Adaptive IIR System Identification using JADE

Subhrajit Roy*, Saúl Zapotecas Martínez† and Carlos A. Coello Coello‡

*Jadavpur University, Dept. of Electronics and Telecommunication Engineering,
Kolkata 700 032, INDIA
†‡CINVESTAV-IPN (Evolutionary Computation Group),
Departamento de Computación, México D.F. 07360, MÉXICO
Email: *roy.subhrajit20@gmail.com, †saul.zapotecas@gmail.com,
‡ccoello@cs.cinvestav.mx

***Abstract*—Conventional mathematical programming techniques have several drawbacks related to lack of stability and premature convergence when applied to the identification of adaptive IIR systems. Additionally, such mathematical methods normally fail when reduced order adaptive models are used for the identification of higher order systems. In this paper, the IIR system identification task is formulated as an optimization problem and a variant of differential evolution (DE) called JADE is adopted to solve the problem. The explorative mutation scheme and innovative parameter adaptation schemes of JADE avoid premature convergence and increase the robustness of the DE algorithm. Both actual and reduced order identification of some benchmark IIR plants is carried out through a simulation study. The results indicate that the method adopted here is able to obtain better results than those found by several state-of-the-art metaheuristics.**

I. Introduction

Infinite Impulse Response (IIR) is an important property of systems dealing with Digital Signal Processing (DSP). Filter systems exhibiting this property are known as IIR filters, which can be defined as those digital filters having an infinite-duration unit sample response. On the other hand, Finite Impulse Response (FIR) filters have fixed-duration impulse responses. Nowadays, adaptive IIR filtering is an emerging field of research due to its many applications in signal processing and communications. To attain a particular level of performance, an IIR filter requires a lower number of coefficients than the corresponding FIR filter. That is because the output of IIR filters is constituted by poles and zeros in comparison with FIR filters that only possess zeros [10].

However, the major snag that arises in adaptive IIR filtering applications is given by the error landscape, which is generally non-quadratic and multimodal in nature with respect to the filter coefficients. Therefore, mathematical programming methods fail to converge to the optimal solution and get stuck at local optima due to the multimodal property of the error surface [12]. Further, during the learning process, the movement of poles outside the unit circle incorporates instability into adaptive IIR filters. Therefore, stability monitoring is very important during identification of higher order adaptive IIR filters. The properties of an adaptive IIR filter are considerably more complex than those of an adaptive FIR filter, and hence, it is more difficult to predict its behavior. Most adaptive IIR filters are realized in direct form due to their relatively simple implementation and analysis. However, some disadvantages of the direct form, such as finite-precision effects and the complexity of stability monitoring, have led to the development of new structures (see for example [8], [9]). The computational complexity and convergence properties of adaptive algorithms depend on the filter realization. This has aroused the interest of many researchers to design suitable algorithms for the adaptive IIR problem, including the use of evolutionary techniques [5], [1].

Recently, Differential Evolution (DE) has emerged as a powerful global optimizer. In this paper, we employ a recent variant of the DE algorithm (named JADE [13]) for solving the adaptive IIR filter identification problem. The study presented here, tests the abilities of JADE components when dealing with this type of problem. The results obtained by JADE are compared with respect to those achieved by five different state-of-the-art metaheuristics. As we will see later on, JADE outperforms the other approaches with respect to which it was compared in all the test problems adopted.

The remainder of this paper is organized as follows. In Section II, we introduce the basic concepts related to IIR filters. Section III shows the details of the JADE algorithm which is adopted in this work. Section IV shows the results obtained in our study. In Section V, we present a short discussion of our results. Finally, Section VI provides our conclusions and some possible paths for future research.

