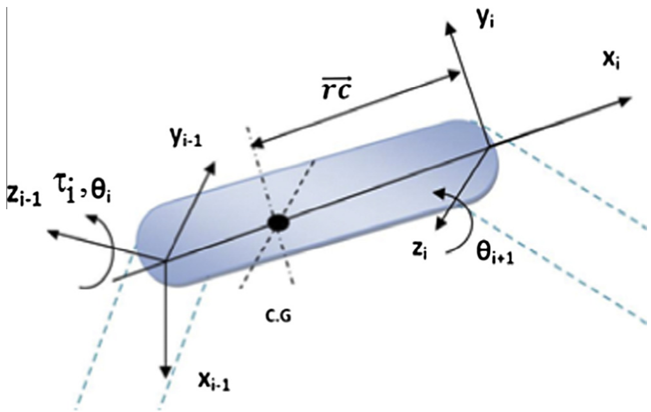


Fig. 2. Coordinate frame assignment of single joint CRS A465.

x_1, the angular velocity x_2, the positive sign of the velocity x_3 (=1 if x_2 is positive, 0 otherwise) and the negative sign of the velocity x_4 (=1 if x_2 is negative, 0 otherwise).

3. Fuzzy adaptive differential evolution algorithm (FADE)

In this algorithm [16], a fuzzy logic control (FLC) has been utilized for controlling the mutation factor F and the crossover rate CR of DE, whereby a fuzzy knowledge-based system is used to update the control parameters on-line, in a dynamic adaptive manner to the inconsistent situation. This algorithm had first been proposed to solve two major problems in DE: premature convergence and stagnation. These problems often arise when using DE or any other optimization method, because the control parameter values of these methods are sometimes not initially well tuned for the problem in hand and having no knowledge about the population's information. The general steps of FADE are as follows:

3.1. Initialization

The initial population is fed with candidate solutions using the standard equation:

$$x_{i,j}^{t=0} = X_{min,j} + \alpha_{i,j}\left(X_{max,j} - X_{min,j}\right) \quad (4)$$

where X_{min} and X_{max} are the lower and upper bound of the parameter values which are specified to define the domain from which the values, $x_{i,j}^t; j \in \{1, \ldots, D\}$ of each Np vector in this initial population are chosen. $\alpha_{i,j}$ is a random number generator that returns a uniformly distributed random number within the range [0,1).

3.2. Mutation

The standard DE/rand/1 mechanism has been used to produce the mutant vector $V_i(t) = \left\{v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,D}^t\right\}$. This operation is applied by differentiating multiple randomly selected members, r_1 and $r_2 \in \{1, 2, \ldots, Np\}$, of the current population. These indices should be mutually different and also different from the current index i. A parameter called scaling factor, F is then multiplied by the difference value $(x_{r_1}^t - x_{r_2}^t)$ to control the amplification of the differential variation. The standard mutation equation is as follows:

$$v_i^{t+1} = x_{r_1}^t + F \cdot (x_{r_2}^t - x_{r_3}^t), \quad (5)$$

$$1 \le i \ne r_1 \ne r_2 \ne r_3 \le Np$$

3.3. Crossover

The perturbation operation known as the binomial crossover (bin) comes into play, as illustrated in Eq. (3), to deliver the so-called trial vector $U_i(t) = \left\{u_{i,1}^t, u_{i,2}^t, \ldots, u_{i,D}^t\right\}$. In this interpretation a user-specified real parameter value known as crossover probability or crossover rate, CR is used to control the mixing process. Then, a component from the donor vector $v_{i,j}^t$ is transferred with probability CR to the offspring $u_{i,j}^t$, and with probability $1 - CR$ from the target vector $x_{i,j}^t$.

$$DE/x/y/bin \quad u_{i,j}^t = \begin{cases} v_{i,j}^t & (\beta_{i,j} \le CR) \text{ or } (j = j_{rand}) \\ x_{i,j}^t & \text{otherwise } j = 1, 2, \ldots, D \end{cases} \quad (6)$$

where $\beta_{i,j}$ is a real number uniformly generated in the range [0,1]. j_{rand} is a randomly generated integer in the range [1, D] and ensures that the trial vector $U_i(t)$ will differ from its corresponding target vector $X_i(t)$ by at least one component.

3.4. Selection

Then, one-to-one greedy selection operation is used to decide whether the trial vector $U_i(t)$ would win the competition against its corresponding target vector $X_i(t)$. The winner becomes a member in the population $X_i(t+1)$ of the next generation. This competition is normally based on the evaluation of the fitness function of both individuals.

$$X_i^{t+1} = \begin{cases} U_i^t & \text{if } f(U_i^t) \le f(X_i^t) \\ X_i^t & \text{otherwise} \end{cases} \tag{7}$$

3.5. Algorithm's control parameters adaptation

The control parameter values of FADE *respond to the population's information*, i.e., function values or population diversity (FC), parameter vectors (PC), and their updates after tth generations. Two FLC systems are used as the basis a fuzzy control mechanism. The main adaptation steps of FADE algorithm can be encapsulated as follows:

- *Step1 (Initialization):* The values of FC and PC are calculated and then used as input to the FLCs as in Eqs. (8) and (9), and the values of the control parameters (i.e. F and CR) are the outputs.

$$\left.\begin{aligned} PC &= \sqrt{\tfrac{1}{NP}\textstyle\sum_{i=1}^{NP}\sum_{j=1}^{D}\left(x_{i,j}^{t} - x_{i,j}^{t-1}\right)^2} \\ FC &= \sqrt{\tfrac{1}{NP}\textstyle\sum_{i=1}^{NP}\left(f_i^t - x_i^{t-1}\right)^2} \end{aligned}\right\} \tag{8}$$

$$\left.\begin{aligned} D_1 &= 1-(1+PC)\cdot e^{-PC} \\ D_2 &= 1-(1+FC)\cdot e^{-FC} \\ D_3 &= 2\cdot(1-(1+PC)\cdot e^{-PC} \\ D_4 &= 2\cdot(1-(1+FC)\cdot e^{-FC} \end{aligned}\right\} \tag{9}$$

  where PC is the parameter vector change in magnitude and is transformed into the range of [0,1] as D_1 and the range of [0,2] as D_3; FC is the function value change and is transformed into [0,1] as D_2 and [0,2] as D_4; f_i^t is the ith component of the function value vector for the tth generation, $i = 1, 2, \ldots, NP$; $x_{i,j}^t$ is the component in the ith row and jth column of the parameter matrix $X_{NP\times D}$ for the tth generation, $i = 1,2,\ldots,NP$, $j = 1,2,\ldots,D$; t is the generation index; NP and D represent the population size and dimensionality of the problem, respectively (see Table 1).
- *Step2 (Fuzzification):* The values of F and CR are assigned to the fuzzy sets membership functions. These membership functions have been stated by Liu and Lampinen [16], based on their empirical study. Fig. 3(a–d) represents four fuzzy membership functions for input variables D_1, D_2, D_3, and D_4. Fig. 3(e and f) shows fuzzy membership functions for output variables F and CR.
- *Step3 (Rule Base):* IF-THEN fuzzy rules statements, the most useful method in modeling complex systems, are used to formulate the conditional statements that comprise fuzzy logic. There are 9 × 2 rules as shown in Table 2 that comprise the fuzzy logic. Each of the variables (D_1, D_2, D_3, D_4, F, CR) has a corresponding fuzzy membership function with 3 fuzzy subsets, where S is "small", M is "middle" and B is "big" as illustrated in Table 2. Fig. 4 shows the fuzzy decision surfaces for all combinations of the inputs and outputs. These surfaces are nonlinear and monolithic surfaces and represent fuzzy decision surfaces for given inputs.
- *Step4 (Fuzzy Inference Engine):* Madmani's fuzzy inference method [22] is used as the fuzzy control strategy. It is the most commonly used fuzzy simulation methodology. In this method the fuzzy control strategy is used to map from the given inputs through the rules to an output (fuzzy sets). These fuzzy sets are then collected into one fuzzy set.
- *Step5 (Defuzzification):* The defuzzification process is held to map from a space of fuzzy output into a space of real output. The input is the collected fuzzy set and the output is a single value. There are many types of defuzzification strategies. In FADE the centroid defuzzification technique was selected (CDT), i.e., the center of gravity of the fuzzy control.