II. IIR System Identification

In the system identification configuration, the adaptive algorithm searches for the adaptive filter coefficients such that its input/output relationship matches closely to that of the unknown system or plant. The block diagram of an adaptive IIR system identification is shown in Fig. 1. An IIR system is defined by the following difference equation:

$$y_0 = H_p(z) \cdot x(n) \tag{1}$$

where $x(n)$ and $y(n)$ are the input and the output signals of the IIR plant, respectively. $H_p(z)$ represents the transfer

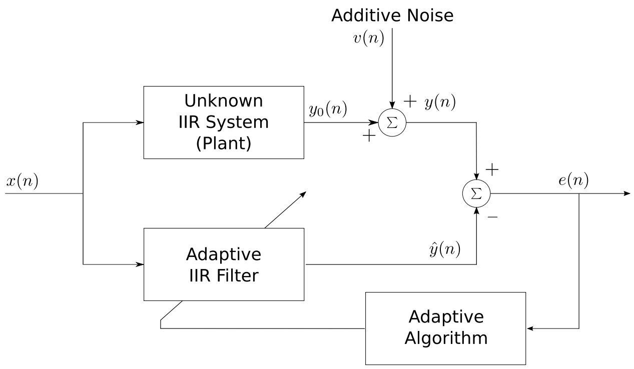

Fig. 1. Block diagram of an adaptive system identifier

function of the unknown plant and is given by:

$$H_p(z) = \left[\frac{A(z)}{B(z)}\right] \tag{2}$$

where $A(z) = \sum_{i=0}^{L} a_i z^{-i}$ and $B(z) = 1 - \sum_{i=0}^{M} b_i z^{-i}$ are z-domain feed-forward and feed-back coefficient polynomials of the IIR plant, respectively. The overall output of the plant is defined then, as:

$$y(n) = y_0(n) + v(n) \tag{3}$$

where $v(n)$ is an additive noise. From equations (1) and (3), we have:

$$y(n) = \left[\frac{A(z)}{B(z)}\right] x(n) + v(n) \tag{4}$$

The adaptive filter is governed by the difference equation:

$$\hat{y}(n) = H_M(z)x(n) \tag{5}$$

where $H_M(z)$ is the transfer function of the IIR model and is stated as:

$$H_M(z) = \left[\frac{\hat{A}(z)}{\hat{B}(z)}\right] \tag{6}$$

In equation (6), $\hat{A}(z) = \sum_{i=0}^{L} \hat{a}_i z^{-i}$ and $\hat{B}(z) = 1 - \sum_{i=0}^{M} \hat{b}_i z^{-i}$ represents the feed-forward and the feed-back coefficient polynomials of the adaptive filter, respectively. $\hat{a}_i$ and $\hat{b}_i$ denote the estimated feed-forward and feed-back coefficient of the model. The transfer function of the IIR plant is identified by using the transfer function of the adaptive filter. This identification task is formulated as an optimization problem, where the Mean Square Error (MSE) (defined in equation (7)) is used as the cost function.

$$J = E[e^2(n)] \approx \frac{1}{N}\sum_{n=1}^{N} e^2(n) \tag{7}$$

In equation (7), $e(n) = y(n) = \hat{y}(n)$ is the error signal, N is the number of input samples to be used and $E(\cdot)$ represents the statistical expectation operator.

III. The JADE Algorithm

A. Standard Differential Evolution

Differential Evolution (DE) is a simple but powerful algorithm originally developed by Rainer Storn and Kenneth Price in the mid-1990s [11]. This population-based evolutionary technique creates new candidate solutions by adding the weighted difference of several randomly selected individuals to another individual. According to [11], [6], the standard DE algorithm consists of the following steps[1].

1) Initialization of the Population: Considering a multi-dimensional optimization problem with D decision variables. DE initializes a population of NP vectors of dimension D, which should cover the entire search space as much as possible within minimum and maximum pre-defined bounds: $\vec{X}_{min} = (x_{1,min}, \ldots, x_{D,min})$ and $\vec{X}_{max} = (x_{1,max}, \ldots, x_{D,max})$, respectively.

Each vector represents a candidate solution to the optimization problem. The subsequent generations are denoted by $G = \{0, 1, \ldots, G_{max}\}$ and the i^{th} vector of the population at the current generation is denoted by $\vec{X}_{i,G} = (x_{1,i,G}, \ldots, x_{D,i,G})$.