Table 1
Membership functions.

Inputs, Outputs	Membership functions
D_1	$S(D_1) = f_g(D_1, 0.25, 0.05)$ $M(D_1) = f_g(D_1, 0.25, 0.5)$ $B(D_1) = f_g(D_1, 0.25, 0.9)$
D_2	$S(D_2) = f_g(D_2, 0.5, 0.1)$ $M(D_2) = f_g(D_2, 0.5, 0.8)$ $B(D_2) = f_g(D_2, 0.5, 1.5)$
D_3	$S(D_3) = f_g(D_3, 0.35, 0.1)$ $M(D_3) = f_g(D_3, 0.35, 0.5)$ $B(D_3) = f_g(D_3, 0.35, 0.9)$
D_4	$S(D_4) = f_g(D_3, 0.5, 0.1)$ $M(D_4) = f_g(D_4, 0.5, 0.8)$ $B(D_4) = f_g(D_4, 0.5,1.5)$
F	$S(F) = f_g(F, 0.5, 0.3)$ $M(F) = f_g(F, 0.5, 0.6)$ $B(F) = f_g(F, 0.5, 0.9)$
CR	$S(CR) = f_g(CR, 0.35, 0.4)$ $M(CR) = f_g(CR, 0.35, 0.7)$ $B(CR) = f_g(CR, 0.35, 1.0)$

Note: f_g = Gaussian curve membership function.

4. Results and discussion

The kinematic and dynamic models of manipulators are nonlinear and coupled. Thus, explicit modeling of noises will make dynamic modal complex. To overcome this problem the fuzzy adaptive Differential Evolution (FADE) has been utilized for improving the parameter estimation of the robot manipulator and to deliver minimum performance variation. As a case study, the single joint arm model of the CRS A465 is considered. The CRS 465 single joint arm has $A = 4$ parameters to be estimated, as discussed in Section 2. In the simulation, a_1 is the inertia, a_2 is the viscous friction coefficient, a_3 is the positive side Coulomb friction, and a_4 is the negative side Coulomb friction.

To develop the new dynamic parameter identification framework based on FADE, attention has to be paid to the following setting points that characterize the FADE algorithm as well as the standard DE algorithm to the robot application:

- *Individual (solution encoding) and Population representation:* A population with pop_{size} ($NP = 30$) of individuals, refers to the number of individuals at each generation. First, we have to encode the necessary information required for the parameter estimation in the individual structure. Each individual should represent a complete solution to the problem at hand. In our application the individual is a vector of 4 real-coded parameters known as *solution parameters.*
- *Parameter control:* The control parameters that are going to be considered are the mutation factor, F and the crossover rate, CR. In the standard DE/rand/1/bin, these parameters have been set to 0.5 and 0.9, respectively. In FADE these parameters undergo evolution via the fuzzy adaptation system, FLC, in such a way that better values of these parame-