2) Mutation: The mutation mechanism creates a donor vector $\vec{V}_{i,G}$ corresponding to each population member or target vector $\vec{X}_{i,G}$ in the generation G. The two most widely used mutation strategies (and referred here) of DE are described below.[2]

- DE/rand/1:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \tag{8}$$

- DE/target-to-best/1:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{9}$$

In equations (8) and (9), the indices $r_1^i \neq r_2^i \neq r_3^i \neq i$ are integers uniformly chosen from the set $\{1, 2, \ldots, NP\}$, and i refers to the vector concerned in the population. The constant $F > 0$ is a scaling factor which controls the amplification of the differential variation. $\vec{X}_{best,G}$ represents the vector with best fitness in the population at generation G.

3) Crossover: After the mutation step, a crossover operation is carried out. There are two different strategies for the crossover: the *binary* (bin) and the *exponential* (exp) schemes. In the binary scheme (which is adopted in this work), the trial vector $\vec{U}_{i,G} = (u_{1,i,G}, \ldots, u_{D,i,G})$ is constituted by an exchange of components between the donor vector and the target vector. That is:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } (rand_{i,j}(0,1) \leq Cr \text{ or } j = j_{rand}) \\ x_{i,j,G} & \text{otherwise} \end{cases}$$

where, $rand_{i,j}(0,1)$ is a uniformly distributed random number for each j^{th} component of the i^{th} parameter vector. j_{rand} is a randomly chosen index and $Cr \in [0, 1]$ represents the crossover probability.

[1] Source codes of differential evolution are available online at http://www.icsi.berkeley.edu/~storn/code.html

[2] More schemes for mutation can be found in [6].

4) Selection: To determine whether the target or the trial vector survives to the next generation (i.e., at $G = G + 1$) the following selection operation is employed:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & \text{if } (f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G})) \\ \vec{X}_{i,G} & \text{otherwise} \end{cases}$$

where $f(\vec{X})$ is the objective function to be minimized.

B. The JADE Algorithm

The *Adaptive Differential Evolution with Optional External Archive* (JADE) was proposed by Zhang and Sanderson in [13]. This algorithm implements a new mutation strategy called "DE/current-to-p-pest with/without external archive" and adaptively controls the parameters associated thereby increasing the convergence speed and accuracy. JADE adopts binary crossover and a *one-to-one* selection scheme. The main aspects of JADE are summarized below.

1) DE/current-to-p-best with/without external archive: "DE/current-to-p-best" is a generalization of the classic "DE/current-to-best" mutation strategy. In this generalized mutation strategy instead of just the best-solution information, the information from other aspiring solutions is adopted as well. The strategy selects (randomly) any solution of the top $100p\%$ ($p \in (0, 1]$) solutions. This plays the same role that the single best solution plays in the "DE/current-to-best" strategy. To increase the exploration of the population and also to avoid getting trapped at local optima, the difference between the archived inferior solutions and the current population can be incorporated into the mutation operator. This is an optional choice for JADE and so the mutation utilized in JADE is called "DE/current-top-best with/without external archive". In this paper, we only consider the variant of JADE with external archive and still denote it as JADE. In this way, the mutation vector is generated in the following manner:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i \cdot (\vec{X}^p_{best,G} - \vec{X}_{i,G}) + F_i \cdot (\vec{X}_{r^i_1,G} - \vec{X}_{r^i_2,G})$$

where $\vec{X}^p_{best,G}$ is a randomly chosen individual from the best $100p\%$ individuals in the current population, and F_i is the mutation scale factor associated with $\vec{X}_{i,G}$ and it is regenerated at each generation by an adaptation strategy. $\vec{X}_{r^i_2,G}$ is randomly selected from the union $P \cup A$, where P represents the current population and A represents the archive. $\vec{X}_{i,G}$, $\vec{X}^p_{best,G}$ and $\vec{X}_{r^i_1,G}$ are selected from P as in equation (9). To achieve the archive operation, at the first iteration, A is empty. After each generation, the parents which are not selected for the next generation are added to A. During this process, when the archive size exceeds a predefined threshold (NP in this work), some solutions are randomly eliminated from A to keep the archive size no larger than NP.

2) Control Parameter Adaptation: In JADE, the crossover rate Cr_i and mutation factor F_i associated with each individual vector $\vec{X}_{i,G}$, are generated using two parameters: μ_{Cr} and μ_F, respectively. At each generation G, a Cr_i of each individual $\vec{X}_{i,G}$ is independently generated according to a normal distribution with mean μ_{Cr} and standard deviation 0.1 (denoted by $N(\mu_{Cr}, 0.1)$), and then truncated to $[0, 1]$. In an analogous way, F_i of each individual $\vec{X}_{i,G}$, is independently generated according to a Cauchy distribution with location parameter μ_F and scale parameter 0.1 (denoted by $Q(\mu_F, 0.1)$), and then truncated to the interval $(0, 1]$. S_{Cr} is denoted as the set of all successful crossover rates Cr_i's at generation G. μ_{Cr} is initialized to be 0.5 and then updated at the end of the generation as follows:

$$\mu_{Cr} = (1 - C)\mu_{Cr} + C \cdot mean_A(S_{Cr}) \tag{10}$$

where C is a constant in the range $[0, 1]$ and $mean_A(\cdot)$ is the arithmetic mean operation. In an analogous way, S_F is denoted as the set of all *successful mutation factors* F_i's at generation G. μ_F is initialized to be 0.5 and then updated at the end of the generation as:

$$\mu_F = (1 - C)\mu_F + C \cdot mean_L(S_F) \tag{11}$$

where $mean_L(S_F)$ is the Lehmer mean, defined by:

$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \tag{12}$$

JADE has shown better or at least competitive performance in terms of both the convergence rate and the reliability, compared to other EAs. For a detailed description of JADE, the interested reader is referred to [13].

IV. Simulation Results

A simulation study was carried out in MATLAB in order to assess the potential of JADE for identifying IIR plants. The input was a white signal with zero mean, unit variance and uniform distribution. The additive noise employed here, corresponds to the Gaussian white signal with low variance. The results obtained by JADE were compared with respect to those reached by five state-of-the-art approaches: DE/rand/1/bin (a standard DE) [11], SaDE (DE with strategy adaptation) [7], PSO (a particle swarm optimizer) [4], DMS-PSO (a dynamic multi-swarm PSO with local search) [14], CMA-ES (an evolution strategy) [2] and a GA (a genetic algorithm) [3]. For a fair comparison, we used the best suited parametric setup chosen with guidelines provided by their authors. The population size chosen for all these algorithms was set in 50. Different tests were carried out for identification of three benchmark IIR systems. Two performance measures (Residual Mean Square Error (RMSE) and Mean Square Deviation (MSD)) were used to compare the performance of the algorithms considered in our study. RMSE is defined as the steady state MSE value and MSD is defined as:

$$MSD = \frac{1}{Q} \sum_{i=0}^{Q-1} (\Phi(i) - \hat{\Phi}(i))^2 \tag{13}$$

where Φ is the desired parameter vector, $\hat{\Phi}$ is the estimated parameter vector and Q represents the total number of parameters to be estimated. An unknown plant can be modeled in two ways: 1) using a filter of order equal to the order of the plant, and 2) using a reduced order filter. The performance of an algorithm is mostly determined by its ability to model a plant using a reduced order model. For each standard test problem, a reduced order model was used to assess the performance of the considered algorithms. For same order modeling, we calculated both the RMSE and MSD metrics; however, for reduced order modeling, only RMSE was computed. This is because the number of coefficients for the original filter and the reduced order filter are not the same, which means that the MSD formula is not valid. Next, we describe the test problems used in our experiments.

A. Example 1

The transfer function of the plant is given by:

$$H_S(z) = \left[\frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}}\right] \tag{14}$$

1) Case 1: This 2^{nd} order plant can be modeled by using a 2^{nd} order IIR filter. Hence, the transfer function of the model is given by:

$$H_S(z) = \left[\frac{a_0 + a_1z^{-1}}{1 - b_1z^{-1} + b_2z^{-2}}\right]$$

In this case, the total number of fitness function evaluations was set to $6,500$, i.e., 130 generations, for all the algorithms. Fig. 2 shows the convergence plot for the algorithms when minimizing MSE.