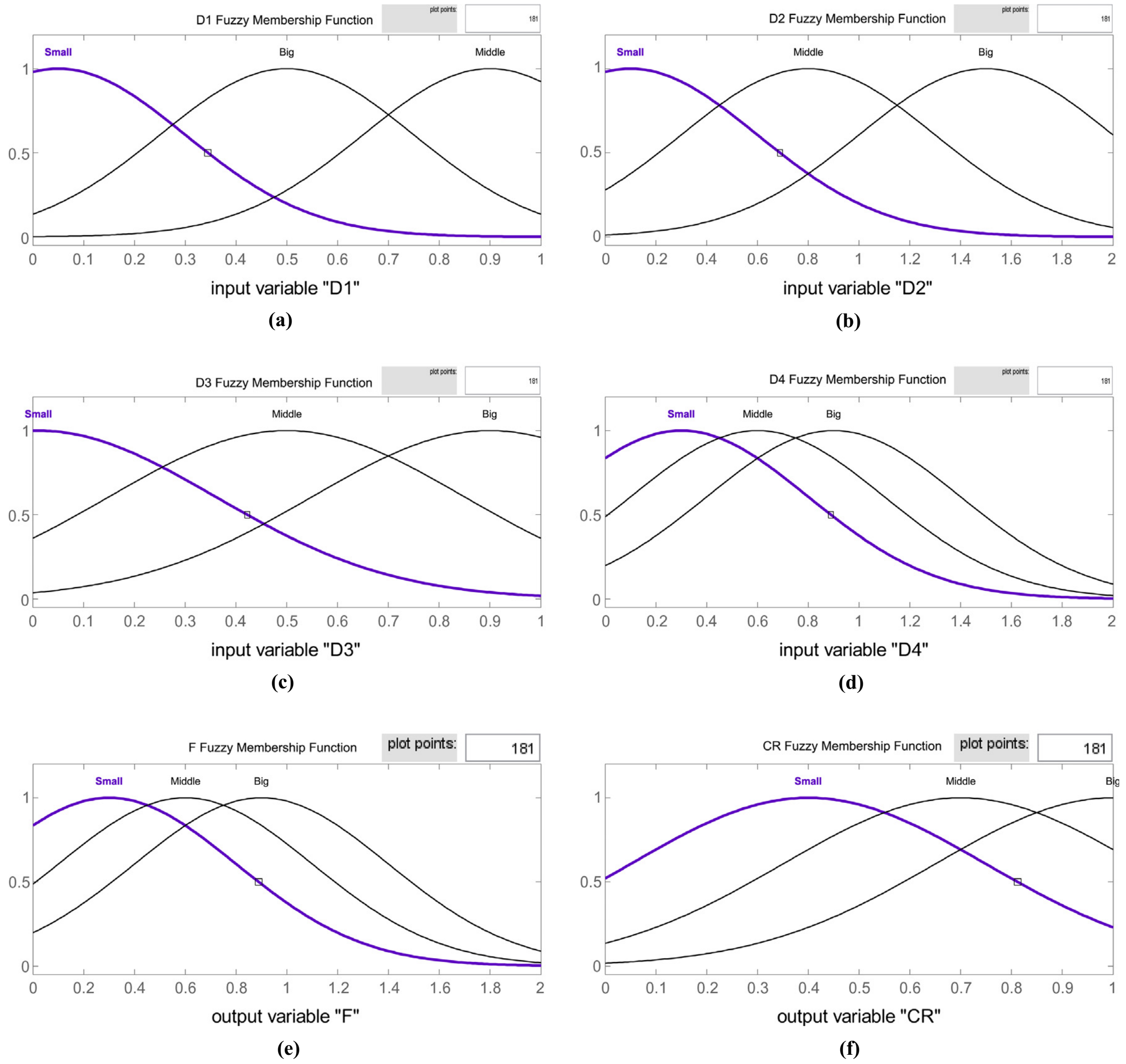


Fig. 3. Fuzzy membership function for the input and the output variables. (a) D_1, (b) D_2, (c) D_3, (d) D_4, (e) F and (f) CR.

ters would lead to better individuals which in turn are more likely to survive and produce offspring and hence propagate these better parameter values to the next generation.