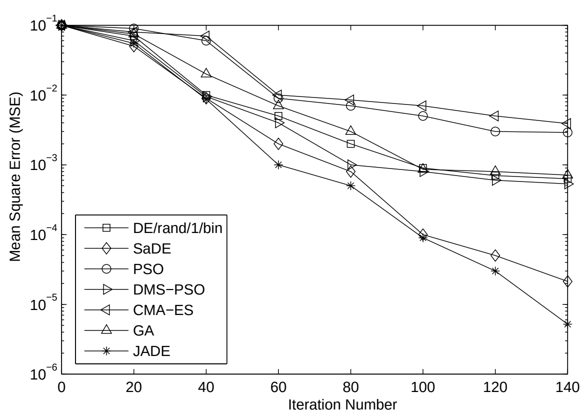

Fig. 2. Convergence plot corresponding to Example 1 using a 2^{nd} order filter.

2) Case 2: This 2^{nd} order plant can also be modeled by using a reduced order IIR filter. Therefore, a 1^{st} order IIR filter can be used for modeling the 2^{nd} order plant. In this case, the transfer function of the model is given by:

$$H_S(z) = \left[\frac{a_0}{1 - b_1z^{-1}}\right]$$

In this case, the total number of fitness evaluations carried out was set in $2,500$ (50 generations). Tables I and II show the estimation parameters for the 2^{nd} order filter, and the metrics for both the 2^{nd} and the reduced (1^{st}) order IIR filters, respectively. For an easy interpretation, the best values reached by the algorithms are displayed in **boldface** for each metric.

B. Example 2

The transfer function of the plant is given by:

$$H_S(z) = \left[\frac{-0.2 - 0.4z^{-1} + 0.5z^{-2}}{1 - 0.6z^{-1} + 0.25z^{-2} - 0.2z^{-3}}\right] \tag{15}$$

1) Case 1: This 3^{rd} order plant can be modeled by using a 3^{rd} order IIR filter. Hence, the transfer function of the model is given by:

$$H_S(z) = \left[\frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 - b_1z^{-1} - b_2z^{-2} - b_3z^{-3}}\right]$$

In this case, the number of fitness function evaluations was set to $10,000$ (300 generations). Fig. 3 shows the convergence plot for the algorithms when minimizing the MSE.

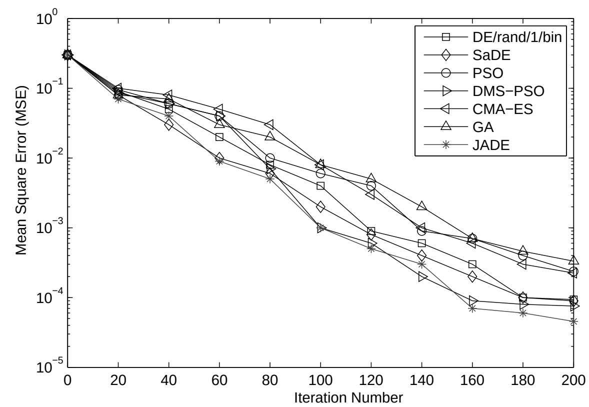

Fig. 3. Convergence plot corresponding to Example 2 using a 3^{rd} order filter.

2) Case 2: The 3^{rd} order plant considered here, can be modeled by using a 2^{nd} order IIR filter. Hence, the transfer function of the reduced order filter is stated as:

$$H_S(z) = \left[\frac{a_0 + a_1z^{-1}}{1 - b_1z^{-1} - b_2z^{-2}}\right]$$

The total number of fitness function evaluations in this case was set to $10,000$ (200 generations) for all the algorithms. Tables III and IV show the estimation parameters for the 3^{rd} order filter, and the metrics for both the 3^{rd} and the reduced (2^{nd}) order IIR filters, respectively.