- *Individual evaluation (solution validation):* The definition of the fitness function is crucially important for a successful application. In this work, we have to evaluate the fitness of each individual based on the mean square error (MSE) of the estimated model, as in Eq. (10).

$$MSE = \sum_{i=1}^{n} (\tau_i - \hat{\tau}_i)^2 \tag{10}$$

where τ represents the measured torque in our application and $\hat{\tau}$ is the estimated torque using the DE and FADE algorithms. n is the dimension of τ vector.

- *Stopping criteria:* The most common stopping condition used in literature is to allow the algorithm to run to a maximum number of iterations. A small number of iterations may not give the algorithm enough time to reach an optimum especially when the size of the search space is large. On the other hand, a very large number of iterations may be unnecessary because there can no further gain once the optimum solution is reached; so, the number of iterations for the standard DE and FADE is set at 100.

Table 2
The fuzzy rules.

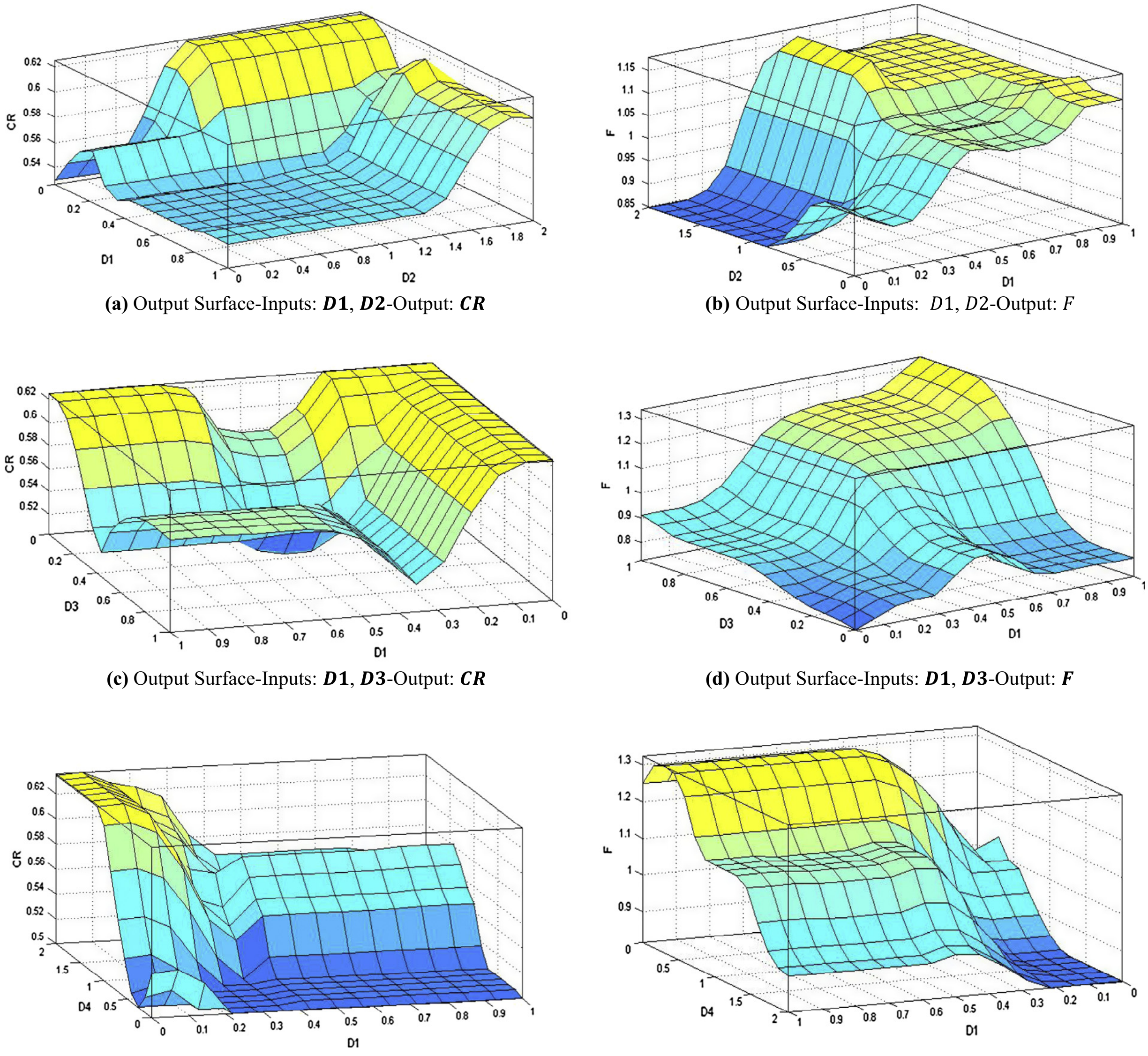
Rules	Fuzzy sets		
	D_1 or D_2	D_3 or D_4	F or CR
1	*S*	*S*	*S*
2	*S*	*M*	*M*
3	*S*	*B*	*B*
4	*M*	*S*	*M*
5	*M*	*M*	*M*
6	*M*	*B*	*B*
7	*B*	*S*	*B*
8	*B*	*M*	*B*
9	*B*	*B*	*B*