C. Example 3

The transfer function of the plant is given by:

$$H_S(z) = \left[\frac{1 - 0.9z^{-1} + 0.81z^{-2} - 0.729z^{-3}}{1 + 0.04z^{-1} + 0.2775z^{-2} - 0.2101z^{-3} + 0.14z^{-4}}\right] \tag{16}$$

1) Case 1: The 4^{th} order plant can be modeled by using a 4^{th} order IIR filter. Hence, the transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3}}{1 - b_1z^{-1} - b_2z^{-2} - b_3z^{-3} - b_4z^{-4}}\right]$$

In this case, the total number of fitness function evaluations was set to $20,000$ (400 generations). Fig. 4 shows the convergence plot for the algorithms when minimizing the MSE.

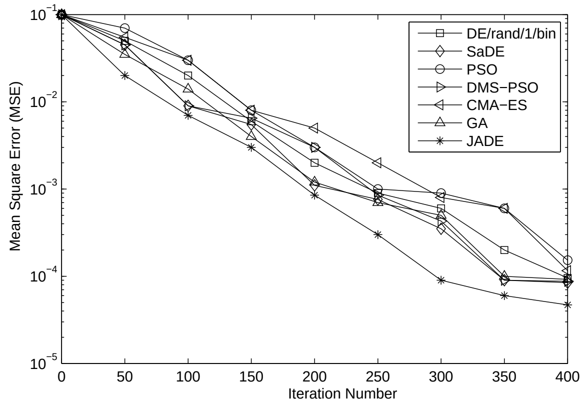

Fig. 4. Convergence plot corresponding to Example 3 using a 3^{rd} order filter.

2) Case 2: The 4^{th} order plant considered here can be modeled by using a 3^{rd} order IIR filter. Hence, the transfer function of the model is given by:

$$H_S(z) = \left[\frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3}}\right]$$

In this case, $15,000$ fitness function evaluations were carried out (i.e., 300 iterations). Tables V and VI show the estimation parameters for the 4^{th} order filter, and the metrics for both the 4^{th} and the reduced (3^{rd}) order IIR filters, respectively. The maximum number of fitness function evaluations was chosen based on the complexity of the plant. Evidently, for higher order plants the complexity increases.

V. Discussion of Results

According to the results presented in Tables II to VI, JADE has clearly shown its superiority in terms of the performance measures considered here. These tables provide a quantitative assessment of the performance of JADE in terms of the RMSE, parameter estimation and MSD. That means that JADE not only provides the best average results from all the algorithms compared in terms of RMSE but also that it has been able to successfully identify the IIR system components, since it outperformed the other approaches in terms of MSD, as well. The remarkable performance of JADE can be attributed to its algorithmic components, namely its parameter adaptation and mutation schemes. The parameter adaptation schemes which are based on recorded historical information, ensure that JADE converges to its minimum MSE with relatively few fitness function evaluations. The explorative mutation scheme of JADE also worked properly in the problems studied here, since no premature convergence occurred in any of our experiments.

VI. Conclusions and Future Work

DSP has emerged as a field of research since several years ago. IIR systems have established themselves as an important component for applications in DSP. In this paper, a variant of Differential Evolution (JADE) has been introduced for IIR system identification. This evolutionary technique has been applied to the identification of system components of some benchmark IIR plants. The performance assessment of JADE was not only carried out with a similar order but also with a reduced order model for each plant. Our preliminary results indicate that JADE constitutes a viable alternative for the identification of IIR plants, being able to outperform several other state-of-the-art metaheuristics. As part of our future work, we intend to hybridize mathematical programming techniques with JADE. In this way, while the evolutionary strategy explores the search space, mathematical programming methods could be used to exploit promissory regions within it.

Acknowledgements

The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The third author acknowledges support from CONACyT project no. 103570.