Note: *S* = *small*; *M* = *medium*; *B* = *big*.

In this comparison the ordinary least square (OLS) identification method (see Eq. (11)) has also been included to estimate the unknown parameters X_{OLS} by minimizing the sum of the squared error between the actual torque τ and the predicted torque $\Phi_{X_{OLS}}$, as follows:

$$X_{OLS} = (\phi^T \phi)^{-1} \phi^T \tau \tag{11}$$

Using the aforementioned methods and their corresponding settings the barycentric parameters of the CRS 465 single joint



(a) Output Surface-Inputs: $\boldsymbol{D1}$, $\boldsymbol{D2}$-Output: $\boldsymbol{CR}$

(b) Output Surface-Inputs: $D1$, $D2$-Output: F

(c) Output Surface-Inputs: $\boldsymbol{D1}$, $\boldsymbol{D3}$-Output: $\boldsymbol{CR}$

(d) Output Surface-Inputs: $\boldsymbol{D1}$, $\boldsymbol{D3}$-Output: $\boldsymbol{F}$

(e) Output Surface-Inputs: $\boldsymbol{D1}$, $\boldsymbol{D4}$-Output: $\boldsymbol{CR}$

(f) Output Surface-Inputs: $\boldsymbol{D1}$, $\boldsymbol{D4}$-Output: $\boldsymbol{F}$

Fig. 4. Fuzzy decision surfaces for all combinations of the inputs and outputs.

Table 3
Barycentric parameters estimation of the single joint CRS A465 robot arm.

Single joint parameters	OLS	Standard DE	FADE
a_1	0.0036	0.0037	0.0038
a_2	0.0164	0.0143	0.0169
a_3	0.0089	0.0594	0.0112
a_4	−0.2582	−0.3060	−0.2261

Table 4
Mean square error and standard deviation of the estimation methods for the estimated model.

OLS MSE (Std Dev)	Standard DE MSE (Std Dev)	FADE MSE (Std Dev)
9.259E−02 (2.099E+00)	8.872E−02 (1.415E+00)	5.233E−02 (1.380E+00)

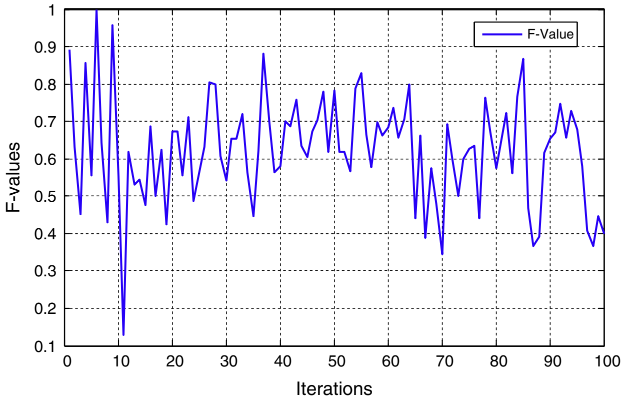
arm are estimated. The results of the three estimation techniques are presented in Table 3. These results have been averaged over 30-independent runs.

A clear comparison among these methods is presented in Table 4 which illustrates the mean square error and the standard deviation of the prediction error. From the same table it can be observed that the FADE could outperform the OLS and the standard DE for both values. This is so because FADE updates the values of the control parameters each generation and this change can deal with difficult problems such as noise.

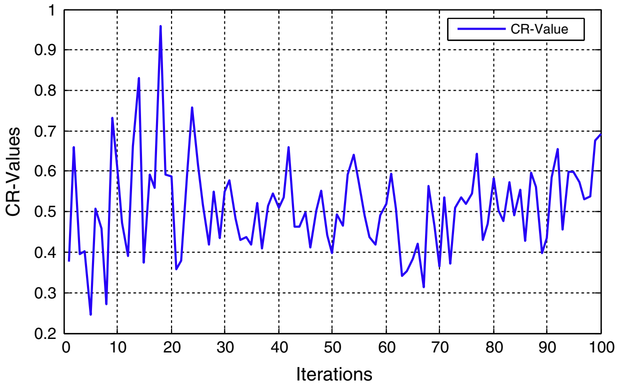
Fig. 5 depicts the different behavior of *F* and *CR* values during the 100 generations due to the population information. The plot of the figure shows a significant high fluctuation at the early stages of the run then begins to stabilize due to stability in the population. This change in *F* and *CR* values helps FADE to escape from the local optimums generated by the noisy components in the dataset.

The distinct performance of the FADE in comparison with the standard DE and the OLS is further proved in the torque prediction, as depicted in Fig. 6.

From this figure, and based on the barycentric parameters, it is worth noting that the standard DE and FADE are both nearer to the measured torque than the OLS. The difference is clearer in the accuracy of the model as already presented in Table 4. However, the difference between the standard DE and FADE performance



(a) F-value



(b) CR-value

Fig. 5. The behavior of the F and CR values in FADE algorithm during 100 generations.

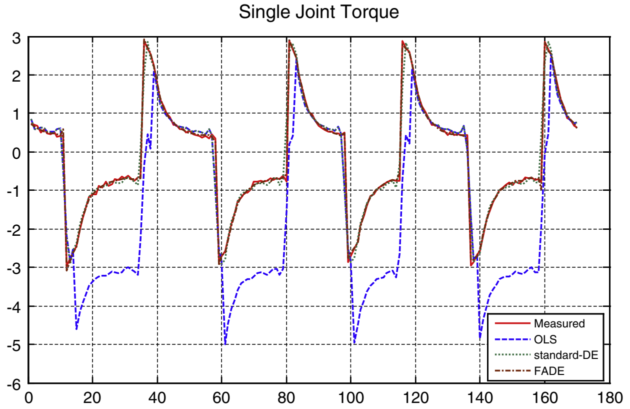


Fig. 6. Measured torque compared with estimated torque using different methods.

plot will be more significant as the number of the estimated parameters is increased.

5. Conclusion

In this paper, the fuzzy adaptive differential evolution algorithm (FADE) is utilized to estimate the barycentric parameters of single joint CRS A465 robot arm dynamics. In this method the values of the control parameters F and CR are adapted using fuzzy logic control. The main advantages of the fuzzy scheme in this approach are: computationally efficient, and well-adaptable with optimization techniques. FADE is not only a simple approach in comparison with other adaptive DE variants, but is also reliable and easy to be implemented in real time applications such as robot system identification.