References

[1] M. Abe and M. Kawamata. Evolutionary digital filtering for iir adaptive digital filters based on the cloning and mating reproduction. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 81(3):398–406, 1998.
[2] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1769–1776, September 2005.
[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[4] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
[5] S. Ng, S. Leung, C. Chung, A. Luk, and W. Lau. The genetic search approach. a new learning algorithm for adaptive iir filtering. *IEEE Signal Processing Magazine*, 13(6):38–46, November 1996.
[6] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin, 2005. ISBN 3-540-20950-6.
[7] A. Qin, V. Huang, and P. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evo. Comp.*, 13(2):398–417, April 2009.
[8] P. Regalia. Stable and efficient lattice algorithms for adaptive IIR filtering. *IEEE Transactions on Signal Processing*, 40(2):375–388, Febrary 1992.
[9] J. Shynk. Adaptive IIR Filtering Using Parallel-form Realizations. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(4):519–533, April 1989.
[10] J. Shynk. Adaptive IIR Riltering. *IEEE ASSP Magazine*, 6(2):4–21, April 1989.
[11] R. M. Storn and K. V. Price. Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI, Berkeley, CA, March 1995.
[12] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1985.
[13] J. Zhang and A. Sanderson. Jade: Adaptive differential evolution with optional external archive. *Evolutionary Computation, IEEE Transactions on*, 13(5):945–958, October 2009.
[14] S. Zhao, J. Liang, P. Suganthan, and M. Tasgetiren. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In *2008 IEEE Congress on Evolutionary Computation*, pages 3845–3852, June 2008.

TABLE I

PARAMETER ESTIMATION FOR EXAMPLE 1 MODELED BY USING A 2^{nd} ORDER IIR FILTER

Parameter	Actual value	Estimated Value						
		DE/rand/1/bin	SaDE	PSO	DMS-PSO	CMA-ES	GA	JADE
a_0	0.05	0.085	0.0509	0.096	0.089	0.099	0.0877	0.0507
a_1	-0.4	-0.425	-0.4108	-0.467	-0.413	-0.487	-0.4112	-0.4035
b_1	1.1314	1.165	1.1232	1.204	1.173	1.186	1.182	1.1242
b_2	-0.25	-0.298	-0.2503	-0.316	-0.307	-0.31	-0.305	-0.2429

TABLE II

RESULTS OF THE RMSE AND MSD METRICS FOR EXAMPLE 1 MODELED BY USING A 2^{nd} ORDER AND THE REDUCED (1^{st}) ORDER IIR FILTER

		DE/rand/1/bin	SaDE	PSO	DMS-PSO	CMA-ES	GA	JADE
RMSE	Average	7.67E-004	2.89E-005	3.67E-003	5.09E-004	3.89E-003	8.87E-004	**4.99E-006**
	Std. Dev	1.64E-005	9.46E-006	1.18E-004	2.02E-005	3.49E-004	6.71E-005	**6.38E-007**
MSD	Average	3.03E-004	8.89E-005	7.87E-003	8.12E-004	7.89E-003	5.10E-004	**5.06E-005**
	Std. Dev	4.67E-005	1.46E-006	7.74E-004	9.18E-005	9.46E-004	1.55E-005	**3.48E-005**
RMSE (reduced order)	Average	1.14E-001	3.95E-002	8.99E-001	7.59E-002	8.84E+003	8.19E-002	**1.62E-002**
	Std. Dev	8.79E-003	9.03E-003	3.31E-003	1.69E-003	5.91E-003	2.29E-002	**1.48E-004**

TABLE III

PARAMETER ESTIMATION FOR EXAMPLE 2 MODELED BY USING A 3^{rd} ORDER IIR FILTER

Parameter	Actual value	Estimated Value						
		DE/rand/1/bin	SaDE	PSO	DMS-PSO	CMA-ES	GA	JADE
a_0	-0.2	-0.2189	-0.1963	-0.2163	-0.2128	-0.1989	-0.2289	-0.1963
a_1	-0.4	-0.3965	-0.4171	-0.3919	-0.3991	-0.4165	-0.4165	-0.4032
a_2	0.5	0.5132	0.4991	0.5192	0.5154	0.4932	0.5132	0.4946
b_1	0.6	0.6121	0.5964	0.6151	0.6117	0.5921	0.6121	0.5869
b_2	-0.25	-0.2431	-0.2472	-0.2483	-0.2473	-0.2331	-0.2431	-0.2489
b_3	0.2	0.2154	0.1982	0.2171	0.2118	0.1854	0.2254	0.1925

TABLE IV

RESULTS OF THE RMSE AND MSD METRICS FOR EXAMPLE 2 MODELED BY USING A 3^{rd} ORDER AND THE REDUCED (2^{nd}) ORDER IIR FILTER

		DE/rand/1/bin	SaDE	PSO	DMS-PSO	CMA-ES	GA	JADE
RMSE	Average	9.37E-005	8.35E-005	2.13E-004	7.68E-005	2.71E-004	3.15E-004	**4.53E-005**
	Std. Dev	1.39E-005	6.55E-006	5.79E-005	7.57E-006	6.48E-005	4.24E-005	**7.80E-007**
MSD	Average	4.55E-005	4.01E-005	4.01E-005	3.99E-005	9.29E-005	8.29E-005	**9.75E-006**
	Std. Dev	6.55E-006	6.13E-006	6.13E-006	7.73E-006	1.73E-005	7.73E-006	**8.96E-006**
RMSE (reduced order)	Average	3.73E-002	1.41E-003	1.79E-003	1.47E-003	2.22323-03	3.26E-002	**1.05E-003**
	Std. Dev	1.88E-002	1.07E-004	2.87E-004	2.73E-004	2.89E-003	1.61E-002	**2.01E-005**

TABLE V

PARAMETER ESTIMATION FOR EXAMPLE 1 MODELED BY USING A 4^{th} ORDER IIR FILTER

Parameter	Actual value	Estimated Value						
		DE/rand/1/bin	SaDE	PSO	DMS-PSO	CMA-ES	GA	JADE
a_0	1	0.8984	1.1421	0.9115	0.9012	0.9821	1.0432	0.9984
a_1	-0.9	-0.9109	-0.9092	-0.9123	-0.8971	-0.8912	-0.8973	-0.8967
a_2	0.81	0.7963	0.7412	0.8075	0.8123	0.8321	0.8072	0.8074
a_3	-0.729	-0.7066	-0.7241	-0.7563	-0.6542	-0.7121	-0.7123	-0.7268
b_1	-0.04	-0.0455	-0.0422	-0.0387	-0.0401	-0.0398	-0.0342	-0.0424
b_2	-0.2775	-0.2987	-0.29711	-0.2763	-0.2741	-0.2512	-0.2534	-0.2826
b_3	0.2101	0.2091	0.2285	0.1901	0.1984	0.2342	0.2011	0.2031
b_4	-0.14	-0.1312	-0.1519	-0.1564	-0.1451	-0.1523	-0.1502	-0.1435

TABLE VI

RESULTS OF THE RMSE AND MSD METRICS FOR EXAMPLE 3 MODELED BY USING A 4^{th} ORDER AND THE REDUCED (3^{rd}) ORDER IIR FILTER

		DE/rand/1/bin	SaDE	PSO	DMS-PSO	CMA-ES	GA	JADE
RMSE	Average	9.56E-005	8.45E-005	1.53E-004	8.77E-005	1.17E-004	9.23E-005	**4.73E-005**
	Std. Dev	6.76E-006	8.82E-006	5.97E-006	7.73E-006	4.74E-005	7.35E-006	**8.02E-007**
MSD	Average	6.85E-004	7.37E-005	4.76E-004	9.36E-005	9.89E-002	8.48E-002	**2.11E-005**
	Std. Dev	5.87E-004	2.28E-005	4.74E-004	4.79E-005	3.49E-002	3.85E-002	**1.01E-005**
RMSE (reduced order)	Average	2.87E-002	6.13E-003	3.65E-002	8.97E-003	1.58E-002	4.66E-002	**5.65E-003**
	Std. Dev	1.95E-002	3.13E-003	3.68E-002	3.46E-003	2.13E-002	2.33E-002	**4.59E-005**