The barycentric parameters of a single joint CRS A465 robot are also estimated using OLS and the standard DE, and the experimental results suggest that fuzzy adaptive DE provides better overall performance than the ordinary least square method and the standard DE with fixed parameters. However, in order to further investigate the performance of the FADE as an estimator technique and any possible shortcomings, further work is considered to increase the number of joints of the robot arm which in turn will increase the number of parameters of the predicted model.

Acknowledgments

The authors would like to thank University of Malaya for providing financial support under the UMRG Research Grants Project No. RP006E-13ICT.

References

[1] Pisla D, Gherman B, Vaida C, Suciu M, Plitea N. An active hybrid parallel robot for minimally invasive surgery. Robotics Comput – Integr Manuf 2013;29:203–21.
[2] Gomes P. Surgical robotics: reviewing the past, analysing the present, imagining the future. Robotics Comput – Integr Manuf 2011;27:261–6.
[3] Mavroidis C, Flanz J, Dubowsky S, Drouet P, Goitein M. High performance medical robot requirements and accuracy analysis. Robotics Comput – Integr Manuf 1998;14:329–38. Oct–Dec.
[4] Gautier M, Poignet P. Extended Kalman filtering and weighted least squares dynamic identification of robot. Control Eng Pract 2001;9:1361–72.
[5] Khalil W., Gautier M., Lemoine P., and Ieee. Identification of the payload inertial parameters of industrial manipulators. In: IEEE international conference on robotics and automation, Rome, Italy; 2007. p. 4943–8.
[6] Karahan O., Binguel Z., and Ieee. Modelling and identification of STAUBLI RX-60 Robot. In: 2008 IEEE conference on robotics, automation, and mechatronics, vols. 1 and 2; 2008. p. 184–9.
[7] Fleming PJ, Purshouse RC. Evolutionary algorithms in control systems engineering: a survey. Control Eng Pract 2002;10:1223–41.
[8] Vuong ND, Ang Jr MH. Dynamic model identification for industrial robots. Acta Polytech Hung 2009;6:51–68.
[9] Bingul Z, Karahan O. Dynamic identification of Staubli RX-60 robot using PSO and LS methods. Expert Syst Appl 2011;38:4136–49.
[10] Zakharov A, Halasz S. Parameter identification of a robot arm using genetic algorithms. Periodica Politehn Ser Eng 2001;45:195–209.
[11] Adamson M, Liu G. IEEE. A genetic algorithms approach to model parameter estimation of a robot joint with torque sensing; 2006.
[12] Wang K. Application of genetic algorithms to robot kinematics calibration. Int J Syst Sci 2009;40:147–53.
[13] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Global Optimization 1997;11:341–59.
[14] Price K, Storn R. Differential evolution: a simple evolution strategy for fast optimization. Dr. Dobb's J Software Tools 1997;22:18–24.
[15] Rout BK, Mittal RK. Optimal design of manipulator parameter using evolutionary optimization techniques. Robotica 2010;28:381–95.
[16] Liu J, Lampinen J. A fuzzy adaptive differential evolution algorithm. Soft Comput Jun 2005;9:448–62.
[17] Tvrdik J. Adaptation in differential evolution: a numerical comparison. Appl Soft Computing Jun 2009;9:1149–55.
[18] Zhu W, Tang Y, Fang J-A, Zhang W. Adaptive population tuning scheme for differential evolution. Inf Sci 2013;223:164–91.
[19] Mittal RK, Nagrath IJ. Robotics and control. New Delhi: Tata McGraw-Hill; 2003.
[20] Kinsheel A, Taha Z, Deboucha A, Ya TMYST. Robust least square estimation of the CRS A465 robot arm's dynamic model parameters. J Mech Eng Res 2012;4:89–99.
[21] Renaud M. Quasi-minimal computation of the dynamic model of a robot manipulator utilizing the Newton-Euler formalism and the notion of augmented body. In: Proceedings of the IEEE international conference robot automation, Raleigh, North Carolina. 1987.
[22] Mamdani EH, Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man–Machine Studies 1975;7:1–13.